BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ QUỐC PHÒNG
HỌC VIỆN KỸ THUẬT QUÂN SỰ

PHÙNG DUY VŨ

NGHIÊN CỨU XÂY DỰNG MÁY TÌM KIẾM

Chuyên ngành: Hệ thống thông tin

LUẬN VĂN THẠC SĨ KỸ THUẬT

Hà Nội - Năm 2014

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ QUỐC PHÒNG
HỌC VIỆN KỸ THUẬT QUÂN SỰ

PHÙNG DUY VŨ

NGHIÊN CỨU XÂY DỰNG MÁY TÌM KIẾM

Chuyên ngành: Hệ thống thông tin

Mã số: 60 48 01 04

LUẬN VĂN THẠC SĨ KỸ THUẬT

Hà Nội - Năm 2014

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI

HỌC VIỆN KỸTHUẬT QUÂN SỰ

Cán bộ hướng dẫn chính: TS Phạm Văn Việt

Cán bộ chấm phản biện 1:.....

Cán bộ chấm phản biện 2:.....

Luận văn thạc sĩ được bảo vệ tại:

HỘI ĐỒNG CHẤM LUẬN VĂN THẠC SĨ

HỌC VIỆN KỸTHUẬT QUÂN SỰ

Ngày ... tháng ... năm 2014

Tôi xin cam đoan:

Những kết quả nghiên cứu được trình bày trong luận văn là hoàn toàn trung thực, của tôi, không vi phạm bất cứ điều gì trong luật sở hữu trí tuệ và pháp luật Việt Nam. Nếu sai tôi hoàn toàn chịu trách nhiệm trước pháp luật.

TÁC GIẢ LUẬN VĂN

Phùng Duy Vũ

MỤC LỤC

Trang

Chương 2

XÂY DỰNG ỨNG DỤNG MÁY TÌM KIẾM

Chương 3

THỰC NGHIỆM

KẾT LUẬN VÀ KHUYẾN NGHỊ

Tóm tắt luận văn

Họ và tên học viên: **Phùng Duy Vũ**

Chuyên ngành: Hệ thống thông tinKhóa 24

Cán bộ hướng dẫn: TS Phạm Văn Việt

Tên đề tài: **Nghiên cứu xây dựng máy tìm kiếm.**

Tóm tắt: nghiên cứu phương pháp thu thập các url từ một web tin tức, phương pháp duyệt url, phương pháp lập index cho nội dung url và phương pháp sắp xếp kết quả tìm kiếm để trả về kết quả mà người dùng muốn tìm.

DANH MỤC CÁC TỪ VIẾT TẮT

STT	KÝ HIỆU VIẾT TẮT	NGHĨA CỦA KÝ HIỆU VIẾT TẮT
1	CSDL	Cơ sở dữ liệu
2	FIFO	First In First Out là vào trước ra trước
3	HTML	HyperText Markup Language là ngôn ngữ đánh dấu siêu văn bản
4	HTTP	Hypertext Transfer Protocol là giao thức truyền tải siêu văn bản
5	PHP	Hypertext Preprocessor là ngôn ngữ lập trình web
6	URL	Uniform Resource Locator là địa chỉ web
7	WWW	World Wide Web

DANH MỤC HÌNH ẢNH

Trang

MỞ ĐẦU

1.Tính cấp thiết của đề tài

Trong thời đại mà mỗi ngày, có không biết bao nhiêu là thông tin mới được tạo ra khắp mọi nơi trên thế giới, các thông tin như: kinh tế, khoa học, kỹ thuật, tài chính,… ai cũng muốn khi lướt web, thì chỉ cần 1 cú click là có được các thông tin mà mình cần, có rất nhiều cỗ máy tìm kiếm trên thế giới làm được điều đó, như là : Google, yahoo, bing,…

Lĩnh vực tìm kiếm thật sự là mảnh đất màu mỡ, việc Google trở thành 1 công ty hàng đầu trên thế giới là nhờ vào cỗ máy tìm kiếm của mình đã chứng mình điều đó, nên việc nghiên cứu về lĩnh vực tìm kiếm không bao giờ là dư thừa.

Làm thế nào để cung cấp thông tin cho người dùng?

Làm thế nào để lưu trữ các thông tin đó?

Làm thế nào để hiểu được người dùng muốn tìm kiếm điều gì?

Làm thế nào để tạo ra một máy tìm kiếm?

…

Rất nhiều câu hỏi được đặt ra khi nghĩ về lĩnh vực này và vì vậy tôi chọn đề tài : "NGHIÊN CỨU XÂY DỰNG MÁY TÌM KIẾM" làm đề tài luận văn tốt nghiệp cao học của mình.

2.Tình hình nghiên cứu

Nghiên cứu về lĩnh vực tìm kiếm hiện nay cũng khá là nhiều, nhưng viết ứng dụng máy tìm kiếm là chưa nhiều.

Hiện nay, có một số tác giả đã và đang nghiên cứu về lĩnh vực tìm kiếm gồm các luận án, luận văn thạc sĩ, một số ít bài báo được công bố.

Về luận văn có các công trình sau:

Nguyễn Quang Huy công bố năm 2010 tại Học viện Công nghệ Bưu chính Viễn thông, với nội dung: Nghiên cứu search engine và các thuật toán đối sánh mẫu cho hệ thống tìm kiếm thông tin trên mạng.

Dương Đình Thiện công bố năm 2012 tại Đại học Đà Nẵng, với nội dung: Nghiên cứu các kỹ thuật xây dựng bộ tìm kiếm.

Nguyễn Huy Kiên công bố năm 2012 tại Học viện Công Nghệ Bưu chính Viễn Thông, với nội dung: Nghiên cứu về khai phá dữ liệu web và ứng dụng xây dựng website tích hợp thông tin.

Về bài báo có các công trình sau:

Huỳnh Đức Việt, Võ Duy Thanh, Võ Trung Hùng công bố năm 2010 trên Tạp chí khoa học và công nghệ, Đại học Đà Nẵng số 4 với tiêu đề "Nghiên cứu ứng dụng mã nguồn mở Lucene để xây dựng phần mềm tìm kiếm thông tin trên văn bản".

Như vậy, đề tài "NGHIÊN CỨU XÂY DỰNG MÁY TÌM KIẾM" là hoàn toàn mới, không trùng với các đề tài nghiên cứu trước đó.

3.Mục tiêu của đề tài

Nghiên cứu các phương phápvà kỹ thuật xây dựng máy tìm kiếm. Từ đó, xây dựng một ứng dụng thử nghiệm về máy tìm kiếm để làm cơ sở cho những nghiên cứu tiếp theo.

4.Phương pháp nghiên cứu

Nghiên cứu tổng quan: các phương pháp thu thập thông tin trên web, nghiên cứu phương pháp lưu trữ, quản lí CSDL, phương pháp lập chỉ mục, thuật toán tìm kiếm của máy tìm kiếm.

Thử nghiệm: thử nghiệm các phương pháp thu thập, lưu trữ và tìm kiếm thông tin trên website http://vnexpress.net.

Phân tích : phân tích các điểm mạnh điểm yếu của từng phương pháp, nhằm tìm ra phương án hiệu quả nhất.

Đánh giá: đánh giá máy tìm kiếm được xây dựng.

5.Nội dung nghiên cứu

Nghiên cứu thuật toán thu thập thông tin.

Nghiên cứu thuật toán tìm kiếm theo chiều sâu, chiều rộng, ngẫu nhiên.

Nghiên cứu phương pháp xây dựng index, xử lý từ khóa, cách thức lưu trữ từ khóa.

Nghiên cứu thuật toán sắp xếp kết quả tìm kiếm.

6.Ý nghĩa của đề tài

Thông qua đề tài, tôi muốn đóng góp một phần nghiên cứu của mình vào lĩnh vực tìm kiếm, hi vọng rằng từ đó, có thể ứng dụng vào những lĩnh vực khác trong cuộc sống.

7.Cấu trúc luận văn

Ngoài lời giới thiệu tham khảo, phụ lục thì luận văn nghiên cứu gồm 3 chương:

CHƯƠNG 1: TÌM HIỂU CÁC KIẾN THỨC TỔNG QUAN.

CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG MÁY TÌM KIẾM.

CHƯƠNG 3 : THỰC NGHIỆM

Do những hiểu biết của bản thân còn hạn chế, nên bài viết của tôi còn nhiều thiếu sót, rất mong nhận được sự đóng góp ý kiến của Quý Thầy Cô và những người quan tâm đến đề tài này.

Tôi xin chân thành cảm ơn.

Chương 1

TÌM HIỂU CÁC KIẾN THỨC TỔNG QUAN

1.1. Tìm hiểu về máy tìm kiếm

1.1.1. World Wide Web

Hình 1. 1 - Logo WWW được tạo bởi Robert Cailliau năm 1990

World Wide Web, gọi tắt là Web hoặc WWW, mạng lưới toàn cầu là một không gian thông tin toàn cầu mà mọi người có thể truy nhập (đọc và viết) qua các máy tính nối mạng Internet. Thuật ngữ này thường được hiểu nhầm là từ đồng nghĩa với chính thuật ngữ Internet. Nhưng Web thật ra chỉ là một trong các dịch vụ chạy trên Internet, chẳng hạn như thư điện tử. Web được phát minh và đưa vào sử dụng vào khoảng năm 1990, 1991 bởi viện sĩ Viện Hàn lâm Anh Tim Berners-Lee và Robert Cailliau (Bỉ) tại CERN, Geneva, Thụy Sĩ.

Các tài liệu trên World Wide Web được lưu trữ trong một hệ thống siêu văn bản (hypertext), đặt tại các máy tính trong mạng Internet. Người dùng phải sử dụng một chương trình được gọi là trình duyệt web (web browser) để xem siêu văn bản. Chương trình này sẽ nhận thông tin tại ô địa chỉ (address) do người sử dụng yêu cầu (thông tin trong ô địa chỉ được gọi là tên miền

(domain name), rồi sau đó chương trình sẽ tự động gửi thông tin đế máy chủ (web server) và hiển thị trên màn hình máy tính của người xem. Người dùng có thể theo các liên kết siêu văn bản (hyperlink) trên mỗi trang web để nối với các tài liêu khác hoặc gửi thông tin phản hồi theo máy chủ trong một quá trình tương tác. Hoạt động truy tìm theo các siêu liên kết thường được gọi là duyệt Web.

Quá trình này cho người dùng có thể lướt các trang web để lấy thông tin. Tuy nhiên độ chính và chứng thực của thông tin không được đảm bảo.

1.1.2. Thế giới Web rộng lớn như thế nào ?

Câu hỏi đặt ra là thế giới Web rộng lớn như thế nào ? Tổng cộng có bao nhiêu tên miền đã được cấp phát trên thế giới ? Đây là một câu hỏi khó mà không một tổ chức quốc tế nào có thể đưa ra được con số thống kế chính xác. Chúng ta chỉ biết rằng:

- Trong năm 1992, có khoảng 150.000 tên miền .com được đăng kí mới.
- Đến năm 1998, theo Google thì số lượng trang web đã đạt tới con số 26 triệu.
- Đến năm 2007, một cuộc khảo sát quy mô lớn được tiến hành thì có đến 29.7 tỷ trang trên World Wide Web theo thống kê của Netcraft.
- Trong năm 2008, Google tiết lộ có tới 1 nghìn tỷ (10^{12}) website nằm trong CSDL của mình.

Như vậy, chúng ta có thể thấy sẽ khó khăn như thế nào trong việc tìm kiếm thông tin mà chúng ta cần trên WWW nếu ko có một cỗ máy tìm kiếm.

1.1.3. Máy tìm kiếm – Search Engine

Máy tìm kiếm - Search Engine là một thư viện thông tin khổng lồ về các Website, cho phép người sử dụng có thể tìm kiếm các Website cần quan

tâm theo 1 chủ đề nào đó căn cứ vào các từ khóa (Keywords) mà người đó yêu cầu Search Engine tìm kiếm. Một số công cụ tìm kiếm mạnh trên thế giới hiện nay: Google.com, Yahoo.com, Altavista.com, Bing.com,…

Sẽ rất khó khăn cho người sử dụng truy cập vào Internet để tìm kiếm 1 Website có chủ đề phục vụ cho mục đích của mình vì hàng ngày có khoảng hơn 150.000 liên kết mới từ các Website được đưa lên mạng. Vì vậy để phục vụ việc tìm kiếm nhanh chóng Website của người sử dụng Internet, Search Engine ra đời.

1.1.4. Các bộ phận cấu thành Search Engine

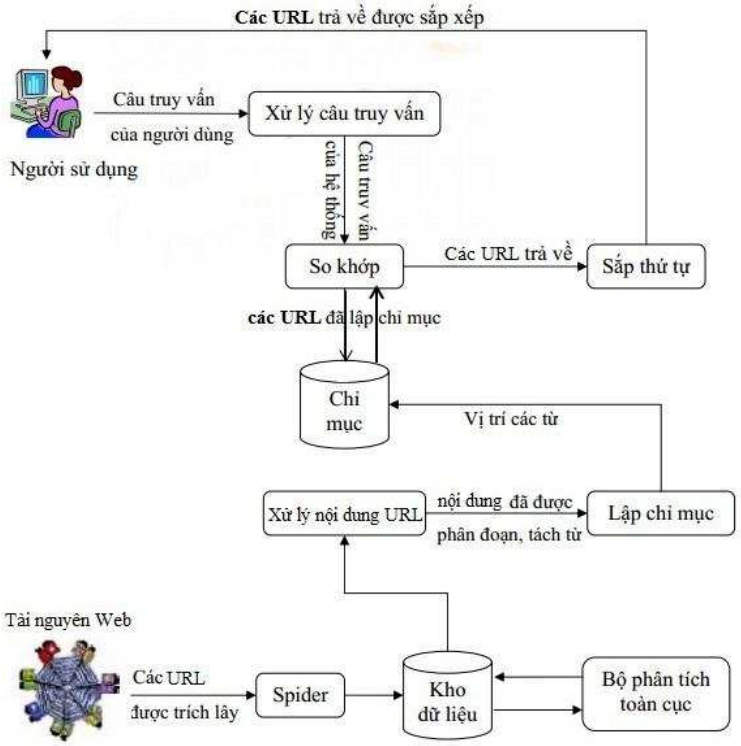
Bộ thu thập thông tin – Robot: Robot được biết đến dưới nhiều tên gọi khác nhau: spider, bot, crawler, hoặc web worm,… Về bản chất robot chỉ là một chương trình duyệt và thu thập thông tin từ các Website trên mạng, nó tự động duyệt qua các cấu trúc siêu liên kết và trả về các danh mục kết quả của công cụ tìm kiếm. Những trình duyệt thông thường không được xem là robot do thiếu tính chủ động, chúng chỉ duyệt web khi có sự tác động của con người.

Bộ lập chỉ mục – Index: hệ thống lập chỉ mục trong các công cụ tìm kiếm thực hiện phân tích, chọn lựa và lưu trữ những thông tin cần thiết (thường là các từ khóa hay cụm từ khóa) một cách nhanh chóng và chính xác từ những dữ liệu mà robot thu được. Hệ thống chỉ mục cho biết các danh mục từ khóa cần tìm nằm ở trang nào.

Bộ tìm kiếm thông tin – Search Engine: search engine hay còn gọi là Web Search Engine là một công cụ tìm kiếm được thiết để tìm kiếm các thông tin trên World Wide Web. Thông tin này có thể bao gồm những trang Web, hình ảnh hay bất cứ một kiểu file nào trên mạng. Nói rộng ra, Search Engine là hệ thống bao gồm cả bộ thu thập thông tin và bộ lập chỉ mục. Các bộ này

hoạt động liên tục từ lúc khởi động hệ thống, chúng phụ thuộc lẫn nhau về mặt dữ liệu nhưng độc lập với nhau về nguyên tắc hoạt động.

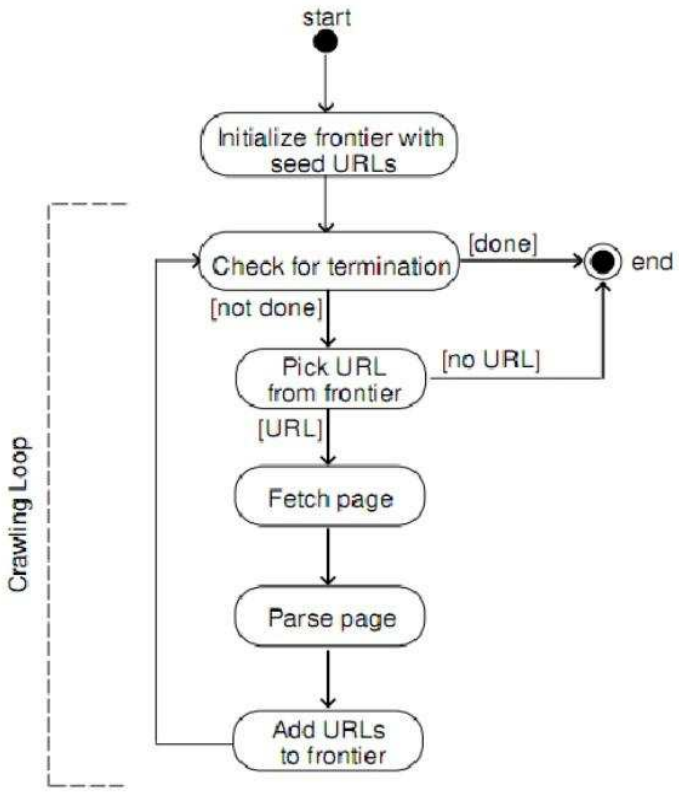
Như vậy, cơ chế tìm kiếm của search engine sẽ như hình dưới đây:



Hình 1. 2 - Cơ chế hoạt động của Search Engine

1.2. Bộ thu thập thông tin

Chu trình thu thập web cơ bản của máy tìm kiếm:



Hình 1. 3 - Vòng thu thập web cơ bản

Có những máy tìm kiếm, thu thập URL như sau:

- Frontier [5]: là danh sách URL thu thập được nhưng chưa được duyệt, việc thu thập các URL để thêm vào danh sách URL tùy thuộc vào bộ nhớ cũng như tốc độ của máy tìm kiếm. Và nó làm việc theo cơ chế FIFO (viết tắt của First In First Out) nghĩa là vào trước ra trước trong trường hợp chúng ta sử dụng thuật toán tìm kiếm theo chiều rộng. Frontier sẽ kết thúc tiến trình duyệt URL nếu nó đạt trạng thái rỗng. Điều này thật sự rất hiếm khi xảy ra. Nên tiến trình này rất dễ gây quá tải cho máy tìm kiếm nếu nó rơi vào bẫy nhện (Bẫy nhện là một tình huống có 2 URL có đường link đến nhau, khiến cho tiến trình thu thập URL rơi vào 1 vòng lặp vô tận trong quá trình duyệt URL

trong thuật toán tìm kiếm URL) của các trang web mà nó thu thập.

- Lược sử và kho lưu trữ trang [5]: khi các URL được lấy về, và sau khi được duyệt và xử lý chỉ mục nếu có, nội dung URL sẽ được lưu trữ thành file .html hoặc .txt để chờ xử lý index và tên của URL sẽ được một hàm băm tạo ra một tên mới, ví dụ với http://vnexpress.net ta sẽ có một chuỗi sau khi băm là 160766577426e1d01fcb7735091ec584 vào file .txt (ví dụ: site.txt) được đánh dấu là đã ghé thăm, đây là giá trị băm 128 bit sau đó được chuyển đổi sang hệ thập lục phân (hexa), bằng cách này, chúng ta sẽ luôn có 1 độ dài cố định cho dù tên có dài bao nhiêu đi chăng nữa. Ta có thể dựa vào chuỗi băm này để biết được rằng URL này đã được xử lý hay chưa.

Với cách thức trên, thì việc thu thu thập URL và lưu trữ sẽ làm tốn một dung lượng ổ đĩa cứng rất lớn và khó quản lí.

Vì vậy, ta sẽ tiến hành lưu trữ các URL và nội dung URL bằng hệ quản trị CSDL, việc lưu trữ này, sẽ giúp chúng ta dễ dàng kiểm soát quá trình thu thập URL và quản lý các URL thu thập được một cách trực quan hơn. Khi thu thập các URL, việc tạo ra vòng lặp thu thập sâu hơn vào bên trong trang web thu thập, vì quá trình thu thập sẽ diễn ra ra rất nhanh chiếm băng thông của trang web mà ta thu thập, sẽ gây ra cơ chế tự vệ (cơ chế này thường được các webmaster cấu hình để tránh trang web bị quá tải hay bị tấn công từ bên ngoài bằng cách đẩy các yêu cầu được gửi từ tiến trình thu thập của máy tìm kiếm sang một trang web khác) từ trang web bị thu thập. Vì vậy, trên thực tế, ta sẽ chỉ thu thập các URL trên bề mặt các trang URL rồi tiến hành lọc lấy các URL chưa có trong CSDL, rồi đưa các URL này vào trong CSDL, sau đó, ta sẽ dùng các chiến lược thu thập URL để duyệt các URL thu thập được.

1.3. Các chiến lược thu thập URL [3,tr1062-1064].

Trước khi các trang web được đánh chỉ mục, tất cả các trang web phải được lấy về máy tìm kiếm để được lưu trữ. Để lấy được tất cả các trang web, các crawler phải có chiến thuật. Từ một số trang web có sẵn, crawler lọc ra danh sách các liên kết, rồi từ đó dò tìm các trang khác. Có 3 chiến thuật tìm kiếm Heuristic sau : tìm kiếm theo chiều sâu, tìm kiếm theo chiều rộng và tìm kiếm ngẫu nhiên.

1.3.1. Chiến thuật tìm kiếm theo chiều sâu

Từ một danh sách chứa các liên kết cần duyệt, thực hiện các bước sau :

Bước 1: Lấy URL đầu tiên trong danh sách (frontier) để thu thập.

- Nếu có qua bước 2.
- Nếu không qua bước 5.

Bước 2: Lấy trang tương ứng với URL qua HTTP.

- Nếu có qua bước 3.
- Nếu không quay lại bước 1.

Bước 3: Kiểm tra xem trang này đã được được thăm chưa?

- Nếu chưa qua bước 4.
- Nếu rồi quay lại bước 1.

Bước 4: Đánh dấu trang này đã được thăm. Bóc tách trang và tìm các liên kết có trong trang này.

- Nếu có, thêm các liên kết vào đầu danh sách. Quay lại bước 3.
- Nếu không, quay lại bước 1.

Bước 5: Kết thúc.

1.3.2. Chiến lược thu thập dữ liệu theo chiều rộng

Từ một danh sách chứa các liên kết cần duyệt, thực hiện các bước sau :

Bước 1: Lấy URL đầu tiên trong danh sách để thu thập.

- Nếu có qua bước 2.

- Nếu không qua bước 5.

Bước 2: Lấy trang tương ứng với URL qua HTTP.

- Nếu có qua bước 3.
- Nếu không quay lại bước 1.

Bước 3: Kiểm tra xem trang này đã được được thăm chưa?

- Nếu chưa qua bước 4.
- Nếu rồi quay lại bước 1.

Bước 4: Đánh dấu trang này đã được thăm. Bóc tách trang và tìm các liên kết có trong trang này.

- Nếu có, thêm các liên kết vào cuối danh sách. Quay lại bước 3.
- Nếu không, quay lại bước 1.

Bước 5: Kết thúc.

1.3.3. Chiến thuật tìm kiếm theo ngẫu nhiên

Từ một danh sách chứa các liên kết cần duyệt, thực hiện các bước sau :

Bước 1: Lấy URL ngẫu nhiên trong danh sách để thu thập.

- Nếu có qua bước 2.
- Nếu không qua bước 5.

Bước 2: Lấy trang tương ứng với URL qua HTTP.

- Nếu có qua bước 3.
- Nếu không quay lại bước 1.

Bước 3: Kiểm tra xem trang này đã được được thăm chưa?

- Nếu chưa qua bước 4.
- Nếu rồi quay lại bước 1.

Bước 4: Đánh dấu trang này đã được thăm. Bóc tách trang và tìm các liên kết có trong trang này.

- Nếu có, thêm các liên kết vào cuối danh sách. Quay lại bước 3.
- Nếu không, quay lại bước 1.

Bước 5: Kết thúc.

1.4. Bộ lập chỉ mục - index

Các URL thu thập được cần phải được xử lý thích hợp trước khi thực hiện việc tìm kiếm. Việc sử dụng các từ khoá hay thuật ngữ để mô tả nội dung của URL theo một khuôn dạng ngắn gọn hơn được gọi là tạo chỉ mục (hay còn gọi là từ khóa) cho URL.

Modul Indexer và Collection Analysis có chức năng tạo ra nhiều loại chỉ mục khác nhau. Modul Indexer tạo ra hai loại chỉ mục chính đó là chỉ mục Text Index (chỉ mục nội dung được tạo ra qua quá trình xử lý nội dung của URL) và Structure Index (cấu trúc chỉ mục là lưu trữ chỉ mục theo dạng cây). Dựa vào hai loại chỉ mục này bộ Collection Analysis tạo ra nhiều loại chỉ mục hữu ích khác:

Link Index: tạo chỉ mục liên kết, các URL đã duyệt được biểu diễn dưới dạng đồ thị với các đỉnh (đỉnh là các URL đã được xử lý nội dung) và các cạnh (cạnh là index liên kết các URL và ta có thể có 1 cạnh nối 2 đỉnh URL nghĩa là có 1 đỉnh URL liên kết đến 1 đỉnh URL còn lại).

Text Index: Phương pháp đánh chỉ mục dựa theo nội dung (text-based) là một phương pháp quan trọng để định danh các trang có liên quan đến yêu cầu tìm kiếm, được tạo ra khi xử lý nội dung của URL, phần này được xử lý tại bộ xử lý index.

Chỉ mục kết hợp: do người lập trình quy định, nó có thể là các con số, ngày tháng, dạng text, số lượng và kiểu của các chỉ mục được quy định bởi bộ Collection Analysis tuỳ thuộc vào chức năng của bộ máy truy vấn và kiểu thông tin mà modul Ranking sử dụng. Nếu chúng ta có bộ từ điển đầy đủ 1 từ và 2 từ, 3 từ, 4 từ thì chúng ta có thể giới hạn được số từ khóa thu thập được từ nội dung các URL, chúng ta có thể không lập chỉ mục với các con số, ký tự đặc biệt…

Nói một cách ngắn gọn là quá trình xử lý nội dung của URL sẽ tạo ra nhiều từ khóa và các từ khóa và các từ khóa này có rất nhiều dạng khác nhau, việc tạo ra từ khóa theo các định dạng như thế nào sẽ do người viết chương trình phân tích và thiết kế bộ lọc từ khóa, bộ lọc này có thể giữ kiểu từ khóa này và loại bỏ kiểu từ khóa kia.

1.4.1. Các bước lập chỉ mục

Bước 1: Xác định các mục từ (là các từ ngắn có từ 1 đến 4 từ) có khả năng đại diện cho nội dung URL sẽ được lưu trữ.

Bước 2: Xác định trọng số cho từng mục từ, trọng số này là giá trị phản ánh tầm quan trọng của mục từ đó trong văn bản.

1.4.2. Xác định mục từ quan trọng

Ta xác định mục từ của một văn bản dựa vào chính nội dung của văn bản đó, hoặc tiêu đề hay tóm tắt nội dung của văn bản đó.

Thông thường việc lập chỉ mục tự động bắt đầu bằng việc khảo sát tần số xuất hiện của từng loại từ riêng rẽ trong văn bản.

Đặc trưng xuất hiện của từ vựng có thể được định bởi "thứ hạng - tần số" (Rank_Frequency).

Các bước để xác định một mục từ quan trọng:

- Cho một tập hợp n tài liệu, thực hiện tính toán tần số xuất hiện của các mục từ trong tài liệu đó.

Ký hiệu Fik (Frequency): là số lần xuất hiện của mục từ k trong tài liệu i.

- Xác định tổng tần số xuất hiện TF_k(Total Frequency) cho mỗi từ bằng cách cộng những tần số của mỗi mục từ duy nhất trên tất cả n tài liệu.

$$TF_k = \sum_{i=1}^{n} F_{ik}$$

- Sắp xếp các mục từ theo thứ tự giảm dần của tần số xuất hiện. Chọn một giá trị làm ngưỡng và loại bỏ tất cả những từ có tổng tần số xuất hiện cao hơn ngưỡng này (stop-word).

1.4.3. Tính trọng số của mục từ

Trọng số của mục từ: là tần xuất xuất hiện của mục từ trong toàn bộ nội dung URL. Phương pháp thường được sử dụng để đánh giá trọng số của từ là dựa vào thống kê, với ý tưởng là những từ thường xuyên xuất hiện trong tất cả các URL thì "ít có ý nghĩa hơn" là những từ tập trung trong một số URL.

Ngược lại khi tần số xuất hiện của mục từ k trong tập URL càng cao thì mục từ đó càng có ý nghĩa.

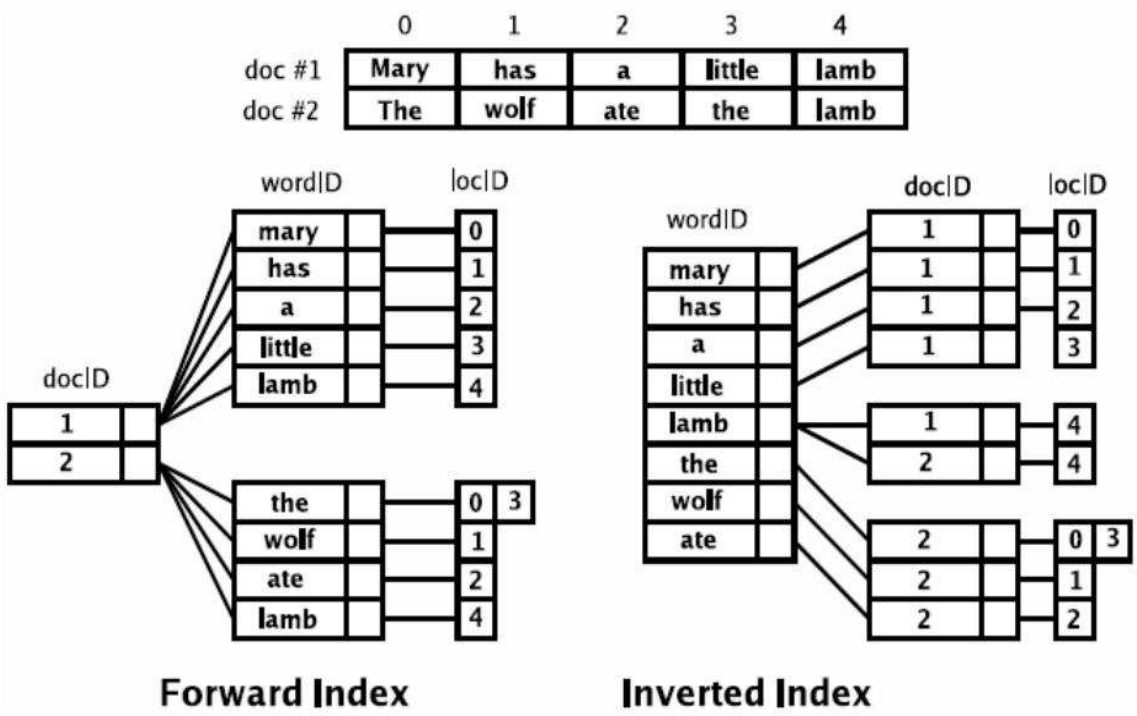
Lập chỉ mục tự động (có thể xem là từ điển) cho URL là xác định tự động mục từ và chỉ mục (các mục từ có thể được kết hợp qua bộ xử lý index thành chỉ mục) cho các URL,dựa vào đó ta có thể loại bỏ các từ stop-word vì những từ này có độ phân biệt kém và không thể sử dụng để xác định nội dung của tài liệu.

Bước tiếp theo là chuẩn hoá mục từ, tức là đưa mục từ về dạng nguyên gốc bằng cách loại bỏ tiền tố, hậu tố, và các biến thể khác của từ như từ ở dạng số nhiều, quá khứ,… được áp dụng cho tiếng Anh.

1.4.4. Cấu trúc của chỉ mục đảo

Sau khi phân tích các trang web, và thực hiện tách các từ, chuẩn hoá các từ về dạng nguyên gốc, loại bỏ các từ stop word. Ta thu được một danh mục các từ mỗi từ được gắn kèm danh sách các trang chứa từ đó. Danh mục này gọi là chỉ mục đảo (inverted index).

Hình minh họa cho cấu trúc chỉ mục nghịch đảo, phục vụ cho việc tìm kiếm.



Hình 1. 4 - Cấu trúc bảng nghịch đảo

1.4.5. Lập chỉ mục cho Tiếng Việt dựa vào từ điển Tiếng Việt

Với các từ một từ, ta có thể dùng danh sách vnstopword để loại, nhưng với từ khóa có 2 từ, điều này là không thể. Vì vậy, ta phải tạo ra từ điển Tiếng Việt[1] dựa vào đó, ta sẽ xác định được từ nào có nghĩa trong Tiếng Việt, từ nào không một cách dễ dàng.

Ví dụ : với câu : "Học sinh đi học sinh học".

Ta thấy là có thể tách ra được các keyword 2 từ: học sinh, sinh học, đi học dựa vào từ điển lập sẵn, ta sẽ loại đi keyword là "đi học" vì từ "đi học" không có trong từ điển Tiếng Việt chỉ có 2 từ mà ta lập sẵn.

Như vậy, giải pháp đề nghị là dùng từ điển được lập sẵn,với một chi phí thấp hơn ta có thể lập được một từ điển tương đối đầy đủ mà kết quả chính xác hơn rất nhiều.

Thuật toán lọc từ khóa có 2 từ viết đơn giản như sau:

Bước 1: lấy phần nội dung, tách ra mảng từ khóa có 1 từ theo thứ tự của nội dung. Sang bước 2

Bước 2 : tạo ra mảng từ khóa 2 từ bằng cách lấy 2 từ liên tiếp theo thứ tự các từ của mảng từ khóa 1 từ để tạo thành từ khóa có 2 từ. Sang bước 3

Bước 3 : đưa mảng từ khóa có 2 từ so sánh với từ điển Tiếng Việt, từ nào không có trong từ điển, thì ta loại từ đó.

Bước 4 : kết thúc.

1.5. Bộ tìm kiếm thông tin

Với một máy tìm kiếm, thì ngoài việc thu thập các thông tin trên các website và việc lập chỉ mục sao cho hợp lí thì việc cho ra kết quả tìm kiếm nhanh chóng chính xác đúng với những gì người dùng tìm kiếm là điều bắt buộc nếu muốn thu hút người dùng.

Hiện nay, trên thế giới, sử dụng 2 thuật toán sắp xếp kết quả tìm kiếm được là thuật toán Pagerank và thuật toán Tf-idf.

1.5.1. Thuật toán Pagerank[2]

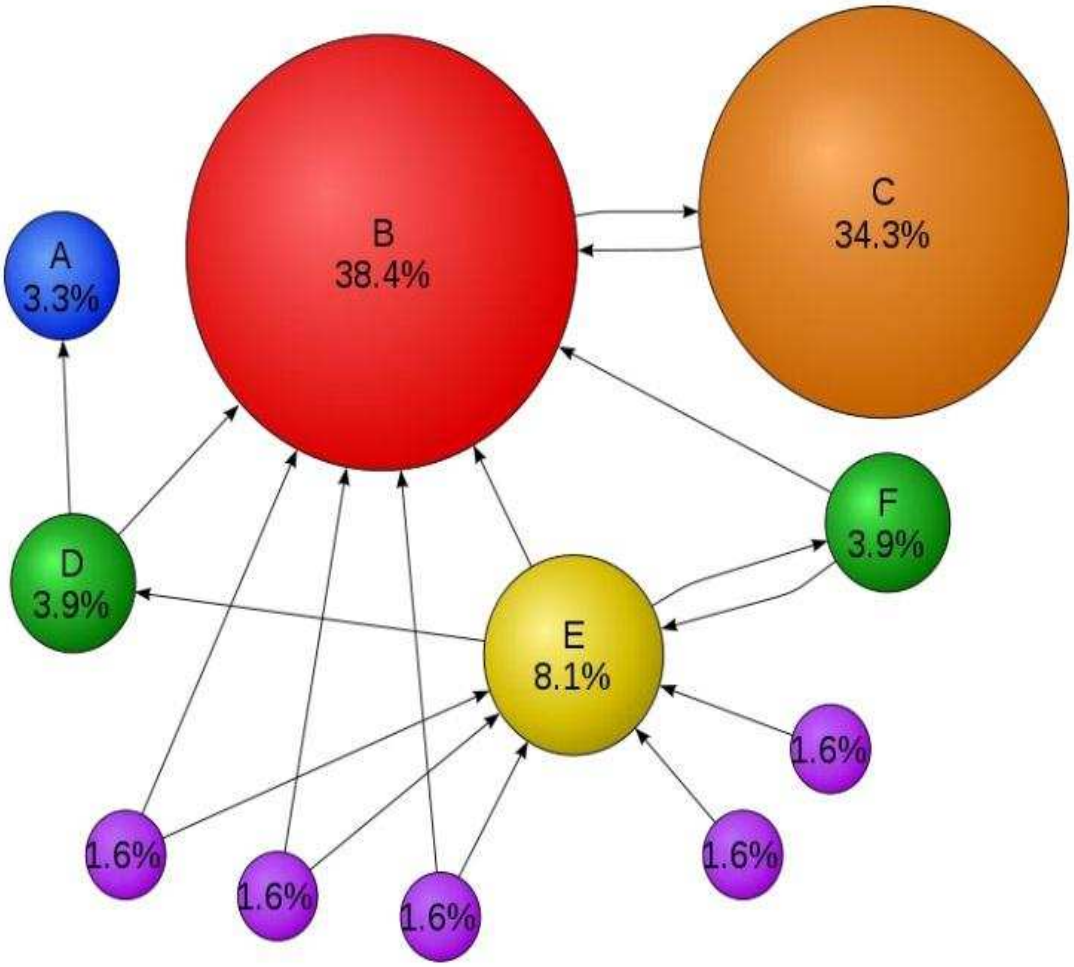
Được phát triển bởi Larry Page và Sergey Brin vào năm 1997, sau đó đã đưa vào sử dụng trong Google, Yahoo! Và các máy tìm kiếm khác.

Hiện tại cũng sử dụng một thuật toán tương tự. Ý tưởng chính của PageRank dựa vào số các liên kết đến trang web. Khi một trang web A có một trang web B liên kết đến nó ta có thể nói tác giả của B quan tâm đến A nói cách khác nội dung của A sẽ liên quan đến một chú đề hoặc một từ khóa. Bên cạnh đó, PageRank còn thêm trọng số vào các trang B liên kết đến A. Cụ thể nếu B có PageRank càng cao cũng sẽ dẫn đến A có PageRank cao. Trọng số này được đưa ra nhằm tránh hiện tượng spam link. Cụ thể ta xét ví dụ:

Giả thuyết không gian các website gồm có 4 phần tử: A, B, C, D. Khi đó xác suất một người dùng truy cập vào một trang bất kì là 0.25.

Tuy nhiên nếu B đặt liên kết đến A, xác suất A được truy cập sẽ tăng lên và bằng 0.25 + 0.25 = 0.5

Giả thuyết thêm C cùng trỏ đến A và D khi đó xác suất A được truy cập sẽ bằng 0.25 + 0.25 + 0.25/2 = 0.625



Hình 1. 5 - Kết quả tính toán từ 1 trang web của Pagerank

Mặc dù C chỉ có liên kết từ B nhưng do B có PageRank cao nên PageRank của C cao. Ngược lại E có nhiều liên kết đến nhất nhưng các trang liên kết đến E có PageRank thấp do đó PageRank của E thấp.

Trên thực tế, PageRank còn dựa vào rất nhiều yếu tố khác để đưa ra đánh giá cuối cùng. Theo công bố của Google, PageRank chứa khoảng 500 triệu biến cùng với 2 tỉ số hạng.

Với độ phức tạp như vậy, chúng ta có thể hiểu vì sao Google lại đứng đầu trong việc tìm kiếm.

1.5.2. Thuật toán tf-idf [4,tr116-137]

Tf-idf, viết tắt của thuật ngữ tiếng Anh term frequency – inverse document frequency, được phát triển bởi Gerard Salton. Tf-idf của một từ, là con số thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản khác.

Tf-idf là 1 phương pháp phổ biến để đánh giá và xếp hạng một từ trong một tài liệu, về cơ bản tf-idf là một kỹ thuật (ranking function) giúp chuyển đổi thông tin dưới dạng văn bản thành một mô hình không gian vector (Vector space model).

Trong đó, chúng ta giả định mỗi một nội dung tài liệu là một vector và mỗi từ tương ứng với mỗi thành phần vector được sắp xếp theo thứ tự bảng chữ cái. Ta phải tính các trọng số tf-idf của các từ trong các tài liệu. Công thức tính tf, idf cho mỗi từ trong tài liệu như sau:

$$tf(i,d) = idf(i,d) = \log_2(N/n_i) \quad (1.1)$$

Trong đó :

- N: là tổng số tài liệu.
- n_i là số lần xuất hiện của 1 từ trong các tài liệu.

Lí do vì sao tf = idf là do idf chính là tần số tài liệu nghịch đảo của tần số hạn tf (hay còn gọi là tần số xuất hiện), thì từ idf ta có được ma trận tần số hạn tf của các tài liệu.

Từ đó, ma trận tf, ta có được vector tf-idf của từng tài liệu.

Với câu truy vấn bất kì, ta tính được idf của mỗi từ có trong câu truy vấn qua công thức sau:

$$tf(i,q) = idf(i,q) = (N_{qi}/n_i)*idf_i \quad (1.2)$$

Trong đó:

N_{qi}: số lần xuất hiện của từ trong câu truy vấn.

n_i : số lần xuất hiện của từ trong các tập tài liệu.

Rồi từ đó, ta nghịch đảo để tìm idf sang tf để tìm vector tf-idf của câu truy vấn.

Ta tiếp tục tính độ dài của từng vector tf-idf của các tài liệu là $lengthD_i$ và độ dài của vector tf-idf câu truy vấn lengthq có công thức là:

$$lengthD_i = \sqrt{\sum_i^n (tf(i,d))^2} \qquad (1.3)$$

$$lengthq = \sqrt{\sum_i^n (tf(i,q))^2}$$

Từ các giá trị có được, ta tính tương đồng CosSim, với công thức tính là:

$$CosSim(\vec{q}, \vec{D_i}) = (\vec{q} * \vec{D_i}) / lengthD_i * lengthq \qquad (1.4)$$

Với $\vec{q}$, $\vec{D_i}$: là vector tf-idf của q và D_i

Khi tính được CosSim xong, thì ta sắp xếp theo chiều giảm dần của giá trị CosSim sẽ ra được số thứ tự của tài liệu cần tìm.

Ví dụ: ta có tài liệu ma trận các tài liệu sau:

D1 = “new york times”

D2 = “new york post”

D3 = “los angeles times”

Như vậy, với công thức tính idf thì ta có bảng tính idf cho mỗi từ là như sau:

	angeles	los	new	post	times	york
idf	1.584	1.584	0.584	1.584	0.584	0.584

Và từ idf ta có các điểm tf trong ma trận giá trị của tập tài liệu, bảng sau là vector tf-idf của mỗi tài liệu:

Vector tf-idf của tài liệu	angeles	los	new	post	times	york
D1	0	0	0.584	0	0.584	0.584
D2	0	0	0.584	1.584	0	0.584
D3	1.584	1.584	0	0	0.584	0

Với câu truy vấn là: "new new times", chúng ta sẽ tính được vector tf-idf của câu truy, theo công thức trên, ta sẽ có bảng sau:

Vector tf-idf	angeles	los	new	post	times	york
q	0	0	(2/2)*0.584 = 0.584	0	(1/2)*0.584 = 0.292	0

Tiếp theo, chúng ta tính độ dài của các tài liệu lengthD và độ dài câu truy vấn lengthq có các từ tìm kiếm được tìm thấy trong các tài liệu.

lengthq	sqrt($0.584^2 + 0.292^2$) = 0.652
lengthD1	sqrt($0.584^2 + 0.584^2 + 0.584^2$) = 1.0.11
lengthD2	sqrt($0.584^2 + 1.584^2 + 0.584^2$) = 1.786
lengthD3	sqrt($1.584^2 + 1.584^2 + 0.584^2$) = 2.316

Chúng ta tiếp tục tính góc tương đồng của câu truy vấn so với các tập tài liệu để tìm ra được thư tự tài liệu cần tìm trong câu truy vấn:

cosSim(D1,q)	(0*0+0*0+0.584*0.584+0*0+0.584*0.292+0.584*0) /(1.0.11*0.652) = 0.776
cosSim(D2,q)	(0*0+0*0+0.584*0.584+1.584*0+0*0.292+0.584*0)

	/(1.786*0.652) = 0.292
cosSim(D3,q)	(1.584*0+1.584*0+0*0.584+0*0+0.584*0.292+0*0) /(2.316*0.652) = 0.112

Với giá trị cosSim tìm được, ta sắp xếp giảm dần giá trị cosSim này, và câu truy vấn sẽ có kết quả là: D1, D2, D3.

Như vậy, tôi chọn thuật toán tf-idf cho quá trình xử lý tìm thông tin cho máy tìm kiếm vì lý do, chi phí cũng như khả năng xử lý cao cho kết quả chính xác một cách nhanh chóng mà không tốn nhiều tài nguyên của bộ nhớ của hệ thống.

1.6. Tổng kết chương 1

Chương 1 của luận văn, tác giả đã trình bày các vấn đề lý luận, của máy tìm kiếm bao gồm :

Trình bày các khái niệm cơ bản, các thuật ngữ liên quan đến máy tìm kiếm.

Đi sâu giải thích các thành phần cấu thành nên một máy tìm kiếm, nguyên lý hoạt động, các thuật toán tìm kiếm các liên kết, thuật toán xử lý chỉ mục, mô hình không gian với phương pháp tf-idf, cũng như sơ lược về phương pháp pagerank.

Chương 2

XÂY DỰNG ỨNG DỤNG MÁY TÌM KIẾM

2.1. Phần mềm hỗ trợ

Để viết thành công ứng dụng, ta cần cài đặt 2 gói phần mềm sau:

- Phần mềm xampp-win32-1.8.3-3-VC11-installer : đây là phần mềm mã nguồn mở là chương trình tạo máy chủ Web (Web Server) trên máy tính cá nhân (Localhost) được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như PHPmyadmin. Để viết được ứng dụng ta cần phải dùng phần mềm này, để tạo và quản trị CSDL, cũng như hỗ trợ ngôn ngữ lập trình PHP.
- Phần mềm phpDesigner : phần mềm hỗ trợ cho việc viết các ứng dụng lập trình web, không chỉ hỗ trợ PHP mà còn các ngôn ngữ web khác như HTML, MySQL, XML, CSS, JavaScript, VBScript, Java, C#, Perl, Python và Ruby Sức mạnh trong PHP nằm ở khả năng ấn định nó vào cùng với các ngôn ngữ và công nghệ web khác.

2.2. Thiết kế CSDL của máy tìm kiếm

2.2.1. Bảng domains

Bảng này dùng để lưu trữ các tên miền mà ta đã thu thập, có thể xem các tên miền là seedUrl đầu tiên làm hạt giống để từ đó thu thập toàn bộ các trang web bên trong của một trang web. Bảng này có tác dụng trong việc thống kê tổng số trang web mà ta thu thập được là bao nhiêu và đánh dấu thông tin là tên miền này đã được thu thập.

Bảng domains		
Tên	Kiểu dữ liệu	Ý nghĩa
domain	Int(10)	khóa chính, id tự sinh ra khi 1 domain được thêm vào.
domain_name	Varchar(50)	Tên của domain được thu thập
domain_date	datetime	lưu lại ngày được thu thập gần nhất

2.2.2. Bảng urls

Bảng urls này dùng để lưu lại tất cả các link thu thập được trong quá trình thu thập trên 1 website.

Bảng urls			
STT	tên	Kiểu dữ liệu	Ý nghĩa
1	url_id	Int(10)	Khóa chính, id tự sinh ra khi được thêm vào.
2	url_name	Varchar(200)	Lưu lại tên các link thu thập được.
3	url_title	Varchar(200)	Lưu lại tiêu đề lấy được từ đường link.
4	url_desc	Text	Lưu lại phần mô tả tin tức của link.
5	url_content	mediumtext	Lưu lại nội dung tin tức của link.
6	url_date	datetime	Lưu lại ngày cập nhật gần nhất của link.
7	url_click	Int(10)	Dùng để đếm số lần người tìm kiếm click vào.

8	flag	Int(1)	Dùng để làm cờ, đánh dấu cho quá trình xử lý.
9	FK_domain_id	Int(10)	Khóa ngoại của urls, dùng cho việc thống kê tổng số URL và đánh dấu URL thu thập được thuộc tên miền nào.

Giải thích rõ hơn về trường flag trong bảng urls này.

Trong trường flag, ta có thể qui định tùy theo người lập trình, dùng để nhận biết tình trạng các dòng thông tin trong bảng urls.

- flag = 0 : nghĩa là link này được thu thập lần đầu tiên, nhưng chưa được duyệt, hỗ trợ cho thuật toán tìm kiếm theo chiều sâu.
- flag = 1 : nghĩa là link này đã được duyệt và đã được xử lý, lấy các link bên trong nó nhưng chưa được xử lý keyword.
- flag = 2 : nghĩa là link này đã được xử lý keyword, có thể phục vụ cho các kết quả tìm kiếm của người dùng.
- flag = 3 : để nhanh chóng cung cấp thông tin cho người tìm kiếm mà ko phải chờ đợi việc duyệt tin, ta có thể bật cờ này lên, tùy cách lập trình của mỗi người, và tính cấp thiết của việc phục vụ người dùng, mà ta có thể phá cách trong việc xử lý của thuật toán tìm kiếm theo chiều sâu.

Flag tùy thuộc hoàn toàn vào ý đồ của người viết ứng dụng.

Lý do vì sao chúng ta phải chia ra nhiều giai đoạn để xử lý như vậy, có nhiều nguyên nhân sau:

- Để tránh server bị quá tải trong quá trình xử lý các liên kết. Lí do là với seedURL đầu tiên, thì ta có thể thu thập được rất nhiều URL, và từ đó, ta sẽ tiếp tục tìm ra rất nhiều các URL khác, và trong một thời gian ngắn việc thu thập này sẽ tạo chiếm rất nhiều

bộ nhớ máy của tính và công suất của CPU máy chủ, nếu ta không kiểm soát được tiến trình này, thì việc quá tải nhất định sẽ xảy ra. Như vậy, việc kiểm soát được số URL thu thập được trong mỗi lần thu thập sẽ tránh được việc quá tải của server.

- Tránh bị trang web mà chúng ta lấy tin ngắt kết nối hoặc đẩy sang trang web khác, khi mà các webmaster nhận thấy chúng ta chiếm lưu lượng băng thông của họ trong một thời gian quá ngắn. Như đã nói ở trên, khi chúng ta thu thập các URL, là ta đã gửi rất nhiều yêu cầu một cách dồn dập và liên tục đến trang web mà ta thu thập URL việc này sẽ làm trang web bị ta thu thập phải trả lời yêu cầu của ta một cách dồn dập và liên tục trong một thời gian ngắn, như vậy sẽ ảnh hưởng đến trang web nên yêu cầu của ta chắc chắn sẽ bị từ chối.
- Chúng ta chủ động trong việc xử lý, và có thể kiểm soát được sự quá tải của server, cũng như việc xử lý của thuật toán. Có nghĩa là khi thu thập các URL ta phải kiểm soát được có bao nhiêu URL được thu thập, duyệt các URL một cách chủ động ta sẽ có 1 khoảng nghỉ và việc gửi các yêu cầu đến trang web bị thu thập một cách có kiểm soát, sẽ giúp ta giải phóng bộ nhớ, giảm tải công suất hoạt động của CPU.

2.2.3. Bảng words

Bảng words dùng để chứa các keyword thu thập được, cũng như chứa các tham số tf-idf .

Bảng words			
TT	tên	Kiểu	Ý nghĩa

1	word_id	Int(10)	Khóa chính, id tự sinh ra khi 1 từ được thêm vào
2	word	Varchar	Tên của từ khóa
3	word_idf	float	Dùng để lưu tần số tài liệu nghịch đảo idf
4	word_date	datetime	Lưu lại ngày cập nhật gần nhất
5	url_id_join	longtext	lưu nhiều id của url, cách nhau dấu “,”
6	total_wordurl	Int(10)	Đếm số url được tìm thấy của 1 từ để tính tf-idf
7	word_count	Int(10)	Đếm số lần người dùng tìm kiếm, để thống kê.

Giải thích rõ hơn về một số trường trong bảng words.

Từ công thức tính trọng số idf của mỗi từ là $\log_2(N/n)$:

- total_wordurl : chính là N và N chính là tổng số link đã được xử lý keyword.
- word_idf : cột này dùng để lưu tần số nghịch đảo idf đã tính toán được dùng cho việc tính ma trận vector của thuật toán tf-idf.
- url_id_join : cột này dùng để tiện cho việc tìm kiếm các link khi mà từ này được tìm kiếm trong bảng words. Ví dụ: ta lưu các url_id là 1,12,13,50,…,1452154.

2.2.4. Bảng words_urls

Bảng words_urls dùng để lưu ma trận vector của thuật toán tf-idf phục vụ cho việc tính toán làm sao để tính được cosSim của mỗi tài liệu so với câu truy vấn của người dùng, để từ đó tính được link nào gần với yêu cầu tìm kiếm của người dùng.

Dùng để lưu chi tiết của mỗi keyword so với mỗi url.

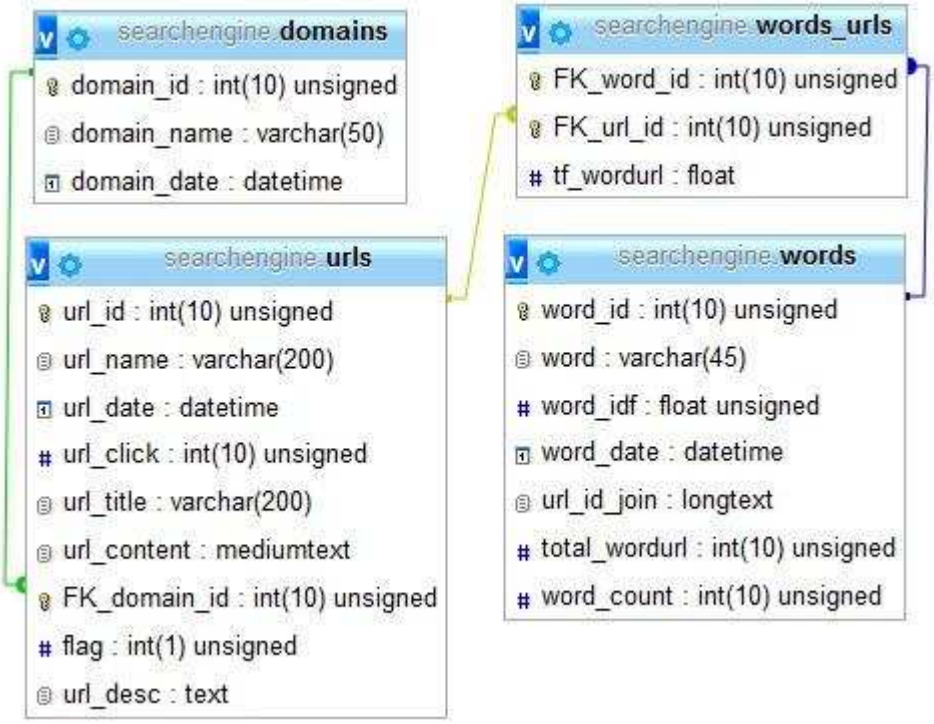
Bảng words_urls			
TT	tên	Kiểu	Ý nghĩa
1	FK_word_id	Int(10)	Khóa ngoại, liên hệ trong ma trận với 1link
2	FK_url_id	Int(10)	Khóa ngoại, liên hệ trong ma trận với 1keyword
3	tf_wordurl	float	Lưu trọng số của mỗi link so với mỗi keyword

Giải thích rõ hơn về trường tf_wordurl trong bảng words_urls :

- tf_wordurl : trường này để lưu trọng số của 1 link đối với 1 keyword trong ma trận vector sắp xếp theo thứ tự abc.

2.2.5. Sơ đồ quan hệ các bảng trong CSDL

Sơ đồ quan hệ dưới đây sẽ cho ta có cái nhìn rõ hơn về mối quan hệ các bảng trong CSDL máy tìm kiếm.



Hình 2. 1 - Sơ đồ quan hệ của các bảng trong máy tìm kiếm

2.3. MySQL

MySQL là hệ quản trị CSDL tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là CSDL tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, ...

2.3.1. Các câu lệnh dùng để tạo kết nối đến CSDL

- mysql_connect() : hàm này dùng để kết nối đến CSDL. Khi kết nối tới MySQL thành công, hàm sẽ trả về giá trị là một số

nguyên định danh của liên kết, ngược lại, hàm trả về giá trị false.

Ta có thể dùng hàm if để kiểm tra xem có kết nối được tới MySQL hay không:

Cú pháp: mysql_connect(host,tên_truy_cập,mật_khẩu);

Trong đó:

host : là chuỗi chứa tên (hoặc địa chỉ IP) của máy chủ cài đặt MySQL.

tên_truy_cập : là chuỗi chứa tên truy cập hợp lệ của CSDL cần kết nối.

mật_khẩu : là chuỗi chứa mật khẩu tương ứng với tên truy cập.

Ví dụ :

```
1.<?php
2.$connect = mysql_connect("localhost", "mysql_user", "mysql_passw
3.ord");
4.  if (!$link)
5.{
6.       echo "Không thể kết nối được tới CSDL MySQL";
7. }
8.?>
```

- mysql_select_db : sau khi kết nối đến CSDL ta có thể chọn CSDL để làm việc.

Cú pháp: mysql_select_db(tên_CSDL, $kết nối)

Ví dụ : tiếp nối với ví dụ trên

```
1.<?php
2.$connect = mysql_connect("localhost", "mysql_user", "mysql_
pas3.s.
4.word");
```

5.mysql_select_db(“searchengine”, $connect);

6.?>

Như vậy là ta đã hoàn thành việc kết nối đến CSDL để làm việc.

2.3.2. Một số lệnh thao tác cơ bản của MySQL

- Select : lệnh này thường dùng lấy ra dữ liệu trong bảng.

Cú pháp : select tên_cột from tên_bảng

Ví dụ :

Select * from urls where url_id=10 : lấy thông tin của 1 url có điều kiện

Select * from urls : lấy tất cả url có trong csdl của bảng url.

- Insert : lệnh này để thêm vào thông tin vào CSDL trong bảng

Cú pháp : insert into tên_bảng(cột 1,cột 2,…) values(giá trị 1,…)

Ví dụ :

Insert into urls(url_date) value(now()) : thêm ngày tháng vào bảng urls.

- Update : lệnh này để cập nhật các giá trị của thông tin đã có.

Cú pháp : update tên_bảng set tên_cột=giá trị mới where tên_cột=giá trị.

Ví dụ :

Update urls set url_date=now() where url_id=10

- Delete : lệnh này dùng để xóa 1 hay nhiều thông tin đã có.

Cú pháp : delete from tên_bảng where tên_cột=giá trị

Ví dụ :

Delete from urls where url_id = 10 : xóa thông tin có id = 10

Delete from urls : xóa hết các thông tin trong bảng urls.

2.4. Ngôn ngữ lập trình web PHP

PHP (viết tắt hồi quy "PHP: Hypertext Preprocessor") là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các

ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

Ngôn ngữ, các thư viện, tài liệu gốc của PHP được xây dựng bởi cộng đồng và có sự đóng góp rất lớn của Zend Inc., công ty do các nhà phát triển cốt lõi của PHP lập nên nhằm tạo ra một môi trường chuyên nghiệp để đưa PHP phát triển ở quy mô doanh nghiệp.

Hiện nay, PHP đã phát triển lên version 6 và facebook là một trong những ứng dụng mạng xã hội nổi tiếng được viết bằng ngôn ngữ PHP.

Một số hàm PHP thường dùng để viết ứng dụng máy tìm kiếm

Dưới đây, là những hàm thường được dùng để ứng dụng cho việc viết ứng dụng máy tìm kiếm cũng như là cho các chương trình khác:

- preg_match() : hàm này dùng để kiểm tra, cũng như lấy chuỗi ký tự, được ứng dụng trong máy tìm kiếm bằng cách để kiểm tra các link rác, cũng như tìm kiếm 1 chuỗi con trong 1 chuỗi lớn, lấy domain của 1 link nào đó. Đây là hàm quan trọng được ứng dụng rất nhiều trong viết ứng dụng.
- parse_url() : hàm này cho ta biết thông tin chi tiết của một link, như là : giao thức, tên host, đường dẫn(path), biến.
- isset() : hàm để kiểm tra 1 biến có tồn tại hay không, nó trả về giá trị boolean, tồn tại thì cho giá trị TRUE, ngược lại là FALSE.

- empty() : kiểm tra xem giá trị có rỗng hay không, nó khác isset() ở chỗ là nó kiểm tra biến có tồn tại một giá trị nào hay không, bất chấp là TRUE hay FALSE.
- is_array() : kiểm tra xem có phải là mảng hay không.
- array_key_exists : kiểm tra 1 biến có tồn tại trong mảng 2 chiều là $key có hay không, trả về giá trị boolean.
- urlencode() : nó chuyển đổi các link lấy được từ mã ASCII sang định dạng mã ASCII hợp lệ, thay thế các ký tự không có thành "%" theo sau bởi 2 chữ số thập lục phân ngẫu nhiên và thay thế khoảng trắng nếu có trong link bằng 1 dấu "+".
- preg_split() : dùng để cắt 1 chuỗi thành từng từ đưa vào mảng, đây là hàm thật sự quan trọng với ứng dụng.

Ví dụ : preg_split("/[\s,.?!]+/i") : thông thường, việc tách 1 ngôn ngữ Tiếng Anh, thì chỉ cần như vậy là đủ, nhưng với ngôn ngữ Tiếng Việt thì sẽ bị lỗi, vì mã của chúng ta là unicode, để có thể tách thành công, chúng ta phải bật tính năng nhận diện unicode bằng cách thay chữ "i = u". Như vậy, để tách thành công 1 từ trong chuỗi Tiếng Việt sẽ là preg_split("/[\s,.?!]+/u").

- array_count_values() : hàm đếm các giá trị trùng nhau trong mảng 1 chiều và chuyển đổi giá trị thành $key.
- array_diff() : hàm so sánh 2 mảng với nhau, và trả về giá trị không có của 1 mảng, được áp dụng để loại bỏ các từ vô nghĩa trong vnstopword.
- array_intersect() : hàm so sánh 2 mảng, và trả về giá trị trùng nhau của 2 mảng, được áp dụng để loại bỏ các từ không thuộc từ có nghĩa trong Tiếng Việt, dùng để lọc keyword có 2 từ.
- array_filter() : hàm dùng để loại bỏ các giá trị rỗng trong mảng.

- array_merger() : hàm dùng để kết nối 2 mảng lại với nhau, được áp dụng cho việc nối mảng có keyword 1 từ và mảng keyword có 2 từ.
- array_unique() : hàm dùng để loại bỏ các giá trị trùng nhau, áp dụng cho việc lọc keyword, cũng như lọc url_id lấy được trong tìm kiếm.
- count() : dùng để đếm trong mảng có bao nhiêu giá trị.
- log() : dùng để tính toán, áp dụng cho việc tính toán các trọng số trong thuật toán tf-idf.
- pow() : đây là hàm tính toán cho lũy thừa cho các con số.
- sqrt() : hàm tính căn bậc 2 cho các con số.

2.5. Thư viện mã nguồn mở simple_html_dom.php

Để viết thành công 1 ứng dụng ngoài các kiến thức cơ bản cần biết, ta cũng không thể bỏ qua những thư viện mã nguồn mở được ứng dụng rộng rãi, trong ứng dụng máy tìm kiếm thì thư viện simple_html_dom.php là thư viện hoàn hảo.

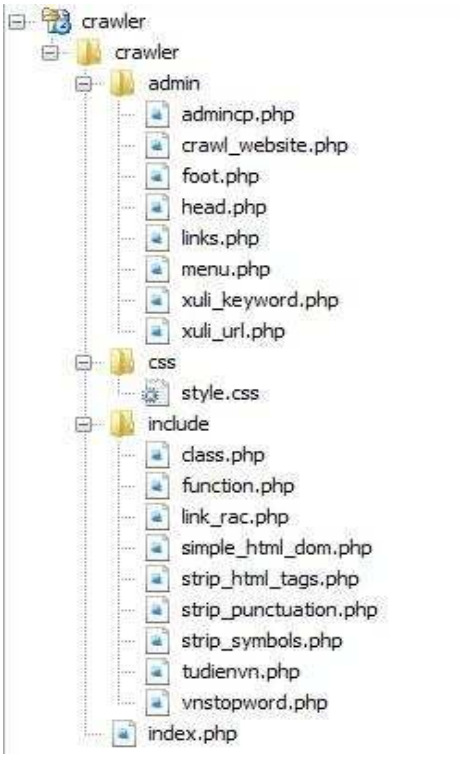
Dưới đây là một ví dụ cho việc ứng dụng mã nguồn mở này

```
1.// chèn thư viện simple_html_dom.php vào
2.Require_once(simple_html_dom.php);
3.// tạo một một kết nối để mở nội dung từ 1 link or 1 file
4.$html = file_get_html('http://www.google.com/');
5.// tìm tất cả hình ảnh có trong nội dung
6.foreach($html->find('img') as $element)
7.      echo $element->src . '<br>';
8.// tìm tất cả link có trong nội dung.
9.foreach($html->find('a') as $element)
10.     echo $element->href . '<br>';
```

Như vậy với thư viện hỗ trợ, thì ta chỉ cần viết vài dòng là có thể lọc lấy toàn bộ những gì chúng ta cần trong nội dung 1 link hay 1 file tài liệu.

2.6. Cấu trúc tổ chức của máy tìm kiếm

Trước khi viết ứng dụng tìm kiếm, ta tổ chức cách lưu trữ sắp xếp sao cho hợp lí. Hình dưới đây là cấu trúc tổ chức của ứng dụng.



Hình 2. 2 - Cấu trúc của ứng dụng máy tìm kiếm chụp từ phpDesigner

Cấu trúc máy tìm kiếm		
TT	tên	Ý nghĩa
Thư mục admin		
1	admincp.php	Trang quản lí chính của máy tìm kiếm. tất cả các trang khi được gọi, sẽ đều chạy và hiển thị nội dung

		sau khi xử lý xong tại trang này.
2	crawl_website.php	Trang có nhiệm vụ thu thập tất cả các đường link thêm vào CSDL từ một seedURL nhập vào.
3	foot.php	trang này hiển thị phần cuối cùng, dùng để ghi những thông tin cuối cùng mà ta muốn mọi người biết, ví dụ như: bản quyền,…
4	head.php	Đây là phần đầu trang, dùng để trang trí, có thể gọi là banner.
5	links.php	Khi click vào 1 link nào đó trên web, trang này có nhiệm vụ gọi trang mà link yêu cầu xử lý, rồi hiển thị vào phần nội dung của trang admincp.php
6	menu.php	Hiển thị menu của trang, chứa các đường link dẫn đến các trang xử lý khác.
7	xuli_keyword.php	Trang xử lý các phần keyword của 1 link sau khi link đó được duyệt trong thuật toán chiều sâu.

8	xuli_url.php	Đây là trang xử lý các link được thêm vào, chưa được duyệt, hay nói đúng hơn, trang này là trang chạy thuật toán duyệt theo chiều sâu.
9	Thư mục css	
10	Style.css	Dùng để trang trí trên toàn trang, mà khi ta muốn chỉnh sửa hay thay đổi phần hiển thị, ta chỉ cần sửa tại trang này là được.
Thư mục include		
1	class.php	Thư viện gọi kết nối cũng như thực hiện các câu truy vấn đến CSDL, được viết theo hướng đối tượng để thuận tiện cho việc viết ứng dụng.
2	function.php	Chứa các hàm xử lý các tác vụ, thuật toán chạy của ứng dụng tìm kiếm.
3	link_rac.php	Dùng để loại bỏ các link chi tiết khi các link lấy về.
4	simple_html_dom.php	Thư viện mã nguồn mở, hỗ trợ ứng dụng máy tìm kiếm
5	strip_html_tags.php	Hàm loại bỏ các thẻ html

		trong nội dung tin tức lấy về
6	strip_punctuation.php	Hàm loại bỏ các dấu chấm câu, dấu phẩy…
7	strip_symbol.php	Hàm loại bỏ các ký tự đặc biệt trong nội dung
8	tudienvn.php	Đây là phần từ điển chứa Tiếng Việt lấy từ trang wikic, chứa 24717 từ có 2 tiếng, dùng để xác định các keyword 2 từ có nghĩa của Tiếng Việt.
9	vnstopword.php	Thư viện chứa các từ ít nghĩa, xuất hiện liên tục trong các nội dung văn bản.
1	index.php	Đây là trang giao diện chính, cũng là trang để hiển thị kết quả tìm kiếm được đến người dùng.

2.7. Trình thu thập web - Crawler

Trình thu thập web – crawler sẽ nhận seedURL và tiến hành tìm kiếm tất cả các link có được từ việc phân tích nội dung của seedURL.

```php
<?php
// bắt đầu crawl với url bất kì được đưa vào
function crawl_site($u)
{
    require_once('simple_html_dom.php'); // thư viện html_dom nổi tiếng, đây là thư viện mã nguồn mở
    $crawled_urls=array(); // khai báo mảng
    $found_urls=array(); // khai báo mảng
    global $crawled_urls;
    $uen=urlencode($u);
    if(array_key_exists($uen,$crawled_urls)==0 ||
    $crawled_urls[$uen] < date("YmdHis",strtotime('-25 seconds', time())))
    {
        $domain = get_domain($u); // lấy domain của $u
        $html = file_get_html($u);
        $crawled_urls[$uen]=date("YmdHis");
        foreach($html->find("a") as $li)
        {
            if(get_domain($li->href) == $domain)
            {
                $url=perfect_url($li->href,$u);
                $enurl=urlencode($url); // mã hóa các dau đặc biệt như dau _ thành %
                if($url!='' && substr($url,0,4)!="mail" && substr($url,0,4)!="java"
                  && array_key_exists($enurl,$found_urls)==0) // kiểm tra key url có bằng ko hay ko
                {
                    $found_urls[] = $url;
                }
            }
        }
    }
    return $found_urls;
} // kết thúc function
?>
```

Hình 2. 3 - Một phần mã crawl_site() chụp từ PhpDesigner

Đoạn mã trên sẽ thu thập toàn bộ link của trang web đưa vào, và chuyển đổi các link từ tương đối(các link không rõ ràng) sang các link tuyệt đối rồi lưu vào mảng. Link lấy đầu tiên này có thể gọi là link gốc hay còn gọi là hạt nhân đầu tiên.

Sau khi các link được lấy về, sẽ được đi qua bộ lọc các link rác trước khi thêm vào CSDL và được gán cho flag = 0 trong bảng urls.

Lúc này, ta có thể dùng vòng lặp để tiếp tục lấy các link từ trong bảng urls ra để xử lý tiếp, hoặc ta có thể chọn giải pháp dừng lại.

Từ cách thức chủ động trên, ta viết tiếp trang xuli_url.php là thuật toán tìm kiếm theo chiều sâu bằng cách dựa vào cờ flag = 0 trong bảng urls để lấy các link có cờ flag = 0 ra để xử lý tiếp, theo thuật toán tìm kiếm theo chiều sâu thì vào trước ra sau, vậy thì ta sẽ lấy các id ngược từ dưới lên mới nhất với flag = 0, ra để xử lý và gọi hàm crawl_site() xử lý link này, ta sẽ thu thập

được các link mới với flag=0, ta thêm các link thu thập được vào CSDL và đánh dấu link vừa xử lý với flag = 1 để không bị gọi xử lý lại lần nữa.

Với cách thức như trên, ta đã hoàn thành việc thu thập 1 trang web theo thuật toán chiều sâu 1 cách chủ động mà không làm cho máy tìm kiếm bị quá tải, tránh bị treo máy, hay bị chính trang web mà chúng ta thu thập ngắt kết nối của chúng ta do chúng ta gửi quá nhiều yêu cầu trong một thời gian ngắn.

2.8. Lập chỉ mục index

Với các link đã được duyệt với cờ flag = 1, ta sẽ lấy ra từ CSDL để xử lý nội dung của trang link đó.

Trước khi xử lý chính xác những gì chúng ta cần lấy, ta cần phân tích nội dung trang. Thông qua trình duyệt firefox hoặc internet explorer để xem các đoạn mã bên trong 1 trang web.

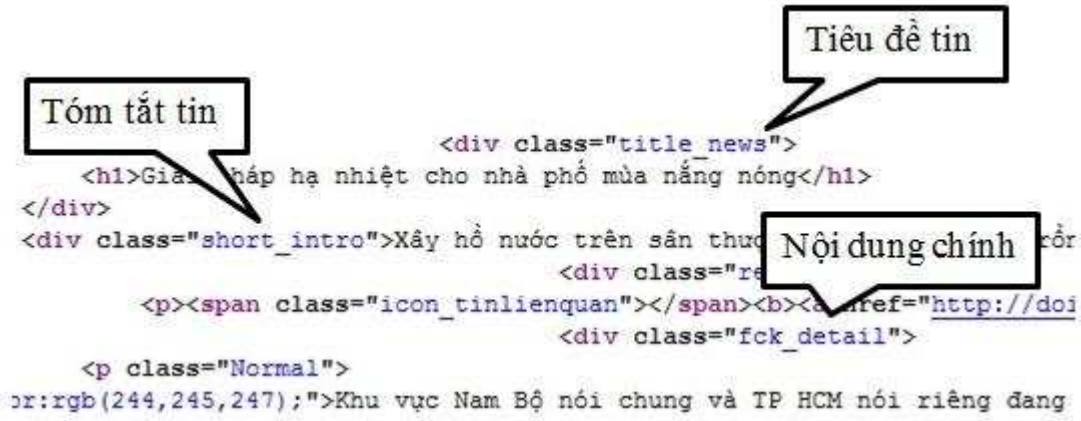
Ví dụ: phân tích 1 trang tin chi tiết của trang vnexpress.net

Hình 2. 4 - Ảnh được cắt từ trang vnexpress.net từ trình duyệt

Phần tiêu đề tin, tóm tắt tin, nội dung tin, là các phần mà ta sẽ lấy.

Tiếp đến, ta tiếp tục bấm ctrl+u để xem phần mã của trang tin chi tiết này.

Với phần này, ta kéo xuống để tìm phần thể hiện của đoạn mã như trong hình 2.5 dưới đây:



Hình 2. 5 - Ảnh chỉ rõ các mã chứa phần text cần lấy

Như vậy, với phần mà ta đã xem xét, thì ta thấy rằng:

- Thẻ : là thẻ chứa tiêu đề tin mà ta cần lấy.
- Thẻ : là thẻ chứa phần tóm tắt của tin.
- Thẻ : là thẻ chứa nội dung chính của tin.

Để lấy được các phần mà ta cần lấy, ta sẽ dùng các biến lưu các thẻ mà ta cần lấy, và một mảng được khai báo để lưu các nội dung mà ta lấy về.

Hình dưới đây sẽ cho ta thấy rõ điều đó hơn.

```
35 $item=array(); // khai báo mảng để chứa các phần tin
36
37 $title = "div.title_news";
38 $desc = "div.short_intro";
39 $content = "div.fck_detail";
40
41 $item['title'] = laythongtin($title,$url_name);
42 $item['desc'] = laythongtin($desc,$url_name);
43 $item['content'] = laythongtin($content,$url_name);
```

Hình 2. 6 - Đoạn mã gán thẻ cần lấy chụp từ PhpDesigner

Như vậy, sau khi khai báo xong, ta sẽ truyền các biến này vào hàm laythongtin() ra để lấy các phần của trang web mà ta định lấy.

Hàm lấy thông tin được thể hiện như hình chụp đoạn mã được viết dưới đây, sẽ cho ta hiểu rõ hơn về cách dùng thư viện simple_html_dom.php

```
198 // bắt đầu lấy thông tin của trang web cần xử lí
199 function laythongtin($data,$url)
200 {
201     // thư viện html_dom nổi tiếng, đây là thư viện mã nguồn mở
202     require_once('../include/simple_html_dom.php');
203     $html = file_get_html($url);
204     // tìm đến tất cả các phần có tag h1 với class=title lấy nội dung bên trong nó
205     foreach($html->find($data) as $element){
206         $item = $element->innertext;
207     }
208     if(isset($item))
209     {
210         return $item; // nếu có giá trị thì trả về.
211     }
212     else
213     {
214         return null; // nếu ko thì tra về null
215     }
216 }
```

Hình 2. 7 - Đoạn mã lấy thông tin chụp từ PhpDesigner

Như vậy, ta đã lấy về được các phần cần thiết để xử lý keyword.

Sau khi lấy về ta sẽ đưa phần nội dung này vào bộ lọc gồm:

- strip_html_tags : loại bỏ các thẻ html còn sót lại trong phần tin.
- strip_punctation : lọc các dấu _, ‘’, “”,…
- strip_symbol : lọc các ký tự đặc biệt.

Với phần tin đã được sàng lọc, chỉ còn lại các từ, ta sẽ đưa phần tin này vào hàm preg_split() để tạo ra mảng 1 chiều.

```
preg_split("/[\s,.?!]+/u",
```

(2.1)

Từ mảng có từ khóa 1 từ, ta truyền vào hàm tao2keyword() mà ta viết trong ứng dụng máy tìm kiếm để tạo ra mảng có từ khóa 2 từ.

Hình dưới đây sẽ cho ta thấy cách tạo mảng 2 từ khóa là như thế nào:

```
103 function tao2keyword($mang1)
104 {
105     // tạo mảng 2 keyword từ mảng 1
106     $n = count($mang1); // đếm có bao nhiu phần tử trong mảng
107     for($i=0;$i<$n-1;$i++) {
108         $mang2[] = $mang1[$i]." ".$mang1[$i+1];
109     }
110     return $mang2; // mảng 2 keyword được tạo
111 }
```

Hình 2. 8 - Đoạn mã tạo từ khóa có 2 từ chụp từ PhpDesigner

Khi có được mảng từ khóa 1 từ và mảng từ khóa 2 từ, ta dùng các hàm so sánh mảng để so sánh với bảng tudienvn và vnstopword để loại bỏ các từ rác, ít nghĩa. Xử lý xong, ta thêm vào bảng words, từ nào có rồi thì không thêm nữa. Mà thay vào đó, ta tiến hành cập nhật idf cho từ đó trong bảng words và tf trong bảng words_urls. Từ nào chưa có thì thêm vào, đồng thời thêm trọng số tf của từ thuộc link đang xử lý vào bảng words_urls. Lưu ý là Trước khi thêm vào ta tính trọng số idf cho các từ trước khi thêm vào.

2.9. Thuật toán tìm kiếm tf-idf

Như đã trình bày ở chương 1, khi người dùng nhập câu truy vấn vào, thì ta sẽ tạo ra mảng từ khóa có 1 từ, rồi từ mảng ban đầu, ta tạo ra mảng từ khóa có 2 từ. Sau đó, dùng bộ tudienvn và vnstopword để loại bỏ các từ rác và ít nghĩa trong câu truy vấn.

Ta có mảng chứa câu truy vấn, từ mảng câu truy vấn, ta tìm được các id của các đường link.

Sau đó, ta sẽ so sánh và đối chiếu để tính toán vector tf-idf của từng link và tính vector tf-idf của câu truy vấn, rồi dùng công thức tính CosSim để sắp xếp các link theo giá trị CosSim giảm dần và trả lại mảng chứa id các đường link.

Và kết quả đạt được là CosSim của từng link, rồi ta dùng hàm sắp sếp mảng giảm dần để sắp xếp mảng CosSim.

Như vậy là ta đã có kết quả tìm kiếm được tính theo mô hình không gian vector theo phương pháp tf-idf.

Hình dưới đây thể hiện việc tính toán của thuật toán cosSim.

```
208     foreach($murl as $klu=>$vlu)
209     {
210         $tamcosSim = 0;
211         foreach($mq as $newkq2=>$newvq2)
212         {
213             $sql = '';
214             $sql = "select * from words where word='".$newkq2."'";
215             $fash->query($sql);
216             $num = $fash->count_rows();
217             if($num>0)
218             {
219                 $row = $fash->fetch();
220                 $sql2 = '';
221                 $sql2 = "select * from words_urls where FK_word_id= '"
222                 .$row['word_id']."' and FK_url_id= '".$klu."'";
223                 $fash->query($sql2);
224                 $num2 = $fash->count_rows();
225                 if($num2>0)
226                 {
227                     $row2 = $fash->fetch();
228                     $tamcosSim = $tamcosSim + $row2['tf_wordurl']*$newvq2;
229                 }
230             }
231         }
232         // đưa cosSim tính được của từng url vào mảng
233         $cosSim_url[$klu]= ($tamcosSim/($vlu*$leng_q));
234     }
235 } // kết thúc hàm function
```

Hình 2. 9 - Đoạn code tính cosSim chụp từ màn hình PhpDesigner

2.10. Tổng kết chương 2

Chương 2 của luận văn, tác giả trình bày về cách thức xây dựng máy tìm kiếm bao gồm:

Trình bày, các bảng trong CSDL, phân tích tác dụng của từng cột trong các bảng, cũng như phân tích các mối quan hệ trong các bảng.

Trình bày thuật toán thu thập các liên kết, duyệt các liên kết theo chiều sâu, cách lấy nội dung 1 trang tin chi tiết, thuật toán xử lý từ khóa, phương pháp sắp xếp tf-idf.

CHƯƠNG 3

THỰC NGHIỆM

Sau khi hoàn thành ứng dụng, tôi đã thử nghiệm với trang web http://vnexpress.net/.

3.1. Mô tả ứng dụng máy tìm kiếm

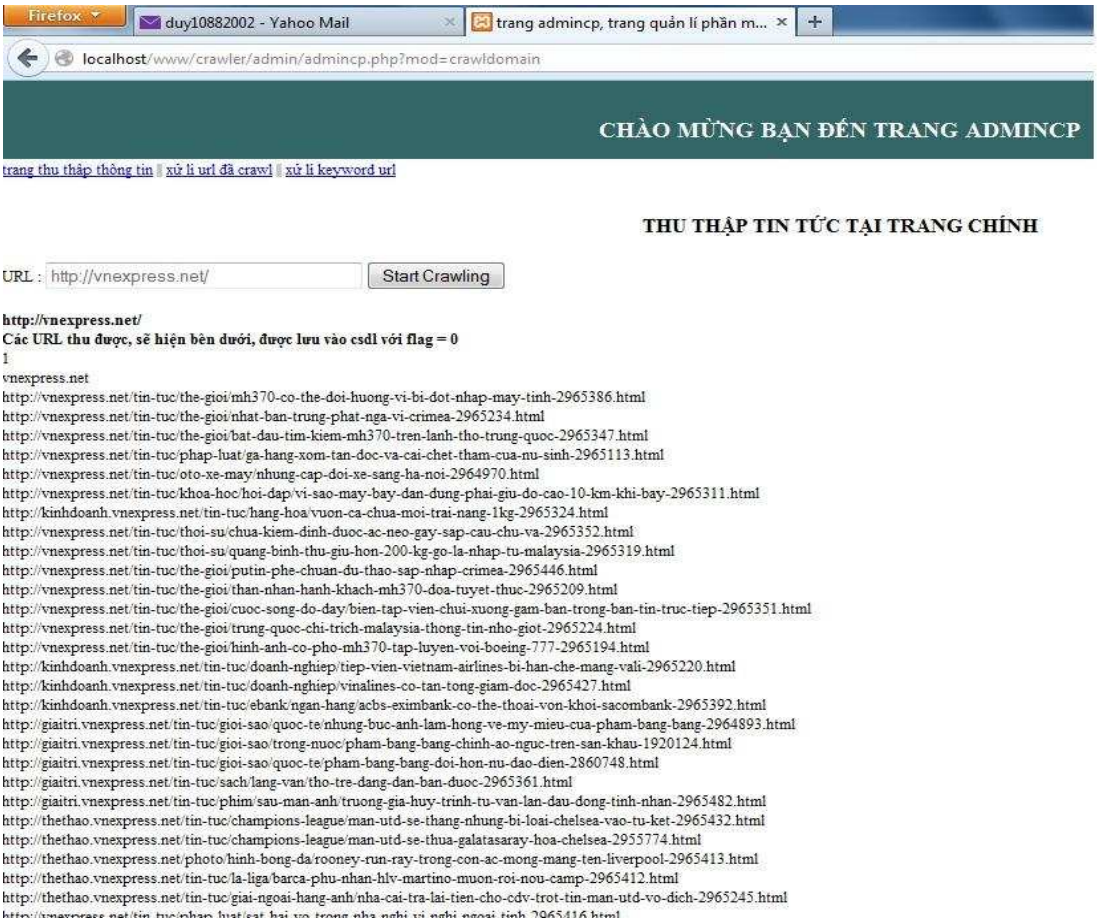
Chương trình có các phần chính sau:

STT	Tên menu	Chức năng
1	Trang thu thập thông tin	Có nhiệm vụ thu thập tiếp nhận seedURL đầu tiên, để làm cơ sở cho việc thu thập các link tiếp theo.
2	Xử lý url đã crawl	Có nhiệm vụ lần theo dấu vết các link thu thập được, để tiếp tục lấy được các link sâu hơn bên trong trang web, xử lý các link có flag=0
3	Xử lý keyword url	Có nhiệm vụ thu thập các từ khóa từ nội dung của link đã được duyệt qua.
4	List url hiện có	Liệt kê tất cả các link thu thập được, không phân biệt trạng thái của từng link.
5	List url chưa duyệt	Liệt kê tất cả các link chưa được duyệt có flag = 0.
6	List url đã duyệt	Liệt kê tất cả các link đã duyệt, có flag=1
7	List url đã xử lý keyword	Liệt kê tất cả các link đã xử lý keyword, có flag = 2

8	List keyword thu thập được	Liệt kê tất cả các keyword thu thập được từ nội dung các link.
9	List url không thể xử lý keyword	Nghiên cứu xem xét các link tại sao không thể thu thập được keyword, để tìm hướng giải quyết, phát triển chương trình hơn nữa.

3.2. Kết quả đạt được

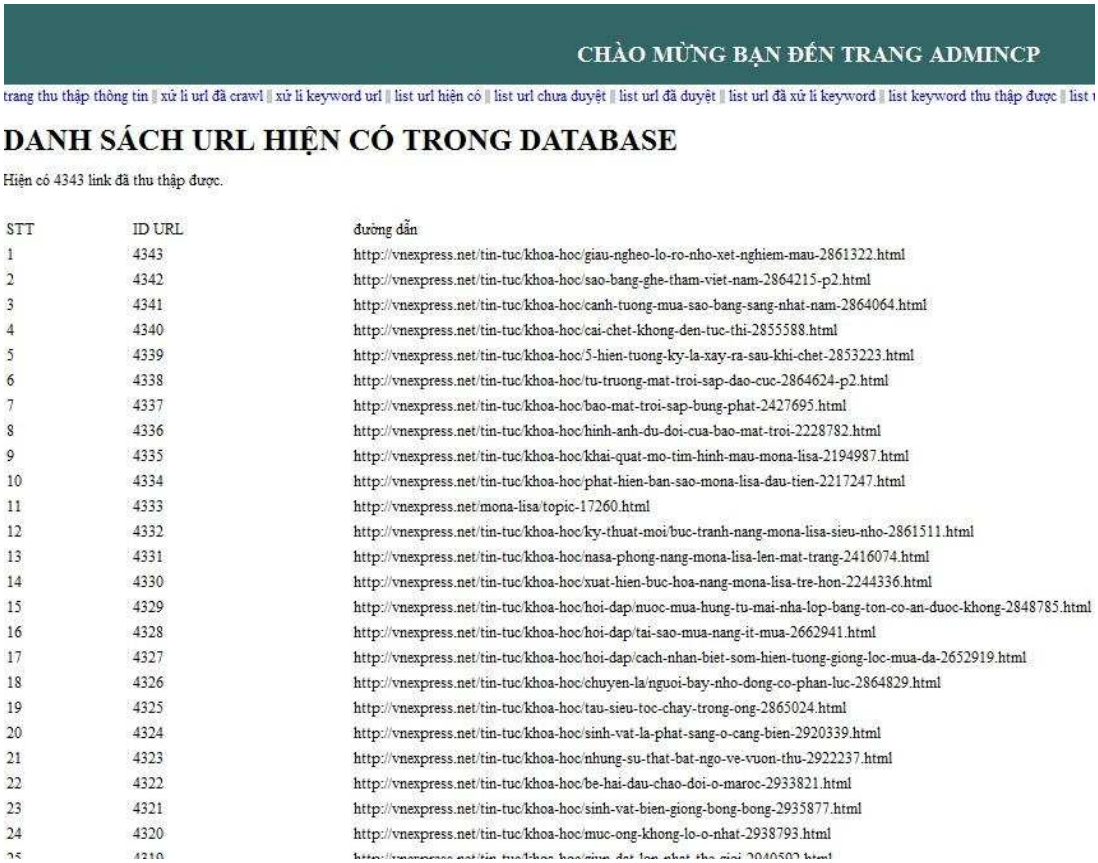
Và kết quả đạt được từ khi ta thu thập các link từ trang link đầu tiên là 142 link đã được lấy từ trang chủ của trang vnexpress.



Hình 3. 1 - Danh sách link lấy được từ link đầu tiên vnexpress

Từ những link đầu tiên này, ta tiếp tục thu thập với thuật toán duyệt theo chiều sâu, thì trong vòng 90 phút, với việc thu thập chủ động, ta đã thu thập được là 4343 link.

Hình chụp từ màn hình ứng dụng tìm kiếm, cho ta thấy điều đó



CHÀO MỪNG BẠN ĐẾN TRANG ADMINCP

trang thu thập thông tin ‖ xử li url đã crawl ‖ xử li keyword url ‖ list url hiện có ‖ list url chưa duyệt ‖ list url đã duyệt ‖ list url đã xử li keyword ‖ list keyword thu thập được ‖ list

DANH SÁCH URL HIỆN CÓ TRONG DATABASE

Hiện có 4343 link đã thu thập được.

STT	ID URL	đường dẫn
1	4343	http://vnexpress.net/tin-tuc/khoa-hoc/giau-ngheo-lo-ro-nho-xet-nghiem-mau-2861322.html
2	4342	http://vnexpress.net/tin-tuc/khoa-hoc/sao-bang-ghe-tham-viet-nam-2864215-p2.html
3	4341	http://vnexpress.net/tin-tuc/khoa-hoc/canh-tuong-mua-sao-bang-sang-nhat-nam-2864064.html
4	4340	http://vnexpress.net/tin-tuc/khoa-hoc/cai-chet-khong-den-tuc-thi-2855588.html
5	4339	http://vnexpress.net/tin-tuc/khoa-hoc/5-hien-tuong-ky-la-xay-ra-sau-khi-chet-2853223.html
6	4338	http://vnexpress.net/tin-tuc/khoa-hoc/tu-truong-mat-troi-sap-dao-cuc-2864624-p2.html
7	4337	http://vnexpress.net/tin-tuc/khoa-hoc/bao-mat-troi-sap-bung-phat-2427695.html
8	4336	http://vnexpress.net/tin-tuc/khoa-hoc/hinh-anh-du-doi-cua-bao-mat-troi-2228782.html
9	4335	http://vnexpress.net/tin-tuc/khoa-hoc/khai-quat-mo-tim-hinh-mau-mona-lisa-2194987.html
10	4334	http://vnexpress.net/tin-tuc/khoa-hoc/phat-hien-ban-sao-mona-lisa-dau-tien-2217247.html
11	4333	http://vnexpress.net/mona-lisa/topic-17260.html
12	4332	http://vnexpress.net/tin-tuc/khoa-hoc/ky-thuat-moi/buc-tranh-nang-mona-lisa-sieu-nho-2861511.html
13	4331	http://vnexpress.net/tin-tuc/khoa-hoc/nasa-phong-nang-mona-lisa-len-mat-trang-2416074.html
14	4330	http://vnexpress.net/tin-tuc/khoa-hoc/xuat-hien-buc-hoa-nang-mona-lisa-tre-hon-2244336.html
15	4329	http://vnexpress.net/tin-tuc/khoa-hoc/hoi-dap/nuoc-mua-hung-tu-mai-nha-lop-bang-ton-co-an-duoc-khong-2848785.html
16	4328	http://vnexpress.net/tin-tuc/khoa-hoc/hoi-dap/tai-sao-mua-nang-it-mua-2662941.html
17	4327	http://vnexpress.net/tin-tuc/khoa-hoc/hoi-dap/cach-nhan-biet-som-hien-tuong-giong-loc-mua-da-2652919.html
18	4326	http://vnexpress.net/tin-tuc/khoa-hoc/chuyen-la/nguoi-bay-nho-dong-co-phan-luc-2864829.html
19	4325	http://vnexpress.net/tin-tuc/khoa-hoc/tau-sieu-toc-chay-trong-ong-2865024.html
20	4324	http://vnexpress.net/tin-tuc/khoa-hoc/sinh-vat-la-phat-sang-o-cang-bien-2920339.html
21	4323	http://vnexpress.net/tin-tuc/khoa-hoc/nhung-su-that-bat-ngo-ve-vuon-thu-2922237.html
22	4322	http://vnexpress.net/tin-tuc/khoa-hoc/be-hai-dau-chao-doi-o-maroc-2933821.html
23	4321	http://vnexpress.net/tin-tuc/khoa-hoc/sinh-vat-bien-giong-bong-bong-2935877.html
24	4320	http://vnexpress.net/tin-tuc/khoa-hoc/muc-ong-khong-lo-o-nhat-2938793.html

Hình 3. 2 - Danh sách link thu được trong 90 phút

Cùng lúc đó, thì số link đã duyệt qua là 1000 link (gồm có 4 link đã được xử lý từ khóa).

CHÀO MỪNG BẠN ĐẾN TRANG ADMINCP

trang thu thập thông tin | xử lí url đã crawl | xử lí keyword url | list url hiện có | list url chưa duyệt | list url đã duyệt | list url đã xử lí keyword | list keyword thu thập được | list url không thể xử lí keyword

DANH SÁCH URL ĐÃ DUYỆT

Hiện có 996 link đã được duyệt.

STT	ID URL	đường dẫn	title	ngày crawl	flag
1	5	http://vnexpress.net/tin-tuc/giao-duc/than-dong-tin-hoc-13-tuoi-gianh-hang-chuc-giai-thuong-2966532.html		21/03/2014	1
2	6	http://vnexpress.net/tin-tuc/phap-luat/de-xuat-them-chu-the-toi-hiep-dam-la-nu-2966158.html		21/03/2014	1
3	7	http://giaitri.vnexpress.net/tin-tuc/gioi-sao/quoc-te/bi-kich-cua-co-be-13-tuoi-bi-cha-giet-vi-cuong-than-tuong-2966822.html		21/03/2014	1
4	8	http://vnexpress.net/tin-tuc/the-gioi/than-nhan-hanh-khach-mh370-tuyet-thuc-2966797.html		21/03/2014	1
5	9	http://kinhdoanh.vnexpress.net/tin-tuc/doanh-nghiep/giam-doc-doanh-nghiep-lan-dan-tim-viec-2966708.html		21/03/2014	1
6	10	http://thethao.vnexpress.net/tin-tuc/cac-giai-khac/tottenham-bi-loai-ngoai-hang-anh-sach-bong-o-europa-league-2966769.html		21/03/2014	1
7	11	http://giaitri.vnexpress.net/tin-tuc/gioi-sao/quoc-te/ve-diu-ngot-mot-thuo-cua-sao-moi-tinh-dau-choi-ji-woo-2966609.html		21/03/2014	1
8	12	http://vnexpress.net/tin-tuc/the-gioi/my-ra-don-kinh-te-voi-nga-2966745.html		21/03/2014	1
9	13	http://vnexpress.net/tin-tuc/oto-xe-may/xe-do-ham-ho-ktm-duke-125-o-sai-gon-2966396.html		21/03/2014	1
10	14	http://kinhdoanh.vnexpress.net/tin-tuc/ebank/ngan-hang/vua-tien-mat-doi-chien-luoc-khi-lai-suat-giam-2966714.html		21/03/2014	1
11	15	http://vnexpress.net/tin-tuc/the-gioi/malaysia-van-tim-kiem-mh370-o-ca-hai-hanh-lang-bay-2966710.html		21/03/2014	1
12	16	http://vnexpress.net/tin-tuc/thoi-su/taxi-du-thu-cuoc-khach-nuoc-ngoai-dat-gap-20-lan-2966855.html		21/03/2014	1
13	17	http://vnexpress.net/tin-tuc/thoi-su/nu-tai-xe-taxi-doi-1-trieu-dong-cho-quang-duong-3-km-2953818.html		21/03/2014	1
14	18	http://vnexpress.net/tin-tuc/thoi-su/nganh-y-that-chat-an-ninh-sau-vu-bat-coc-tre-em-2966878.html		21/03/2014	1
15	19	http://vnexpress.net/tin-tuc/thoi-su/tp-hcm-lai-kien-nghi-khong-xay-san-bay-long-thanh-2966630.html		21/03/2014	1
16	20	http://vnexpress.net/tin-tuc/thoi-su/nu-sinh-bi-oto-dam-o-xa-dan-da-tinh-lai-2966703.html		21/03/2014	1
17	21	http://vnexpress.net/tin-tuc/thoi-su/10-nguoi-ngo-doc-nam-nguy-ki-ch-vi-khong-co-gan-de-ghep-2966740.html		21/03/2014	1

Hình 3. 3 - 1000 link đã được duyệt trong 90 phút

Với 50s cho 1 lần thu thập các từ khóa trên 1 đường link, thì việc thu thập từ khóa là 1 thách thức thật sự. Như vậy, để có thể thu thập hết từ khóa từ 4343 link thu thập được, thì ta phải mất khoảng 60h30’.

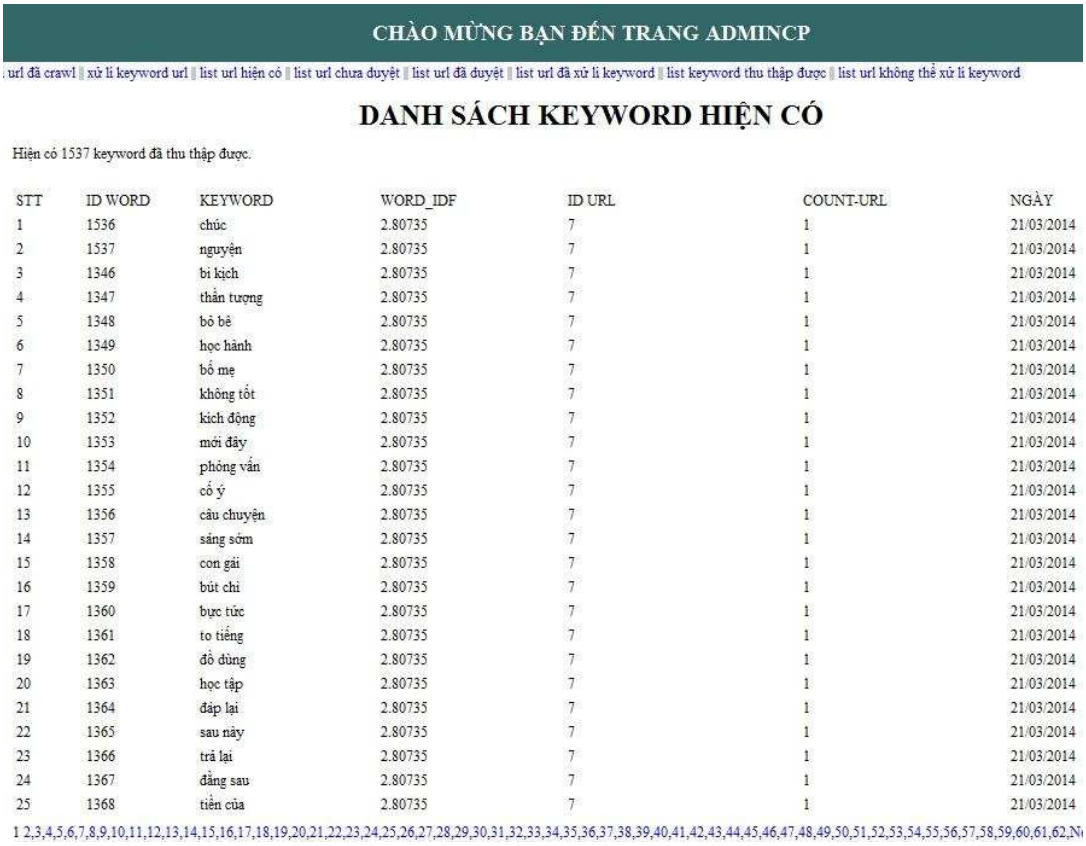
Ảnh dưới đây minh họa cho kết quả lọc keyword trên 1 đường link.

Các key word thu thập được 421

Array ([bi kịch] => 1 [thần tượng] => 1 [bỏ bê] => 1 [học hành] => 1 [bố mẹ] => 1 [không tốt] => 1 [kích động] => 1 [trung quốc] => 1 [mới đây] => 1 [phỏng vấn] => 1 [cố ý] => 1 [giết người] => 1 [câu chuyện] => 1 [gia đình] => 1 [bắt đầu] => 1 [sáng sớm] => 1 [con gái] => 1 [bút chì] => 1 [bực tức] => 1 [to tiếng] => 1 [đồ dùng] => 1 [học tập] => 1 [điện thoại] => 1 [di động] => 1 [đáp lại] => 1 [sau này] => 1 [trả lại] => 1 [tiết lộ] => 1 [đằng sau] => 1 [tiền của] => 1 [ham mê] => 1 [ngôi sao] => 1 [như vậy] => 1 [ảnh hưởng] => 1 [câu nói] => 1 [mâu thuẫn] => 1 [ban đầu] => 1 [tiếp tục] => 1 [tranh cãi] => 1 [sau đó] => 1 [trại giam] => 1 [thừa nhận] => 1 [đau lòng] => 1 [nghĩ rằng] => 1 [làm gì] => 1 [phòng không] => 1 [ăn uống] => 1 [ngày càng] => 1 [quần áo] => 1 [mê mẩn] => 1 [tranh ảnh] => 1 [liên quan] => 1 [ngoài ra] => 1 [tham gia] => 1 [bận rộn] => 1 [tổ chức] => 1 [hoạt động] => 1 [theo dõi] => 1 [tình hình] => 1 [cho phép] => 1 [người khác] => 1 [máy tính] => 1 [cá nhân] => 1 [ra đi] => 1 [thậm chí] => 1 [thậm tệ] => 1 [nghỉ hè] => 1 [vấn đề] => 1 [thần kinh] => 1 [tỏ ra] => 1 [ít khi] => 1 [tại sao] => 1 [không có] => 1 [có một] => 1 [có người] => 1 [đam mê] => 1 [vô ích] => 1 [thành viên] => 1 [hiện nay] => 1 [tạp chí] => 1 [thứ năm] => 1 [năm trong] => 1 [danh sách] => 1 [quyền lực] => 1 [biểu diễn] => 1 [nhân dân] => 1 [thu nhập] => 1 [tuy nhiên] => 1 [nhất quyết] => 1 [chủ động] => 1 [dự định] => 1 [tiền thế] => 1 [cuối cùng] => 1 [bà ngoại] => 1 [thỏa lòng] => 1 [mong ước] => 1 [thế giới] => 1 [chỉ có] => 1 [cuộc đời] => 1 [kết thúc] => 1 [cô gái] => 1 [khấu đầu] => 1 [theo đuổi] => 1 [giấc mơ] => 1 [giao lưu] => 1 [dừng lại] => 1 [nói chuyện] => 1 [tự tử] => 1 [di chúc] => 1 [yêu cầu] => 1 [nguyện vọng] => 1 [diễn viên] => 1 [từ chối] => 1 [kịch] => 1 [tuổi] => 1 [giết] => 1 [cuồng] => 1 [thần] => 1 [tượng] => 1 [tiểu] => 1 [nhóm] => 1 [bỏ] => 1 [học] => 1 [hành] => 1 [màng] => 1 [ngủ] => 1 [nghe] => 1 [bố] => 1 [mẹ] => 1 [tốt] => 1 [kich] => 1 [động] => 1 [chết] => 1 [tờ] => 1 [fawan] => 1 [trung] => 1 [quốc] => 1 [đăng] => 1 [bài] => 1 [phỏng] => 1 [vấn] => 1 [khải] => 1 [khởi] => 1 [tố] => 1 [hôm] => 1 [tội] => 1 [cố] => 1 [câu] => 1 [chuyện] => 1 [lộ] => 1 [đinh] => 1 [bắt] => 1 [đầu] => 1 [cuối] => 1 [sáng] => 1 [sớm] => 1 [gái] => 1 [tìm] => 1 [gọt] => 1 [bút] => 1 [chì] => 1 [bực] => 1 [tức] => 1 [tiếng] => 1 [lấy] => 1 [gạt] => 1 [đồ] => 1 [dùng] => 1 [tập] => 1 [xuống] => 1 [nền] => 1 [chiếc] => 1 [điện] => 1 [thoại] => 1 [vỡ] => 1 [tát] => 1 [mắng] => 1 [mày] => 1 [mạng] => 1 [đêm] => 1 [chịu] => 1 [dậy] => 1 [tiêu] => 1 [tiền] => 1 [đáp] => 1 [thôi] => 1 [trả] => 1 [tiết] => 1 [đăng] => 1 [cái] => 1 [dịu] => 1 [giọng] => 1 [nổi] => 1 [cần] => 1 [đừng] => 1 [ngôi] => 1 [đẹp] => 1 [đẽ] => 1 [mấy] => 1 [vậy] => 1 [ảnh] => 1 [hưởng] => 1 [ngờ] => 1 [yêu] => 1 [gặp] => 1 [vạn] => 1 [đẩy] => 1 [khiến] => 1 [thêm] => 1 [mâu] => 1 [thuẫn] => 1 [bếp] => 1 [dọa] => 1 [tiếp] => 1 [tục] => 1 [tranh] => 1 [cãi] => 1 [dẫn] => 1 [kẻ] => 1 [liền] => 1 [chém] => 1 [cầm] => 1 [cửa] => 1 [cổ] => 1 [trái] => 1 [trại] => 1 [giam] => 1 [thừa] => 1 [đau] => 1 [lòng] => 1 [nghĩ] => 1 [chẳng] => 1 [sống] => 1 [đời] => 1 [nữa] => 1 [nhạc] => 1 [hàn] => 1 [lời] => 1 [kể] => 1 [khoảng] => 1 [vùi] => 1 [internet] => 1 [phòng] => 1 [buồn] => 1 [uống] => 1 [thức] => 1 [khuya] => 1 [càng] => 1 [đủ] => 1 [kiểu] => 1 [giày] => 1 [túi] => 1 [xách] => 1 [quần] => 1 [dán] => 1 [đầy] => 1 [xếp] => 1 [đĩa] => 1 [vợ] => 1 [mẫn] => 1 [liên] => 1 [quan] => 1 [ngoài] => 1 [tham] => 1 [hội] => 1 [suốt] => 1 [bận] => 1 [rộn] => 1 [tổ] => 1 [chức] => 1 [hoạt] => 1 [đối] => 1 [tình] => 1 [hình] => 1 [phép] => 1 [bước] => 1 [gần] => 1 [máy] => 1 [tính] => 1 [nhân] => 1 [hết] => 1 [cút] => 1 [chạm] => 1 [thậm] => 1 [chí] => 1 [tệ] => 1 [lúc] => 1 [nghỉ] => 1 [dưỡng] => 1 [mất] => 1 [màn] => 1 [thiết] => 1 [để] => 1 [kinh] => 1 [nhỏ] => 1 [tỏ] => 1 [thích] => 1 [gọi] => 1 [tự] => 1 [thể] => 1 [sợ] => 1 [trai] => 1 [khuyên] => 1 [bớt] => 1 [đam] => 1 [ích] => 1 [gồm] => 1 [thành] => 1 [viên] => 1 [nhỉ] => 1 [hiện] => 1 [tạp] => 1 [forbes] => 1 [thứ] => 1 [danh] => 1 [sách] => 1 [quyền] => 1 [lực] => 1 [showbiz] => 1 [bắc] => 1 [biểu] => 1 [diễn] => 1 [phần] => 1 [khích] => 1 [buổi] => 1 [dân] => 1 [triệu] => 1 [đồng] => 1 [vượt] => 1 [nhập] => 1 [nhiên] => 1 [quyết] => 1 [đòi] => 1 [chủ] => 1 [dự] => 1 [định] => 1 [quả] => 1 [hộ] => 1 [nghèo] => 1 [kiếm] => 1 [ngoại] => 1 [cháu] => 1 [thỏa] => 1 [mong] => 1 [ước] => 1 [đắm] => 1 [chìm] => 1 [giới] => 1 [cuộc] => 1 [kết] => 1 [thúc] => 1 [nhát] => 1 [khấu] => 1 [gợi] => 1 [nhớ] => 1 [lưu] => 1 [đức] => 1 [dương] => 1 [lệ] => 1 [quyên] => 1 [quê] => 1 [tỉnh] => 1 [túc] => 1 [nỗi] => 1 [bán] => 1 [thận] => 1 [đuổi] => 1 [giấc] => 1 [hong] => 1 [kong] => 1 [giao] => 1 [dừng] => 1 [tử] => 1 [chúc] => 1 [dài] => 1 [cầu] => 1 [ứng] => 1 [nguyện] => 1 [vọng] => 1 [chối] => 1 [hải] => 1)

Hình 3. 4 - Các từ khóa thu thập từ nội dung của 1 link

Và dưới đây là danh sách từ khóa thu thập từ nội dung các link, sẽ được hiển thị trong hình dưới đây.



CHÀO MỪNG BẠN ĐẾN TRANG ADMINCP

url đã crawl || xử lí keyword url || list url hiện có || list url chưa duyệt || list url đã duyệt || list url đã xử lí keyword || list keyword thu thập được || list url không thể xử lí keyword

DANH SÁCH KEYWORD HIỆN CÓ

Hiện có 1537 keyword đã thu thập được.

STT	ID WORD	KEYWORD	WORD_IDF	ID URL	COUNT-URL	NGÀY
1	1536	chúc	2.80735	7	1	21/03/2014
2	1537	nguyện	2.80735	7	1	21/03/2014
3	1346	bi kịch	2.80735	7	1	21/03/2014
4	1347	thần tượng	2.80735	7	1	21/03/2014
5	1348	bỏ bê	2.80735	7	1	21/03/2014
6	1349	học hành	2.80735	7	1	21/03/2014
7	1350	bố mẹ	2.80735	7	1	21/03/2014
8	1351	không tốt	2.80735	7	1	21/03/2014
9	1352	kích động	2.80735	7	1	21/03/2014
10	1353	mới đây	2.80735	7	1	21/03/2014
11	1354	phỏng vấn	2.80735	7	1	21/03/2014
12	1355	cố ý	2.80735	7	1	21/03/2014
13	1356	câu chuyện	2.80735	7	1	21/03/2014
14	1357	sáng sớm	2.80735	7	1	21/03/2014
15	1358	con gái	2.80735	7	1	21/03/2014
16	1359	bút chì	2.80735	7	1	21/03/2014
17	1360	bực tức	2.80735	7	1	21/03/2014
18	1361	to tiếng	2.80735	7	1	21/03/2014
19	1362	đồ dùng	2.80735	7	1	21/03/2014
20	1363	học tập	2.80735	7	1	21/03/2014
21	1364	đáp lại	2.80735	7	1	21/03/2014
22	1365	sau này	2.80735	7	1	21/03/2014
23	1366	trả lại	2.80735	7	1	21/03/2014
24	1367	đằng sau	2.80735	7	1	21/03/2014
25	1368	tiền của	2.80735	7	1	21/03/2014

1 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,N

Hình 3. 5 - Danh sách từ khóa thu thập được

Và để kiểm tra kết quả tìm kiếm có đúng như thuật toán tìm kiếm đã trình bày ở trên, ta sẽ chọn lấy 1 đường link đã được thu thập từ khóa để kiểm tra việc tính toán tìm kiếm có cho kết quả đúng như chúng ta mong muốn hay không.

Ta chọn ngẫu nhiên đường link như có tên sau:

http://vnexpress.net/tin-tuc/giao-duc/than-dong-tin-hoc-13-tuoi-gianh-hang-chuc-giai-thuong-2966532.html

Vào trình duyệt, ta đọc từ nội dung của link trên, và chọn lấy vài từ có trong nội dung mà ta nhớ. Ví dụ: với câu truy vấn " thần đồng tin học".

Ta sẽ có kết quả như trong hình sau:


tìm được từ câu truy vấn thần đồng tin học
có 481 kết quả.
thần đồng tin học tuổi giành hàng chục giải thưởng
thao tác thuần thục của thần đồng tin học ngoài thời ....thao tác thuần thục của thần đồng tin h
....phần mềm công nghệ tin học ấm trên giải thưởng ....

lưu đức hoa từ chối gặp cô gái mất cha
chỉ ôm ấp hình ảnh thần tượng lưu đức hoa ....hy vọng lưu đức hoa đồng ý gặp cô cô ....viên kiên

lưu đức hoa phải đền tiền vì làm hỏng trực thăng
hoa lương gia huy ngô thần quân hải lan ....số tiền tương đương tỷ đồng qq đưa ....kong phải bồi

những khoảnh khắc bikini thiêu đốt của candice swanepoel
trong studio thiên thần victoria....đường cong nóng bỏng xem tiếp song ngư ảnh ....

chi pu trẻ trung trong bộ ảnh mới
ảnh của nữ diễn viên thần tượng phủ sóng rộng ....cộng giúp người đẹp đốn tim fan chi pu ....hội

từ nhược tuyên hòa hợp với hai con riêng của chồng
mình chồng và hai thiên thần nhỏ đáng yêu tôi ....chồng dự định tổ chức tiệc cưới vào tháng ....

Hình 3. 6 - Kết quả tìm kiếm từ câu truy vấn "thần đồng tin học"

Ta thấy là với câu truy vấn trên, ta có được 481 link được tìm thấy và link mà ta chọn đã đứng ở vị trí đầu tiên trong kết quả tìm kiếm nhờ vào thuật toán tf-idf .

Với kết quả thu được khi thu thập thực tế từ trang vnexpress.net, ứng dụng máy tìm kiếm hoàn toàn có thể đáp ứng được việc thu thập thông tin cũng như hiển thị kết quả tìm kiếm chính xác đến cho người dùng.

3.3.Tổng kết chương 3

Chương 3 của luận văn, tác giả trình bày các kết quả đạt được của ứng dụng khi chạy trong thực tế bao gồm:

Kết quả thu được khi nhập seedURL đầu tiên là http://vnexpress.net/

Kết quả duyệt liên kết theo chiều sâu từ những liên kết đầu tiên thu thập được.

Kết quả các từ khóa thu thập được từ các liên kết.

Kết quả tìm kiếm với một câu tìm kiếm, được chọn từ 1 liên kết ngẫu nhiên để kiểm tra kết quả đạt được.

KẾT LUẬN VÀ KHUYẾN NGHỊ

1. Kết luận

Đề tài: "Nghiên cứu xây dựng máy tìm kiếm" được thực hiện nhằm nghiên cứu những phương pháp xây dựng máy tìm kiếm, có khả năng ứng dụng vào hoạt động tìm kiếm thông tin, trong một thế giới thông tin rộng lớn như hiện nay.

Về cơ bản, luận văn đã trình bày được những nội dung như sau:

Phương pháp thu thập các đường link của máy tìm kiếm.

Phương pháp duyệt link theo chiều sâu, chiều rộng, ngẫu nhiên.

Phương pháp xây dựng chỉ mục với từ khóa có 1 từ, và từ khóa có 2 từ, dựa vào từ điển Tiếng Việt được lập sẵn.

Mô hình không gian vector có rất nhiều phương pháp khác nhau được áp dụng rộng rãi trên nhiều lĩnh vực trong cuộc sống, chứ không riêng gì trong lĩnh vực tìm kiếm. Ví dụ như : sắp xếp các sản phẩm trong siêu thị, tìm kiếm đường đi, nhận diện khuôn mặt,…

Tóm lại, có thể mọi người thường dùng Google để tìm kiếm vì những gì Google làm được là không tưởng, nhưng không thể vì điều đó, mà chúng ta bỏ qua lĩnh vực tìm kiếm, chúng ta vẫn có thể viết ứng dụng tìm kiếm để áp dụng cho những lĩnh vực chuyên biệt như: chuyên về thông tin kinh tế, thể thao, ngân hàng…

2. Khuyến nghị

Để viết thành công 1 ứng dụng, cần có nhiều người tham gia nghiên cứu, thảo luận thì ứng dụng mới có thể đi vào cuộc sống, chứ riêng một cá nhân thì thật sự rất khó khăn.

Nên hướng phát triển của đề tài là tập trung nhiều chuyên gia để tham gia viết ứng dụng tìm kiếm hoàn hảo hơn, một ứng dụng tìm kiếm có thể hiểu được thói quen tìm kiếm, cũng như tập quán sử dụng ngôn ngữ của từng vùng miền trong nước, để từ đó, cho ra kết quả chính xác hơn.

Và một để ứng dụng chạy tốt thì không thể thiếu một hệ thống các máy chủ mạnh mẽ, để xử lý việc thu thập, lập index, cũng như việc trả kết quả tìm kiếm nhanh chóng chính xác đáp ứng được yêu cầu của người dùng.

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] Chu Bích Thu, Nguyễn Ngọc Trâm, Nguyễn Thị Thanh Nga, Nguyễn Thúy Khanh, Phạm Hùng Việt, (2002) T*ừ điể*n Tiếng Việt ph*ổ* thông, NXB TP. Hồ Chí Minh.

Tiếng Anh

[2] Sergey Brin and Lawrence Page**,** (1998), "The Anatomy of a Large-Scale Hypertextual Web Search Engine", Computer Networks, 30 (1-7), 107-117.

[3] Zhumin Chen, Jun Ma, Jingsheng Lei, Bo Yuan, Li Lian, Ling Song, (2008), A cross-language focused crawling algorithm based on multiple relevance prediction strategies, Elsevier Ltd, Netherlands.

[4] G.Al-Gaphari, (2008), "Building an efficient indexing for crawling the website with an efficient spider", International Journal of Information Science & Technology, 6 (2), 116-137.

[5] Gautam Pant, Padmini Srinivasan and Filipo Menczer, (2004), Crawling the Web, Springer, USA.

Các trang web

http://www.sirgroane.net/google-page-rank/

http://simplehtmldom.sourceforge.net/manual_api.htm

http://www.butchiso.com/2013/10/tim-hieu-ve-mo-hinh-khong-gian-vector.html

LÝ LỊCH TRÍCH NGANG

Họ và tên: Phùng Duy Vũ

Ngày tháng năm sinh: 22/02/1981 Nơi sinh: Hậu Giang

Địa chỉ liên lạc: F6/5A, đường Quách Điêu, ấp 6, Vĩnh Lộc A, Huyện Bình Chánh, Tp.HCM

QUÁ TRÌNH ĐÀO TẠO:

-Năm 2008: học liên thông đại học từ xa tại Học viện Bưu chính Viễn thông.

-Năm 2012: học cao học tại Học viện Kỹ thuật quân sự.

QUÁ TRÌNH CÔNG TÁC:

-Năm 2007-đến nay: làm việc tại Trường Đh Ngân Hàng Tp.HCM.

XÁC NHẬN QUYỂN LUẬN VĂN ĐỦ ĐIỆU KIỆN NỘP LƯU CHUYỂN

CHỦ NHIỆM KHOA (BỘ MÔN)	**CÁN BỘ HƯỚNG DẪN**
QUẢN LÝ CHUYÊN NGÀNH	(Ký và ghi rõ họ tên)
(Ký và ghi rõ họ tên)	