LỜI CẢM ƠN

Trước tiên, tôi xin cảm ơn gia đình tôi đã luôn cổ vũ, động viên, giúp đỡ tôi trong quá trình hoàn thiện luận văn này.

Dưới sự chỉ bảo của TS.Nguyễn Văn Vinh trường Đại học Công nghệ - Đại học Quốc Gia, tôi đã hoàn thiện các nhiệm vụ đề ra của luận văn. Tôi xin gửi lời cảm ơn sâu sắc nhất tới TS.Nguyễn Văn Vinh đã tận tình hướng dẫn cho tôi những định hướng và những ý kiến rất quý báu trong suốt quá trình thực hiện luận văn này.

Tôi xin chân thành cảm ơn các thầy, cô giáo trong Bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin, Phòng Đào tạo Sau đại học - Nghiên cứu Khoa học, Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội đã tạo mọi điều kiện tốt nhất để tôi hoàn thành khóa học này.

Tôi cũng xin cảm ơn bạn bè, những người luôn khuyến khích và giúp đỡ tôi trong mọi hoàn cảnh khó khăn. Tôi xin cảm ơn cơ quan và các đồng nghiệp đã hết sức tạo điều kiện cho tôi trong suốt quá trình học tập và làm luận văn này.

Hà Nội, ngày 22 tháng 05 năm 2017

Tác giả luận văn

Nguyễn Thị Loan

LỜI CAM ĐOAN

Tôi xin cam đoan bản luận văn "***Nghiên cứu công nghệ tìm kiếm (Mã nguồn mở) Lucene áp dụng giải quyết bài toán tìm kiếm trong hệ thống Văn bản***" là công trình nghiên cứu của tôi dưới sự hướng dẫn khoa học của TS.Nguyễn Văn Vinh, tham khảo các nguồn tài liệu đã chỉ rõ trong trích dẫn và danh mục tài liệu tham khảo. Các nội dung công bố và kết quả trình bày trong luận văn này là trung thực và chưa từng được ai công bố trong bất cứ công trình nào.

Hà Nội, ngày 22 tháng 05 năm 2017

Tác giả luận văn

Nguyễn Thị Loan

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT

Từ viết tắt	Nghĩa tiếng Việt
CSDL	Cơ sở dữ liệu
DBMS	Hệ quản trị cơ sở dữ liệu (Database Management Systems)
Search Engine	Máy tìm kiếm
Index	Chỉ mục
Crawl	Thu thập dữ liệu
API	Application Programming Interface
Rank	Hạng
Stop word	Là những từ xuất hiện nhiều nhưng không mang nhiều ý nghĩa (và, vẫn, vậy, nhưng, nếu, đáng lẽ, đang, thì, thế…)

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH VẼ

MỞ ĐẦU

Với sự phát triển không ngừng của công nghệ thông tin, số lượng các tài liệu điện tử do con người tạo ra ngày càng phong phú và đa dạng, nhu cầu khai thác dữ liệu trong kho tài liệu là rất lớn, đây là một trong những nhu cầu thường ngày và thiết thực của người sử dụng. Tuy nhiên, một trong những khó khăn con người gặp phải trong việc khai thác thông tin là: Khả năng tìm kiếm chính xác thông tin cần tìm trong kho tài liệu, khả năng tìm kiếm nhanh với số lượng dữ liệu lớn. Nếu dùng các hệ quản trị cơ sở dữ liệu quan hệ để tìm kiếm dữ liệu thì sẽ gặp phải các hạn chế như: Bị giới hạn ở cú pháp của ngôn ngữ SQL, tốc độ tìm kiếm chậm khi tìm kiếm gần đúng (dùng LIKE) trong cơ sở dữ liệu lớn…Điều này đã thúc đẩy cho sự ra đời của các hệ thống tìm kiếm, điển hình nhất cho các hệ thống này là các máy tìm kiếm như Google và Yahoo…Tuy nhiên, phần lớn các công cụ tìm kiếm này đều là những sản phẩm thương mại và mã nguồn được giữ bí mật. Vì vậy, nhiều đơn vị phát triển phần mềm đã tự mình xây dựng từ đầu một công cụ tìm kiếm bằng cách sử dụng các thư viện mã nguồn mở.

Trên thế giới hiện nay có một số thư viện mã nguồn mở hỗ trợ xây dựng hệ thống tìm kiếm thông tin như: Lucene, Egothor, Xapian, MG4J, Sphinx…Trong số các mã nguồn mở này thì Lucene là thư viện mã nguồn mở được nhiều tổ chức, cá nhân sử dụng nhất, cụ thể: CNET sử dụng Lucene để tìm kiếm danh sách thể loại sản phẩm, Wikipedia dùng lucene để tìm kiếm nội dung toàn văn bản. ElasticSearch và Sorl là hai một công cụ tìm kiếm rất mạnh cũng được xây dựng và phát triển dựa trên nền tảng Lucene,… Vì vậy, trong đề tài này tôi đã lựa chọn Lucene để xây dựng thử nghiệm hệ thống tìm kiếm thông tin.

Đề tài luận văn "***Nghiên cứu công nghệ tìm kiếm (Mã nguồn mở) Lucene áp dụng giải quyết bài toán tìm kiếm trong hệ thống Văn bản***" sẽ cố gắng giải quyết các vấn đề nêu trên. Luận văn kế thừa thư viện mã nguồn mở Lucene để xây dựng hệ thống tìm kiếm với hai thành phần chính là Tạo chỉ mục và Tìm kiếm.

Luận văn tập trung nghiên cứu công nghệ mã nguồn mở Lucene áp dụng cho bài toán quản lý Văn bản, đưa ra các hướng phát triển trong tương lai. Do thời gian có hạn, việc xử lý văn bản, theo dõi tiến độ xử lý, đánh giá kết quả xử lý… là phức tạp nên luận văn chỉ tập trung hoàn thiện các chức năng về quản lý văn bản và áp dụng công nghệ Lucene để đánh chỉ mục, tìm kiếm văn bản.

Nội dung mà luận văn nghiên cứu bao gồm: Tìm hiểu tổng quan về các hệ thống tìm kiếm thông tin. Tìm hiểu tổng quan về công nghệ tìm kiếm mã nguồn mở Lucene. Phân tích, thiết kế, xây dựng ứng dụng thử nghiệm Quản lý Văn bản.

Bố cục của luận văn như sau:

Chương 1: Nghiên cứu tổng quan về hệ thống tìm kiếm thông tin, các thành phần và nguyên lý hoạt động của hệ thống tìm kiếm thông tin.

Chương 2: Nghiên cứu các tính năng và hoạt động của mã nguồn mở Lucene, sử dụng mã nguồn mở Lucene.NET để xây dựng thử nghiệm hệ thống tìm kiếm thông tin.

Chương 3: Trên cơ sở nghiên cứu về Hệ thống tìm kiếm thông tin và mã nguồn mở Lucene, chúng tôi đề xuất xây dựng thử nghiệm hệ thống tìm kiếm Văn bản với hai thành phần chính là: Tạo chỉ mục và Tìm kiếm.

Chương 4: Trình bày các kết quả đạt được, những hạn chế của luận văn và hướng phát triển cho hệ thống quản lý Văn bản ứng dụng công nghệ Lucene trong tương lai.

CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN VỀ HỆ THỐNG TÌM KIẾM THÔNG TIN

Với những hệ thống có số lượng lớn các tài liệu thì việc tra cứu, tìm kiếm thông tin thông thường chưa đáp ứng được nhu cầu tìm kiếm của người dùng. Hệ thống chủ yếu tìm kiếm một cách chính xác dựa trên tiêu đề của tài liệu, cơ sở dữ liệu tìm kiếm đơn giản, tốc độ tìm kiếm chậm, chưa chính xác và chưa hỗ trợ các phép toán tìm kiếm. Vậy đây chính là các vấn đề cần cải thiện để cải thiện cho các hệ thống tra cứu tìm kiếm thông tin.

1.1. Khái niệm về hệ thống tìm kiếm thông tin

Theo lý thuyết, hệ thống tìm kiếm thông tin là một hệ thống thông tin. Nó được sử dụng để lưu trữ, xử lý, tra cứu, tìm kiếm và phổ biến các yếu tố thông tin đến người sử dụng. Hệ thống tìm kiếm thông tin thường thao tác với các dữ liệu dạng văn bản và không có sự giới hạn về các yếu tố thông tin trong văn bản.

Hệ thống thông tin bao gồm một tập hợp các yếu tố thông tin, một tập các yêu cầu và một vài cơ chế tìm kiếm để quyết định yếu tố thông tin nào liên quan đến các yêu cầu. Theo nguyên tắc, mối quan hệ giữa các câu truy vấn và tài liệu có được từ sự so sánh trực tiếp. Nhưng trên thực tế, sự liên quan giữa các câu truy vấn và tài liệu xác định không phải được quyết định trực tiếp mà gián tiếp bằng cách: các tài liệu, yếu tố thông tin phải chuyển sang ngôn ngữ chỉ mục trước khi xác định mức độ liên quan. Người sử dụng có thể đưa vào những câu hỏi, những yêu cầu và hệ thống sẽ tìm trong các tập chỉ mục để tìm ra các tài liệu liên quan, sau đó sắp xếp các tài liệu theo mức độ liên quan giảm dần và trả về cho người sử dụng.

1.2. Các bộ phận cấu thành hệ thống tìm kiếm thông tin

1.2.1. Bộ thu thập thông tin

Bộ phận thu thập thông tin là một chương trình chạy tự động dùng để đi thu thập, lấy dữ liệu và lưu trữ các nội dung từ các trang web trên Internet. Bộ phận này có các thành phần chính: Một thành phần để theo dõi và phát hiện các URL mới, phát hiện các URL thay đổi. Một thành phần dùng để đọc đệ quy nội dung tài liệu của tất cả các trang web từ một tập các URL đã có, phân tích tài liệu, trích xuất nội dung tài liệu dưới các định dạng như html, pdf, excel...và lưu trữ về cơ sở dữ liệu thu thập.

1.2.2. Bộ lập chỉ mục

Hệ thống lập chỉ mục là để tối ưu hóa tốc độ và hiệu suất trong việc tìm kiếm các tài liệu có liên quan cho một truy vấn tìm kiếm. Nếu không có chỉ mục, công cụ tìm kiếm sẽ quét tất cả các tài liệu trong thư viện, đòi hỏi thời gian và sức mạnh tính toán đáng kể. Chẳng hạn, trong khi một chỉ mục 10.000 tài liệu có thể được truy vấn trong vòng mili giây thì việc quét theo từng phần của mỗi từ trong 10.000 tài liệu lớn có thể mất hàng giờ.

Chỉ mục nghịch đảo:

Nhiều công cụ tìm kiếm kết hợp một chỉ số đảo ngược khi đánh giá một truy vấn tìm kiếm để nhanh chóng tìm các tài liệu chứa các từ trong một truy vấn và sau đó sắp xếp các tài liệu này theo sự liên quan. Bởi vì chỉ mục nghịch đảo chứa danh sách các tài liệu chứa mỗi từ, công cụ tìm kiếm có thể sử dụng truy cập trực tiếp để tìm các tài liệu liên quan đến mỗi từ trong truy vấn để lấy các tài liệu phù hợp nhất. Sau đây là một minh hoạ đơn giản của một chỉ mục nghịch đảo:

Chúng ta có 5 tài liệu với nội dung như sau;

Tài liệu 1: **Giáo dục** là quốc sách hàng đầu

Tài liệu 2: **Tin học** là một ngành **khoa học**

Tài liệu 3: Đầu tư cho **giáo dục**, đào tạo và **khoa học**, công nghệ là đầu tư cho phát triển

Tài liệu 4: Sở **giáo dục** và đào tạo **Hải Dương**

Tài liệu 5: **Giáo dục** là tương lai của dân tộc

Vậy chỉ mục nghịch đảo của tập các tài liệu trên với các từ: Giáo dục, Hải Dương, Tin học và Khoa học là:

Bảng 1.2.2.1: Bảng chỉ mục nghịch đảo

Từ	Các tài liệu	Ký hiệu
Giáo dục	Tài liệu 1, Tài liệu 3, Tài liệu 4, Tài liệu 5	D1, D3, D4, D5
Hải Dương	Tài liệu 4	D4
Tin học	Tài liệu 2	D2
Khoa học	Tài liệu 2, Tài liệu 3	D2, D3

Từ bảng lưu chỉ mục nghịch đảo ở trên ta có thể thấy được việc tìm kiếm sẽ nhanh hơn rất nhiều so với việc không lưu trữ dữ liệu dưới dạng chỉ mục nghịch đảo. Ví dụ để tìm từ khóa "Giáo dục" chúng ta phải duyệt qua tất cả các nội dung của 5 tài liệu ở trên, nếu tài liệu nào có thì hiển thị kết quả cho người dùng. Còn

đối với chỉ mục nghịch đảo, người dùng tìm từ khóa "Giáo dục" hệ thống sẽ hiển thị ra kết quả là các tài liệu: Tài liệu 1, Tài liệu 3, Tài liệu 4 và Tài liệu 5 (ở bảng trên) mà không cần phải đọc nội dung của tất cả các tài liệu.

Ngoài ra, giả sử chúng ta muốn tìm kiếm cụm từ: "Giáo dục", cụm từ "Khoa học" và tìm kiếm cụm từ "Giáo dục" AND "Khoa học".

Kết quả tìm kiếm với các từ khóa trên cho tập kết quả như sau:

Giáo dục: {D1, D3, D4, D5}

Khoa học: {D2, D3}

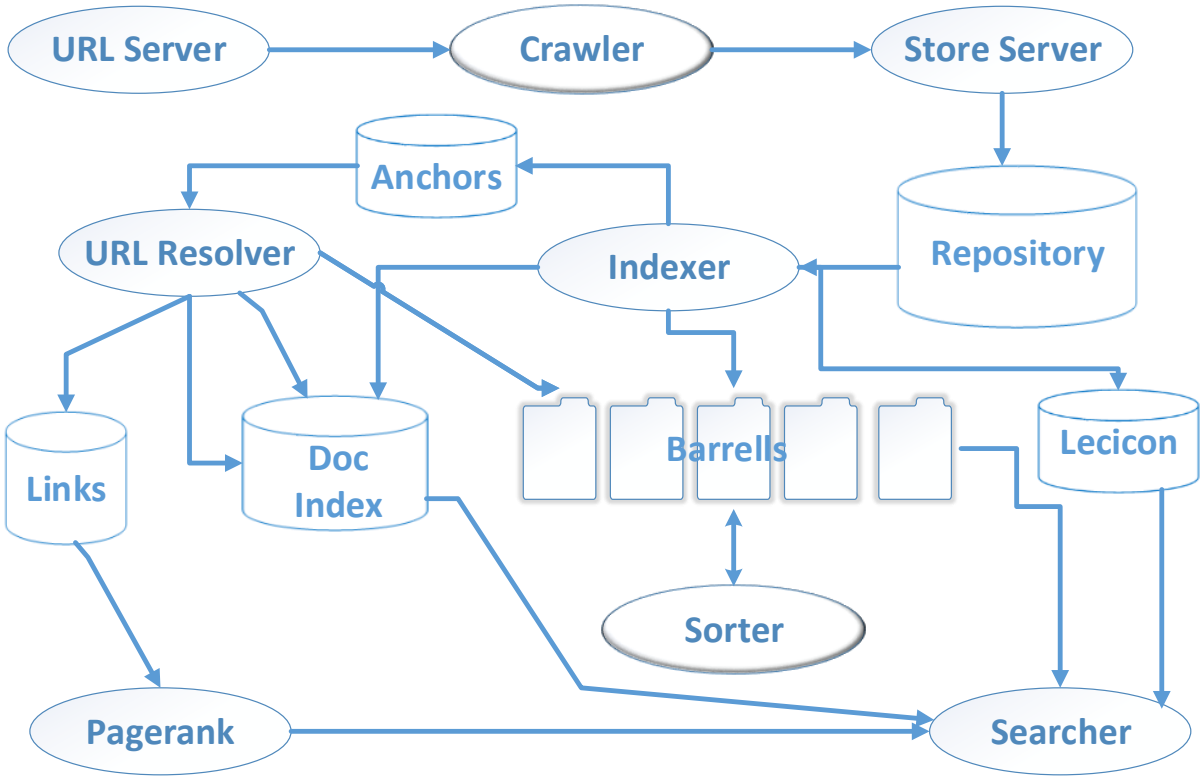
"Giáo dục" AND "Khoa học" : {D1, D3, D4, D5} ∩ {D2, D3} = {D3}

1.2.3. Bộ tìm kiếm thông tin

Bộ phận này chịu trách nhiệm tìm kiếm các tài liệu từ yêu cầu của người sử dụng, sau đó trả về danh sách các tài liệu chính xác với yêu cầu nhất. Do số lượng các trang web là rất lớn, và thông thường người dùng chỉ đưa vào một vài từ khóa trong câu truy vấn nên tập kết quả thường rất lớn. Vì vậy bộ xếp hạng (ranking) có nhiệm vụ sắp xếp các tài liệu này theo mức độ hợp lệ với yêu cầu tìm kiếm và hiển thị kết quả cho người sử dụng. Khi muốn tìm kiếm các trang web về một vấn đề nào đó, người sử dụng đưa vào một số từ khóa liên quan để tìm kiếm. Module truy vấn dựa theo các từ khóa này để tìm kiếm trong bảng chỉ mục nội dung địa chỉ các url có chứa từ khóa này. Sau đó, module truy vấn sẽ chuyển các trang web cho module xếp hạng để sắp xếp các kết quả theo mức độ giảm dần của tính hợp lệ giữa trang web và câu truy vấn rồi hiển thị kết quả cho người sử dụng.

1.3. Hệ thống tìm kiếm thông tin của Google

Google là một công ty Internet có trụ sở tại Hoa Kỳ, được thành lập vào năm 1998. Sản phẩm chính của công ty này là công cụ tìm kiếm Google, được nhiều người đánh giá là công cụ tìm kiếm hữu ích và mạnh mẽ nhất trên Internet. Trong khuôn khổ của đề tài, tôi đề xuất nghiên cứu mô hình tìm kiếm thông tin của Google để hiểu rõ hơn về kiến trúc của một Hệ thống tìm kiếm thông tin. Mô hình kiến trúc tổng thể của hệ thống tìm kiếm Google như sau:



Hình 1.3.1: Mô hình kiến trúc của hệ thống tìm kiếm Google [6]

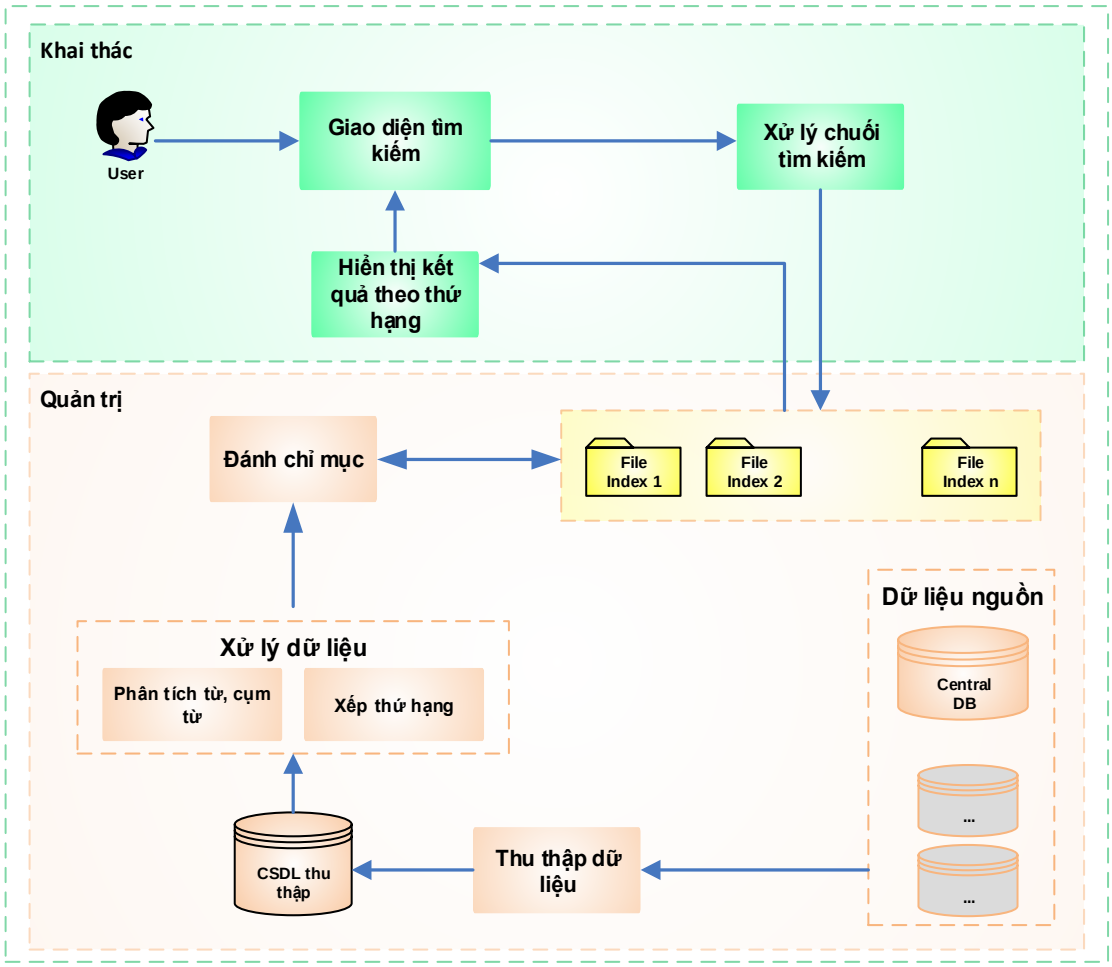
- Quy trình làm việc của hệ thống và chức năng của từng thành phần được mô tả như sau:

URL server gửi cho Crawler (được tổ chức phân tán, làm việc song song) một tập hợp các địa chỉ URLs. Các tài liệu (WebPages, hay Document) được Crawler tải xuống đưa vào Store Server, tại đây chúng được nén lại theo chuẩn Zlib (RFC 1950) và lưu trữ vào hệ thống lưu trữ tập trung Repository. Tại Repository, mỗi tài liệu được gán cho một số number: DocID, Indexer đọc tài liệu từ Repository, giải nén và phân tích chúng. Tài liệu sau đó được chuyển đổi sang một tập các từ khóa xuất hiện bên trong nó gọi là Hits, mỗi hits là một bản ghi gồm: từ khóa, vị trí xuất hiện, font size, chữ hoa/thường. Indexer phân bổ các hits vào trong tập các kho chứa nhỏ hơn Barrels. Đồng thời nó cũng phân tích toàn bộ các đường link có trong mỗi trang và lưu trữ quan trọng vào AnchorsFile: text của link, link from, link to.

URLresolver đọc AnchorsFile rồi chuyển đổi đường dẫn tương đối về tuyệt đối và ánh xạ tương ứng các đường dẫn tuyệt đối này với DocIDs, sau đó thông tin này sẽ được đưa vào Barrels tương ứng theo DocID. Đồng thời

cũng sản sinh Database link (lưu từng cặp DocIDs có mối liên kết với nhau). Sorter sắp xếp dữ liệu (hits) trong Barrels bởi DocID và sắp xếp lại bởi WordID để tạo ra Inverted Index (index nghịch đảo). Bộ phận từ điển Lexicon lấy danh sách WordID tạo ra mục từ mới. Searcher chạy bởi một WebServer sử dụng các từ điển (Lexicon) và thông tin index đảo (invert index) trong Barrels cùng với kết quả tính rank (từ PageRank) để trả về kết quả tìm kiếm.

1.4. Kiến trúc của hệ thống tìm kiếm thông tin



Hình 1.4.1.1: Mô hình kiến trúc hệ thống tìm kiếm thông tin

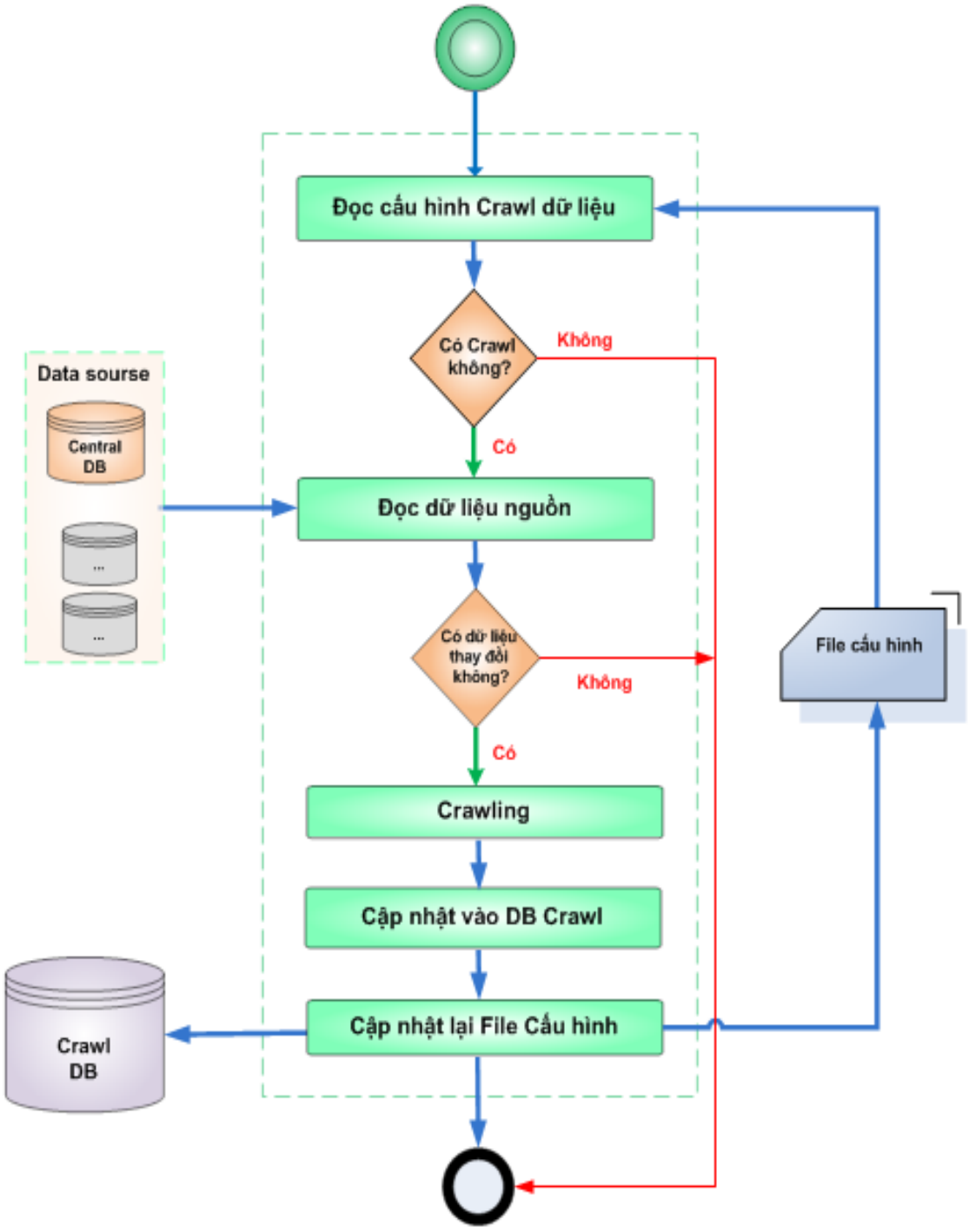
Dựa trên ý tưởng của Google và các hệ thống tìm kiếm thông tin khác chúng ta có thể hiểu về cơ bản một hệ thống tìm kiếm thông tin luôn có ba thành phần như sau:

- *Thành phần Thu thập dữ liệu:* thực hiện thu thập toàn bộ dữ liệu sẽ tìm kiếm đưa về một nguồn tập trung để phục vụ quá trình phân tích và đánh chỉ mục dữ liệu, thành phần này được quản lý bởi môđun thu thập dữ liệu, môđun này sẽ thực hiện một số chức năng chính như: Quản lý kết nối tới

nguồn cần thu thập, quản lý chi tiết đến từng loại (đối tượng) dữ liệu cần thu thập. Sau đó thực hiện thu thập dữ liệu theo từng loại dữ liệu này. Thiết lập và quản lý các kết nối tới nguồn dữ liệu cần thu thập, mỗi kết nối sẽ tương ứng với một nguồn dữ liệu, đại diện là một CSDL, và một CSDL có thể có một hoặc nhiều đối tượng dữ liệu cần thu thập. Mỗi thông tin kết nối tới nguồn dữ liệu sẽ bao gồm thông tin cơ bản sau như: Thông tin kết nối tới máy chủ, thông tin kết nối tới CSDL. Quản lý các đối tượng dữ liệu cần thu thập theo từng nguồn dữ liệu đã đưa vào hệ thống quản lý. Mỗi đối tượng dữ liệu cần quản lý các thông tin đặc tả như: Nhóm các bảng (table) liên quan đến đối tượng dữ liệu cần thu thập, tại mỗi table phải chỉ ra các trường (field) đại diện cho table đó, quan hệ giữa các table và khoá quan hệ giữa các table. Việc quản lý các thông tin đặc tả của từng đối tượng dữ liệu để phục vụ việc xây dựng các câu truy vấn (query) dữ liệu nguồn cần thu thập.

Thu thập dữ liệu lần đầu: Thực hiện thu thập dữ liệu lần đầu tiên (ngay sau khi thiết lập các kết nối tới nguồn dữ liệu, và xác định các đối tượng dữ liệu cần thu thập). Chức năng này sẽ lấy toàn bộ dữ liệu nguồn (theo từng đối tượng đã xác đinh trong hệ thống) về dữ liệu thu thập. Theo dõi thay đổi dữ liệu nguồn: Sử dụng kỹ thuật trigger để ghị lại sự thay đổi các thông tin (mang tính chỉ dẫn) theo từng đối tượng dữ liệu nguồn bị thay đổi (thêm mới, cập nhật, xoá) vào LOG FILE, làm cơ sở cho chức năng Thu thập dữ liệu định kỳ thực hiện cập nhật lại cơ sở dữ liệu thu thập.

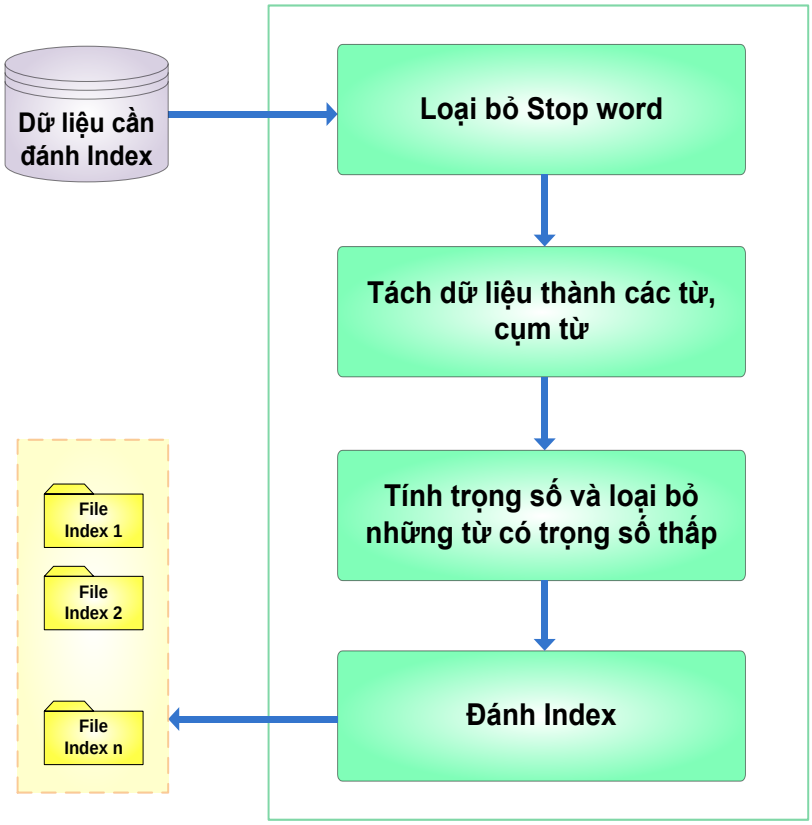
Thu thập dữ liệu định kỳ: Dựa vào thông tin chỉ dẫn trong LOG FILE (được cập nhật bởi chức năng theo dõi thay đổi dữ liệu nguồn), chức năng này sẽ thu thập bổ sung dữ liệu nguồn về cơ sở dữ liệu thu thập. Ghi nhận lại quá trình thu thập dữ liệu (ghi log) phục vụ cho mục đích phân tích, đánh giá và có thể có những điều chỉnh cần thiết nhằm khắc phục sự cố hoặc tăng hiệu xuất của hệ thống sau này.



Hình 1.4.1.2: Quy trình thu thập dữ liệu

- *Thành phần Đánh chỉ mục dữ liệu:* thực hiện phân tích, tiền xử lý nội dung dữ liệu, sau đó tiến hành đánh chỉ mục dữ liệu theo cách thức, cơ chế và yêu cầu của từng máy tìm kiếm cụ thể, thực hiện đánh chỉ mục dữ liệu này lưu vào các File index. Thành phần (môđun) này sẽ thực hiện các chức năng chính như: Tiền xử lý dữ liệu cần đánh index: Thực hiện loại bỏ các từ dư thừa, vô nghĩa, tách dữ liệu thành các từ, cụm từ,

xử lý Tiếng Việt có dấu và Tiếng Việt không dấu. Phân tích và xác định các từ, cụm từ thích hợp có khả năng đại diện cho nội dung của tài liệu. Thực hiện đánh index cho dữ liệu sau khi thu thập dữ liệu lần đầu. Sau một thời gian dữ liệu nguồn có sự thay đổi, bộ phận thu thập tiếp tục quá trình thu thập dữ liệu và bộ phận đánh chỉ mục sẽ tiến hành đánh chỉ mục, cập nhật file index, cập nhật quá trình đánh index, cập nhật lại kết quả và quá trình đánh index dữ liệu phục vụ công tác tra cứu, tìm kiếm và phân tích khi cần thiết. Xếp hạng (ranking) cho tài liệu theo tần suất xuất hiện của các từ chỉ mục trong tài liệu, tần suất nghịch đảo của tài liệu, số term (field) trong câu truy vấn tìm thấy trong tài liệu. Chức năng đánh chỉ mục dữ liệu đã qua tiền xử lý sử dụng API sẵn có của Lucene, hỗ trợ khả năng cập nhật lại file index mỗi khi có sự thay đổi mà không phải đánh index lại từ đầu.



Hình 1.4.1.3: Quy trình đánh chỉ mục

- Thành phần *Tìm kiếm dữ liệu:* thực hiện phân tích câu truy vấn và thực hiện tìm kiếm tài liệu trên các file index, sau đó kết hợp với thông tin xếp hạng

(Rank) để trả lại kết quả tìm kiếm cho người dùng, thành phần này có một số chức năng chính như: Tiền xử lý khoá tìm kiếm, thực hiện phân tích từ khoá tìm kiếm, xử lý các toán tử tìm kiếm cơ bản (AND, OR, NOT,...), xử lý tìm kiếm chính xác, và xây dựng câu truy vấn dữ liệu. Truy vấn cụm từ, truy vấn boolean, truy vấn bắt đầu bằng các ký tự đại diện, truy vấn giới hạn thời gian, số lượng… Tính năng này sẽ kết hợp với nhiều API do Lucence cung cấp để thực hiện truy vấn dữ liệu từ File index, tìm trên nhiều Field, xếp hạng kết quả tìm kiếm và trả về kết quả tốt nhất lên đầu.

Ngoài các chức năng cơ bản của ba thành phần tìm kiếm trên, hệ thống còn có nhóm các chức năng liên quan đến việc thiết lập, cấu hình hệ thống như: Lập lịch thu thập dữ liệu, chỉ ra các thông tin cấu hình liên quan đến việc thu thập dữ liệu như: Hình thức thu thập dữ liệu (tự động, hay không tự động), định kỳ bao lâu thì thực hiện thu thập dữ liệu, lập lịch thu thập dữ liệu định kỳ.

Lập lịch đánh index dữ liệu, chức năng này chỉ ra các thông tin cấu hình liên quan đến việc đánh index dữ liệu như: Hình thức đánh index (tự động hay không tự động), lịch đánh index dữ liệu định kỳ, vị trí các tệp lưu trữ file Index.

Quản lý nhật ký thu thập dữ liệu, ghi nhận lại kết quả thu thập dữ liệu nguồn, bao gồm cả thu thập dữ liệu lần đầu, thu thập dữ liệu định kỳ. Cung cấp các chức năng tra cứu, tìm kiếm, thống kê nhật ký thu thập dữ liệu, hỗ trợ người quản trị hệ thống phân tích kết quả thu thập dữ liệu khi cần thiết. Quản lý nhật ký đánh chỉ mục dữ liệu: Ghi nhận lại kết quả quá trình đánh chỉ mục dữ liệu, cung cấp các chức năng tra cứu, tìm kiếm, thống kê quá trình đánh chỉ mục dữ liệu, hỗ trợ người quản trị hệ thống phân tích kết quả đánh chỉ mục dữ liệu khi cần thiết.

Từ những nghiên cứu trên chúng ta có thể nhận thấy hệ thống tìm kiếm thông tin có những ưu điểm vượt trội hơn so với chức năng tìm kiếm trong cơ sở dữ liệu thông thường như: Hệ quản trị CSDL thông thường không thể đánh chỉ mục cho dữ liệu dạng file trong khi đó hệ thống tìm kiếm thông tin có thể đánh chỉ mục cho tất cả các tập tin dạng: pdf, html, MS Word, Excel,...Các câu truy vấn của các hệ quản trị CSDL bị giới hạn bởi cú pháp của SQL query, trong khi câu truy vấn của Hệ thống tìm kiếm gần với yêu cầu tìm kiếm của người dùng, chúng ta có thể

dùng các phép toán tìm kiếm AND, OR, NOT, tìm kiếm chính xác cụm từ, cụm từ…Ngoài ra với những dữ liệu lớn thì tốc độ tìm kiếm của Hệ thống tìm kiếm thông tin nhanh hơn nhiều so với chức năng tìm kiếm của các hệ Quản trị CSDL thông thường.

CHƯƠNG 2: NGHIÊN CỨU TỔNG QUAN VỀ MÃ NGUỒN MỞ LUCENE

Lucene là thư viện mã nguồn mở cho phép xử lý các văn bản đầu vào ở dạng văn bản (text) để tạo ra tập chỉ mục và cung cấp phương thức tìm kiếm trên tập chỉ mục đó. Nó cũng cho phép người dùng kế thừa và phát triển để phù hợp với nhiều ngôn ngữ khác nhau. Chúng tôi đề xuất nghiên cứu ứng dụng Lucene để phát triển hệ thống tìm kiếm trên các văn bản lưu trữ [2].

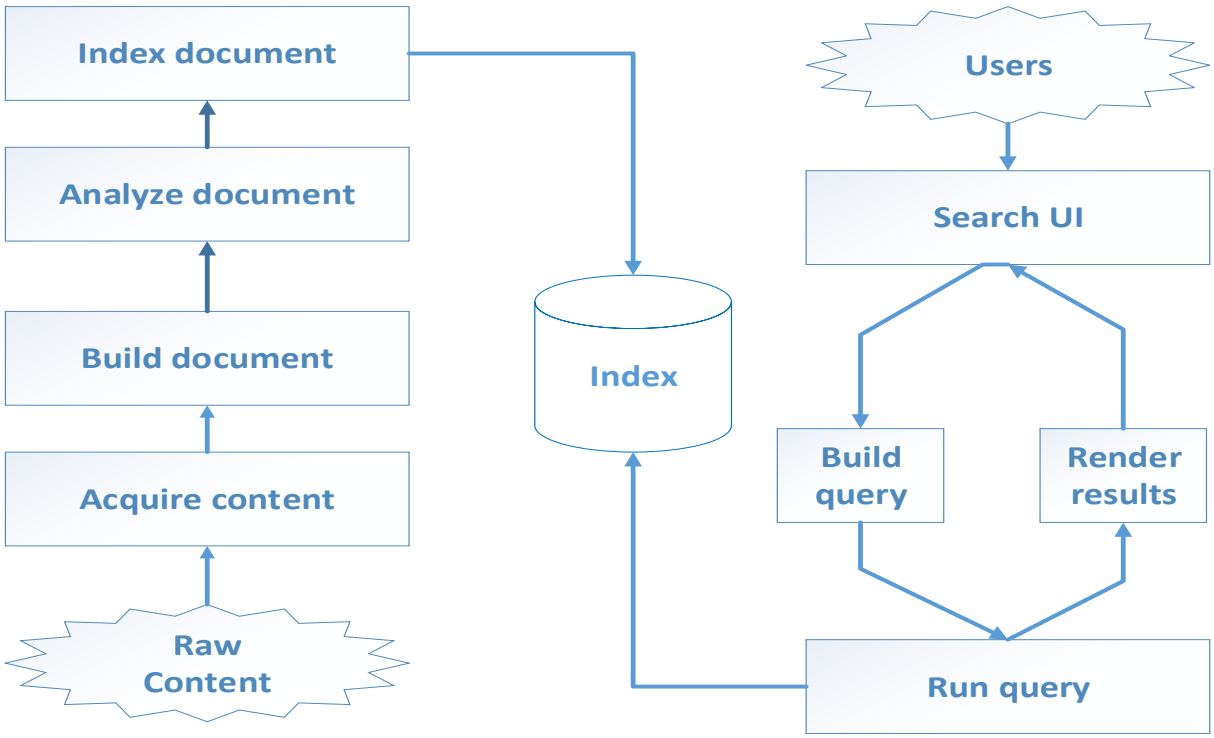
2.1. Giới thiệu về thư viện Lucene

Lucene là phần mềm mã nguồn mở, dùng để phân tích, đánh chỉ mục và tìm kiếm thông tin với hiệu suất cao bằng Java. Lucene được phát triển đầu tiên bởi Doug Cutting được giới thiệu đầu tiên vào tháng 8 năm 2000. Tháng 9 năm 2001 Lucene gia nhập vào tổ chức Apache và hiện tại được Apache phát triển và quản lý. Lucene không phải là một ứng dụng mà chỉ là một công cụ đặc tả API cần thiết cho việc xây dựng một search engine. Được xây dựng và thiết kế theo hướng hướng đối tượng nên các API cũng được cung cấp theo dạng hướng đối tượng. Mặc dù thiết kế và xây dựng ban đầu từ java nhưng hiện nay cũng đã có một số phiên bản cho các ngôn ngữ khác: .NET, C++, Perl, …[10]

- **Các phiên bản ngôn ngữ khác nhau của Lucene:**
  - Lucene4c – C
  - Clucene – C++
  - MUTIS – Delphi
  - Nlucene -.NET
  - Lucene.Net -.NET
  - Plucene – Perl
  - Pylucene – Python
  - Ferret and RubyLucene – Ruby
  - Zend Framework (Search) – PHP
  - Montezuma – Common Lisp
- **<u>Những sản phẩm sử dụng Lucene:</u>**
  - Beagle dùng một nhánh của Lucene phát triển trong C#, có tên gọi Lucene.Net làm chỉ mục.

- Docco dùng Lucene trong việc tìm kiếm trong máy tính cá nhân.
- CNET dùng Lucene để tìm kiếm danh sách thể loại sản phẩm.
- LjFind dùng Lucene để tìm kiếm hơn 110.000.000 bài post ở LiveJournal.
- Nutch là một máy tìm kiếm dùng Lucene.
- Red-Piranha cũng là một máy tìm kiếm khác dựa trên Lucene
- Wikipedia dùng Lucene để tìm kiếm nội dung toàn bộ văn bản.
- Trình duyệt web Flock dùng Clucene, một phiên bản trong C++, để tìm kiếm toàn văn hoặc tìm kiếm lịch sử của trình duyệt.
- Ants P2P dùng Lucene trong lựa chọn tìm kiếm trong chương trình chia sẻ file khuyết danh của nó.
- Solr một máy chủ tìm kiếm nguồn mở dựa trên Lucene với XML/HTTP APIs, lưu trữ (cache), sao chép, và một giao diện web quản trị.
- LIRE – Lucene Image Retrieval Thư viện CBIR, dùng máy tìm kiếm Lucene.

- **<u>Lucene trong hệ thống tìm kiếm thông tin:</u>**

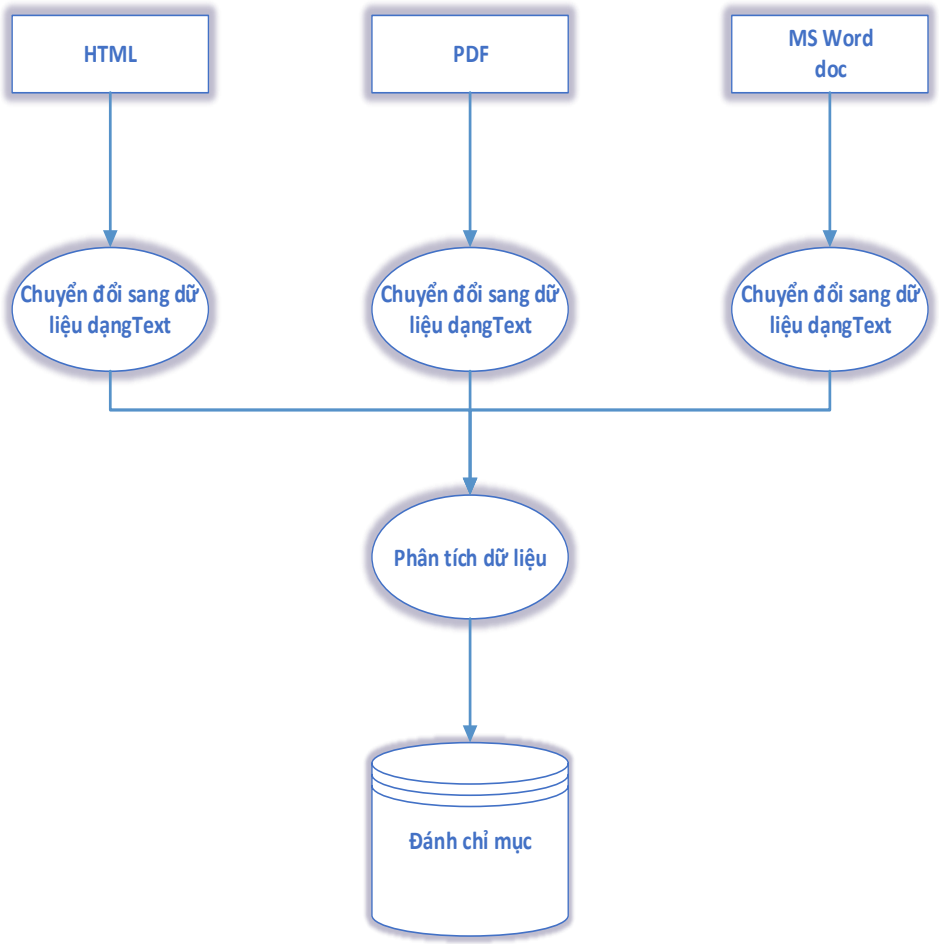


Hình 2.1.1: Lucene trong hệ thống tìm kiếm thông tin [5]

Thành phần chức năng chính của Lucene bao gồm hai phần: Thành phần tạo chỉ mục và thành phần tìm kiếm. Đây là hai thành phần quan trọng cho một hệ thống tìm kiếm thông tin.

- **Thành phần Tạo chỉ mục**: Bao gồm các chức năng xử lý và phân tích dữ liệu để đánh chỉ mục. Lucene cho phép thiết lập các trường thông tin cần thiết để đánh chỉ mục phục vụ cho thành phần tìm kiếm, các thư viện phục vụ đánh chỉ mục mà Lucene hỗ trợ. Thành phần này bao gồm các lớp đối tượng chính như: Lớp Directory, lớp này cho phép người dùng định nghĩa vùng nhớ, xác định nơi lưu trữ trên bộ nhớ trong quá trình tạo chỉ mục. Lớp Document và Field, lớp này định nghĩa các tài liệu và các trường thông tin của tài liệu sử dụng cho việc lập chỉ mục, nó cũng dùng cho việc lấy kết quả trả về cho thành phần Tìm kiếm dữ liệu. LớpAnalyzer thực hiện chức năng xử lý và phân tích nội dung văn bản để lấy nội dung, chuẩn hóa, loại bỏ mục từ không cần thiết,… để chuẩn bị cho việc lập chỉ mục. Lớp IndexWriter là thành phần chính trong thành phần tạo chỉ mục, nó thực hiện việc tạo mới, cập nhật hoặc xóa chỉ mục.
- **Thành phần Tìm kiếm:** bao gồm các phần chức năng xử lý tìm kiếm, trả về kết quả tìm kiếm cho người dùng, thông qua biên dịch và so khớp để lấy về kết quả tốt nhất. Lucene hỗ trợ nhiều loại truy vấn boolean thuận tiện cho người sử dụng như: Query: bao gồm nhiều loại truy vấn khác nhau, chứa các phương thức phục vụ các tiêu chí truy vấn của người dùng. IndexSearcher: Tìm kiếm dữ liệu trên các file chỉ mục do IndexWriter tạo ra, đây là thành phần chỉ thực hiện nhiệm vụ mở tập chỉ mục, không cho phép chỉnh sửa hay thay đổi. Có nhiều phương thức tìm kiếm, một trong số đó là lớp thành phần thực thi Searcher, với cách đơn giản là cung cấp một Query truy vấn, số lượng các liên kết cần trả về, và kết quả trả về sẽ là tập các đối tượng TopDoc. TopDoc dùng cấu hình số bản ghi có thứ hạng cao nhất trả về cho người dùng. Với mỗi đối tượng trong danh sách này sẽ cho một docID dùng để liên kết đến tài liệu nhận về.

2.2. Quy trình đánh chỉ mục



Hình 2.2.1: Quy trình đánh chỉ mục Lucene [3]

Để tiến hành đánh chỉ mục cho tài liệu, trước hết chúng ta phải chuyển đổi toàn bộ nội dung trong các file dữ liệu như HTML, PDF, MS WORD…sang các nội dung chỉ chứa dữ liệu dạng text. Lucene sẽ tiến hành phân tích và xử lý dữ liệu, loại bỏ những từ không có nghĩa, tách từ, cụm từ,….Sau khi dữ liệu được phân tích sẽ chuyển sang cho việc index. Lucene chứa dữ liệu này theo cấu trúc inverted index (chỉ mục có thể nghịch đảo). Cấu trúc này sẽ có hiệu quả để tiết kiệm dung lượng ổ đĩa và cho phép tìm kiếm nhanh hơn các từ khóa trong quá trình tìm kiếm. Nguyên tắc đó là thay vì phải tìm kiếm các từ nào chứa trong tài liệu đó thì với cấu trúc này sẽ tối ưu hóa việc tìm ra câu trả lời "tài liệu nào chứa từ khóa này".

2.3. Các toán tử đánh chỉ mục cơ bản

Lucene hỗ trợ các toán tử giúp thực hiện việc đánh chỉ mục như: Thêm tài liệu mới (Document) cùng các trường (Fields): Keyworks, UnIndexed, UnStored và Text. Trong mỗi tài liệu lại có thể có chứa nhiều Fields cùng tồn tại và trong

mỗi Fields lại có thêm nhiều giá trị khác nhau. Xóa tài liệu ra khỏi chỉ mục (Remove Documents), lớp này sử dụng lớp IndexReader với phương thức delete() ta có thể dễ dàng xóa bỏ tài liệu được chỉ định ra khỏi chỉ mục. Lucene sẽ xem như các tài liệu này được đánh dấu như là sẽ xóa. Tuy nhiên việc này chỉ có thể thực hiện khi gọi hàm close(). Cập nhật tài liệu: Lucene không hỗ trợ thực hiện việc cập nhật tài liệu, thay vào đó sẽ xóa bỏ tài liệu và sau đó thêm lại tài liệu mới thay thế. Để đảm bảo tốc độ thực thi thì tốt nhất việc xóa bỏ và thêm tài liệu mới nên thực hiện theo khối, không nên xen lẫn giữa việc xóa và thêm tài liệu mới.

2.4. Tối ưu hóa việc đánh chỉ mục

Việc tối ưu hóa tiến trình đánh chỉ mục là tiến trình trộn nhiều file chỉ mục lại với nhau để giảm thiểu thời gian đọc chỉ mục trong quá trình tìm kiếm. Bằng việc sử dụng API của Lucene mà cụ thể là hàm optimize() của đối tượng IndexWriter ta có thể dễ dàng tối ưu điêu này. Tuy nhiên việc làm này chỉ có hiệu quả tăng tốc độ tìm kiếm trên chỉ mục đã có, mà không có tác động tới tốc độ đánh chỉ mục.

2.5. Tính đồng thời, an toàn tiến tình,ngăn chặn các thực thi

Các luật đồng thời: Lucene cung cấp cho người dùng nhiều toán tử liên quan tới việc đánh chỉ mục tài liệu như: xóa, cập nhật. Do đó trong quá trình thực hiện chúng ta phải tuân theo một số luật cụ thể để tránh việc đụng độ trong quá trình thực thi. Điều này là cấn thiết khi mà có nhiều thực thi diễn ra một cách thường xuyên trước những yêu cầu gởi từ web tới ứng dụng của bạn. Sau đây là một số luật cơ bản: Bất kì toán tử chỉ đọc nào cũng có thể thực thi đồng thời, chẳng hạn là nhiều tiến trình có thể tìm kiếm cùng một chỉ mục tại một thời điểm. Bất kì toán tử chỉ đọc nào cũng có thể thực thi đồng thời trong khi một chỉ mục nào đó đang được cập nhật. Ví dụ: người dùng có thể tìm kiếm trong chỉ mục trong khi nó đang được cập nhật, thêm tài liệu mới hoặc là được xóa khỏi chỉ mục. Chỉ có duy nhất 1 toán tử cập nhật chỉ mục có thể thực thi tại một thời điểm. Một chỉ mục chỉ có thể được mở bởi chỉ một đối tượng IndexWriter hoặc là IndexReader tại một thời điểm mà thôi.

Khóa chỉ mục: nhằm tránh gây ra sự đụng độ trong quá trình sử dụng các hàm API của Lucene, thư viện này đã tạo ra các file lock bên cạnh các segment để đảm bảo rằng việc thực thi một chương trình tại một thời điểm. Mỗi chỉ mục có một tập các file lock. Có hai loại file lock được tao ra: File Write.lock: được

tạo ra trong quá trình đánh chỉ mục tài liệu nhằm đảm bảo các tiến trình thực thi một cách có tuần tự trong quá trình cập nhật chỉ mục. Hơn nữa nó được tồn tại khi dối tượng IndexWriter đang duy trì và giữ cho tới khi nó đóng mới thôi. Ngoài ra nó tồn tại khi đối tượng IndexReader đang được sử dụng để xóa, hủy bỏ việc xóa, hay cài đặt các mục của trường nào đó. Nó còn giúp khóa chỉ mục mà diễn ra lâu hơn mong đợi. File commit.lock: tạo ra trong quá trình merge các segment. Nó được dùng bất kể khi nào những segment đang được đọc hoặc trộn lại với nhau, nắm giữ bởi đối tượng IndexReader trước khi nó đọc các segment và chỉ giải phóng sau khi IndexReader đã mở và đọc các segment. Vô hiệu hóa tính năng khóa chỉ mục: nhiều khi ta cần vô hiệu hóa tính năng khóa chỉ mục của Lucene. Chẳng hạn là ứng dụng cần đọc chỉ mục nằm trên ổ CD-ROM. Có nghĩa là ứng dụng chỉ ở chế độ đọc mà thôi, tức là chỉ tìm kiếm trên đó mà không hề có cập nhật chỉ mục. Để vô hiệu hóa ta chỉ cần thay đổi thuộc tính disableLuceneLocks thành true là được.

2.6. Bộ chuyển đổi câu truy vấn của người dùng: QueryParser

Hai yêu cầu quan trọng trong ứng dụng tìm kiếm đòi hỏi là: chuyển đổi câu truy vấn và truy xuất thông tin trả về. Hầu hết các phương thức Lucene đòi hỏi đối tượng Query. Việc chuyển đổi câu truy vấn là việc biểu diễn câu truy vấn của người dùng thành đối tượng Query phù hợp để sau đó truyền vào hàm tìm kiếm của lucene. Lucene có thể tìm ra kết quả chỉ khi câu truy vấn truyền vào là đúng định dạng của nó.

Để thực hiện được việc chuyển đổi câu truy vấn của người dùng, QueryParser cần thêm một đối tượng khác gọi là bộ phân tích Analyzer. Tùy vào việc chọn lựa bộ Analyzer để phân tích chuỗi truyền vào thì kết quả sẽ khác nhau.

2.7. Các biểu thức truy vấn của QueryParser

Bảng 2.7.1: Bảng các biểu thức truy vấn của QueryParser

Biểu thức truy vấn	Tìm những tài liệu với biểu thức truy vấn
Quy	Tìm kiếm những tài liệu có chứa từ “Quy” trong trường mặc định tìm kiếm
Giáo dục (hoặc Giáo OR dục)	Tìm kiếm những tài liệu có chứa từ “Giáo” hoặc “dục”, hoặc tìm những tài liệu có chứa đồng thời cả hai từ “Giáo dục”
Giáo AND dục	Tìm kiếm những tài liệu vừa có từ “Giáo” và từ “dục”

BGD*	Tìm kiếm những tài liệu có chứa những từ bắt đầu bằng từ "BGD"

QueryParser sử dụng nhiều toán tử luận lí để thực hiện việc chuyển đổi câu truy vấn như: OR, AND, NOT. Mặc định là OR. Chẳng hạn câu truy vấn sau: abc xyz thì sẽ được phân tích thành là abc or xyz or (abc and xyz). Để thay đổi tham số mặc định này, ta cần đặt lại toán tử cho đối tượng QueryParser.

2.8. Bộ phân tích – Analyzer:

Trong Lucene, phân tích (*analysis*) là quá trình chuyển đổi các field văn bản về dạng trình bày chỉ mục cơ bản nhất (*term*). Các terms thì được sử dụng để xác định rõ tài liệu nào sẽ phù hợp với một câu truy vấn trong quá trình tìm kiếm. Bộ phân tích (*analyzer*) là cách nói tóm lược quá trình phân tích. Analyzer phân tích trong đoạn văn bản thành *tokenizes*, đó là quá trình rút trích các từ, bỏ đi hệ thống các dấu chấm câu, chuyển toàn bộ các chữ trong văn bản về dạng chữ thường (*lowercasing* hay còn gọi là *normalizing*), loại bỏ các từ chung (*common words hay stop words*), giảm số lượng từ từ văn bản đưa vào (*root form* hay còn gọi *stremming*). Quá trình này còn được gọi là *tokenization*,chuyển đoạn văn bản thành nhiều khúc văn bản được gọi là các *token*.Tokens được kết hợp với các field name của chúng được gọi là *terms*. Sau quá trình tạo ra terms, terms sẽ là những khối dữ liệu được dùng để tìm kiếm trực tiếp. Vì vậy chọn bộ phân tích đúng đắn là cốt yếu quan trọng của quá trình phát triển phần mềm tìm kiếm. Ngôn ngữ là một yếu tố phải được nghĩ đến để chọn bộ phân tích, bởi vì đều có đặc trưng riêng và duy nhất của từng ngôn ngữ.

2.9. Sử dụng lớp IndexSearcher

Sau khi tạo ra đối tượng IndexSearcher, ta sẽ gọi phương thức search để thực hiện việc tìm kiếm. Có ba phương thức chính để tìm kiếm. Song ta chủ yếu sử dụng phương thức search(Query), tức tham số là câu truy vấn Query. Các phương thức tìm kiếm đều trả về là các Hits –chứa các thông tin đã tìm kiếm được, kết quả được sắp xếp theo thứ tự độ chính xác. Thông qua đối tượng này ta có thể truy xuất thêm nhiều thông tin về kết quả tìm kiếm.

Mặc định 100 Documents sẽ tự động được khởi tạo ban đầu và sẽ được xử lý. Bộ Hits sẽ tự nó thêm vào khi người dùng truy vấn tới những tài liệu ở mức trên.

Phân trang kết quả tìm kiếm là điều hết sức cần thiết trong việc trình bày kết quả trả về. Có hai hướng cài đặt chính: Giữ đối tượng Hits và IndexSearcher trong khi người dùng chuyển kết quả tìm kiếm. Thực hiện truy vấn lại mỗi khi người dùng chuyển đến trang mới. Truy vấn lại thường được dùng hơn và là giải pháp tối ưu hơn. Việc này đòi hỏi phải lưu trữ trạng thái người dùng. Trong ứng dụng web, nơi người dùng gõ truy vấn ta cần lưu lại chuỗi truy vấn ban đầu, có thể lưu giữ trong các hidden field hoặc là cookie và sau mỗi lần truy vấn lại thì phải cập nhật lại câu truy vấn của người dùng.

Một điểm cần lưu ý là mặc dù việc tìm kiếm diễn ra trên thư mục chứa dữ liệu index, song để tăng tốc độ tìm kiếm nên nạp dữ liệu đọc được từ index và đẩy lên RAMDirectory.

2.10. Cú pháp truy vấn Lucene

Lucene có một cú pháp truy vấn tùy chỉnh để truy vấn các chỉ mục của nó. Trong hấu hết các ứng dụng ta sử dụng đối tượng QueryParser để chuyển đổi câu truy vấn theo từng loại thích hợp. Lucene cung cấp bốn loại Query: QueryParse, BooleanQuery, RangeQuery và TermQuery. Sau đây ta sẽ tìm hiểu từng loại Query và lúc nào QueryParse sẽ chuyển đổi câu truy vấn thành dạng nào [7].

- **Đối sánh từ khóa**

  Tìm từ "quyết" trong trường tiêu đề.

  Tiêu đề : quyết

  Tìm cụm từ "quyet dinh" trong trường tiêu đề.

  Tiêu đề : "quyết định"

  Tìm từ "quyết" và không chứa từ "định" trong trường tiêu đề.

  Title : quyết - title : định

- **Kết hợp ký tự đại diện**

  Tìm kiếm bất kỳ từ nào bắt đầu bằng "BGD" trong trường tiêu đề.

  Tiêu đề : BGD *

Tìm kiếm bất kỳ từ nào bắt đầu bằng "quy" và kết thúc bằng “định” trong trường tiêu đề.

Tiêu đề : quy * định

Lưu ý rằng Lucene không hỗ trợ sử dụng ký hiệu * làm ký tự đầu tiên của tìm kiếm.

- **Kết hợp gần**

Lucene hỗ trợ tìm các từ nằm trong một khoảng cách cụ thể.

Tìm kiếm "kỷ niệm" trong vòng 6 từ. "kỷ niệm" ~ 6

2.11. Các máy tìm kiếm phát triển dựa trên Lucene

- **Apache Solr:**

Solr là một máy chủ tìm kiếm mã nguồn mở phát triển dựa trên Apache Lucene có khả năng cung cấp các thư viện cho việc index (đánh chỉ mục) và search (tìm kiếm) dữ liệu. Solr nhập dữ liệu dưới dạng XML thông qua HTTP, hoặc sử dụng thư viện để nhập khối lượng lớn dữ liệu. Người dùng có thể truy vấn dữ liệu này thông qua HTTP GET và nhận về một kết quả dạng XML. Solr chạy bên trong một Java servlet container như Tomcat, Jetty hay Resin.

- **Elasticsearch:**

ElasticSearch là một máy tìm kiếm cấp doanh nghiệp (enterprise-level search engine). Mục tiêu của nó là tạo ra một công cụ, nền tảng hay kỹ thuật tìm kiếm và phân tích trong thời gian thực, có thể áp dụng hay triển khai một cách dễ dàng vào nguồn dữ liệu (data sources) khác nhau. ElasticSearch được phát triển bởi Shay Banon và dựa trên Apache Lucene, là một bản phân phối mã nguồn mở cho việc tìm kiếm dữ liệu trên máy chủ. Đây là một giải pháp mở rộng, hỗ trợ tìm kiếm thời gian thực mà không cần có một cấu hình đặc biệt. Nó đã được áp dụng bởi một số công ty, bao gồm cả StumbleUpon và Mozilla. ElasticSearch được phát hành theo Giấy phép Apache 2.0.

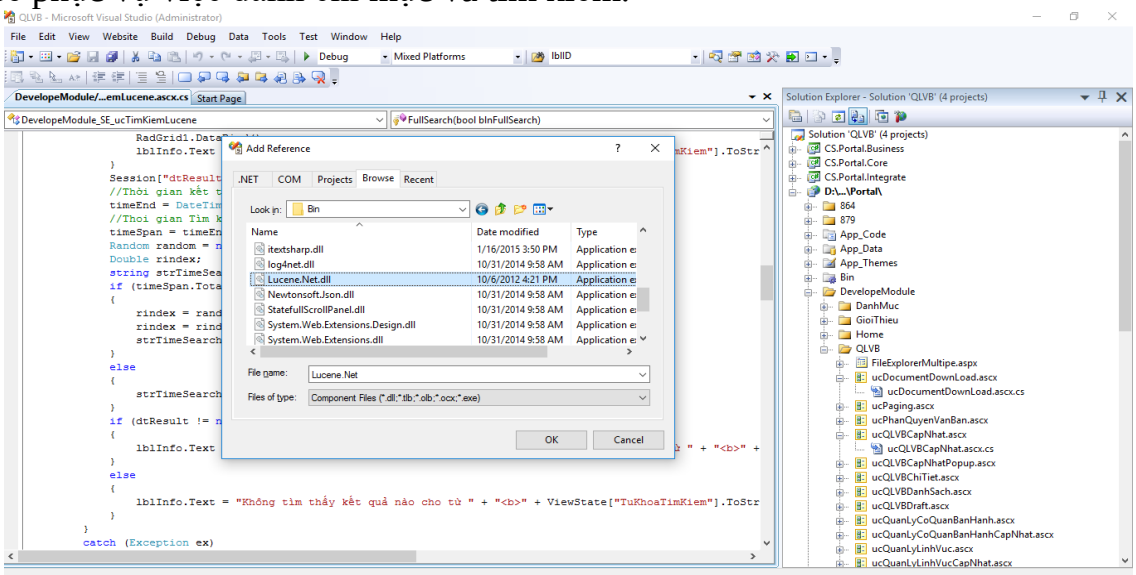
CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG THỬ NGHIỆM

Trên cơ sở nghiên cứu, chúng tôi đề xuất xây dựng ứng dụng thử nghiệm Lucene vào trong hệ thống tìm kiếm Văn bản. Trong đó, chúng tôi kế thừa mã nguồn mở Lucene.NET để xây dựng hệ thống tìm kiếm với hai thành phần chính là: Tạo chỉ mục và Tìm kiếm. Do chỉ xây dựng hệ thống tìm kiếm nhỏ với dữ liệu văn bản ít nên hệ thống không có thành phần Thu thập dữ liệu .

Lucene.Net là một thư viện để xây dựng công cụ tìm kiếm của Lucene, được viết bằng C# và nhắm mục tiêu đến người dùng chạy trên nền tảng .NET. Thư viện tìm kiếm Lucene.NET cũng dựa trên chỉ mục nghịch đảo để tìm kiếm. Lucene.NET ra đời với ba mục tiêu chính: Duy trì Lucene hiện tại theo dòng từ Java sang C#, tự động hóa đầy đủ và sắp xếp quá trình để dự án có thể dễ dàng đồng bộ hóa với lịch trình phát hành của Java Lucene. Duy trì các yêu cầu hiệu suất cao dự kiến của thư viện công cụ tìm kiếm C#. Tối đa hóa khả năng sử dụng cung cấp API rất tiện dụng và được thiết kế cẩn thận, tận dụng nhiều tính năng đặc biệt trên nền tảng .NET.

3.1. Tài mã nguồn Lucene.NET

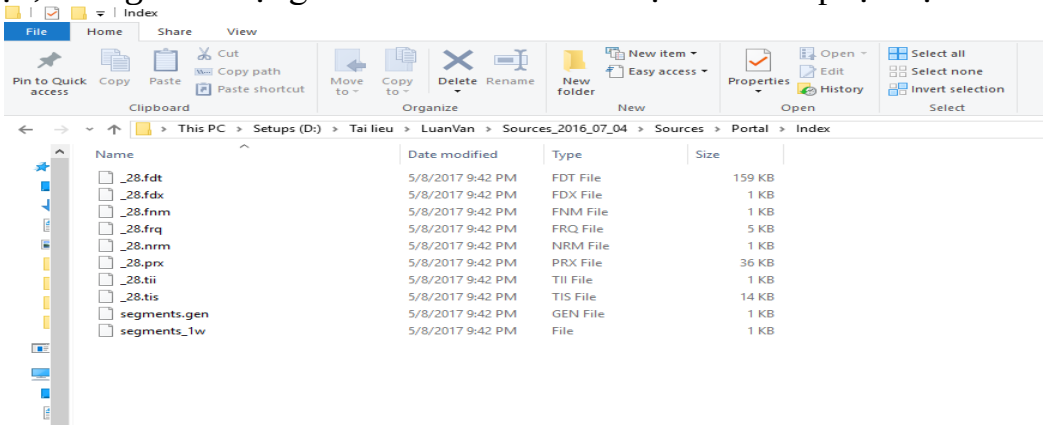
Trước tiên, chúng ta truy cập vào website https://lucenenet.apache.org/ để tải phiên bản mới nhất của Lucene. Sau khi tải về giải nén vào thư mục làm việc mà không cần cài đặt. Sau khi giải nén, kiểm tra lại để đảm bảo trong thư mục vừa giải nén có chứa thư viện Lucene.net.dll. Thư viện này sẽ được tích hợp vào trong công cụ phát triển phần mềm Microsoft Visual Studio. Ứng dụng thử nghiệm xây dựng các giao diện tương tác với người dùng, mã nguồn của các giao diện này là các dòng lệnh gọi các phương thức do bộ thư viện Lucene.NET cung cấp để phục vụ việc đánh chỉ mục và tìm kiếm.



Hình 3.1.1: Hình tích hợp thư viện mã nguồn mở Lucene.net

3.2. Dữ liệu văn bản thử nghiệm

Do không có thành phần thu thập dữ liệu nên dữ liệu phục vụ thử nghiệm sẽ được đưa vào hệ thống thông qua chức năng nhập dữ liệu văn bản của ứng dụng. Dữ liệu này được lưu trữ trong hệ quản trị CSDL SQL Server 2008. Sau khi có dữ liệu, chúng ta sử dụng Lucene để đánh chỉ mục văn bản phục vụ tìm kiếm.

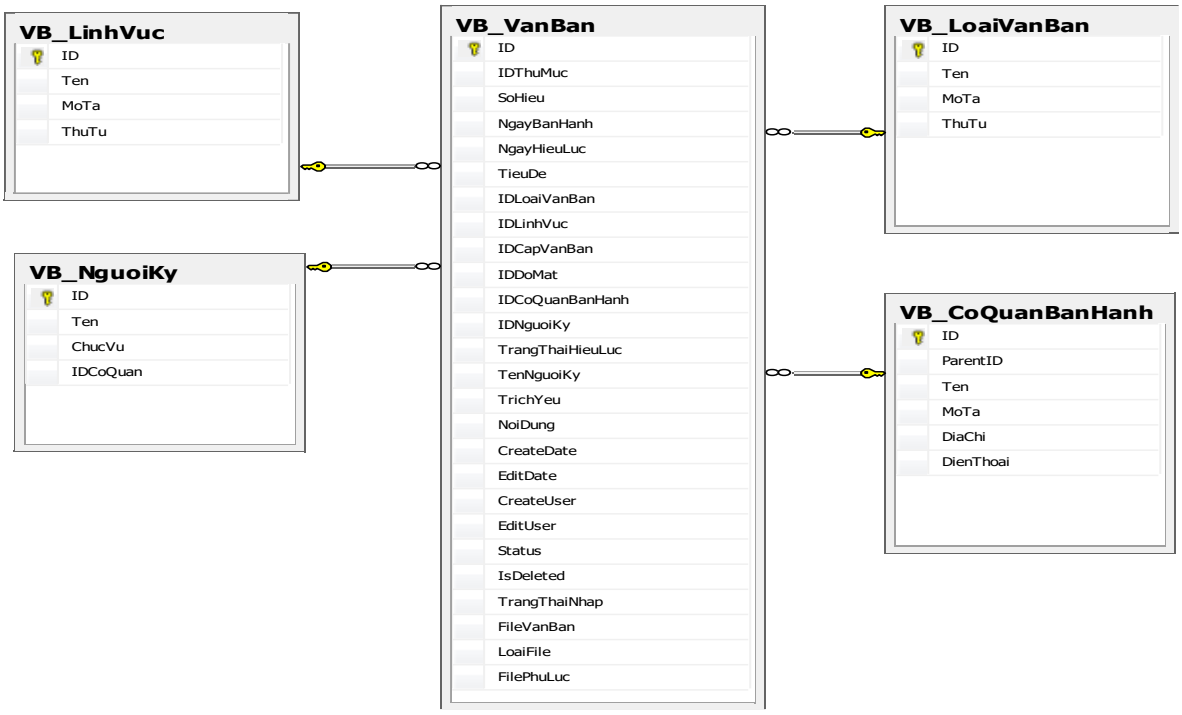


Hình 3.2.1: Hình Các tập tin kết xuất sau khi lập chỉ mục

3.3. Mô hình cơ sở dữ liệu

3.3.1. Lược đồ cơ sở dữ liệu

Lược đồ mô hình CSDL cung cấp một cách tổng quan về cấu trúc dữ liệu cũng như các mối quan hệ dữ liệu giữa các bảng trong CSDL. Mô tả chi tiết các trường dữ liệu, kiểu dữ liệu, khóa chính, khóa phụ trong các bảng. Đây là mô hình CSDL lưu trữ Văn bản phục vụ việc đánh chỉ mục và tìm kiếm của ứng dụng thử nghiệm.



Hình 3.3.1: Hình lược đồ cơ sở dữ liệu

3.3.2. Danh sách bảng

Bảng 3.2.2.1: Bảng danh sách các bảng trong CSDL

STT	Tên bảng	Thông tin mô tả
1	VB_LinhVuc	Bảng danh mục lĩnh vực văn bản
2	VB_NguoiKy	Bảng danh mục người ký
3	VB_VanBan	Bảng thông tin văn bản
4	VB_LoaiVanBan	Bảng danh mục loại văn bản
5	VB_CoQuanBanHanh	Bảng danh mục cơ quan ban hành

3.3.3. Mô tả bảng

3.3.3.1. Bảng VB_LinhVuc

Bảng 3.2.3.1: Bảng Lĩnh vực

STT	Mã trường	Kiểu trường	Độ lớn	Null	PK/FK	Giá trị mặc định	Mô tả
1	ID	Int	4	No	PK		Khóa chính
2	Ten	Nvarchar	200	No			Tên lĩnh vực
3	MoTa	Nvarchar	500	Yes			Mô tả
4	ThuTu	Int	4	Yes			Thứ tự

3.3.3.2. Bảng VB_NguoiKy

Bảng 3.3.3.2: Bảng Người ký

STT	Mã trường	Kiểu trường	Độ lớn	Null	PK/FK	Giá trị mặc định	Mô tả
1	ID	Int	4	No	PK		Khóa chính
2	Ten	Nvarchar	200	No			Tên người ký
3	ChucVu	Nvarchar	500	Yes			Chức vụ người ký

3.3.3.3. Bảng VB_VanBan

Bảng 3.3.3.3: Bảng Văn bản

STT	Mã trường	Kiểu trường	Độ lớn	Null	PK/FK	Giá trị mặc định	Mô tả
1	ID	Int	4	No	PK		Khóa chính
2	SoHieu	Nvarchar	500	No			Số hiệu
3	NgayBanHanh	DateTime	8	No			Ngày ban hành
4	NgayHieuLuc	DateTime	8	Yes			Ngày hiệu lực
5	TieuDe	Nvarchar	500	No			Tiêu đề văn bản
6	IDLoaiVanBan	Int	4	No	FK		Mã loại văn bản
7	IDLinhVuc	Int	4	No	FK		Mã lĩnh vực
8	IDCoQuanBanHanh	Int	4	No	FK		Mã cơ quan ban hành văn bản

9	IDNguoiKy	Int	4	No	FK		Mã người ký
10	TrichYeu	Nvarchar	500	No			Trích yếu
11	NoiDung	Ntext	16	Yes			Nội dung
12	FileVanBan	Nvarchar	200	Yes			File nội dung
13	FilePhuLuc	Nvarchar	200	Yes			File phụ lục
14	NgayTao	DateTime	8	Yes			Ngày tạo
15	IDNguoiTao	Int	4	No			Mã người tạo

3.3.3.4. Bảng VB_LoaiVanBan

Bảng 3.3.3.4: Bảng Loại văn bản

STT	Mã trường	Kiểu trường	Độ lớn	Null	PK/FK	Giá trị mặc định	Mô tả
1	ID	Int	4	No	PK		Khóa chính
2	Ten	Nvarchar	200	No			Tên loại văn bản
3	MoTa	Nvarchar	500	Yes			Mô tả loại văn bản
4	ThuTu	Int	4	Yes			Thứ tự

3.3.3.5. Bảng VB_CoQuanBanHanh

Bảng 3.3.3.5: Bảng Cơ quan ban hành

STT	Mã trường	Kiểu trường	Độ lớn	Null	PK/FK	Giá trị mặc định	Mô tả
1	ID	Int	4	No	PK		Khóa chính
2	ParentID	Int	4	No			Mã cha
3	Ten	Nvarchar	200	No			Tên cơ quan
4	MoTa	Nvarchar	500	Yes			Mô tả cơ quan
5	DiaChi	Nvarchar	500	Yes			Địa chỉ cơ quan
6	DienThoai	Nvarchar	20	Yes			Điện thoại

3.4. Giao diện chính

3.4.1. Giao diện trang Quản lý lĩnh vực

Giao diện này cho phép người dùng có thể quản lý các danh mục lĩnh vực văn bản. Thêm mới lĩnh vực văn bản, xóa hoặc cập nhật các lĩnh vực văn bản đã có trong hệ thống.

Hình 3.4.1: Giao diện trang Quản lý lĩnh vực

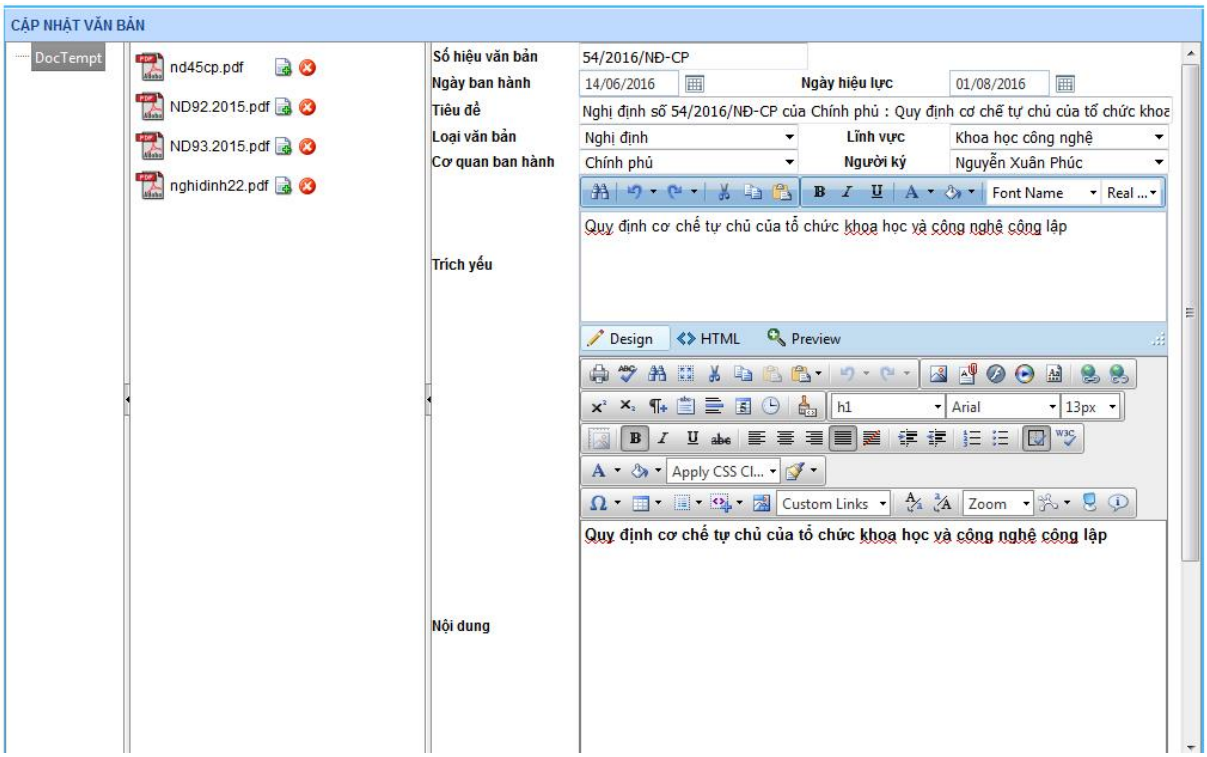
3.4.2. Giao diện trang Quản lý văn bản

Giao diện này cho phép người dùng thêm mới, cập nhật hoặc xóa các văn bản đã có trong hệ thống. Đây chính là chức năng tạo nên Cơ sở dữ liệu văn bản phục vụ việc đánh chỉ mục để tạo ra cơ sở dữ liệu Lucene. Hệ thống cho phép đánh chỉ mục toàn bộ văn bản đã lưu trữ trong CSDL, hoặc chỉ đánh chỉ mục những văn bản vừa được thêm mới, cập nhật hoặc xóa mà không cần phải đánh chỉ mục lại từ đầu.

Hình 3.4.2: Giao diện trang Quản lý văn bản

3.4.3. Giao diện trang Cập nhật văn bản

Chức năng này cho phép người dùng thực hiện thêm mới hoặc cập nhật văn bản đã có trong hệ thống. Thông tin về văn bản bao gồm các trường thông tin như: Số hiệu văn bản, ngày ban hành, ngày hiệu lực, trích yếu, nội dung văn bản…



Hình 3.4.3: Giao diện trang Cập nhật văn bản

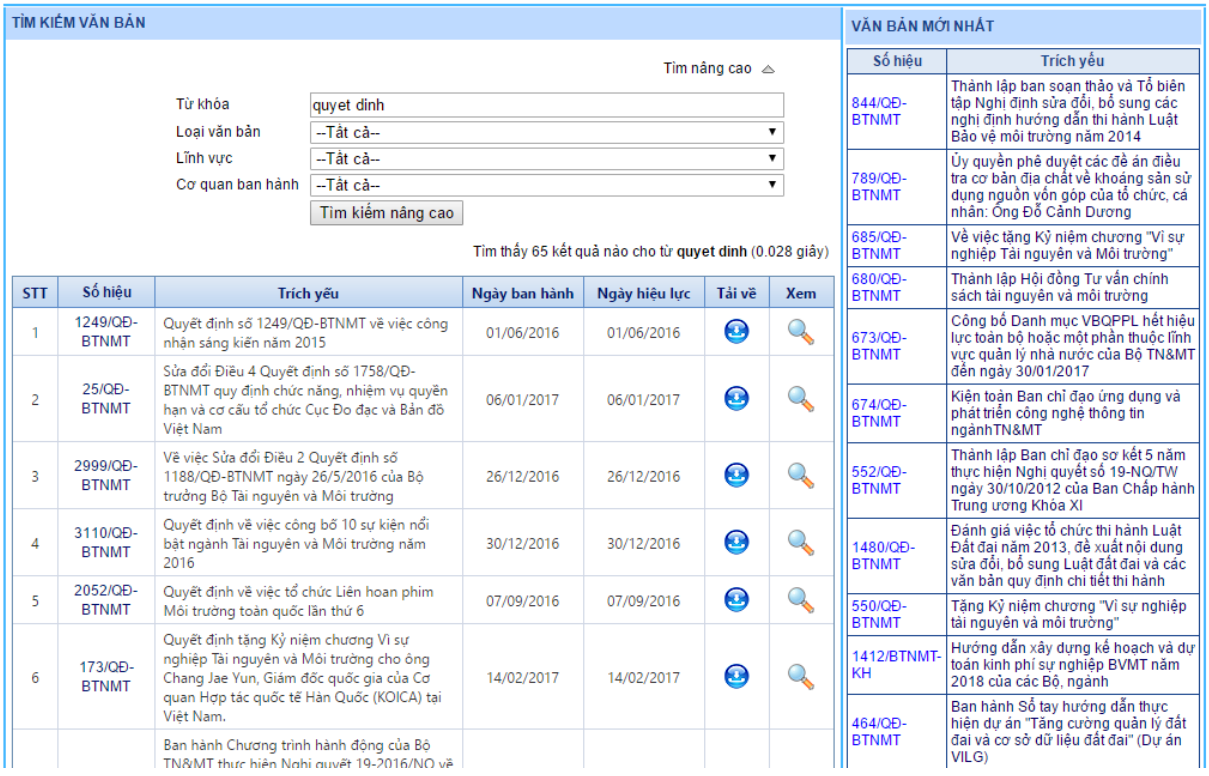
3.4.4. Giao diện trang Tìm kiếm văn bản

Giao diện này cho phép người dùng gõ các từ khóa tìm kiếm, hệ thống thực hiện tìm kiếm các từ khóa trong file chỉ mục, sắp xếp các kết quả và trả về danh sách các kết quả theo mức độ liên quan giữa câu truy vấn và các tài liệu trong cơ sở dữ liệu chỉ mục. Chức năng này cho phép người dùng gõ trực tiếp từ khóa kết hợp với các phép toán mà Lucene hỗ trợ (AND, OR, NOT…), tìm kiếm chính xác, tìm kiếm gần đúng hoặc tìm với các ký tự đại diện.

Hình 3.4.4: Giao diện trang Tìm kiếm văn bản

3.4.5. Giao diện trang Tìm kiếm nâng cao văn bản

Ngoài chức năng tìm kiếm tổng quát thì hệ thống còn cho phép người dùng tra cứu, tìm kiếm nâng cao văn bản theo các tiêu chí tìm kiếm như: Tìm theo thể loại văn bản, lĩnh vực văn bản hoặc tìm theo cơ quan ban hành văn bản.



Hình 3.4.5: Giao diện trang Tìm kiếm nâng cao văn bản

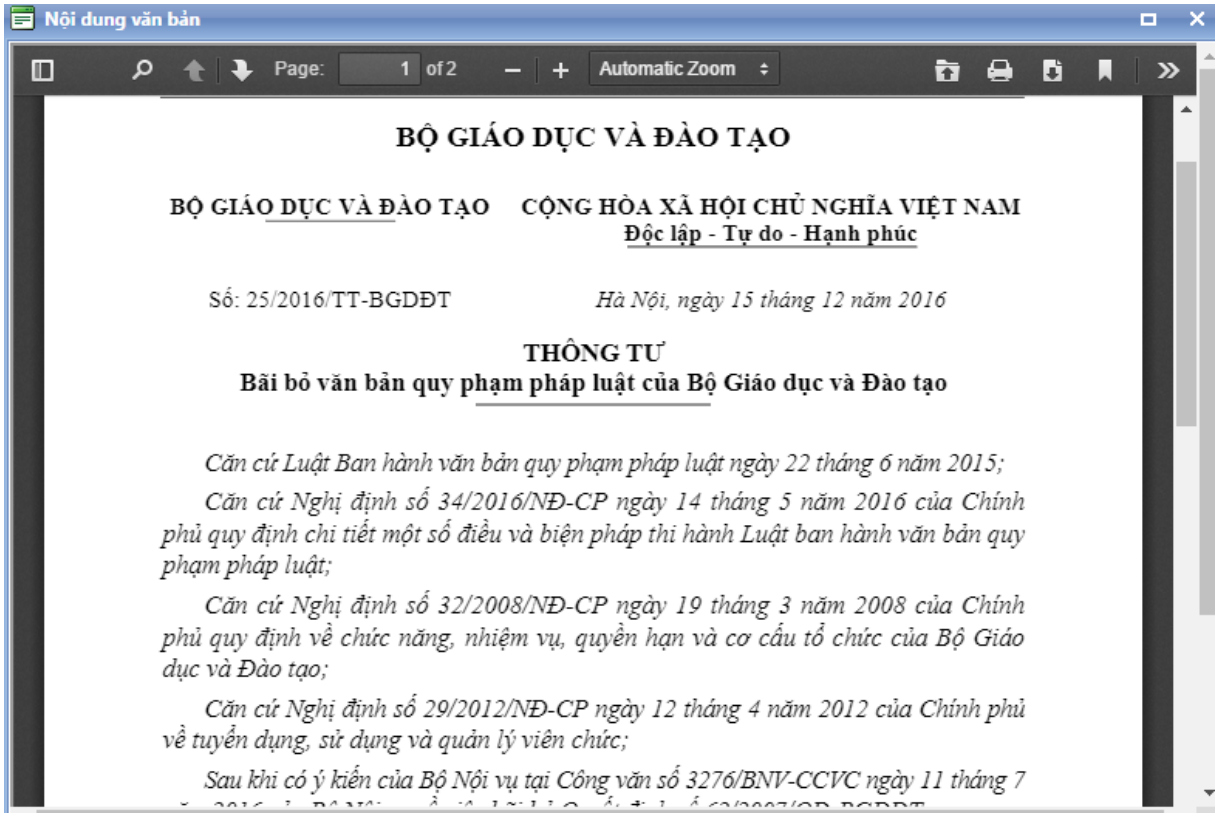
3.4.6. Giao diện trang Xem chi tiết văn bản

Sau khi người dùng tra cứu tìm kiếm văn bản, hệ thống hiển thị danh sách các văn bản theo các tiêu chí đã tìm kiếm. Tại danh sách văn bản ở kết quả tìm kiếm, người dùng click chọn một văn bản để xem thông tin chi tiết về văn bản, tải văn bản hoặc xem nội dung gốc của văn bản.

Hình 3.4.6: Giao diện trang Xem chi tiết văn bản

3.4.7. Giao diện trang Xem nội dung file văn bản

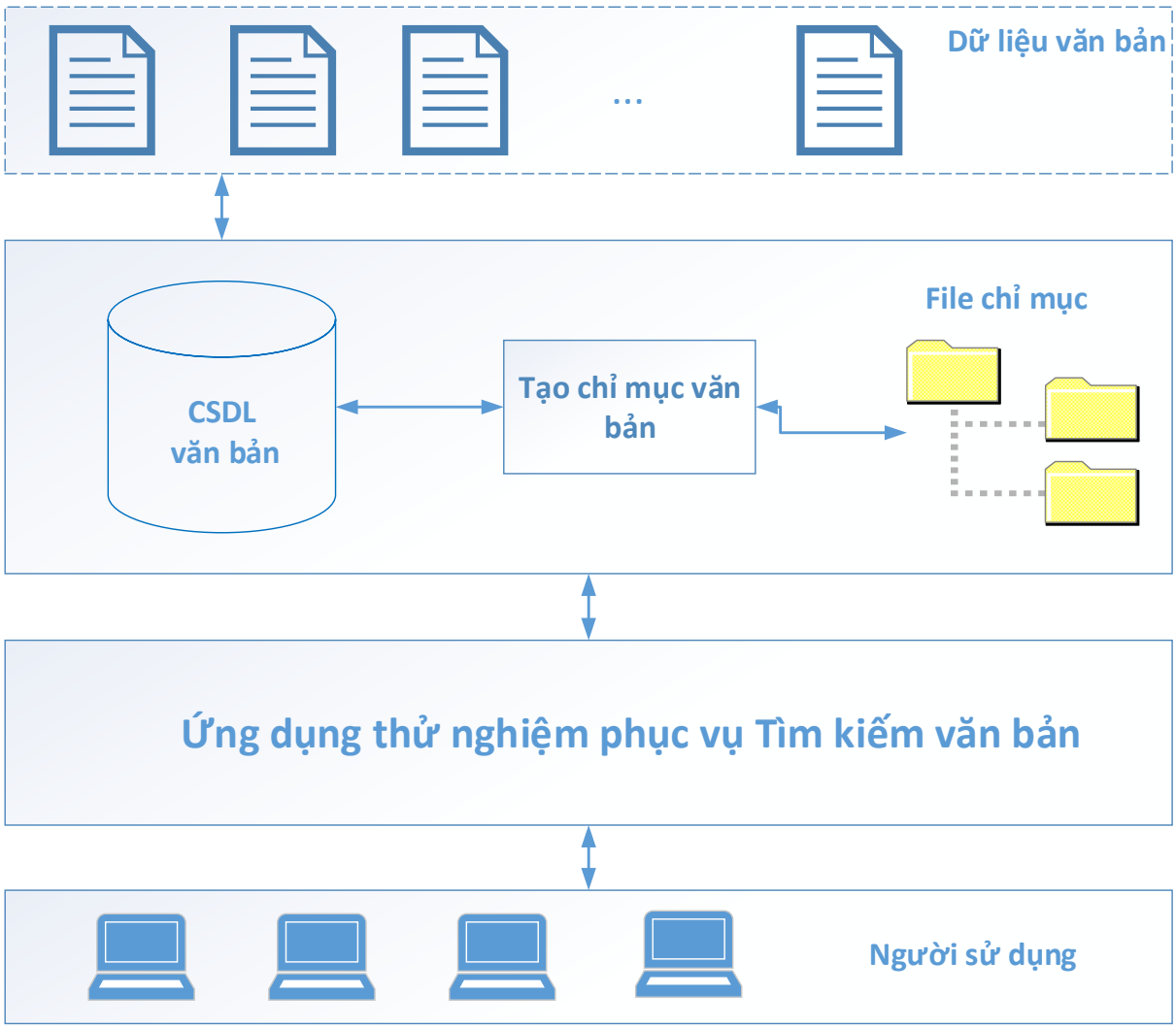
Với các văn bản đã được đưa vào hệ thống, ngoài việc xem thông tin chi tiết văn bản thì người dùng có thể xem trực tiếp nội dung của văn bản đã được lưu trữ trong file pdf. Đây chính là nội dung toàn văn bản cần được đưa vào hệ thống để đánh chỉ mục phục vụ chức năng tra cứu, tìm kiếm văn bản.



Hình 3.4.7: Giao diện trang Xem nội dung file văn bản

3.5. Đánh giá và thử nghiệm

3.5.1. Mô hình kiến trúc ứng dụng thử nghiệm



Hình 3.5.1: Hình Kiến trúc ứng dụng thử nghiệm

Hệ thống thử nghiệm có 3 thành phần chính:

- **Dữ liệu văn bản thử nghiệm**: Tất cả các văn bản được đưa vào hệ thống và lưu trữ trong hệ Quản trị CSDL SQL Server 2008 gọi là cơ sở dữ liệu Văn bản. Với mô hình như trên chúng tôi đã thử nghiệm với số lượng khoảng hơn 300 văn bản.
- **Tạo chỉ mục văn bản**: Từ những dữ liệu văn bản đã được đưa vào hệ thống, chúng tôi xây dựng chức năng đánh chỉ mục cho văn bản, chức năng này sẽ tạo ra các file chỉ mục lưu trữ trên đĩa cứng máy tính, đây chính là cơ sở dữ liệu chỉ mục hay cơ sở dữ liệu Lucene.

- **Tìm kiếm văn bản:** Người dùng truy cập vào đường dẫn phần mềm thử nghiệm và gõ từ khóa tìm kiếm văn bản. Hệ thống sẽ thực hiện tìm kiếm ở trong cơ sở dữ liệu Lucene hay tìm trong các file chỉ mục và trả về kết quả cho người dùng**.** Từ danh sách kết quả tìm kiếm người dùng có thể sắp xếp tăng dần, giảm dần theo số hiệu văn bản, ngày ban hành, ngày hiệu lực. Hoặc click vào một văn bản để xem thông tin chi tiết.

Công cụ phát triển ứng dụng:

- Bộ thư viện mã nguồn mở Lucene.NET phiên bản 3.0
- Visual Studio 2008, ngôn ngữ C#
- Microsoft SQL Server 2008 R2

3.5.2. Kịch bản và kết quả

Với dữ liệu văn bản đã được đánh chỉ mục, chúng tôi đã thử nghiệm sử dụng chức năng tìm kiếm văn bản với các kịch bản như sau:

- Tìm kiếm theo từ khóa bất kỳ
- Tìm kiếm theo phép toán AND
- Tìm kiếm theo phép toán OR
- Tìm kiếm chính xác từ khóa
- Tìm kiếm từ khóa là tiếng việt không có dấu (tìm kiếm gần đúng)
- Tìm kiếm với ký tự đại diện
- Tìm kiếm gợi ý từ khóa (autocomplete)

Hệ thống đã thực hiện thành công trên máy tính cá nhân với cấu hình Intel Core i5-3210M, CPU 2.5GHz, 8GB RAM với kết quả như sau:

Bảng 3.5.2.1: Bảng Kịch bản tìm kiếm của Hệ thống tìm kiếm thông thường

Kịch bản tìm kiếm	Từ khóa tìm kiếm	Số lượng kết quả	Thời gian tìm kiếm (giây)
Tìm kiếm theo từ khóa bất kỳ	quy	67	1.25
Tìm kiếm theo từ khóa bất kỳ	Ban hành	34	1.76
Tìm kiếm theo phép toán AND	Môi AND trường	0	0.84

Tìm kiếm theo phép toán OR	Môi OR trường	0	0.74
Tìm kiếm chính xác từ khóa	“quy chế”	0	1.026
Tìm kiếm từ khóa là tiếng việt không có dấu (tìm kiếm gần đúng)	Quy che	0	0.311
Tìm kiếm với ký tự đại diện	BGD*	0	0.24
Tìm kiếm gợi ý từ khóa (autocomplete)	Ban hành Quy chế công tác sinh viên đối với chương trình đào tạo đại học hệ chính quy	1	0.53

Bảng 3.5.2.2: Bảng Kịch bản tìm kiếm của Hệ thống tìm kiếm thông tin

Kịch bản tìm kiếm	**Từ khóa tìm kiếm**	**Số lượng kết quả**	**Thời gian tìm kiếm (giây)**
Tìm kiếm theo từ khóa bất kỳ	quy	45	0.027
Tìm kiếm theo từ khóa bất kỳ	Ban hành	76	0.016
Tìm kiếm theo phép toán AND	Môi AND trường	117	0.041
Tìm kiếm theo phép toán OR	Môi OR trường	130	0.023
Tìm kiếm chính xác từ khóa	“quy chế”	17	0.012
Tìm kiếm từ khóa là tiếng việt không có dấu (tìm kiếm gần đúng)	Quy che	48	0.017

Tìm kiếm với ký tự đại diện	BGD*	4	0.029
Tìm kiếm gợi ý từ khóa (autocomplete)	Ban hành Quy chế công tác sinh viên đối với chương trình đào tạo đại học hệ chính quy	251	0.023

Qua kết quả thống kê trên cho thấy chức năng tìm kiếm của hệ thống cơ sở dư liệu thông thường không hỗ trợ các toán tử tìm kiếm, không hỗ trợ tìm kiếm chính xác trong dấu "" và tìm kiếm tiếng việt không có dấu. Đặc biệt tốc độ tìm kiếm của hệ thống tìm kiếm thông tin nhanh hơn rất nhiều so với chức năng tìm kiếm của hệ quản trị cơ sở dữ liệu thông thường. Kết quả thử nghiệm tìm kiếm trên hệ thống tìm kiếm thông tin tương đối chính xác và hiệu quả, hỗ trợ đầy đủ các phép toán tìm kiếm với thời gian tìm kiếm rất nhanh. Như vậy hệ thống thử nghiệm tìm kiếm văn bản sử dụng mã nguồn mở Lucene đáp ứng được mục tiêu đặt ra của đề tài.

CHƯƠNG 4: KẾT LUẬN

Với giải pháp nâng cao hiệu quả của việc tra cứu, tìm kiếm dữ liệu bằng cách nghiên cứu bộ thư viện mã nguồn mở Lucene để xây dựng thử nghiệm Hệ thống tìm kiếm thông tin trên các văn bản đã được lưu trữ trong kho dữ liệu. Với thuận lợi lớn là hệ thống đã kế thừa toàn bộ các chức năng từ bộ thư viện mã nguồn mở Lucene.NET.

Về lý thuyết, luận văn tìm hiểu về các thành phần cơ bản của một hệ thống tìm kiếm thông tin bao gồm: Thành phần Thu thập dữ liệu: thực hiện thu thập toàn bộ dữ liệu sẽ tìm kiếm đưa về một nguồn tập trung để phục vụ quá trình phân tích và đánh chỉ mục dữ liệu. Thành phần Đánh chỉ mục dữ liệu: thực hiện phân tích, tiền xử lý nội dung dữ liệu, sau đó tiến hành đánh chỉ mục dữ liệu theo cách thức, cơ chế và yêu cầu của từng máy tìm kiếm cụ thể. Thành phần Tìm kiếm dữ liệu: thực hiện phân tích câu truy vấn và tìm kiếm tài liệu trên các file index, sau đó kết hợp với thông tin xếp hạng để trả lại kết quả tìm kiếm cho người dùng.

Luận văn cũng tìm hiểu một cách hệ thống các tính năng và hoạt động của mã nguồn mở Lucene như: Lucene cung cấp khả năng phân tích dữ liệu, tạo chỉ mục cho các tài liệu để xây dựng nên hệ thống chỉ mục, cung cấp khả năng tiếp nhận các xâu truy vấn của người dùng, thực hiện tìm kiếm dựa trên hệ thống chỉ mục đã có và trả về kết quả.

Thực nghiệm, từ cơ sở lý thuyết, luận văn đã xây dựng và cài đặt thành công ứng dụng thực nghiệm Lucene vào trong hệ thống tìm kiếm Văn bản. Trong đó, ứng dụng bộ thư thư viện mã nguồn mở Lucene.NET để xây dựng hệ thống tìm kiếm với hai thành phần chính là: Tạo chỉ mục và Tìm kiếm văn bản. Hệ thống được kết thừa toàn bộ thư viện mã nguồn mở Lucene.NET nên tính hiệu quả rất lớn và không mất chi phí bản quyền sử dụng.

4.1. Đánh giá kết quả nghiên cứu

4.1.1. Kết quả đạt được:

Về cơ bản luận văn đã thực hiện tốt các nội dung đề ra và đạt được một số kết quả nhất định: Luận văn đã trình bày cơ sở lý thuyết và nguyên lý vận hành của một hệ thống tìm kiếm thông tin, trình bày một cách hệ thống các tính năng và hoạt động của mã nguồn mở Lucene. Luận văn đã ứng dụng thành công mã nguồn mở Lucene trong công tác tìm kiếm thông tin trên hệ thống Quản lý văn bản.

4.1.2. Hạn chế:

Bên cạnh những kết quả đạt được thì đề tài còn có những mặt hạn chế như: Phần thực nghiệm mới chỉ dừng lại ở phạm vi nhỏ với số lượng văn bản còn hạn chế. Đề tài chưa nghiên cứu được các bộ thư viện mã nguồn mở khác giống thư viện Lucene, chưa nghiên cứu một số tính năng liên quan đến tìm kiếm tiếng Việt. Đề tài cần nâng cao hiệu quả tìm kiếm cũng như tính tiện dụng cho người sử dụng, giao diện hiển thị tốt trên mọi nền tảng thiết bị như Desktop, laptop, tablet, mobile…

4.2. Hướng phát triển

Hướng nghiên cứu tiếp theo của đề tài là tập trung nghiên cứu tìm hiểu các mã nguồn mở khác giống thư viện mã nguồn mở Lucene, để có thể áp dụng thử nghiệm và đưa ra được những nhận xét, đánh giá so sánh hiệu quả với thư viện Lucene. Nghiên cứu, xây dựng các module chức năng khác để hoàn thiện hệ thống Quản lý văn bản. Ngoài ra cần nghiên cứu một số tính năng xử lý nâng cao cho việc tìm kiếm tiếng Việt áp dụng vào hệ thống Quản lý văn bản.

Tôi nhận thấy rằng, với việc phát triển nhanh chóng của công nghệ thông tin cũng như nhu cầu tìm kiếm thông tin của người dùng ngày một nhiều thì sẽ có rất nhiều hệ thống tìm kiếm thông tin ứng dụng mã nguồn mở Lucene ra đời.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Đỗ Phúc, Đỗ Hoàng Cường, Nguyễn Tri Tuấn, Huỳnh Thụy Bảo Trân, Nguyễn Văn Khiết, Nguyễn Việt Hoàng, Nguyễn Việt Thành, Phạm Phú Hội, Dương Ngọc Long Nam, Nguyễn Phước Thanh Hải, "Phát triển một Hệ thống S.E" Hỗ trợ Tìm kiếm Thông tin, thuộc lãnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt", Đại học Khoa Học Tự Nhiên, TP.HCM, 2004
2. Huỳnh Đức Việt, Võ Duy Thanh, Võ Trung Hùng, "Tạp chí khoa học và công nghệ", Đại học Đà Nẵng, 2010

Tiếng Anh

3. Michael McCandless, Erik Hatcher, Otis Gospodnetic, Lucene in action, 2010
4. Haralambos Marmanis and Dmitry Babenko, Algorithms of the Intelligent Web, 2009
5. Chris Manning and Pandu Nayak, Introduction to Information Retrieval
6. http://infolab.stanford.edu/~backrub/google.html
7. http://www.lucenetutorial.com
8. https://www.tutorialspoint.com
9. https://lucenenet.apache.org/
10. https://en.wikipedia.org/wiki/Lucene