LỜI CẢM ƠN

Đầu tiên, chúng em xin gởi lời cảm ơn đến Thầy, Cô khoa Công nghệ Thông tin trường Đại học Khoa học Tự nhiên đã tận tình dạy dỗ, dìu dắt chúng em suốt bốn năm đại học.

Chúng em cảm ơn Cô Nguyễn Thị Diễm Tiên, người tận tình hướng dẫn, giúp đỡ, động viên chúng em hoàn thành luận văn này.

Chúng tôi cảm ơn các anh Trần Nguyễn Hoàng Phương, Bùi Ngọc Tuấn Anh, Đoàn Hữu Quang Vinh và các bạn Nguyễn Huy Hoàng, Phan Anh Đức đã giúp đỡ, đóng góp ý kiến cho chúng tôi trong quá trình cài đặt, thử nghiệm chương trình.

Cuối cùng, chúng con cảm ơn Ba, Mẹ và những người thân đã khích lệ, động viên chúng con trong thời gian học tập, nghiên cứu để có được thành quả như ngày nay.

Tháng 7 năm 2004

Sinh viên

Lê Thuý Ngọc – Đỗ Mỹ Nhung

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Ngày…… tháng……năm 2004

Ký tên

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Ngày…… tháng……năm 2004

Ký tên

MỤC LỤC

MỞ ĐẦU

Trong thời đại ngày nay, thông tin là nhu cầu thiết yếu đối với mọi người trên mọi lĩnh vực. Mỗi phút trôi qua hàng triệu triệu trang web được đẩy lên nhằm làm giàu nguồn tài nguyên vô tận này. Tuy nhiên tồn tại một nghịch lý là dù được ví như thư viện toàn cầu, internet vẫn không thoả mãn nhu cầu thông tin của con người. Xung quanh vấn đề này có nhiều nguyên nhân nhưng quan trọng nhất là sự thông hiểu giữa con người và công cụ tìm kiếm trên mạng – search engine – chưa đạt đến mức có thể giao tiếp tốt với nhau.

Hơn nữa, mỗi search engine sẽ mang đặc thù của ngôn ngữ mà nó hiển thị như search engine Tiếng Việt phải giải quyết những vấn đề đặc trưng của Tiếng Việt, cụ thể là vấn đề bảng mã, ngữ pháp trong Tiếng Việt.

Nếu ta hiểu cách thức search engine tổ chức thông tin, thực thi một câu truy vấn và đặc trưng của ngôn ngữ mà search engine sẽ tiếp cận thì ta có thể tối ưu hoá cơ hội nhận được các thông tin hữu ích. Đây là mục tiêu chính của luận văn.

Phần 1 : TÌM HIỂU VẤN ĐỀ

Chương 1: TỔNG QUAN VỀ HỆ THỐNG SEARCH ENGINE

1. Các bộ phận cấu thành hệ thống search engine

1.1 Bộ thu thập thông tin – Robot

Robot là một chương trình tự động duyệt qua các cấu trúc siêu liên kết để thu thập tài liệu & một cách đệ quy nó nhận về tất cả tài liệu có liên kết với tài liệu này.

Robot được biết đến dưới nhiều tên gọi khác nhau : spider, web wanderer hoặc web worm,… Những tên gọi này đôi khi gây nhầm lẫn, như từ 'spider', 'wanderer' làm người ta nghĩ rằng robot tự nó di chuyển và từ 'worm' làm người ta liên tưởng đến virus. Về bản chất robot chỉ là một chương trình duyệt và thu thập thông tin từ các site theo đúng giao thức web. 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.

1.2 Bộ lập chỉ mục – Index

Hệ thống lập chỉ mục hay còn gọi là hệ thống phân tích và xử lý dữ liệu, thực hiện việc phân tích, trích chọn những thông tin cần thiết (thường là các từ đơn , từ ghép , cụm từ quan trọng) từ những dữ liệu mà robot thu thập được và tổ chức thành cơ sở dữ liệu riêng để có thể tìm kiếm trên đó một cách nhanh chóng, hiệu quả. Hệ thống chỉ mục là danh sách các từ khoá, chỉ rõ các từ khoá nào xuất hiện ở trang nào, địa chỉ nào.

1.3 Bộ tìm kiếm thông tin – Search Engine

Search engine là cụm từ dùng chỉ toàn bộ hệ thống bao gồm bộ thu thập thông tin, bộ lập chỉ mục & bộ tìm kiếm thông tin. 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 mặt hoạt động.

Search engine tương tác với user thông qua giao diện web, có nhiệm vụ tiếp nhận & trả về những tài liệu thoả yêu cầu của user.

Nói nôm na, tìm kiếm từ là tìm kiếm các trang mà những từ trong câu truy vấn (query) xuất hiện nhiều nhất, ngoại trừ stopword (các từ quá thông dụng như mạo từ a, an, the,…). Một từ càng xuất hiện nhiều trong một trang thì trang đó càng được chọn để trả về cho người dùng. Và một trang chứa tất cả các từ trong câu truy vấn thì tốt hơn là một trang không chứa một hoặc một số từ. Ngày nay, hầu hết các search engine đều hỗ trợ chức năng tìm cơ bản và nâng cao, tìm từ đơn, từ ghép, cụm từ, danh từ riêng, hay giới hạn phạm vi tìm kiếm như trên đề mục, tiêu đề, đoạn văn bản giới thiệu về trang web,…..

Ngoài chiến lược tìm chính xác theo từ khoá, các search engine còn cố gắng ‘ hiểu ‘ ý nghĩa thực sự của câu hỏi thông qua những câu chữ do người dùng cung cấp. Điều này được thể hiện qua chức năng sửa lỗi chính tả, tìm cả những hình thức biến đổi khác nhau của một từ. Ví dụ : search engine sẽ tìm những từ như speaker, speaking, spoke khi người dùng nhập vào từ speak.

2. Nguyên lý hoạt động

Search engine điều khiển robot đi thu thập thông tin trên mạng thông qua các siêu liên kết (hyperlink). Khi robot phát hiện ra một site mới, nó gởi tài liệu (web

page) về cho server chính để tạo cơ sở dữ liệu chỉ mục phục vụ cho nhu cầu tìm kiếm thông tin.

Bởi vì thông tin trên mạng luôn thay đổi nên robot phải liên tục cập nhật các site cũ. Mật độ cập nhật phụ thuộc vào từng hệ thống search engine. Khi search engine nhận câu truy vấn từ user, nó sẽ tiến hành phân tích, tìm trong cơ sở dữ liệu chỉ mục & trả về những tài liệu thoả yêu cầu.

Chương 2: BỘ THU THẬP THÔNG TIN – ROBOT

1. Ứng dụng của Robot

Robot thường được sử dụng cho những mục đích sau :

1.1 Phân tích, thống kê – Statistical Analysis

Robot đầu tiên được dùng để đếm số lượng web server, số tài liệu trung bình của một server, tỉ lệ các dạng file khác nhau, kích thước trung bình của một trang web, độ kết dính, …

1.2 Duy trì siêu liên kế - Maintenance

Một trong những khó khăn của việc duy trì một siêu liên kết là nó liên kết với những trang bị hỏng (dead links) khi những trang này bị thay đổi hoặc thậm chí bị xóa. Thật không may vẫn chưa có cơ chế nào cảnh báo các bộ duy trì về sự thay đổi này. Trên thực tế khi các tác giả nhận ra tài liệu của mình chứa những liên kết hỏng, họ sẽ thông báo cho nhau, hoặc thỉnh thoảng độc giả thông báo cho họ bằng email.

Một số robot, chẳng hạn MOMspider có thể trợ giúp tác giả phát hiện các liên kết hỏng cũng như duy trì các cấu trúc siêu liên kết cùng nội dung của một trang web. Chức năng này lặp lại liên tục mỗi khi một tài liệu được cập nhật, nhờ đó mọi vấn đề xảy ra sẽ được giải quyết nhanh chóng.

1.3 Ánh xạ địa chỉ web - Mirroring

Mirroring là một kỹ thuật phổ biến trong việc duy trì các kho dữ liệu của FPT. Một ánh xạ (mirror) sẽ sao chép toàn bộ cấu trúc cây thư mục và thường xuyên cập

nhật những file bị thay đổi. Điều này cho phép nhiều người cùng truy xuất một nguồn dữ liệu, giảm số liên kết bị thất bại, nhanh hơn và ít chi phí hơn so với truy cập trực tiếp vào site thực sự chứa các dữ liệu này.

1.4 Phát hiện tài nguyên – Resource Discovery

Có lẽ ứng dụng thú vị nhất của robot là dùng nó để phát hiện tài nguyên. Con người không thể kiểm soát nổi một khối lượng thông tin khổng lồ trong môi trường mạng. Robot sẽ giúp thu thập tài liệu, tạo và duy trì cơ sở dữ liệu, phát hiện và xoá bỏ các liên kết hỏng nếu có, kết hợp với công cụ tìm kiếm cung cấp thông tin cần thiết cho con người.

1.5 Kết hợp các công dụng trên- Combined uses

Một robot có thể đảm nhận nhiều chức năng. Ví dụ RBSE Spider [4] vừa thống kê số lượng tài liệu thu được vừa tạo cơ sở dữ liệu. Tuy nhiên những ứng dụng như thế còn khá ít ỏi.

2. Robot chỉ mục – Robot Indexing

Trong quá trình thu thập thông tin phục vụ cho bộ lập chỉ mục, ta cần giải quyết những vấn đề sau :

Một là : Trong môi trường mạng, robot lấy thông tin từ các site. Vậy robot sẽ bắt đầu từ site nào ? Điều này hoàn toàn phụ thuộc vào robot. Mỗi robot khác nhau sẽ có những chiến lược khác nhau. Thường thì robot sẽ viếng thăm các site phổ biến hoặc những site có nhiều liên kết dẫn đến nó.

Hai là : Ai sẽ cung cấp địa chỉ của các site này cho robot ?

Có 2 nguồn :

Robot nhận các URL ban đầu từ user.

Robot phân tích các trang web để lấy các URL mới, đến lượt các URL này trở thành địa chỉ đầu vào cho robot. Quá trình này được lặp lại liên tục.

Ba là : Chọn dữ liệu nào trong tài liệu để lập chỉ mục ?

Quyết định chọn dữ liệu nào trong tài liệu cũng hoàn toàn phụ thuộc vào robot, thường thì những từ được liệt kê như sau được xem là quan trọng :

- Ở góc cao của tài liệu.
- Trong các đề mục
- Được in đậm (inktomi)
- Trong URL.
- Trong tiêu đề (quan trọng)
- Trong phần miêu tả trang web (description) .
- Trong các thẻ dành cho hình ảnh (ALT graphisc).
- Trong các thẻ chứa từ khóa.
- Trong các text liên kết.

Một số robot lập chỉ mục trên tiêu đề, hoặc một số đoạn văn bản đầu tiên hoặc toàn bộ tài liệu (full text). Một số khác lại lập chỉ mục trên các thẻ META(META tags) hoặc các thẻ ẩn, nhờ vậy tác giả của trang web được quyền ấn định từ khoá cho tài liệu của mình. Tuy nhiên chức năng này bị lạm dụng quá nhiều do đó các thẻ META không còn giữ được giá trị ban đầu của chúng nữa.

3. Các chiến thuật thu thập dữ liệu [II.1]

Trước khi các trang web được đánh chỉ mục, tất cả các trang web phải được lấy về máy của robot. Để lấy được tất cả các trang web, robot phải có chiến thuật. Từ một số trang web có sẵn, robot 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.

3.1 Chiến thuật tìm kiếm theo chiều sâu

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

(1) Cho danh sách = {trang đầu tiên}

(2) Lấy trang đầu tiên trong danh sách.

Nếu có qua (3)

Nếu không qua (5)

(3) Trang này đã xét tới chưa ?

Nếu rồi, quay lại (2)

Nếu chưa, qua (4)

(4) Đánh dấu đã tới rồi. Phân tích và tìm xem liên kết có trong trang đó không?

(4a) Nếu có, thêm liên kết này vào đầu danh sách. Quay lại (4)

(4b) Nếu không, quay lại (2).

(5) Kết thúc.

3.2 Chiến thuật tìm kiếm theo chiều rộng

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

(1) Cho danh sách = {trang đầu tiên}

(2) Lấy trang đầu tiên trong danh sách.

Nếu có qua (3)

Nếu không qua (5)

(3) Trang này đã xét tới chưa ?

Nếu rồi, quay lại (2)

Nếu chưa, qua (4)

(4) Đánh dấu đã tới rồi. Phân tích và tìm xem liên kết có trong trang đó không?

(4a) Nếu có, thêm liên kết này vào cuối danh sách. Quay lại (4)

(4b) Nếu không, quay lại (2).

(5) Kết thúc.

3.3 Chiến thuật tìm kiếm theo ngẫu nhiên

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

(1) Cho danh sách = {trang đầu tiên}

(2) Lấy ngẫu nhiên một trang trong danh sách.

Nếu có qua (3)

Nếu không qua (5)

(3) Trang này đã xét tới chưa ?

Nếu rồi, quay lại (2)

Nếu chưa, qua (4)

(4) Đánh dấu đã tới rồi. Phân tích và tìm xem liên kết có trong trang đó không?

(4a) Nếu có, thêm liên kết này vào cuối danh sách. Quay lại (4)

(4b) Nếu không, quay lại (2).

(5) Kết thúc.

4. Những vấn đề cần lưu ý của web robot

4.1 Chi phí và hiểm hoạ

Việc sử dụng các Robot tốn khá nhiều chi phí, đặc biệt là khi chúng được điều khiển từ xa trên internet. Phần này chúng ta sẽ cùng thảo luận về những hiểm hoạ do robot gây ra.

4.1.1 Qúa tải mạng và server – Network resource and server load

Sau một khoảng thời gian dài, thường là một tháng, robot sẽ bắt đầu hoạt động một cách liên tục. Để tăng tốc nhiều robot được phóng ra đồng thời do đó cần có băng thông lớn. Tài nguyên mạng bị khai thác quá mức khi robot yêu cầu một lượng lớn thông tin trong khoảng thời gian quá ngắn (rapid fire). Kết quả là thiếu băng thông cho những ứng dụng khác. Server vừa phải phục vụ yêu cầu của robot vừa cung cấp dịch vụ cho user, do đó yêu cầu của robot tăng lên bao nhiêu thì dịch vụ sẽ giảm xuống bấy nhiêu. Tác giả của một con robot đã thử nghiệm bằng cách cho thi hành 20 lượt truy cập đồng thời vào server của anh ta. Những lúc robot thu thập thông tin, server bị chậm

lại. Trong vòng một tuần robot đã viếng thăm site này với một yêu cầu kinh khủng. Chỉ sau 170 lượt truy xuất liên tục, thử nghiệm thất bại do server bị quá tải.

Rapid fire thực sự là thảm hoạ. Hiệu quả truyền tải thông tin dạng này bằng giao thức web hay HTTP sụt giảm thấy rõ. Những giao thức mới đang được xem xét nhằm cứu vãn tình thế.

4.1.2 Sự cập nhật quá mức- Updating overhead

Người ta cho rằng các cơ sở dữ liệu do web robot tạo ra có thể được cập nhật tự động nhưng cho đến thời điểm này vẫn chưa có cơ chế kiểm soát sự thay đổi trên web một cách hiệu quả. Cập nhật thông tin rất quan trọng nhưng qúa thường xuyên là điều không cần thiết.

Xuất phát từ thực tế đó HTTP đưa ra kỹ thuật 'if – Modified – Since' giúp các user – agent xác định được thời điểm tài liệu thay đổi. Robot phát hiện điều này chỉ khi nó lưu lại các thông tin cũ nhưng sẽ tốn nhiều bộ nhớ & cần dữ liệu phức tạp.

Một trong những đặc tính phổ biến của robot là khả năng tiếp nhận các từ cần tìm trong khi vẫn thu thập dữ liệu. Tuy nhiên một số người cho rằng đặc tính này không đáng hoan nghênh bởi hai lý do :

- Đầu tiên, các tác vụ tìm kiếm của người sử dụng cuối (end - user) góp phần đẩy server vào chỗ quá tải.
- Thứ hai, không có cơ sở đảm bảo có mối quan hệ giữa các từ cần tìm, đúng chính tả và tối ưu đối với cơ sở dữ liệu. Ví dụ, nếu bộ tìm kiếm không hỗ trợ các toán tử boolean, một user cần dữ liệu về xe máy muốn có được thông tin đúng thay vì nhập vào cụm từ 'Ford and garage' phải nhập vào từ 'car'. Nhưng người đó không hề ý thức được điều này.

Một khía cạnh nguy hiểm nữa bắt nguồn từ sự định hướng sai lầm của end – user. Một số người sử dụng công cụ của mình rất tốt như dự đoán được lượng tài liệu lớn nhất có thể có, biết chính xác nơi cần tìm dữ liệu, giới hạn thời gian sử dụng robot, trong khi đó một số khác lại lạm dụng khả năng của robot một cách vô tình hoặc cố ý. Vì vậy các tác giả viết robot đề nghị chỉ nên phân phát sản phẩm của mình cho những end-user 'hiểu' được web robot và những khó khăn trong môi trường mạng.

4.1.3 Những tình huống không mong đợi – Bad implementations

Thay vì kiểm tra trên máy cục bộ trước, một số tác giả lần đầu tiên viết robot cho thử ngay trên các server thực sự, điều này làm đau đầu không ít nhà quản trị web (web master).

Truy xuất trùng lặp có thể xảy ra khi robot không lưu lại dấu vết những nơi nó đã đi qua hoặc nó không nhận diện được các URL mặc dù khác nhau về tên nhưng lại cùng dẫn đến một địa chỉ, ví dụ địa chỉ DSN & IP.

Đôi khi, robot lãng phí thời gian và tài nguyên chỉ để thu về những tài liệu mà sau đó phải vứt đi. Ví dụ hệ thống chỉ quan tâm đến file văn bản (text file) nhưng robot lại nhận cả những loại file khác như file hình ảnh, file thực thi, …

Trong môi trường mạng có những vùng gần như vô tận. Ví dụ, cứ mỗi lần phân tích một trang robot nhận về cùng một URL nhưng xa hơn một cấp, '/cgi-bin/pit/', và tiếp tục '/cgi-bin/pit/a/', '/cgi-bin/pit/a/a', …. . Sự lặp lại không có điểm dừng này được gọi là các lỗ đen (black holes)

4.2 Tiêu chuẩn loại trừ robot

Trong quá trình xử lý robot không thể tự quyết định tài liệu nào được lập chỉ mục, tài liệu nào không do đó nó lấy tất cả những gì có thể. Thậm chí dù xác định được

tài liệu vô ích thì nó cũng đã bỏ ra một chi phí đáng kể cho hoạt động thu thập. Tiêu chuẩn loại trừ robot ra đời. Các chuẩn này chẳng những chỉ ra URL nào cần tránh mà còn cảnh báo robot về các lỗ đen.

4.2.1 File robot.txt

Robot.txt là một file cấu trúc được đặt tại thư mục gốc của server, gồm 2 trường User-agent và Disallow.

- User-agent : cho biết robot nào sẽ bị kiểm soát.
- Disallow : cho biết robot có được phép kết nối vào URL này hay không.
- Xét các ví dụ sau :

Ví dụ	Ý nghĩa
# / robots.txt file for http://webcrawler.com/	Ký tự # bắt đầu một chú thích
User-agent: webcrawler Disallow:	Robot có tên là webcrawler có thể đi đến bất cứ trang nào của site
User-agent: lycra Disallow: /	Robot có tên là lycra bị cấm trên tất cả các trang của site

User-agent: * Disallow: /tmp Disallow: /logs	Mọi robot đều không được truy xuất vào 2 thư mục tmp và logs

Bảng 2.1 :Ví dụ về chuẩn loại trừ robot dùng file robot.txt

4.2.2 Thẻ META dành cho robot – Robot META tag

META tag là sự mở rộng của chuẩn loại trừ robot, hỗ trợ cho tác giả của những trang web không có quyền admin.

Vị trí	Nằm trong phần HEAD của file HTML
Cú pháp	<*meta name* = 'robots' *content* = *'index,follow'*>
Tên trường	Ý nghĩa
Meta	Thẻ báo hiệu bắt đầu
Name	Tên robot sẽ bị kiểm soát
Content	Cờ định hướng cho robot, các cờ này có thể kết hợp với nhau & được phân cách bằng dấu phẩy.

Bảng 2.2 : Bảng thông tin về META tag trong chuẩn loại trừ robot

Các cờ của thuộc tính Content	Ý nghĩa
[NO]INDEX	Robot không nên lập chỉ mục cho trang này.
[NO]FOLLOW	Robot không nên lấy các liên kết ở trang này
ALL = INDEX, FOLLOW	
NONE= NOINDEX, NOFOLLOW	

Bảng 2.3 : Bảng giá trị các cờ của thuộc tính Content trong META tag

4.2.3 Nhược điểm của file robot.txt

Người ta cho rằng việc liệt kê các trang hoặc các thư mục trong file robot.txt sẽ là nguyên nhân thu hút sự chú ý từ các 'vị khách không mời'. Thực ra chuẩn loại trừ robot chỉ là dấu hiệu cảnh báo, không là biện pháp cấm robot cho nên việc tuân theo hay không hoàn toàn là vấn đề tự nguyện. Tuy nhiên ta vẫn có cách khắc phục :

Một là :

- Tạo một thư mục chứa tất cả các file quan trọng.
- Trường Disallow chỉ liệt kê tên thư mục vừa tạo.
- Cấu hình server sao cho các trang không chứa đường dẫn đến thư mục này.

Đáng buồn trên thực tế cách này không đạt được kết quả mong đợi do một trong các nguyên nhân sau :

- Các server có robot không bị cấm có thể dẫn đường các robot bị cấm khác đến những file này.
- Các file quan trọng có thể nằm trong log file (file được tự do truy xuất)
- Khi cấu hình lại server, admin có thể 'quên' các thư mục này phải cấm robot!

……

Hai là: chứng thực (athorization). Đây là biện pháp hữu hiệu, được sử dụng trong nhiều lĩnh vực, đặc biệt trong những môi trường mà sự an toàn dữ liệu trở nên rất cần thiết.

Tóm tắt :

Có thể nói web robot là con dao 2 lưỡi, sử dụng đúng sẽ giải quyết được nhiều vấn đề, sử dụng sai sẽ để lại những hậu quả khó đoán. Sau đây là tóm tắt cho những vấn đề cần lưu ý của web robot

- Tránh lãng phí tài nguyên
  - ✓ Chỉ tải về những tài liệu cần thiết.
  - ✓ Nếu hệ thống chỉ quan tâm đến các file text (.html, .htm, .xml, …), web robot nên bỏ qua các liên kết dẫn đến những file thực thi (.exe, …), file ảnh (.gif, .bmp, …).
  - ✓ Bỏ qua các trường dữ liệu hệ thống không dùng đến.
  - ✓ Đừng lấy về các trang giống nhau nhiều hơn một lần.
- Tránh cập nhật lại các site cũ quá thường xuyên bằng cách :

- ✓ Ghi nhớ những địa chỉ web robot đã duyệt qua.
- ✓ Dựa vào trường LastModified, trường head. Nếu các trường này khác với dữ liệu ta đã có thì đó là những thông tin cần ghi nhận.
- ✓ Không nên duyệt hết một site, chỉ cần duyệt đến một độ sâu (deep link) cần thiết.

➢ Tránh làm quá tải server

- ✓ Duy trì một khoảng thời gian đợi giữa các lần truy xuất liên tiếp.
- ✓ Kết nối với server vào những thời điểm thích hợp. Tham khảo ý kiến của admin để biết thông tin này.
- ✓ Kiểm tra web robot trên máy cục bộ, sửa lỗi trước khi chạy trên server thực sự.

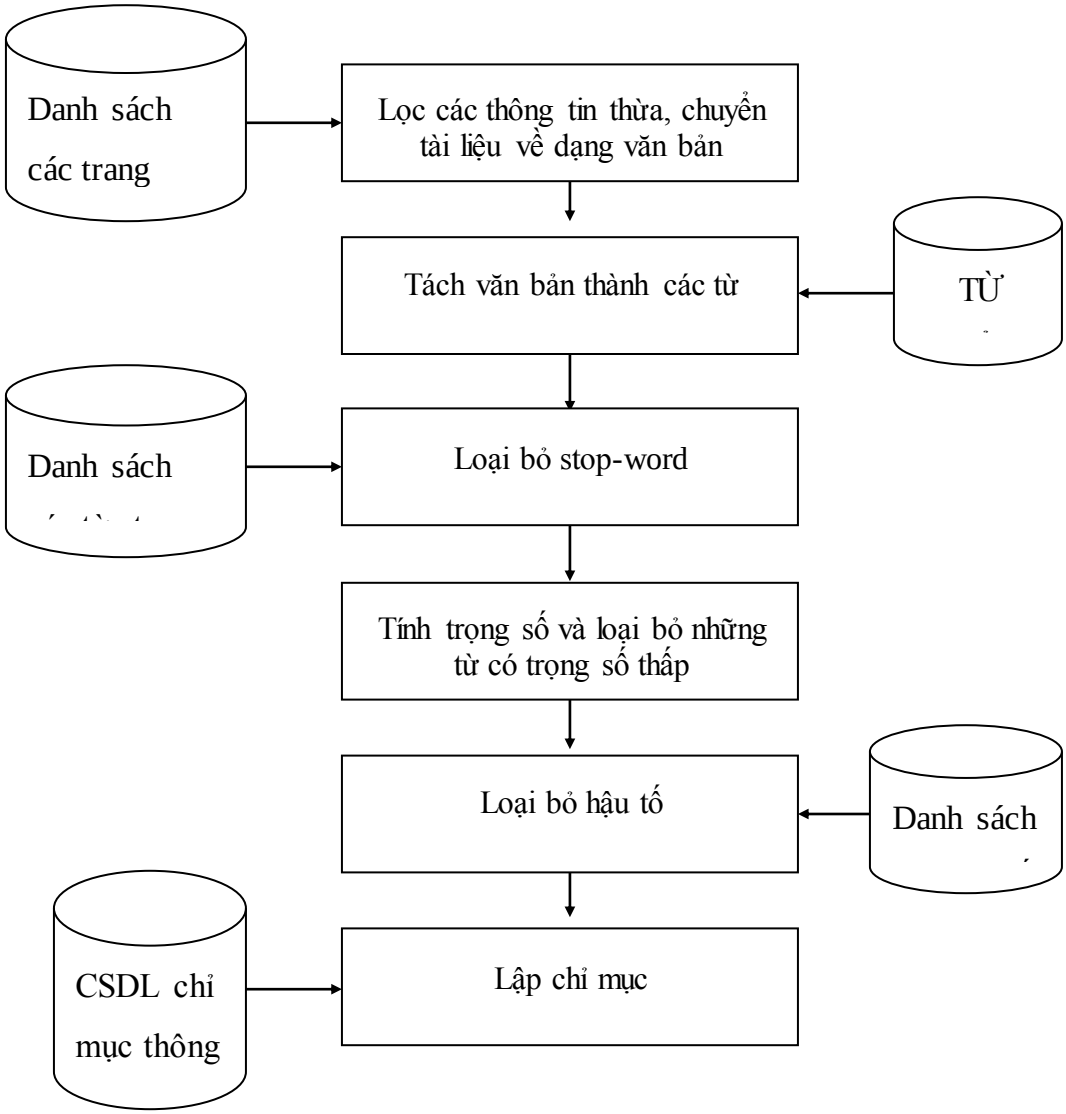
➢ Tuân theo các luật loại trừ robot.

Chương 3: BỘ LẬP CHỈ MỤC – INDEX

1. Khái quát về hệ thống lập chỉ mục

Các trang Web sau khi thu thập về sẽ được phân tích, trích chọn những thông tin cần thiết (thường là các từ đơn , từ ghép , cụm từ quan trọng) để lưu trữ trong cơ sở dữ liệu nhằm phục vụ cho nhu cầu tìm kiếm sau này.

Mô hình xử lý tổng quát của một hệ thống được trình bày như sau:



Hình 3.1 Lưu đồ xử lý cho hệ thống lập chỉ mục

Lập chỉ mục là quá trình phân tích và xác định **các từ , cụm từ thích hợp cốt lõi có khả năng đại diện cho nội dung của tài liệu** . Như vậy, vấn đề đặt ra là phải rút trích ra những thông tin chính, có khả năng đại diện cho nội dung của tài liệu. Thông tin này phải "vừa đủ", nghĩa là không thiếu để trả ra kết quả đầy đủ so với nhu cầu tìm kiếm, nhưng cũng phải không dư để giảm chi phí lưu trữ và chi phí tìm kiếm và để loại bỏ kết quả dư thừa không phù hợp. Việc rút trích này chính là việc lập chỉ mục trên tài liệu. Trước đây , quá trình này thường được các chuyên viên đã qua đào tạo thực hiện một cách "thủ công " nên có độ chính xác cao. Nhưng trong môi trường hiện đại ngày nay, với lượng thông tin khổng lồ thì việc lập chỉ mục bằng tay không còn phù hợp, phương pháp lập chỉ mục tự động mang lại hiệu quả cao hơn.

Một thủ tục lập chỉ mục tự động cơ bản cho các tài liệu tiếng Anh có thể được xử lý như sau: [III.1]

1. *Step of tokenization:* Tách văn bản ra thành các chuỗi nhờ vào khoảng trắng, mỗi chuỗi xem như là một từ.
2. *Step of removal of stop words:* bỏ những từ thường xuyên xuất hiện trong hầu hết các tài liệu nhưng lại không quan trọng trong các tài liệu như tính từ, đại từ.
3. *Step of stemming:* loại bỏ các hậu tố (suffixes) để đưa về các từ gốc

Các từ thu được sẽ được lập chỉ mục. Tuy nhiên hai bước đầu cũng cần cho quá trình lập chỉ mục cho các tài liệu tiếng Việt, bước thứ ba không cần vì tiếng Việt thuộc dòng ngôn ngữ đơn thể.

2. Tổng quan về phương pháp lập chỉ mục ([I.1], [I.2], [II.1])

Phương pháp lập chỉ mục gồm 2 phần chính yếu sau :

- đầu tiên là xác định các **mục từ** , khái niệm mà có khả năng đại diện cho văn bản sẽ được lưu trữ (bao gồm cả việc tách từ, loại bỏ stop-word, xử lý hậu tố…)
- thứ hai là 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

2.1 Xác định mục từ quan trọng cần lập chỉ mục ([I.1])

Mục từ hay còn gọi là mục từ chỉ mục, là đơn vị cơ sở cho quá trình lập chỉ mục. Mục từ có thể là từ đơn, từ phức hay một tổ hợp từ có nghĩa trong một ngữ cảnh cụ thể. Ta xác định mục từ của 1 văn bản dựa vào chính nội dung của văn bản đó , hoặc dựa vào tiêu đề hoặc tóm tắt nội dung của văn bản đó.

Hầu hết việc lập chỉ mục tự động bắt đầu với việc khảo sát tần số xuất hiện của từng loại từ riêng rẽ trong văn bản. Nếu tất cả các từ xuất hiên trong tập tài liêu với những tần số băng nhau, thì không thể phân biệt các mục từ theo tiêu chuẩn định lượng. Tuy nhiên, trong văn bản ngôn ngữ tự nhiên, tần số xuất hiện của từ có tính thất thường, Do đó những mục từ có thể được phân biệt bởi tần số xuất hiên của chúng.

Đặc trưng xuất hiện của từ vựng có thể được định bởi hằng số "thứ hạng - tần số" (Rank_Frequency) theo luật của Zipf :

Tân số xuất hiên * thứ hạng = Hằng

Biểu thức luật Zipf có thể dẫn ra những hệ số ý nghĩa của từ dựa vào những đặc trưng của tần số xuất hiện của mục từ riêng lẽ trong những văn bản tài liệu.

Một đề xuất dựa theo sự xem xét chung sau:

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

   F_{ik} (Frequency): tần số xuất hiện của mục từ k trong tài liệu i

2. Xác định tổng số tập 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.

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

3. Sắp xếp những thứ tự giảm theo tập tần số xuất hiện của chúng. Quyết định giá trị ngữong cao và loại bỏ tất cả những từ có tập tần số xuất hiện cao trên ngững nay. Những từ bị loại bỏ là những từ xuất hiện phổ biến ở hầu hết các tài liệu. Đó chính là các stop-word.

4. Tương tư, loại trừ những từ được xem là có tần số xuất hiện thấp. Việc xoá những mục từ như vậy hiếm khi xảy ra trong tập hợp mà sự mặt của chúng không làm ảnh hưởng lớn đến việc thực hiện truy vấn.

5. Những từ xuất hiện trung bình còn lại bây giờ được dùng cho việc ấn định tới những tài liệu như những mục từ chỉ mục.

Chú ý: một khái niệm xuất hiện ít nhất hai lần trong cùng một đoạn thì được xem là một khái niệm chính. Một khái niệm xuất hiện trong hai đoạn văn liên tiếp cũng được xem là một khái niệm chính mặc dù nó chỉ xuất hiện duy nhất một lần trong đoạn

đang xét. Tất cả những chú giải về những khái niệm chính được liệt kê theo một tiêu chuẩn nhất định nào đó.

Thực tế cho thấy rằng ý tưởng trên khá cứng nhắc , vì nếu loại bỏ tất cả những từ có tần số xuất hiện cao sẽ làm giảm giá trị recall (độ tương tự), tức giảm hiệu quả trong việc trả về số lượng lớn của những mục tin thích đáng. Ngược lại, sự loại bỏ những mục từ có tần số xuất hiện thấp có thể làm giảm giá trị của độ chính xác . Một vấn đề khác là sự cần thiết để chọn những ngưỡng thích hợp theo thứ tự để phân biệt những mục từ hữu ích có tần số xuất hiện trung bình trong phần còn lại

2.2 Một số hàm tính trọng số mục từ. ([I.1])

Trọng số của mục từ: là sự tần xuất xuất hiện của mục từ trong toàn bộ tài liệu. 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 cảcác tài liệu thì "**ít có ý nghĩa hơn**" là những từ tập trung trong một số tài liệu.

Ta xét các khái niệm sau:

- Gọi T={t1,t2,...,tn} là **không gian chỉ mục**, với ti là các mục từ.
- Một tài liệu D được lập chỉ mục dựa trên tập T sẽ được biểu diễn dưới dạng:

T(D)={w1,w2,...wn} với wi là trọng số của ti trong tập tài liệu D. Nếu wi=0 nghĩa là ti không xuất hiện trong D hoặc mục từ ti ít quan trọng trong tài liệu D ta không quan tâm tới.

T(D) được gọi là vector chỉ mục của D, nó được xem như biểu diễn cho nội dung của tài liệu D và được lưu lại trong cơ sở dữ liệu của hệ thống tìm kiếm thông tin để phục vụ cho nhu cầu tìm kiếm.

Mặc dù T(D) biểu diễn nội dung của tài liệu D nhưng không phải bất cứ từ nào có trong D đều xuất hiện trong T(D) mà chỉ có **những từ có trọng lượng** (có ý nghĩa quan trọng trong tài liệu D) mới được lập chỉ mục cho D.

Sau đây ta xét một số hàm tính trọng số của mục từ

2.2.1 Nghịch đảo trọng số tần số tài liệu (*The Inverse Document Frequency Weight*)

w_k : là trọng lượng của mục từ k.

$nDoc_k$: tổng số tài liệu mà mục từ k xuất hiện.

n_{ki} : số lần xuất hiện mục từ k trong tài liệu i.

n_k : số lần xuất hiện mục từ k trong toàn tập tài liệu.

nDoc : tổng số tài liệu.

idf_k : giá trị nghịch đảo tần số tài liệu. (*Inverse Document Frequency*)

Trọng lượng mục từ k :

$$Wk = \text{idf}_k = \log_2 \frac{nDoc}{nDoc_k} + 1$$

Như vậy, trọng số của mục từ k sẽ tăng lên khi tần số xuất hiện của mục từ k trong các tài liệu i tăng lên nhưng giảm xuống khi tấn số xuất hiện của mục từ k trong tập tài liệu ($nDoc_k$) tăng lên.

Biểu thức tổng hợp :

$$w_k = n_{ik} * [\log_2 (n) - \log_2(nDOC_k) + 1]$$

Hàm này gán độ quan trọng cao cho những mục từ chỉ xuất hiện trong một số ít tài liệu của một tập hợp tài liệu (đề cao độ phân biệt)

2.2.2 Độ nhiễu tín hiệu (*Signal Noise*):

Trọng số của từ được đo lường bằng sự **tập trung hay phân tán** của từ. Ví dụ từ "hardware" xuất hiện 1000 lần nhưng trong 200 tài liệu (tập trung) thì có trọng lượng cao hơn từ "computer" cũng xuất hiện 1000 lần nhưng trong 800 tài liệu.

Độ nhiễu của một mục từ k:

$$noise_k = \sum (n_{ki} / n_k).\log_2 (n_{ki} / n_k) \quad \forall i=1,nDoc$$

Hàm số nghịch đảo của độ nhiễu được gọi là **độ signal** có thể được dùng để tính trọng lượng của mục từ k :

$$w_k = signal_k = \log_2(n_k) - noise_k$$

2.2.3 Giá trị độ phân biệt của mục từ :

Không ai muốn kết quả của việc tìm kiếm lại trả về tập tất cả các tài liệu có trong tập hợp (nghĩa là tập chỉ mục của các tài liệu chứa nhiều từ giống nhau). Độ phân biệt của mục từ là giá trị phân biệt mức độ tương đương giữa các tài liệu. Nếu một mục từ có trong chỉ mục mà làm cho độ tương tự của các tài liệu cao thì nó có độ phân biệt kém (nghĩa là từ này thường xuyên xuất hiện trong các tài liệu) và ngược lại. Như vây các **mục từ có độ phân biệt cao nên được chọn để lập chỉ mục**. Thực chất việc sử dụng độ phân biệt này cũng **cho kết quả tương đương với việc sử dụng tần số nghịch đảo và tỉ lệ tín hiệu nhiễu**.

Gọi Sim(Di,Dj) là độ tương tự của cặp tài liệu Di, Dj.

Độ tương tự trung bình được tính trên tất cả các cặp tài liệu:

$$Arv\_Sim = \sum Sim(Di,Dj) \quad \forall i \neq j.$$

Gọi Arv_Simk là độ tương tự trung bình được tính trong trường hợp mục từ k bị loại bỏ khỏi tập chỉ mục.

Khi đó **trọng lượng mục từ k** có thể được **tính theo giá trị độ phân biệt DiscValueK** theo công thức:

$$w_k = DiscValue_K = Arv\_Sim_k - Arv\_Sim$$

Phép tính $DiscValue_K$ cho tất cả những mục từ k, những mục từ có thể được xếp theo thứ tự giảm của giá trị phân biệt $DiscValue_K$. Những mục từ chỉ mục có thể thuộc một trong ba nhóm dựa theo giá trị độ phân biệt của chúng như sau:

- Độ phân biệt tốt đối vơi $DiscValue_K$ **dương,** những mục từ có độ phân biệt cao.
- Đối với $DiscValue_K$ **gần bằng 0**, độ phân biệt giữa các tài liệu không khác nhau khi thêm vào hay bớt đi những mục từ đó.
- Độ phân yếu khi $DiscValue_K$ **âm**, những mục từ có độ phân biệt thấp (độ tương tự cao).

2.2.4 Kết hợp tần số xuất hiện mục từ và nghịch đảo tần số tài liệu

Phần này sẽ đề cập đến một số biến thể tần số xuất hiện của mục từ tf(Term Frequency) và sự kết hợp với idf để xác định tầm quan trọng của một mục từ.

f(t,d) : tần số xuất hiện của mục từ t trong tài liệu d

N : tổng số tài liệu trong tập dữ liệu

n : tổng số tài liệu có mục từ t

Max[f(t,d)]: số lần xuất hiện cao nhất của mục từ t trong toàn tập tài liệu

tf(Term Frequency) vẫn là tần số xuất hiện của một mục từ trong tập tài liệu, có thể được xác định bởi nhiều công thức khác nhau:

$tf = f(t,d)$

$tf = f(t,d)/Max[f(t,d)]$

$tf = \log_2(f(t,d))$

$tf = \log_2(f(t,d) +1)$

$tf = \ln f(t,d) +1$

….

idf(Inverse Document Frequency) : là tần số nghịch đảo tần số xuất hiện của các tài liệu và được tính như sau:

$idf = \log_2(N/n)$

$idf = \log_2((N-n)/n)$

$idf = \log_2(N/n)*2$

……

Mỗi công thức trên đều mang một ý nghĩa riêng trong từng trường hợp cụ thể, sự kết hợp của tf và idf sẽ xác định mức độ quan trọng của mục từ cần xét.

$$W_{it} = tf * idf$$

Sự kết hợp hai tiêu chuẩn này cho biết: tầm quan trọng của một mục từ (do tf mang lại) và sự phân biệt giữa các mục từ (do idf mang lại). Một mục từ có tầm quan trọng lớn hơn thì giá trị W_{it} của nó phải lớn hơn.

2.3 Lập chỉ mục tự động cho tài liệu

Vấn đề chính của lập chỉ mục tự động là xác định tự động mục từ chỉ mục cho các tài liệu. Trong các ngôn ngữ gốc Ấn – Âu thì **tách từ có thể nói là đơn giản** vì khoảng trắng là ký tự để phân biệt từ. Vấn đề cần quan tâm là xác định những từ này là từ khoá, có thể đại diện cho toàn bộ nội dung của tài liệu. **Loại bỏ các từ stop-word** có tần số xuất hiện cao, những từ này thường chiếm đến 40-50% trong số các từ của một văn bản. Những từ này có độ phân biệt kém và không thể sử dụng để xác định nội dung của tài liệu. Trong tiếng Anh, có khoảng 250 từ. Số lượng từ này không nhiều lắm nên giải pháp đơn giản nhất là lưu các từ này vào trong một tự điển, và sau đó chỉ cần thực hiện so sánh từ cần phân tích với từ điển để loại bỏ.

Bước tiếp theo là nhận ra các chỉ mục tốt. Để giảm bớt dung lượng lưu trữ, các mục từ cần được biến đổi về nguyên gốc (*step of stemming* đối với tiếng Anh), Phải loại bỏ đi các tiền tố, hậu tố, các biến thể số nhiều, quá khứ…Giải pháp là sử dụng một danh sách các hâu tố. Trong khi loại bỏ hậu tố thì những **hậu tố dài** được ưu tiên loại bỏ trước, rồi sau đó mới loại bỏ những **hậu tố ngắn** hơn. Sau đây là một số vấn đề khi loại bỏ trong tiếng Anh:

1. Chỉ rõ **chiều dài tối thiểu của một từ gốc** sau khi loại bỏ hậu tố. Ví dụ: việc loại bỏ hậu tố “ability” ra khỏi “computability” hay loại bỏ “ing” ra

khỏi "singing" là hợp lý. Tuy nhiên, những hậu tố đó không cần phải loại bỏ trong các từ "ability" và "sing".

2. Nếu nhiều hậu tố được kết hợp vào một gốc thì ta sẽ áp dụng **đệ quy** cho quá trình loại bỏ hậu tố vài lần hoặc lập từ điển hậu tố rồi loại bỏ những hậu tố dài hơn trước rồi đến các hậu tố ngắn sau. Ví dụ: "effectiveness" → "effective" → "effect".
3. Trong tiếng Anh, **từ gốc có thể bị biến đổi** sau khi đã loại bỏ hậu tố. Do đó, ta cần phải có những luật nhất định để phục hồi từ gốc. Chẳng hạn loại bỏ một trong hai kí tự trùng nhau của những từ có sự xuất hiện b, d, d, l, m, n, p, r, s, t ở cuối của các từ gốc sau khi đã loại bỏ hậu tố. Ví du như "beginning" → "beginn" → "begin".
4. Một số **ngoại lệ** phụ thuộc vào ngữ cảnh đặc biệt phải được chú ý, sử dụng các quy tắc cảm ngữ cảnh. Ví dụ: một quy tắc cho hậu tố "allic" chỉ rõ chiều dài cực tiểu của từ gốc là ba và không loại bỏ hậu tố sau "met" hoặc "ryst", hoặc quy tắc chỉ loại bỏ hậu tố "yl" sau "n" hoặc "r".

Tóm lại, giải quyết vấn đề hậu tố không quá khó nếu chúng ta có sẵn một danh sách chứa các hậu tố, một danh sách chứa các luật thêm các hậu tố và phục hồi từ gốc sau khi thêm hậu tố.

3. Lập chỉ mục cho tài liệu tiếng Việt ([III.1], [II.1], [II.2], [II.3], [II.4], [IV.11], [IV.12])

3.1 Khó khăn cho việc lập chỉ mục tiếng Việt

Các điểm khó khăn khi thực hiện quá trình lập chỉ mục cho tài liệu tiếng Việt so với tài liệu tiếng Anh mà chúng ta phải giải quyết :

- Xác định **ranh giới** giữa các từ trong câu. Đối với tiếng Anh điều này quá dễ dàng vì khoảng trắng chính là ranh giới phân biệt các từ ngược lại tiếng Việt thì khoảng trắng không phải là ranh giới để xác định các từ mà chỉ là ranh giới để xác định các tiếng.

- Chính tả tiếng Việt còn một số điểm chưa thống nhất như sử dụng "**y" hay "i**" (ví dụ "quý" hay "quí"), **cách bỏ dấu** ("lượng" hay "lượng"), **cách viết hoa** tên riêng("Khoa học Tự nhiên" hay "Khoa Học Tự Nhiên")... đòi hỏi quá trình hiệu chỉnh chính tả cho văn bản cần lập chỉ mục và cho từ điển chỉ mục.

- Tồn tại nhiều **bảng mã tiếng Việt** đòi hỏi khả năng xử lý tài liệu ở các bảng mã khác nhau. Cách giải quyết là đưa tất cả về bảng mã chuẩn của hệ thống.

- Sự phong phú về nghĩa của một từ (**từ đa nghĩa**). Một từ có thể có nhiều nghĩa khác nhau trong những ngữ cảnh khác nhau nên việc tìm kiếm khó có được kết quả với độ chính xác cao.

- **Từ đồng nghĩa hoặc từ gần nghĩa:** có nhiều từ khác nhau nhưng lại có cùng ý nghĩa. Do đó, việc tìm kiếm theo từ khoá thường không tìm thấy các websites chứa từ đồng nghĩa hoặc gần nghĩa với từ cần tìm. Vì vậy, việc tìm kiếm cho ra kết quả không đầy đủ.

- Có quá nhiều từ mà mật độ xuất hiện cao nhưng không mang ý nghĩa cụ thể nào mà chỉ là những từ nối, từ đệm hoặc chỉ mang sắc thái biểu cảm như những từ láy. Những từ này cần phải được xác định và loại bỏ ra khỏi tập các mục từ. Nó giống như **stop-word** trong tiếng Anh.

- Các văn bản có nội dung chính là một vấn đề cụ thể, một đề tài nghiên cứu khoa học nhưng đôi khi trọng số của các từ chuyên môn này thấp so với toàn

tập tài liệu. Vì vậy, một số thuật toán tính trọng số bỏ sót những trường hợp như vậy. Kết quả là các từ chuyên môn đó không được lập chỉ mục.

Trong các vấn đề trên thì vấn đề **xác định ranh giới từ** trong câu là quan trọng nhất vì nó ảnh hưởng lớn đến hiệu quả của quá trình lập chỉ mục (**nếu quá trình tách từ sai có nghĩa là nội dung của câu bị phân tích sai**) và **cũng là vấn đề khó khăn nhất**. Các vấn đề còn lại chỉ là thuần tuý về mặt kỹ thuật mà hầu như chúng ta có thể giải quyết một cách triệt để.

3.2 Đặc điểm về từ trong tiếng Việt và việc tách từ

3.2.1 . Đặc điểm về từ trong tiếng Việt:

Tiếng Việt là ngôn ngữ đơn lập. Đặc điểm này bao quát tiếng Việt cả về mặt ngữ âm, ngữ nghĩa, ngữ pháp. Khác với các ngôn ngữ Ấn-Âu, mỗi từ là một nhóm các ký tự có nghĩa được cách nhau bởi một khoảng trắng. Còn tiếng Việt, và các ngôn ngữ đơn lập khác, thì khoảng trắng không phải là căn cứ để nhận diện từ.

3.2.1.a Tiếng:

➢ Trong tiếng Việt trước hết cần chú ý đến đơn vị xưa nay vẫn quan gọi là tiếng. Về mặt ngữ nghĩa, ngữ âm, ngữ pháp, đều có giá trị quan trọng.

➢ Sử dụng tiếng để tạo từ có hai trường hợp:

✓ Trường hợp một tiếng: đây là trường hợp một tiếng được dùng làm một từ, gọi là từ đơn. Tuy nhiên không phải tiếng nào cũng tạo thành một từ.

✓ Trường hợp hai tiếng trở lên: đây là trường hợp hai hay nhiều tiếng kết hợp với nhau, cả khối kết hợp với nhau gắn bó tương đối chặt chẽ, mới có tư cách ngữ pháp là một từ. Đây là trường hợp từ ghép hay từ phức.

3.2.1.b Từ:

Có rất nhiều quan niệm về từ trong tiếng Việt , từ nhiều quan niệm về từ tiếng Việt khác nhau đó chúng ta có thể thấy đặc trưng cơ bản của "từ " là sự hoàn chỉnh về mặt nội dung, từ là đơn vị nhỏ nhất để đặt câu.

Người ta dùng "từ" kết hợp thành câu chứ không phải dùng "tiếng" do đó quá trình lập chỉ mục bằng cách tách câu thành các "từ" cho kết qua tốt hơn là tách câu bằng "tiếng".

3.2.2 Tách từ

Việc xác định từ trong tiếng Việt là rất khó và tốn nhiều chi phí. Do đó, cách đơn giản nhất là **sử dụng từ điển được lập sẵn**. Tách tài liệu thành các từ, loại bỏ các từ láy, từ nối, từ đệm, các từ không quan trọng trong tài liệu. Một câu gồm nhiều từ ghép lại, tuy nhiên trong một câu có thể có nhiều cách phân tích từ khác nhau.

Ví dụ : xét câu "Tốc độ truyền thông tin sẽ tăng cao" có thể phân tích từ theo các cách sau:

Tốc độ / truyền/ thông tin / sẽ / tăng cao.

Tốc độ / truyền thông / tin / sẽ / tăng cao.

Hiện đã có nhiều giải pháp cho vấn đề này với kết quả thu được rất cao. Tuy nhiên thời gian, chi phí tính toán, xử lý lớn không thích hợp cho việc lập chỉ mục cho hệ thống tìm kiếm thông tin vì số lượng tài liệu phải xử lý là rất lớn.

Cách giải quyết: **lập chỉ mục cho các từ có thể có trong một tài liệu**. Ví dụ câu trên ta nên lập xem xét các từ : tốc độ, truyền , truyền thông ,thông tin, tin , sẽ , tăng cao.

Sau đó sẽ **dùng ngưỡng** chặn để loại bỏ các từ, giả sử từ "truyền thông" không phải là một từ xuất hiện thật sự trong tài liệu (chỉ có được do sự kết hợp ngẫu nhiên từ "truyền" và "thông tin") thì xác suất xuất hiện của từ này trong tài liệu sẽ không cao nên khi tính toán trọng lượng thì từ này sẽ bị loại bỏ. Một từ trong tiếng Việt là sự kết hợp của hai hay nhiều tiếng. Phương pháp xác định một từ được ghép lại thông qua nhiều tiếng dựa trên việc xem xét độ gắn kết (cohesion) giữa chúng:

$$Cohension(n_{ij}) = size\_factor * pair\_freq_{ij}/(n_i * n_j)$$

Trong đó:

size_factor: kích thước tập chỉ mục

$pair\_freq_{ij}$: tần số xuất hiện từ

n_i , n_j : tần số xuất hiện tiếng i, j

Hai tiếng có khả năng tạo thành một từ cao khi chúng thường xuất hiện chung với nhau, nghĩa là cohension của chúng cao.

Phương pháp này không tách từ chính xác hoàn toàn nhưng có thể chấp nhận trong hệ thống tìm kiếm thông tin vì trong quá trình lập chỉ mục **chỉ cần xác định đúng các từ có trọng lượng cao**, trong trường hợp việc tách từ là sai thì từ sai chỉ được lập chỉ mục khi nó có trọng lượng cao, **việc lập chỉ mục một từ sai sẽ làm tăng chi phí lưu trữ nhưng có lẽ không ảnh hưởng lớn tính chính xác kết quả tìm kiếm** vì dù sao từ này cũng có trọng lượng lớn.

Còn trong trường hợp một từ ghép được tách thành nhiều từ đơn ví dụ từ "thông tin" khi được lập chỉ mục sẽ luôn có 3 từ "thông", "tin", "thông tin" , điều này gây ảnh hưởng đến tính chính xác của việc lập chỉ mục vì thực sự các từ "thông", "tin" không

cần thiết lập chỉ mục.Ta giải quyết vấn đề này bằng cách nếu từ "thông tin" được lập chỉ mục thì khi đó số lần xuất hiện của các từ "thông" và "tin" sẽ được tính toán lại bằng cách **trừ đi các trường hợp đã xuất hiện trong từ "thông tin"** để tính toán trọng lượng cho các từ đơn. Nếu từ đơn "tin" chỉ luôn xuất hiện trong từ "thông tin" thì số lần xuất hiện của từ "tin" và "thông tin" là bằng nhau nên khi lập chỉ mục cho từ "thông tin" thì số lần xuất hiện riêng của từ đơn "tin" sẽ bằng 0 nên không được lập chỉ mục.

3.3 Giải quyết các vấn đề hiển thị của tiếng Việt (vấn đề chính tả)

3.3.1 Vấn đề bảng mã

Sự tồn tại của nhiều bảng mã (TCVN3, VNI ...) dẫn đến việc phải **chuyển nội dung các tài liệu được viết trên các bảng mã khác về bảng mã chuẩn** cho hệ thống tìm kiếm thông tin xử lý (lập chỉ mục), việc chuyển đổi này là đơn giản trong trường hợp ta biết bảng mã của tài liệu, nhưng vấn đề khó khăn là làm sao cho hệ thống tìm kiếm thông tin nhận ra một tài liệu đang sử dụng bảng mã nào?

Khi phân tích một trang tài liệu HTML, dựa vào thông tin <charset> thì có thể biết được bảng mã nào đang được sử dụng, ví dụ: charset = UTF-8 thì đó là bảng mã Unicode

Tuy nhiên, trong một tài liệu có thể sử dụng nhiều bảng mã khác nhau nên không thể xác định bảng mã của tài liệu theo cách trên. Ta có thể áp dụng phương pháp phân tích từ khoá để xác định bảng mã như sau: dựa trên sự thống kê số lần xuất hiện của các ký tự đặc biệt theo từng bảng mã, bảng mã nào có tần số sử dụng nhiều nhất thì xem như trang đó sử dụng bảng mã đó.

Thật ra, không cần phải xác định bảng mã vẫn có thể lập chỉ mục cho hệ thống bằng cách chuyển mã tài liệu sang một kiểu định dạng, theo bảng mã quy định của hệ thống tìm kiếm. Trong thực tế, các bảng mã đều có một phần chung và một phần các ký tự đặc biệt là khác nhau. Do đó, nhằm hạn chế thời gian và chi phí xử lý, ta sẽ chuyển những ký tự khác nhau về bảng mã quy định. Các bước thực hiện như sau: Đọc một từ, nếu là từ mang dấu tổ hợp nguyên âm hay dấu thanh thì thực hiện so sánh với tất cả các bảng mã chuẩn để xác định bảng mã của từ đó. Nếu bảng mã đó không trùng với bảng mã quy định của hệ thống thì thực hiện chuyển từ bảng mã đó sang bảng mã quy định. Cứ vậy, tiếp tục cho đến hết văn bản và dừng.

Có thể dùng một bảng mã thông dụng nào đó để làm bảng mã quy định cho hệ thống, chẳng hạn Unicode vì hiện nay theo xu hướng chung thì số lượng các trang web, tài liệu dùng Unicode rất lớn và đang tăng nhanh, nên sẽ hạn chế được số lượng các trang web cần chuyển đổi.

3.3.2 Vấn đề dấu thanh

Do cách bỏ dấu tiếng Việt chưa thống nhất nên có khi cùng một từ lại có nhiều các bỏ dấu khác nhau, ví dụ "thuý" và "thúy", rõ ràng hệ thống tìm kiếm thông tin cần nhận ra hai từ này là một. Phương pháp giải quyết dựa trên đặc điểm **một từ đơn tiếng Việt chỉ có một dấu** nên **ta sẽ chuyển dấu từ ra sau cùng**, ví dụ:

quý -> thuy1

qúy -> thuy1

Khi đó tất cả các từ giống nhau cho dù bỏ dấu khác nhau thì qua quá trình xử lý đều cho chuỗi kí tự giống nhau thuận tiện cho việc so sánh từ.

3.3.3 Vấn đề dấu tổ hợp nguyên âm

Một tài liệu hay một câu truy vấn không thể tránh khỏi trường hợp bỏ thiếu dấu tổ hợp nguyên âm. Ví dụ: nước(nước), trừong(trường),…Như vậy, ta cần phải xây dựng một module xác định và sửa lỗi cho từ. Giải pháp đề nghị ở đây là chuyển các từ về một định dạng riêng, gồm hai phần: phần đầu là các ký tự không dấu, phần sau là dấu tổ hợp nguyên âm và dấu thanh. Giai đoạn chuyển mã sẽ thực hiện chuyển các dấu tổ hợp nguyên âm và dấu thanh ra cuối của từ.

Ví dụ: hường → truong772

hừong → truong772

hừong → truong772

huờng → truong772

Như vậy, dù dấu thanh có bỏ ở vị trí nào thì cũng cho chuỗi ký tự giống nhau sau khi xử lý. Ngoài ra, nó còn có khả năng phát hiện ra những từ mà người dùng gõ thiếu dấu tổ hợp nguyên âm. Ví dụ: huờng → huong72, chương trình sẽ tìm kiếm trong cơ sở dữ liệu và sẽ thấy đúng được phần đầu, còn về dấu thanh thì sẽ chọn một trong các tổ hợp gần nhất có thể có trong từ điển như:

hương → huong77

hướng → huong771

hường → huong772

hưởng → huong773

hượng → huong775

Trong tự điển tiếng Việt, không thể có các từ như trương hay trương. Nên người dùng chắc chắn đã gõ thiếu và do đó phải là từ truong772 (trường).

3.4 Giải quyết các vấn đề về từ của tiếng Việt

3.4.1 Luật xác định các từ láy

Từ láy không là từ có ý nghĩa quan trọng trong tài liệu. Vì vậy, ta cần xác định từ láy để giảm số lượng các từ ghép cần lập chỉ mục. Trong các loại từ láy thì láy hai là nhiều nhất. Vì vậy, ta cần phải xây dựng luật để xác định nó. Mỗi luật tương ứng với một loại từ láy:

Từ láy hoàn toàn. Ví dụ: xanh xanh =>Luật xác định từ láy hoàn toàn

Từ láy phụ âm đầu. Ví dụ: biêng biếc=>Luật xác định từ láy phụ âm đầu

Từ láy vần. Ví dụ: chót vót =>Luật xác định từ láy vần

3.4.2 Luật xác định các liên từ

Liên từ đầu câu cũng không đóng vai trò quan trọng trong tài liệu. Hầu hết các trường hợp thì theo sau các liên từ đầu câu là dấu phẩy. Ví dụ: Vì thê,…Do đó,…Ta có thể dựa vào cú pháp này để xây dựng luật xác định các liên từ để giảm số lượng từ cần lập chỉ mục.

3.5 Xây dựng từ điển tiếng Việt

Việc xác định từ cho tiếng Việt thì phương pháp giải quyết là **dùng từ điển từ được lập sẵn.**

Tuy nhiên không thể có một từ điển đầy đủ được vì có những từ có thể "sinh ra thêm" trong tương lai. Ví dụ do sự ra đời của nhiều ngành khoa học công nghệ mới đòi hỏi phải phát sinh thêm từ mới để mô tả chúng..., hoặc do nhu cầu sử dụng tiếng nước ngoài ngày càng tăng dẫn đến tình trạng Việt hoá các từ thông dụng như bit, bite, inch,…

Do đó bên cạnh việc sử dụng từ điển đòi hỏi phải có phương pháp để phát hiện thêm từ tiếng Việt mới chưa có trong từ điển để **bổ sung cho từ điển**. Một "từ" tiếng Việt là sự kết hợp của hai hay nhiều "tiếng" có thể dễ dàng xác định bằng khoảng trắng, phương pháp xác định "từ" gồm nhiều "tiếng" ghép lại dựa trên việc xem xét độ gắn kết (cohesion) của chúng:

$$cohesion_{ij} = size\_factor * pair\_freq_{ij} / (n_i * n_j)$$

Trong đó:

size_factor : kích thước tập chỉ mục.

$pair\_freq_i$: tần số xuất hiện từ.

n_i, n_j : tần số xuất hiện của tiếng i, j.

Sự kết hợp 2 tiếng có khả năng cho ra từ cao khi cohesion của chúng cao (2 tiếng thường xuất hiện chung với nhau thì nó có khả năng là một từ).

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.

Quá trình xác định thêm số từ thiếu có thể được cài đặt bằng phương pháp xác định từ ghép tự động như đã nêu trên với tập tài liệu mẫu cho việc xác định từ ghép được chỉ định, hoặc bổ sung thêm từ mới vào từ điển

Tuy nhiên, khi xác định một từ mới thì không thể thêm ngay nó vào từ điển. Vì làm như vậy sẽ dẫn đến tình trạng bùng nổ dữ liệu. Bởi vì một từ là mới do quá trình tách từ chưa hẳn là từ mới thật sự. Do đó, ta sẽ xây dựng thêm từ điển phụ để chứa các từ mới này, sau một khoảng thời gian kiểm tra các thông số như:

- Số lần xuất hiện trong tất cả các tài liệu mà hệ thống xử lý
- Số tài liệu mà từ đó xuất hiện
- …..

Nếu các thông số trên đạt một tiêu chuẩn nào đó thì mới chính thức thêm nó vào từ điển chính và xoá nó ra khỏi từ điển phụ.

Chương 4: BỘ TÌM KIẾM THÔNG TIN – SEARCH ENGINE

1. Vì sao ta cần một công cụ tìm kiếm (SE) ?

Tưởng tượng ta muốn tìm vài quyển sách trong một thư viện rất lớn. Với sức lực cá nhân ta không thể xem qua hết tất cả sách, vì vậy ta cần một danh mục sách. Tương tự, tồn tại hàng triệu trang web trên thế giới và mỗi phút trôi qua số lượng được đẩy lên càng nhiều hơn, cho dù ta có trong tay một công cụ lướt web tuyệt vời đến đâu cũng không thể duyệt hết. Tuy nhiên, với sự trợ giúp của SE, ta có thể thậm chí xác định được vị trí của những từ cần tìm trong các trang web khắp nơi trên thế giới.

2. Các phương thức tìm kiếm

2.1 Tìm theo từ khoá – Keyword searching

Đây là phương pháp được áp dụng với hầu hết các search engine. Trừ khi tác giả của trang web xác định từ khóa cho tài liệu của mình, ngược lại điều này phụ thuộc vào search engine. Như vậy các search engine sẽ tự mình chọn và đánh chỉ mục cho những từ mà chúng cho quan trọng có thể giúp phân biệt các tài liệu khác nhau. Các từ được đề cập trong phần II chương II hoặc các từ lặp lại nhiều lần đều được chú ý. Một số site lập chỉ mục cho tất cả các từ có trong một trang web, một số khác chỉ chọn một số đoạn văn bản.

Các hệ thống đánh chỉ mục trên toàn văn bản (full-text indexing systems) đếm số lần xuất hiện của mỗi từ trong tài liệu ngoại trừ các từ stopword. Có những công cụ tìm kiếm còn phân biệt cả chữ hoa lẫn chữ thường.

2.2 Những khó khăn khi tìm theo từ khoá

Search engine thường gặp rắc rối với những từ đồng âm khác nghĩa (ví dụ hard cider, hard stone, a hard exam, hard drive) hoặc những từ có các biến thể khác nhau do có tiền tố và hậu tố như big, bigger, student, students, …. Bên cạnh đó search engine cũng không thể trả về các tài liệu chứa những từ đồng nghĩa với các từ trong câu truy vấn.

2.3 Tìm theo ngữ nghĩa – Concept-based searching

Excite đã từng nổi tiếng với chiến thuật tìm theo ngữ nghĩa nhưng giờ đây chiến thuật này không còn được sử dụng nữa. Không giống các hệ thống tìm theo từ khoá, hệ thống tìm theo ngữ nghĩa sẽ 'đoán' ý muốn của người dùng thông qua câu chữ. Tìm theo ngữ nghĩa hoạt động dựa trên hình thức gom nhóm tài liệu, phức tạp hơn thì dựa vào ngôn ngữ học, các thuyết về trí tuệ nhân tạo. Excite tiếp cận dựa vào phương pháp tính toán bằng cách đếm số lần xuất hiện của các từ quan trọng. Khi nhiều từ hoặc những cụm từ có nghĩa đặt gần nhau trong tài liệu thì Excite sẽ cho rằng chúng đang ám chỉ một chủ đề nào đó.

Ví dụ, khi từ 'heart' đứng gần các từ như 'attack' (cơn đau tim), 'blood' (sự sống), 'stroke' (sự say nắng), thì search engine sẽ xếp những trang chứa các từ này vào chủ đề y học và sức khoẻ. Ngược lại, khi từ 'heart' đứng gần các từ 'flowers', 'candy', … thì search engine sẽ xếp những trang chứa các từ này vào chủ đề trữ tình.

3. Các chiến lược tìm kiếm

Mọi người đều nhận xét rằng web là nơi mà ta luôn có được thông tin về bất kỳ chủ đề gì. Nhưng kết quả cuối cùng thường là lãng phí thời gian cho những URL vô ích. Do đó đã đến lúc ta nghĩ đến các chiến lược tìm kiếm.

Ta khởi đầu với một đống thông tin trên một chủ đề khá rộng ? Hoặc ta đã

hình dung được cụ thể những gì cần tìm ? Hay ta muốn tìm địa chỉ của ai đó ?

Nếu phạm vi quan tâm của ta quá rộng, ta nên xem xét các thư mục web (web directory). Nếu sau đó ta thu hẹp phạm vi cần tìm, hãy xem xét việc lựa chọn một công cụ tìm kiếm thích hợp.

3.1 Tìm thông tin với các thư mục chủ đề

Giống như tìm sách trong thư viện, cân nhắc giữa tìm theo tác giả, tiêu đề, chủ đề, ta thường chọn chủ đề để có thể bao quát một vùng thông tin rộng hơn.

Ví dụ : ta muốn tạo một trang chủ (home page) nhưng không biết cách viết một file HTML như thế nào, thậm chí chưa từng tạo một file ảnh, và cũng không biết cách đẩy một trang lên mạng. Tóm lại ta cần những thông tin cho một chủ đề khá rộng - xuất bản một trang web (web publishing).

Khi hoàn toàn xác định mình cần tìm những gì ta nên bắt đầu từ một thư mục web như thư mục của Yahoo hoặc Google,…vì thư mục web tập trung nhiều vào chủ đề đang được quan tâm hơn là một công cụ tìm kiếm.

Gần đây các web site thường kết hợp thư mục web và các công cụ tìm với nhau. Ví dụ nếu ta sử dụng Google để tìm thông tin và một trong những kết quả này nằm

trong thư mục web của Google, Google sẽ cung cấp cho ta một liên kết dẫn vào thư mục.

3.2 Tìm thông tin với các công cụ tìm kiếm

Một số công cụ tìm kiếm gặp rắc rối với dữ liệu đầu vào của người dùng. Ví dụ : những từ chứa các ký tự đặc biệt như *C++* , những từ chứa stopword như *to be or not to be*. Xét ví dụ khác ít rõ ràng hơn. Giả sử ta là một người rất thích tiểu thuyết trinh thám và muốn tìm những trang nói về các tác giả yêu thích. Nếu đơn giản chỉ nhập vào các từ ‘mystery‘ và ‘writer’, phần lớn các search engine sẽ trả về các liên kết dẫn đến các trang chứa một trong 2 từ trên hoặc cả 2. Như vậy có khả năng hàng trăm, thậm chí hàng ngàn URL không mong muốn. Tuy nhiên nếu ta nhập vào 1 cụm từ, kết quả sẽ khả quan hơn.

3.3 Tối ưu câu truy vấn

Rất nhiều search engine áp dụng các toán tử Boolean (Boolean operators) hoặc các bộ định vị trí (proximai locators) để tối ưu câu truy vấn. Đó là các từ khoá sau :

STT	Từ khóa	Ý nghĩa
1	AND / phép toán +	Mọi từ trong câu truy vấn phải có trong tài liệu
2	OR	Tài liệu chứa ít nhất một từ cần tìm
3	NOT / phép toán -	Tài liệu không chứa [các] từ sau từ khoá

4	NEAR	Các từ cần tìm cách nhau bao nhiêu ký tự trong tài liệu
5	FOLLOWED BY / ADJ	Các từ cần tìm phải đứng cạnh nhau trong tài liệu

Bảng 4.1 : Các từ khóa giúp tối ưu câu truy vấn

3.4 Truy vấn bằng ví dụ

Một điểm đáng kích lệ nữa của search engine là truy vấn bằng ví dụ. Sau khi liệt kê một loạt các tài liệu được cho là thoả yêu cầu người dùng, search engine còn 'gợi ý' một vài site có liên quan đến chủ đề ta đang quan tâm. Nếu có thể ta hãy theo sau các liên kết này, biết đâu sẽ có kết quả khả quan!

Chương 5: MỘT SỐ SEARCH ENGINE THÔNG DỤNG TRÊN THẾ GIỚI VÀ VIỆT NAM

Vài nét về các đặc trưng của một số search engine thông dụng trên thế giới

Search Engine	**Google**	**AlltheWeb**	**AltaVista**	**Teoma**
Database	google.com	alltheweb.com	altavista.com	teoma.com
Kích thước(# trang)	Khoảng 4 tỉ (1 tỉ không đánh chỉ mục trên toàn văn bản)	Khoảng 3 tỉ, chỉ mục trên toàn văn bản.	Khoảng 1 tỉ	Khoảng 1 tỉ
Đa phương tiện (multimedia)	Hỗ trợ	Hỗ trợ	Hỗ trợ	Không hỗ trợ
Toán tử				
Mặc định	AND	AND	AND	AND
Loại trừ	-	-	-	-

Cụm từ	Dùng dấu “ “	Dùng dấu “ ”	Dùng dấu “ ”	Dùng dấu “ “
Rút gọn	Không hỗ trợ Dùng ký tự * để thay thế cho các ký tự trong dấu “ “	Không hỗ trợ	Dùng ký tự *	Không hỗ trợ
Boolean	OR (chỉ dùng cho danh từ riêng)	AND, OR, ANDNOT, RANK, ()	AND, OR, ANDNOT, NEAR, ()	OR (chỉ dùng cho tên riêng)
Stop words	Thông thường bỏ qua các từ thông dụng + nếu muốn tìm và phải đặt trong cặp dấu “ “		Dùng dấu “ “ trong search cơ bản Bỏ qua trong search nâng cao	Thông thường bỏ qua các từ thông dụng + nếu muốn tìm
Danh từ riêng	Không hỗ trợ	Không hỗ trợ	Hỗ trợ	Không hỗ trợ
Giới hạn field cần tìm	intitle:inurl: allintitle: allinurl: filetype:	normal.title: url.all: link.all:	title:domain: link:image: text:url:host:	intitle:inurl: site:geoloc:lang: last: afterfate:

	link:site: Trong search nâng cao : cache:info:	link.extension:	anchor:applet:	
Các đặc tính đặc biệt	~ tìm từ đồng nghĩa Giới hạn bởi ngôn ngữ Nhiều kiểu file : pdf, doc,… Caches : trang web khi đánh chỉ mục	Duyệt qua các URL Trong tìm nâng cao : giới hạn bởi ngày, domain, địa chỉ iP	Giới hạn bởi ngày, vị trí, ngôn ngữ Trong tìm nâng cao : sử dụng *sortby* để lọc và sắp xếp kết quả.	Dùng *refine* để tối ưu kết quả. *Resource* để có được các trang và liên kết tập trung trên chủ đề cần tìm.
Ưu điểm				
Ưu điểm chính	Rất tốt với những trang có độ phổ biến cao. Các trang tin tức gần đây	Tốt như Google. Không có stop word.	Dùng nhiều toán tử Boolean trong tìm kiếm. Trong tìm nâng cao hỗ trợ hiển thị kết quả theo độ phổ biến của	Tính độ phổ biến tốt, dựa vào số lượng trang web cùng chủ đề với các trang đang xét. Thường đạt kết quả đáng khích lệ.

			từ.	
Search Engine	Google	AlltheWeb	AltaVista	Teoma

Bảng 5.1 : Bảng hướng dẫn nhanh về cách sử dụng các search engine phổ biến trên thế giới

Search engine	Cơ sở dữ liệu	Toán tử	Lực chọn tìm kiếm	Linh tinh
Google http://www.google.com Hỗ trợ tìm kiếm nâng cao Hệ thống thư mục chủ đề (Subject Directory) Hệ thống thư mục mở (Open	Toàn văn bản của các trang web, .pdf, .doc, .xls, .ps, .wpd (4.3B, + 1B một phần của chỉ mục URLs) Tin tức : cập nhật thường xuyên (4500 nguồn). Các dạng file	AND (mặc định) OR (danh từ riêng) + cho các stop word thông dụng, cho các URL hoặc các trang cụ thể (ví dụ +edu) - loại trừ	Dùng * để rút gọn. Dùng “” tìm cụm từ. fields : intitle:, inurl:, link:, site: Tìm trên hệ thống danh mục các chủ đề trong thư mục web. Tìm các trang web tương tự.	Kiểm lỗi chính tả. Lưu trữ các trang đã lập chỉ mục. Tốt cho tìm các trang hay bị lỗi 404. Phiên dịch đến 5 ngôn ngữ. ~ tìm từ đồng nghĩa.

Directory)	ảnh Nhóm : Usenet từ 1981 đến nay			
AlltheWeb http://allthewe b.com Hỗ trợ tìm kiếm nâng cao	Toàn bộ văn bản các trang web, .pdf, Flash, (3.1B toàn bộ chỉ mục URLs) Tin tức : cập nhật thường xuyên (3000 nguồn) Tranh ảnh Video Audio FPT	AND (mặc định) OR, phải đặt các từ trong dấu " ". ANDNOT, RANK - để loại bỏ	Không rút gọn. Dùng dấu " " cho cụm từ. Field intitle:inurl: link:site: Trong tìm nâng cao : giới hạn theo ngày, ngôn ngữ, domain, file format, địa chỉ iP.	Kiểm lỗi chính tả. Tìm nâng cao : tranh ảnh, video. Hỗ trợ sử dụng kỹ thuật "clusters" để tối ưu câu truy vấn.
AltaVista http://altavist	Toàn bộ văn bản các trang web (khoảng	AND (mặc định)	Dấu * để rút gọn. Dấu "" cho cụm	Kiểm lỗi chính tả.

a.com Hỗ trợ tìm kiếm nâng cao Hệ thống thư mục chủ đề (Subject Directory) Hệ thống thư mục mở (Open Directory)	1B) và file .pdf. Tin tức (3000 nguồn), ảnh, MP3/Audio, Video.	Trong tìm nâng cao hoặc danh từ riêng trong tìm cơ bản : AND, OR, ANDNOT, NEAR, dấu () lồng nhau. - cho loại trừ.	từ. Tìm nâng cao : giới hạn ngày, ngôn ngữ.	Phiên dịch : 8 ngôn ngữ của Châu Âu & các ngôn ngữ của Châu Á. AltaVistaPrima : tối ưu câu hỏi.
Teoma http://teoma.com Hỗ trợ tìm kiếm nâng cao	Toàn bộ văn bản trang web (khoảng 1B)	AND (mặc định) OR (danh từ riêng) + hoặc “” cho stopword - để loại bỏ	Không rút gọn. Dùng dấu “ “ cho cụm từ. Field intitle:inurl: site:geoloc:lang:last: afterdate:beforedate: betweendate: Trong tìm nâng cao :	Kiểm lỗi chính tả. Gom nhóm kết quả *Refine* để tối ưu câu hỏi. *Resource* để có các trang hoặc liên kết tập trung vào chủ đề.

			giới hạn theo ngày, ngôn ngữ, domain, file format, địa chỉ iP.	
AskJeeves www.ask.com	Nhận kết quả từ CSDL của Teoma. Tìm sản phẩm : PriceGrabber.com, Tìm tranh ảnh : Picsearch.com Tìm tin tức : Moreover.com.	Giống Teoma. Đối với những câu hỏi đơn giản, xuất hiện cửa sổ đối thoại.	Giống Teoma. Click vào *Remove Frame* để thấy URLs của các trang.	Kiểm lỗi chính tả.

AskJeeves for Kids www.ajkids.com	Trả lời tốt các câu hỏi đơn giản. Games cho trẻ em, Tin tức theo từng nhóm tuổi.	Hỏi bằng ngôn ngữ tự nhiên. Không sử dụng các toán tử Boolean.	Click vào *No frames* để thấy URL của trang kết quả.	Dẫn đến các trang phục vụ học tập : tự điển, vật lý, khoa học, bản đồ, lịch sử,…

Bảng 5.2 : Sơ lược về các đặc trưng của một số search engine thông dụng trên internet

Meta-search engine	**Cơ sở dữ liệu**	**Toán tử**	**Lực chọn tìm kiếm**	**Linh tinh**
Vivisimo http://vivisimo.com	Netscape, MSN, Lycos,LookSmart, …	AND(mặc định), OR, -	Tìm trên chủ đề : tin tức, thương mại, kỹ thuật, thể thao.	Gom nhóm kết quả. Tốt đối với chủ đề về các sự kiện & nhiều khía cạnh khác.
Dopgpile http://dopgpi	Google, Yahoo,	Tìm nâng cao : AND, OR,	Sắp xếp theo kết quả.	Kiểm lỗi chính tả.

le.com	AltaVista, Teoma/AskJeeves, About.com, FAST, FindWhat, LookSmart	ANDNOT.	Xoá các kết quả trùng.	Highlight từ cần tìm trong kết quả. Gom nhóm kết quả. Tốt đối với chủ đề về các sự kiện & nhiều khía cạnh khác.

Bảng 5.3 : Các meta-search engine thông dụng trên internet

Thư mục chủ đề	Cơ sở dữ liệu	Toán tử	Lực chọn tìm kiếm	Linh tinh
Yahoo http://dir.yahoo.com	Xem xét các trang web (khoảng 13K)	AND(mặc định) OR -	Cụm từ : “” Rút gọn : * Fields t: title, u:URL	Nhiều dịch vụ trong Yahoo: Tin tức : từng giờ. Thể thao :tỉ số,.. Bản đồ, thời tiết, mua sắm.

Academic info http://academicinfo.net	Mức độ cao đẳng hoặc nghiên cứu (cũng hữu ích cho trung học). Được chọn và chú thích bởi thủ thư Michael Madin (khoảng 25K)	AND, OR(mặc định), NOT, dấu () lồng nhau.		Dẫn đến các chương trình mức độ cao đẳng hoặc các site, các nguồn tài nguyên khác hữu ích cho sinh viên.

Bảng 5.4 : Các hệ thống thư mục theo chủ đề thông dụng trên internet

1.1 Thư mục của Yahoo, Google

- Về bản chất là các danh mục chủ đề.
- Sắp xếp các trang theo mức độ quan trọng của chúng.
- Tìm theo đề tài hoặc chủ đề.

Google là một trong những công cụ tìm kiếm mới nhưng nhanh chóng được ưa chuộng nhờ khả năng tìm nhanh và chính xác. Ý tưởng chính của công cụ này là đo lường độ quan trọng của một trang dựa vào số liên kết đến trang đó. Nói cách khác nếu

nhiều webmaster cùng quan tâm đến một website thì website đó xứng đáng được đánh giá cao. Yahoo đã từng dùng Google như một chức năng của mình trong một thới gian dài. Kỹ thuật tìm kiếm của Microsoft và MSN.com dựa trên kỹ thuật inktomi. (inktomi đã được áp dụng cho một trong những search engine nổi tiếng trong những năm 90 là Hobot) Microsoft đã rất nỗ lực trong việc tìm ra một kỹ thuật cho riêng mình nhưng vẫn chưa thành công.

1.2 Alltheweb

Alltheweb cũng là một trong những công cụ tìm kiếm mới, được cho là công cụ dò tìm nhanh hơn và hiệu quả hơn các search engine khác nhờ một lượng chỉ mục rất lớn. Alltheweb đã được sử dụng bởi Yahoo.

1.3 AltaVista

Đã từng là một trong những công cụ tìm kiếm được ưa chuộng nhất nhưng bị đánh bại bởi Google. Mặc dù vậy nó vẫn là một search engine cho kết quả chính xác và từng được Yahoo sử dụng.

1.4 Lycos

Được mô tả như là những cổng truy cập web (web portal) hay những trung tâm truy cập, là nơi mà người dùng đi vào để lấy thông tin cho mọi lĩnh vực, kể cả tán gẫu, gởi thư điện tử,…

1.5 HotBot

Đã đề cập ở trên, HotBot dựa trên kỹ thuật inktomi, là công cụ tìm kiếm chuyên biệt, cung cấp nhiều thông tin chính xác, nhanh chóng cho lĩnh vực thương mại và các

mục đích chuyên môn, hứa hẹn một sự thay thế cho các công cụ thường dùng khác khá tốt.

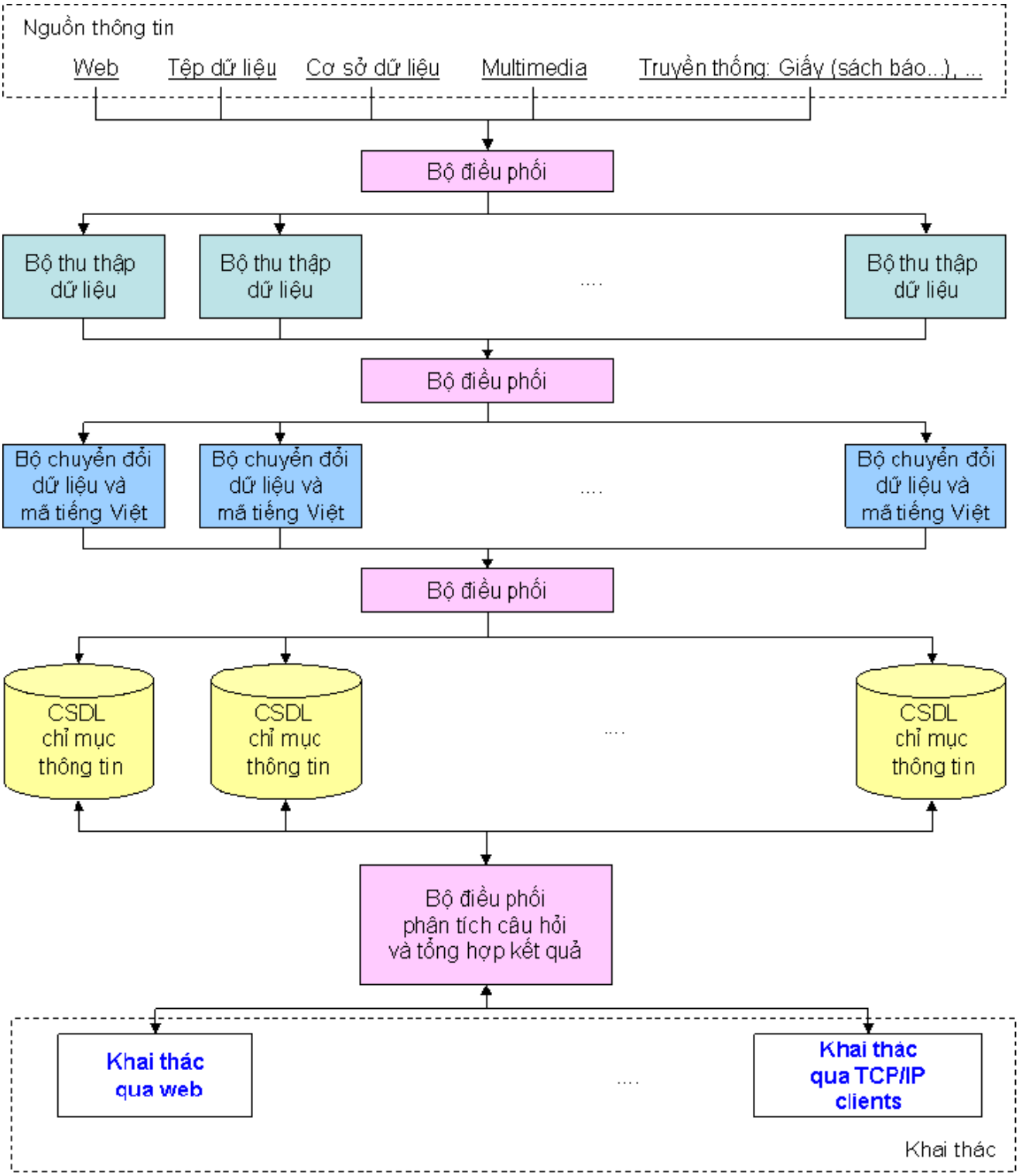
2. Một số search engine thông dụng ở Việt Nam

2.1 Netnam [IV.12]

Là một đơn vị thuộc viện hàn lâm - Viện Công nghệ Thông tin, Netnam đặc biệt chú trọng đến việc thiết kế hệ thống phù hợp với điều kiện cơ sở vật chất - hạ tầng còn khiêm tốn ở Việt Nam. Vì vậy, một trong những ưu tiên quan trọng trong các yêu cầu xây dựng hệ thống là khả năng tiết kiệm chi phí đầu tư cơ sở hạ tầng kỹ thuật, đồng thời phải đáp ứng được nhu cầu mở rộng cao. Do đó Netnam SE được thiết kế theo kiến trúc sử lý song song.với các khối chức năng như hình dưới đây. Kiến trúc này cho phép hệ thống có thể phân tán trên từ một đến hàng trăm máy tính, cho phép sử dụng các máy tính PC cỡ nhỏ thay cho các hệ máy tính chủ cao cấp. Từ đó hệ thống cho phép tiết giảm chi phí tối đa trong việc xây dựng hạ tầng ban đầu, đồng thời khi nhu cầu tính toán hoặc yêu cầu phục vụ liên tục tăng, chỉ cần thêm các máy tính vào hệ thống để tăng cường khả năng xử lý và khả năng phục vụ liên tục mà không cần bổ sung bất cứ thành phần nào khác.

Phần kiến trúc này sẽ giới thiệu về mô hình chia sẻ tính toán song song của hệ thống.

Về mặt vật lý, các máy tính được có thể kết nối với nhau đơn giản bằng hệ thống mạng Ethernet 10/100/1000Mbps. Hệ thống cho phép thay đổi nóng (hotswap) một hoặc một vài đơn vị vật lý (máy tính) mà không làm ảnh hưởng đến hoạt động của toàn hệ thống, cũng như cho phép thực hiện thay thế tự động một hoặc một vài đơn vị vật lý của hệ thống khi chúng gặp sự cố bất ngờ.



Hình 5.1 Sơ đồ hệ thống Search Engine của Netnam

Hệ thống được chia thành ba tầng chính, gồm tầng Thu thập thông tin, Nhận dạng và chuyển đổi thông tin thành dạng text, Lập cơ sở dữ liệu cho các thông tin text. Mỗi tầng được chia thành nhiều đơn vị độc lập hoạt động theo kiểu chia sẻ tính toán và/hoặc dự trữ (redundant), từ đó tính tin cậy và hiệu năng của hệ thống cho phép rất cao đối với các hệ thống đòi hỏi tính tin cậy và hiệu năng cao.Đơn vị khai thác dữ liệu được tích hợp cùng với phần lập chỉ mục cơ sở dữ liệu, cho phép khai thác qua các clients sử dụng giao thức TCP/iP trên bất cứ hệ thống nào (Windows, Unix...) Bằng việc chia hệ thống thành các khối chức năng phối hợp với nhau thông qua các Bộ điều phối, hệ thống có thể được phân tán để xử lý trên nhiều máy tính nhỏ thay vì tập trung toàn bộ hệ thống trên một máy tính lớn. Vì vậy, một mặt hệ thống cho phép sử dụng các máy tính cỡ nhỏ (PC hoặc PC server) cùng phối hợp tính toán xử lý, do đó làm giảm rất nhiều chi phí đầu tư so với các hệ máy cỡ mini hay mainframe, và có thể đầu tư dần dần theo sự gia tăng của nhu cầu thay vì đầu tư toàn bộ một lần ban đầu. Mặt khác, nó cho phép, về mặt nguyên tắc, năng lực tính toán, phục vụ thông tin của hệ thống là không hạn chế ? khi nhu cầu tăng lên chỉ cần thêm máy tính vào hệ thống mà không phải thay đổi lại hệ thống. Vì vậy, lượng dữ liệu mà hệ thống có thể phục vụ, về mặt nguyên tắc thiết kế hệ thống, cho phép lên đến hàng trăm triệu tài liệu.

2.1.1 Phương pháp Netnam SE lập chỉ mục dữ liệu

Thông thường, NetNam lấy tất cả các từ trong tài liệu để lập chỉ mục, và khi trả kết quả tìm kiếm, NetNam Search Engine tìm ra tất cả các từ trong một trang tài liệu đó, và hiển thị một số từ đầu tiên như một bảng tóm tắt ngắn. Với Netnam ta thể dùng thẻ META trong trang web để :

- Cung cấp thêm các từ khoá có ảnh hưởng đến kết quả tìm kiếm của NetNam Search Engine (tác dụng tìm ra trang mà ta cần tìm).
- Đưa ra các miêu tả để hiển thị kết quả tìm kiếm.

<META name="keywords" content="xe máy, ô tô, mới 100%">

Ví dụ, giả sử ta có một trang web quảng cáo bán ô tô, xe máy mới 100%, trang search của NetNam sẽ tự động chỉ ra các từ trong trang. Tuy nhiên, có một số từ hoặc cụm từ miêu tả dịch vụ lại không xuất hiện trong trang. Sử dụng thẻ META và ghi rõ tên="từ khoá" để thêm các cụm từ vào chỉ mục và làm tăng cơ hội tìm kiếm cho người sử dụng khi muốn tìm trang web.

Miêu tả thẻ META cho phép ta tìm được cái mà ta muốn tìm trong bản tóm tắt kết quả tìm kiếm. Với trang bán xe của mình, ta có thể muốn một cụm từ quảng cáo ngắn như sau: <META name="description".

content="Bán xe máy và ô tô với chất lượng cao, bảo hành chu đáo, giá phù hợp.">

NetNam Search Engine chỉ ra các từ trong thẻ miêu tả cùng với những thẻ từ khoá. Do đó trong ví dụ này, người sử dụng sẽ có thể tìm ra trang web của ta bằng cách tìm từ "chất lượng cao" cũng như "giá phù hợp", "bảo hành chu đáo".
Thay vì hiển thị một số dòng đầu của trang web, kết quả tìm kiếm sẽ hiển thị văn bản của thẻ miêu tả:
Car Leasing Corp.
Bán xe máy và ô tô với chất lượng cao, bảo hành chu đáo, giá phù hợp.
http://www.vnmotors.com.vn/ - 3K ? 01/11/2001
Chú ý: các thẻ miêu tả và các thẻ từ khoá có thể dài tối đa là 1024 ký tự.

2.1.2 Cú pháp tìm kiếm

Cả hai chức năng tìm kiếm đơn giản và nâng cao đều sử dụng những quy tắc cú pháp giống nhau đối với các cụm từ, phân biệt dạng chữ, và tìm những từ liên quan.

Từ và cụm từ

NetNam Search Engine định nghĩa một từ cũng như bất cứ một chuỗi những chữ cái và con số được tách rời nhau :

- Ký tự trắng, như dấu cách, dấu tab, chấm xuống dòng, hoặc chỗ bắt đầu hoặc kết thúc của một tài liệu.
- Các ký tự đặc biệt và hệ thống chấm câu, ví dụ như %, $, /, #, và _

Ví dụ, hệ thống tìm kiếm của NetNam sẽ giải thích và chỉ rõ những từ Proliant, 60258, www, http, và XeMayCu như những từ riêng lẻ, vì chúng là những chuỗi ký tự liên tiếp, được bao quanh bởi các ký tự không phải là chữ cũng không phải là số. Phần mềm tìm kiếm sẽ chỉ ra tất cả các từ mà nó tìm được trong một trang tài liệu web không quan tâm liệu từ đó có trong từ điển hay đánh vần sai hay không.

Tìm kiếm cụm từ

Ta có thể tìm thấy các cụm từ, hoặc một nhóm từ liên quan xuất hiện ngay cạnh nhau. Để tìm được một cụm từ, ta đóng mở ngoặc kép ở đầu và cuối cụm từ đó để tạo thành một cụm từ. Cụm từ đảm bảo rằng NetNam Search Engine sẽ tìm được các từ đúng như thế (vị trí, thứ tự, không có từ chen giữa...), chứ không phải là tìm được riêng từng từ một.

Hệ thống chấm câu

NetNam Search Engine sẽ bỏ qua hệ thống chấm câu trừ trường hợp phải thể hiện hệ thống chấm câu đó là một dấu chia cách giữa các từ. Đặt hệ thống chấm câu hoặc các ký tự đặc biệt giữa các từ, và giữa chúng không có dấu cách, cũng là một cách để tìm một cụm từ. Một ví dụ cho thấy hệ thống chấm câu rất hữu dụng trong việc tìm một cụm từ đó là trường hợp tìm số điện thoại. Ví dụ để tìm được một số điện thoại

0903401357 ta gõ 09-0340-1357 thì sẽ dễ tìm hơn là gõ theo kiểu "09 0340 1357", mặc dù đây cũng là một cú pháp có thể chấp nhận được nhưng ít phổ biến. Các từ có dấu nối ở giữa như CD-ROM, cũng tự động làm thành một cụm từ do có dấu gạch nối ở giữa.

Tuy nhiên, thông thường, sử dụng dấu ngoặc kép để tìm một cụm từ là cách được khuyến khích dùng hơn là sử dụng hệ thống chấm câu, vì một số ký tự đặc biệt còn có nghĩa phụ:

- Dấu + và - là những toán tử giúp lọc kết quả của một tìm kiếm đơn giản.
- &, |, ~ và ! là những toán tử giúp lọc kết quả của một tìm kiếm nâng cao

Phân biệt chữ thường/hoa

Phân biệt dạng chữ là một loại tìm kiếm dựa vào loại chữ mà ta gõ yêu cầu tìm kiếm của mình vào.

- Một yêu cầu bằng chữ thường sẽ có kết quả tìm kiếm không theo dạng chữ ta gõ vào. Ví dụ, nếu ta gõ chữ yết kiêu vào ô yêu cầu, NetNam Search Engine sẽ tìm tất cả các biến thể của từ yết kiêu, gồm có yết kiêu, Yết Kiêu, YếT KiÊU, v.v...
- Nếu yêu cầu có cả chữ hoa, thì kết quả tìm kiếm sẽ là tìm kiếm theo dạng chữ. Ví dụ, nếu ta điền Yết Kiêu vào ô yêu cầu, NetNam Search Engine sẽ tìm tất cả các biến thể của Yết Kiêu chỉ với chữ đầu tiên là chữ hoa. Nó sẽ không trả về các văn bản có chữ YếT KiÊU hay yết kiêu.

2.1.3 Sử dụng từ khoá để lọc các tìm kiếm

Cả giao diện của công cụ tìm kiếm đơn giản và nâng cao đều hỗ trợ việc sử dụng các từ khoá để hạn chế các tìm kiếm tới các trang đáp ứng tiêu chuẩn được định

rõ về nội dung và cấu trúc của một trang web. Sử dụng từ khoá, ta có thể tìm kiếm dựa vào URL hoặc một phần của một URL, hoặc dựa vào các liên kết, hình ảnh, văn bản, mã hoá của một trang web. Các từ khoá sẽ rất có ích trong trường hợp:

- Tìm các trang trên một máy chủ nào đó hoặc trong một tên miền chỉ định
- Tìm các trang có chứa các liên kết trỏ tới trang web của ta.
- Tìm các trang có chứa một lớp Java applets.

Tìm kiếm dựa vào từ khoá, gõ một yêu cầu bằng từ khoá:lệnh tìm kiếm Gõ từ khoá bằng chữ thường, sau đó là dấu hai chấm. Quy ước để tìm một cụm từ trong lệnh tìm kiếm sẽ giống với quy ước để tìm một cụm từ trong một yêu cầu bình thường: phương pháp thường được sử dụng nhất là cho cụm từ vào trong ngoặc kép. title:"**thời trang**"

Từ khoá	Chức năng
applet:class	Tìm các trang có chứa một ứng dụng nhỏ (applet) Java hoặc Java class
Domain:domainname	Tìm các trang có từ hoặc cụm từ trong tên miền của máy chủ web nơi có trang cần tìm. (Phần.... của tên máy chủ internet là tên miền)
host:name	Tìm các trang có từ hoặc cụm từ trong tên của máy chủ web, nơi có các trang cần tìm.
image:filename	Tìm các trang có chứa ảnh filename

Bảng 5.5 : Bảng miêu tả các từ khoá sử dụng trong việc tìm kiếm.

Các từ khoá url, host, domain, đều có một mục đích là tìm kiếm các URL dựa vào một phần URL, hoặc dựa vào tên máy chủ hoặc tên miền nơi có các trang web cần tìm.

Các từ khoá link và anchor cũng tương tự như khi chúng tìm kiếm thông tin về liên kết. Từ khóa link tìm các văn bản trong một URL là đích của một liên kết (ví dụ, http://www.abc.org.vn/help.htm), trong khi từ khoá anchor lại tìm các văn bản hiện tại của một siêu liên kết khi người dùng nhìn thấy nó trên một trang web (ví dụ, click here).

Thẻ title sẽ tìm kiếm nội dung tiêu đề của một tài liệu. Từ khoá tiêu đề sẽ giới hạnviệc tìm kiếm tới văn bản mà tác giả của tài liệu đã mã hoá như một phần của thẻ <title>. Tiêu đề là cụm từ sẽ xuất hiện trong đầu đề cửa sổ trong trình duyệt web. Từ khoá tiêu đề có thể sẽ là một cách tốt để giới hạn tìm kiếm chỉ tới các trang về một chủ đề, gồm các trang được đặt tiêu đề một cách thông minh. Tuy nhiên với các trang mà người lập nên không quan tâm đến tiêu đề trang web hoặc đặt tên kém thì cách tìm này không dùng được. Hơn nữa, hệ thống tìm kiếm của NetNam có thể cấu hình để nhận biết các thuộc tính phụ khác của tài liệu có các thẻ HTML META do người dùng quy định.

Điều kiện	Định nghĩa
url:http://www.un.org.vn/about	Tìm tất cả các trang có các từ http://www.un.org.vn/about trong URL.
host:netnam.vn	Thoả mãn tất cả các trang có host:netnam.vn trong tên máy chủ web.
domain:org.vn	Thoả mãn tất cả các trang có tên miền org.vn trong tên máy chủ của máy chủ web.
image:about.jpg	Thoả mãn tất các trang có một thẻ hình ảnh liên quan tới image:about.jpg
anchor:"click here"	Thoả mãn tất cả các trang với cụm từ click here trong đoạn văn bản của một liên kết hoặc một thẻ anchor (<A>) khác.
link:http://www.abc.org.vn/mypage.html	Thoả mãn tất cả các trang có ít nhất một liên kết tới một trang có URL http://www.abc.org.vn/mypage.html
link:http://myhost.abc.org/mypage.html	Chỉ tìm các trang có các liên kết tới URL chỉ định.

title:"NetNam Lifestyle"	Thoả mãn các trang có cụm từ NetNam Lifestyle trong tiêu đề
applet:flower	Thoả mãn các trang chứa Java applet có tên gọi flower.

Bảng 5.6 : Ví dụ tìm kiếm thông tin của Netnam

2.2 Vinaseek ([IV.11])

Vinaseek được phát triển từ năm 1997 theo mô hình của các search engine như Google, AltaVista, bổ sung khả năng tìm kiếm chính xác theo từ khoá cho Tiếng Việt, theo mọi bảng mã (TCVN3, VNi, TVCN-6909, ViQR...), theo mọi định dạng tài liệu văn bản (html, xml, rtf, word, pdf, PostScript...), theo mọi cách bỏ dấu khác nhau ("hoà" hay "hòa"), tìm kiếm hình ảnh và âm thanh, tìm kiếm gần đúng, tìm kiếm mờ (fuzzy search), tìm kiếm đồng âm và đồng nghĩa, đang lưu trữ chỉ mục và toàn văn của tất cả các trang Web Tiếng Việt trên internet (ước chừng 10 triệu văn bản), và nhận được hàng trăm ngàn lượt truy cập mỗi ngày.

Cú pháp tìm kiếm của Vinaseek tương tự như Netnam về tìm kiếm từ, cụm từ, cách phân biệt hoa thường nhưng khác về :

Hệ thống chấm câu bao gồm : +, -, khoảng trắng,...

Toán tử : AND, OR, NOT

Field : link:, site:, url:, title:

Điểm qua một vài tính năng nổi bật của Vinaseek ta có thể liệt kê như sau :

- Tốc độ tìm kiếm nhanh và chính xác và hiển thị kết quả đầy đủ.
- Hỗ trợ toàn bộ các bảng mã Tiếng Việt cả trong nước và ngoài nước.
- Số lượng đánh index khoảng 10 triệu trang và liên tục được cập nhật.
- Cung cấp đầy đủ các tính năng tìm kiếm nâng cao với khả năng hiểu chính xác tiếng Việt.
- Cho phép thực hiện các yêu cầu tìm kiếm phức tạp dạng tổ hợp một cách thông minh, hiệu quả
- Có thể đặt ô tìm kiếm Vinaseek tại các website trong và ngoài nước một cách dễ dàng. Ô tìm kiếm Vinaseek hiện đang được đặt trên 40 trang web tiếng Việt hàng đầu trong và ngoài nước.
- Dịch vụ Vinaseek có thể được tối ưu cho các trang web và mạng intranet của khách hàng làm công cụ tìm kiếm nội tại.

Phần 2 : THIẾT KẾ VÀ CÀI ĐẶT

- Ngôn ngữ lập trình : Java, HTML
- Công cụ lập trình : JBuilder X, Microsoft Fontpage
- Web Server : Resin
- Hệ quản trị CSDL : Mircrosoft SQL Server 2000

Chương 6: THIẾT KẾ DỮ LIỆU

Khi thiết kế hệ thống tìm kiếm thông tin vấn đề khó khăn nhất phải đối mặt là **tổ chức cấu trúc dữ liệu** . Vì khối lượng dữ liệu phải lưu trữ của hệ thống tìm kiếm thông tin là rất lớn, và khối lượng yêu cầu tìm kiếm phải xử lý cũng rất lớn (trên môi trường Web) nên cấu trúc dữ liệu phải được tổ chức tối ưu cho việc đáp ứng (về thời gian) đối với yêu cầu tìm kiếm của người sử dụng.

Dữ liệu của hệ thống được xây dựng dựa trên **mô hình vector,** sử dụng phương pháp **tập tin nghịch đảo**

1. Cơ sở dữ liệu trong SQL

CSDL trong SQL server phục vụ cho toàn bộ hệ thống gồm bảng Url : chứa các thông tin cần thiết cho ứng dụng về 1 URL.

Bảng	Tên thuộc	Ý nghĩa	Kiểu dữ liệu	Miền giá trị	Ghi chú

UrlSTT	**tính**				
1	Depth	Độ sâu của FromURL	Integer		0 : đây là StartURL
2	LastModified	Ngày cập nhật nội dung URL gần nhất	Bigint		
3	ContentLength	Kích thước trang web mà URL chỉ tới	Bigint		
4	Status	Trạng thái của URL	Integer	UNDOWNLOAD	URL chưa được download
				GOOD	URL đã được download về

				BROKEN	URL bị hỏng trong khi kết nối với server
				GOOD-INDEXING	URL đã download về và đang được lập chỉ mục
				BROKEN-INDEXING	URL đang được lập chỉ mục nhưng bị hỏng liên kết
5	Title	Tiêu đề của trang web mà URL chỉ	Varchar(255)		

		tới			
6	ContentType	Cho biết nội dung của URL thuộc dạng nào	Varchar(50)		
7	**UrlId**	**Định danh của URL**	**Bigint**		**Khóa chính**
8	UrlName	Địa chỉ của 1 URL	Varchar(255)		Duy nhất
9	OutputPath	Tên file download về	Varchar(255)		
10	RootId	Định danh của StartURL	Bigint		

Bảng 6.1 : Bảng URL

2. Hệ thống tập tin

Do hệ thống dùng Webserver là Resin nên toàn bộ cơ sở dữ liệu được lưu trong thư mục làm việc của Resin "\doc\se\working". Ngoài dữ liệu được lưu trữ trong Hệ quản trị Cơ sở dữ liệu SQL Server 2000, ứng dụng còn có hệ thống file như sau:

"Index.txt": chính là từ điển chỉ mục, chứa thông tin về một mục từ như trọng số, số tài liệu có chứa mục từ này, là từ tiếng Anh hay tiếng Việt, trang bắt đầu và trang kết thúc trong tập tin nghịch đảo, và sẽ được trình bày cụ thể ở phần dưới.

"Inverse.dat": tập tin nghịch đảo, chứa các thông tin về các tài liệu và trọng số của các mục từ trong tài liệu đó, xem cụ thể trong phần tập tin nghịch đảo ở phần dưới.

"UnicodetoUTF8.txt": font chữ Unicode.

Chương 7: THU THẬP THÔNG TIN

1. Cấu trúc dữ liệu

Với mong muốn không chỉ đảm bảo được các nhiệm vụ của web robot mà còn giúp cho quản trị chủ động hơn nữa trong công việc của mình, module web robot sẽ hỗ trợ những chức năng sau :

- URL bắt đầu (StartURL)
  - ✓ Định độ sâu liên kết
  - ✓ Các tuỳ chọn khi phân tích một URL : cùng site, cùng thư mục gốc, khác site.
- Project
  - ✓ Mỗi project có thể có nhiều StartURL. Các project khác nhau có thể có cùng một / nhiều StartURL.
  - ✓ Chỉ phân tích URL để tạo CSDL hoặc download file.
  - ✓ Download vớI 2 tuỳ chọn.
  - ✓ Quy định các dạng và kích thước file cần download.
  - ✓ Không quy định các dạng và kích thước file cần download.
  - ✓ Tạm dừng 1 StartURL để xử lý 1 project khác hoặc 1 StartURL khác cùng project.

- Hệ thống
  - ✓ Định số spider hoạt động đồng thờI, thờI gian đợI giữa 2 lần truy xuất server liên tiếp.
  - ✓ Định số connection ban đầu, thờI gian đợI được cấp tài nguyên, số lần truy xuất CSDL khi hệ thống bận.
  - ✓ Định chu kỳ lưu thông tin một project.

Nhằm đáp ứng những chức năng đã nêu ra, hệ thống thu thập thông tin sẽ được bổ sung thêm các cấu trúc dữ liệu như :

1.1 Cấu trúc UrlInfo

UrlInfo là ánh xạ của bảng URL trong CSDL. Ngoài những thuộc tính kể trên, UrlInfo còn có các thuộc tính :

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Miền giá trị	Ghi chú
1	Depth	Độ sâu của URL	Integer		
2	ErrorCode	Mã lỗI truy xuất CSDL	integer	0 1	Không có lỗI SQLException
3	RootId	Định danh của URL liên kết tới	Long		

		nó			

Bảng 7.1 : Cấu trúc URLInfo

1.2 Cấu trúc StartUrlInfo

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Miền giá trị	Ghi chú
1	Alias	Tên khác của URL	String		
2	Account	Tên để truy cập URL	String		
3	MaxDepth	Độ sâu lớn nhất của StartURL	Integer		
4	ProcessStatus	Trạng thái xử lý của StartURL	Integer	NONE	Chưa được xử lý
				BEING	Đang xử lý
				DONE	Đã xử lý
5	Password	Password truy cập StartURL	String		

Bảng 7.2 : Cấu trúc StartURLInfo

1.3 Cấu trúc FileRetrieval

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Miền giá trị	Ghi chú
1	Description	Miêu tả dạng file cần lấy	String		
2	Extensions	Danh sách các đuôi file cần lấy	String		
3	MaxSize	Kích thước file lớn nhất	Integer		
4	MinSize	Kích thước file nhỏ nhất			

Bảng 7.3 : Cấu trúc FileRetrieval

1.4 Cấu trúc ProjectInfo

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Miền giá trị	Ghi chú

1	ConnDBTimes	Số lần truy xuất CSDL cho mỗI thao tác thêm, xoá, sửa	Integer		
2	LastUpdate	Ngày cập nhật project gần nhất	Long		
3	HasRun	Project đã được thực thi lần nào chưa	Boolean		
4	NumSpiders	Số spider dùng cho project	Integer		
5	NumResource	Số kết nối ban đầu của project	Integer		
6	PrjName	Tên project	String		Duy nhất

7	SpiderTimeout	Thời gian delay giữa 2 lần truy xuất liên tiếp vào server	Long		
8	StartUrl	Danh sách các StartURL	StartUrlVector		
9	ResourceTimeout	Thời gian delay để được cấp phát tài nguyên	Long		
10	RetrievableExt	Những đuôi file cần xử lý	FileRetrievalVector		
11	Outputpath	Tên file chứa thông tin project lưu trên đĩa	String		

12	UpdatingMode	Kiểu cập nhật các StartURL của project	Integer		21 giá trị

Bảng 7.4 : Cấu trúc ProjectInfo

StartUrlVector là một vector mà mỗi phần tử là một biến cấu trúc kiểu StartUrlInfo.

Tương tự FileRetrievalVector cũng là một vector mà mỗi phần tử là một biến cấu trúc kiểu FileRetrieval.

2. Xử lý của web robot

Thiết kế module phải thoả các yêu cầu sau :

- Đảm bảo vai trò của web robot trong hệ thống : tìm kiếm liên kết, thu thập thông tin, tạo và duy trì cơ sở dữ liệu.
- Không ảnh hưởng đến hoạt động của các module khác.

Nhìn chung quy trình hoạt động của web robot đều giống nhau. Tuy nhiên, các ứng dụng hỗ trợ những tính năng khác nhau sẽ có sự thay đổi về quy trình hoạt động.

Các bước xử lý :

(1) Khởi tạo.

(2) Nếu vẫn còn URL chưa xử lý và user không chọn chức năng tạm dừng (pause)

Qua (3)

ngược lại qua (14)

(3) Lấy URL đầu tiên.

Nếu lấy được qua (4)

ngược lại quay lại (2)

(4) Lưu thông tin cũ

(5) Kết nối vớI server

Nếu kết nối được qua (6)

ngược lại

Thêm URL vào danh sách hỏng

Cập nhật trạng thái của URL trong CSDL = BROKEN

Đánh dấu URL đã xử lý trong CSDL.

Nếu truy xuất CSDL không được

Đưa URL này trở lại hàng đợi.

Quay lại (2)

(6) Huỷ URL ra khỏi danh sách hỏng nếu URL nằm trong danh sách đó.

(7) So sánh với thông tin cũ

Nếu giống qua (8)

ngược lại

Cập nhật thông tin mới

(8) Đánh dấu URL tốt (trạng thái = GOOD)

(9) Thêm URL vào danh sách đã xử lý.

(10) Đây là file HTML ?

Nếu cần phân tích lại thì tiến hành phân tích.

(11) Thoả yêu cầu download của quản trị ?

Nếu thoả

Download

Trả kết quả về để hiện thị thông tin đã xử lý

Qua (12)

ngược lại

Quay lại (2)

(12) Ghi nhận thông tin mới xuống CSDL gồm :

Cập nhật thông tin mới cho URL

Đánh dấu URL đã được xử lý.

Nếu truy xuất CSDL không được

Gán lại thông tin cũ cho URL

Xóa URL khỏi danh sách đã xử lý.

Thêm URL vào lại hàng đợi

Quay lại (2)

(13) Đủ số spider chưa ?

Nếu chưa

Tạo thêm

Tạm dừng

Quay lại (2)

(14) Đã hết URL chưa ?

Nếu chưa (ứng dụng dừng do user chọn pause)

Qua (15)

ngược lại

Đánh dấu StartURL đã thực hiên xong (processStatus = DONE)

(15) Kết thúc.

Giải thích thêm về thuật toán :

- Khi phân tích file HTML, ta dò tìm các liên kết và những thông tin cần thiết để cập nhật bảng URL trong CSDL.

Những liên kết tìm được muốn vào hàng đợi trước tiên nó phải là URL chưa được xử lý lần nào và không có trong danh sách đang đợi xử lý, sau đó phải qua tiền xử lý.

Thuật toán chỉ xem xét danh sách đợi và danh sách đã thực hiện nhưng không xét danh sách bị hỏng nhằm tạo điều kiện để sửa chữa URL hỏng nếu có 1 URL khác liên kết tới nó. Trong trường hợp không có URL nào liên kết tới nó, quản trị vẫn biết nó bị hỏng do trạng thái này đã được ghi nhận trước đó.

- Số spider tạo thêm = min (số liên kết hiện có, số spider theo yêu cầu). Ta luôn có lượng spider vừa đủ dùng, nhờ vậy mà tránh lãng phí tài

nguyên hệ thống do mỗi spider cần khá nhiều tài nguyên cho hoạt động của mình.

Các bước tiền xử lý 1 URL tìm được :

(1) Tuỳ theo yêu cầu của quản trị mà ta kiểm tra URL mới, ví dụ : cùng thư mục gốc, cùng site,…

Nếu thoả yêu cầu

Qua (2)

ngược lại

Trả về thông tin cho biết không tiếp tục xét URL này.

(2) Kiểm tra độ sâu liên kết

Nếu chưa vượt quá giới hạn

Qua (3)

ngược lại

Trả về thông tin cho biết không tiếp tục xét URL này.

(3) Kiểm tra dạng file (content type)của URL có nằm trong danh sách các dạng file cần download hay không ?

Nếu có

Trả về thông tin cho biết tiếp tục xét URL này.

ngược lại

Trả về thông tin cho biết không tiếp tục xét URL này.

3. Giải quyết các vấn đề của web robot

3.1 Tránh sự lặp lại

Trong qúa trình hoạt động của robot ứng dụng dùng 3 danh sách :

- Hàng đợi URL : chứa các URL chưa xử lý theo nguyên tắc FIFO.
- Danh sách các URL kết nốI được với server.
- Danh sách các URL không kết nối được với server.

Nhờ lưu lại dấu vết của mỗi spider nên ứng dụng sẽ không xử lý một liên kết nhiều lần nhưng chưa khắc phục được các liên kết tồn tại dưới nhiều tên khác nhau (DSN, IP, …)

3.2 Tránh làm qúa tải server

Các spider hoạt động đồng hành nhưng bắt đầu từ những địa chỉ khác nhau. Kết hợp sự đồng bộ spider và duy trì thời gian đợi giữa 2 lần truy xuất liên tiếp đến một server nên server tránh bị áp lực quá mức. Tuy nhiên biện pháp này không thể khắc phục triệt để vấn đề do các URL cùng site thường được đặt cạnh nhau trong hàng đợi. Áp dụng chiến lươc duyệt ngẫu nhiên sẽ cho kết quả tốt hơn.

3.3 Tránh truy xuất đến các dạng tài nguyên không thích hợp

Ứng dụng chỉ lập chỉ mục trên những file có thể đánh được chỉ mục, cụ thể là dạng file text, sẽ download tài liệu nếu cần. Trong quá trình download chỉ lấy về các file thoả yêu cầu do đó tránh lãng phí tài nguyên cho những tài liệu không dùng đến.

3.4 Tránh các lỗ đen(black holes)

Ứng dụng chỉ theo dấu các URL còn trong giới hạn độ sâu cho phép nên luôn đảm bảo có điểm dừng.

3.5 Tránh những nơi cấm robot

Như đã trình bày trong những phần trước, các chuẩn loại trừ robot không hiệu quả do bị lạm dụng hoặc do thiếu tính chặt chẽ nên hầu hết các site trên thế giới đều không hỗ trợ chuẩn này vì vậy vấn đề xem như được thông qua.

4. Các thuật toán phân tích cấu trúc file HTML

4.1 Thuật toán lấy liên kết

Để tạo một liên kết trong file HTML người ta thường dùng một trong các dạng sau :

Tên thẻ	Thuộc tính kết hợp
A	Href
AREA	Href
BASE	Href
BODY	Background

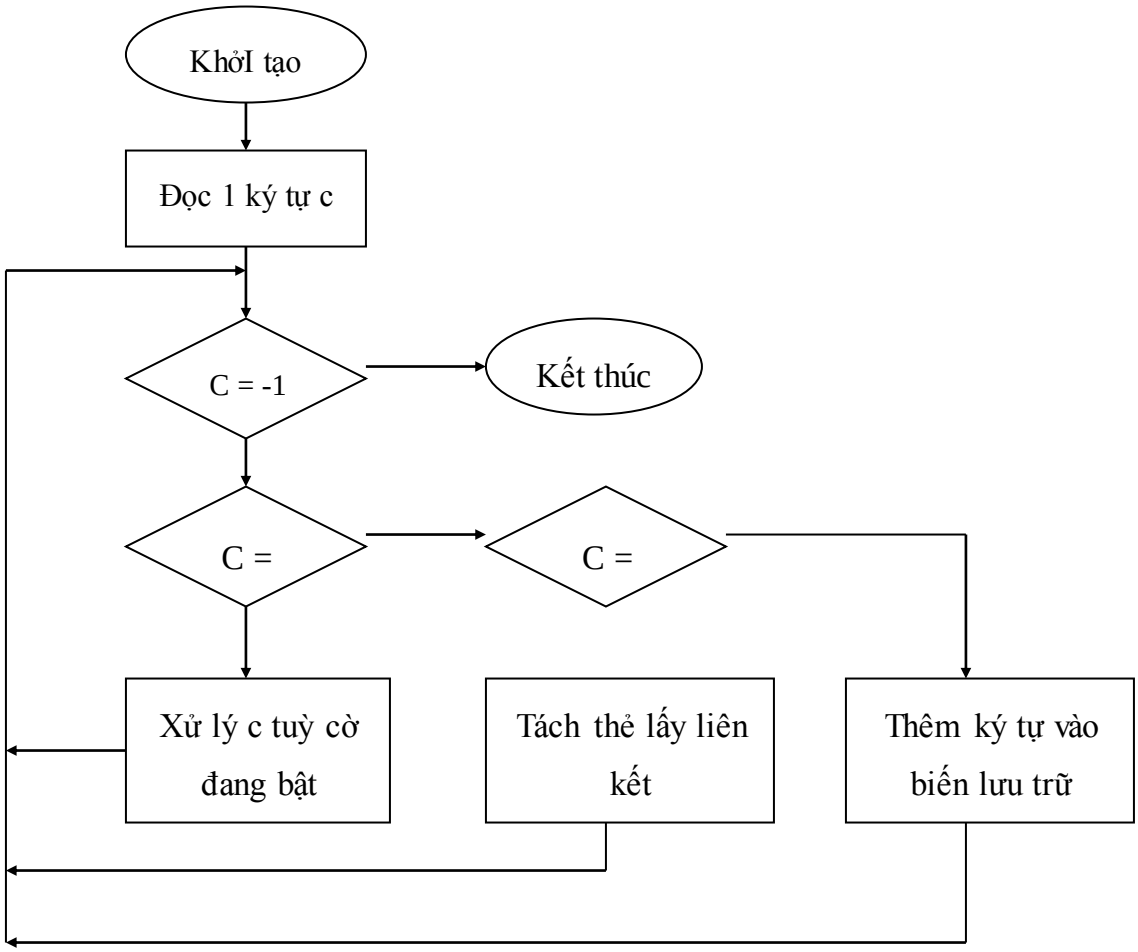
IMG	Src
INPUT TYPE	Src
FRAME	Src
FORM ACTION	
LINK	Href
TD	Bacground
SCRIPT	Src

Bảng 7.5 : Danh sách các thẻ thường dùng tạo tạo liên kết

4.1.1 Thuật toán ứng dụng cũ đã cài đặt

Thuật toán cờ trạng thái

- Ý tưởng : duyệt qua từng ký tự, bật cờ tương ứng khi gặp ký tự đặc biệt hoặc các thẻ chứa liên kết.
- Lưu đồ thuật toán :



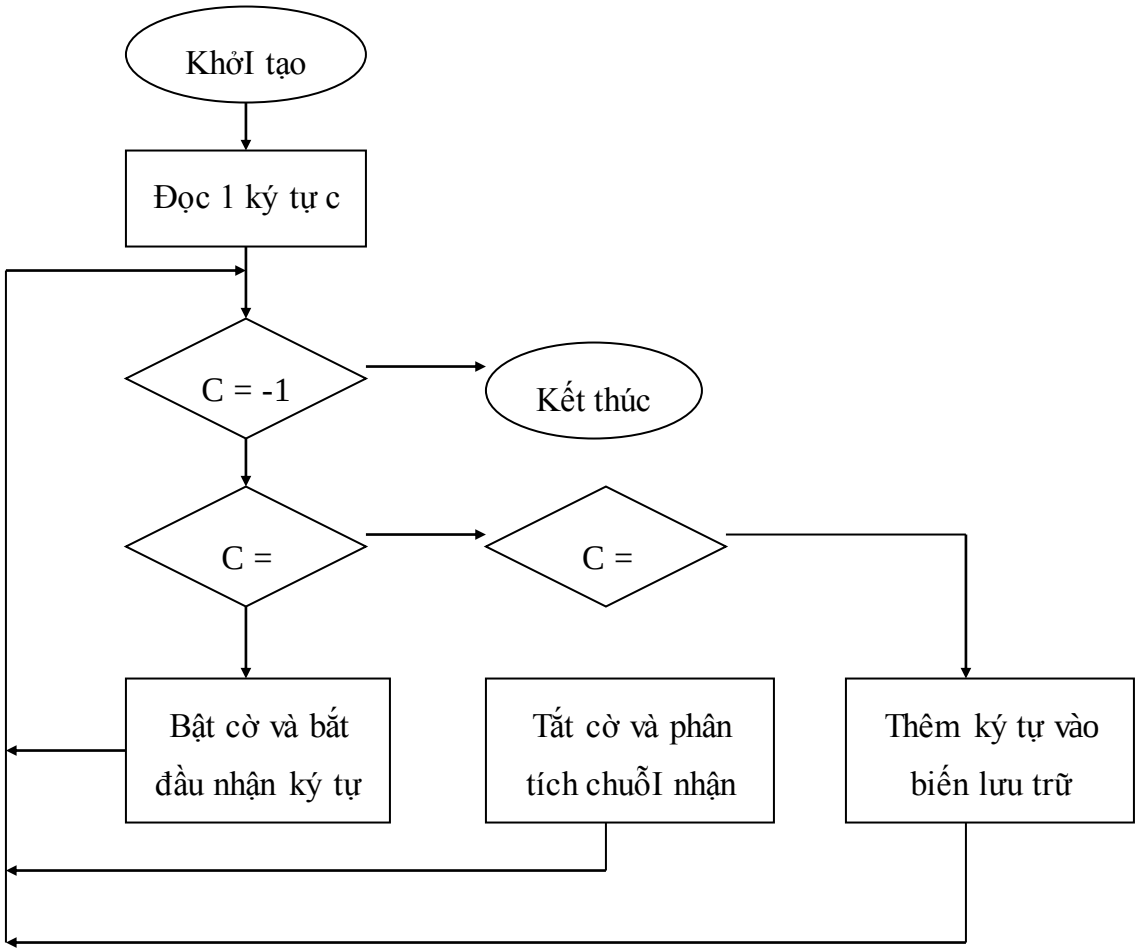
Hình 7.1 Lưu đồ thuật toán cờ trạng thái

- Ưu điểm : lấy chính xác các liên kết theo đúng chuẩn HTML.
- Khuyết điểm : không lấy được liên kết nhúng trong các đoạn script.

Thuật toán dựa vào đuôi file

- Ý tưởng : các thẻ trong file HTML đều bắt đầu bằng ký tự '<', kết thúc bằng ký tự '>' nên ứng dụng lấy nộI dung giữa cặp dấu này. Duyệt qua từng phần tử trong danh dách đuôi file ban đầu, nhận liên kết nếu nó có mặt trong danh sách đã cho.

- Lưu đồ thuật toán :



Hình 7.2 Lưu đồ thuật toán dựa vào đuôi file

- Các bước phân tích như sau :

VớI mỗi đuôi file

(1) Tìm vị trí đuôi file

(2) Xác định biên phải, trái dựa vào các ký tự giớI hạn ‘ ‘, #, =, \n, \t, \r, ….

(3) Lấy liên kết giữa 2 biên, nếu có.

- Ưu điểm : khắc phục nhược điểm cách 1

- Khuyết điểm : phải có danh sách đuôi file ban đầu.

4.1.2 Chọn lựa của ứng dụng mới

Ứng dụng cũ đã chọn thuật toán 2 nên vẫn mắc phải nhược điểm nêu trên. Ứng dụng mới không có sự cải tiến gì đối với thuật toán phân tích lấy liên kết, chỉ khắc phục nhược điểm này bằng cách :

- Kết hợp 2 thuật toán : nếu không có danh sách đuôi file ban đầu ứng dụng sẽ thi hành thuật toán 1.
- Hỗ trợ thêm chức năng user defined : khi phát hiện các dạng file mới, ta có thể bổ sung thông qua chức năng này. Sau đó có thể thi hành thuật toán 2 để giới hạn phạm vi thu thập thông tin của robot.

4.2 Thuật toán lấy tiêu đề

- Áp dụng thuật toán cờ trạng thái.
- Xét ví dụ :

<html>

<title = “Trang chủ”> </title>

<body> Chào mừng bạn đến với trang web của chúng tôi </body>

</html>

Ta lần lượt bật các cờ như sau :

ST_GROUND (cờ bắt đầu)

ST_LT

ST_T

ST_TI

ST_TIT

ST_TITL

ST_TITLE

ST_TITLE_EQUALS → lấy tiêu đề

4.3 Thuật toán lấy nội dung

Từ ví dụ trên ta nhận thấy phạm vi ảnh hưởng của một thẻ nằm trong cặp dấu '< >' do đó để lấy nội dung ta sẽ rút trích phần nằm giữa cặp dấu '><'. Sau khi lấy nội dung ta tiến hành loại bỏ các ký tự đặc biệt như rồi lưu xuống CSDL.

Các bước thực hiện

(1) Khởi tạo

(2) Biên trái = vị trí ký tự '>'

Nếu biên trái > -1 (chưa hết file) thì qua bước (3)

ngược lại qua (5)

(3) Biên phải = vị trí ký tự '<'

Nếu biên phải > -1 qua (4)

ngược lại qua (5)

(4) Trích chuỗi giữa 2 biên.

Quay lại (2)

(5) Lọc ký tự đặc biệt

Lưu vào CSDL

5. Duy trì thông tin cho CSDL

Mục đích của việc duy trì thông tin cho CSDL

- Đảm bảo thông tin trong CSDL là những thông tin mới nhất.
- Phát hiện các URL hỏng mới để có biện pháp xử lý.
- Sửa chữa các URL hỏng.

Thuật toán duy trì thông tin cho CSDL là một phần trong các bước xử lý của web robot, xin xem phần trước.

6. Resume project

Mục đích :

- Tối thiểu hoá lượng công việc mà robot phải thực hiện lại
- Linh động hơn trong quá trình xử lý project, ví dụ : ưu tiên xử lý project quan trọng hơn, tạm dừng project vì một lý do nào đó,…

Project bị dừng lại do 2 nguyên nhân chính :

- Sự cố hệ thống
- Người quản trị chủ động dừng

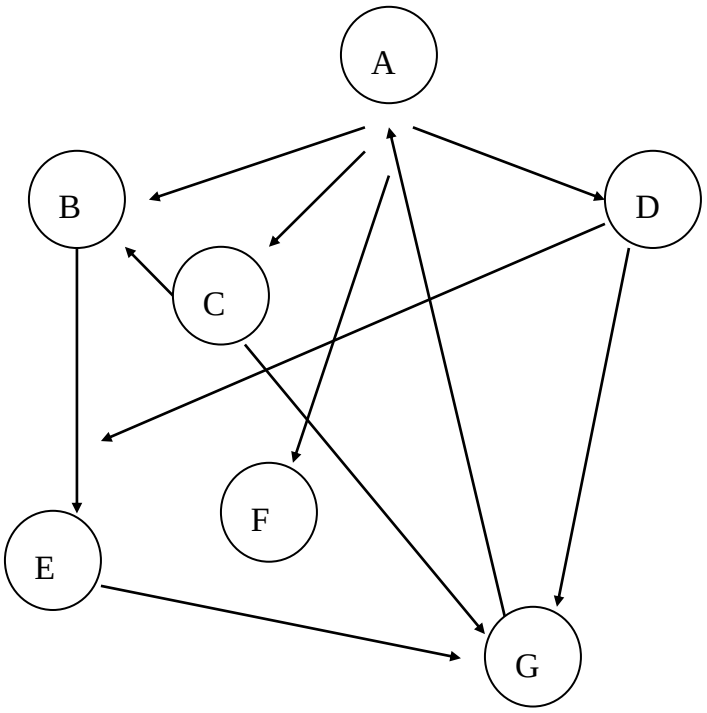
6.1 Nguyên tắc resume của ứng dụng cũ[11]

Khi project được kích hoạt lại, nếu project trước & sau kích hoạt giống nhau thì mọi tài nguyên đã cấp cho nó vẫn còn do đó ứng dụng chỉ cần tạo lại các spider để tiếp tục công việc. Nhưng nếu là project khác thì lúc khởi động lại cần phục hồi trạng thái của project trước điểm dừng. Ứng dụng sử dụng danh sách dự phòng với số phần tử bằng số spider. Khi lấy 1 URL ra khỏi hàng đợi, đầu tiên nó đưa vào danh sách dự phòng sau đó mới tiến hành xử lý. Nếu danh sách đầy, phần tử đầu sẽ bị loại bỏ do đó luôn đảm bảo lưu lại URL mới nhất. Mỗi chu kỳ t giây, thông tin được lưu xuống đĩa để khi cần có thể dùng nó phục hồi hàng đợi.

- Ưu điểm : đảm bảo mục đích resume.
- Khuyết điểm :
  - ✓ Bỏ sót URL.
  - ✓ Xử lý cùng 1 URL nhiều hơn 1 lần.

Sau đây là ví dụ minh hoạ nhược điểm của thuật toán phân tích liên kết dựa vào đuôi file. Xét ví dụ : giả sử ta có cây liên kết như sau

[1] Ứng dụng cũ là luận văn tốt nghiệp năm 2003" Xây dựng công cụ hỗ trợ quá trình tiền xử lý cho hệ thống Search Engine" – SVTH: Đoàn Hữu Quang Vinh .



Hình 7.3 Cây liên kết

Dùng thuật toán duyệt theo chiều sâu & số spider = 3

Hàng đợi : E, G

Đã xử lý : A, B

Đang xử lý : C, F, D

Sự cố xảy ra……

Khi hệ thống khởi động lại, hàng đợi sẽ có : C, F, D

→ mất 2 trang E, G

→ xử lý lại A, B

Project càng có nhiều URL, khuyết điểm này càng phải được khắc phục.

6.2 Cải tiến của ứng dụng mới

Ứng dụng mới cho phép project có nhiều URL ban đầu (StartURL) do đó khi resume là bắt đầu lại 1 StartURL chứ không phải 1 project.

Các bước phục hồi như sau :

(1) Phục hồi danh sách hàng đợi, danh sách đã xử lý, danh sách liên kết đã xử lý nhưng bị hỏng (kết nối với server bị thất bại).

(2) Lấy 1 URL cần xử lý.

Đánh dấu nó trong CSDL.

(3) Tiến hành xử lý

Nếu quá trình xử lý trọn vẹn → xoá đánh dấu.

Quay lại (2)

- Ưu điểm : tránh được nhược điểm của ứng dụng cũ.
- Khuyết điểm : phải tốn thêm một field để đánh dấu trong CSDL. Tuy nhiên trong môi trường mạng dạng liên kết như ví dụ trên rất nhiều cho nên sử dụng thêm field này là cần thiết.

Tóm tắt so sánh những chức năng chính giữa ứng dụng cũ và mới

Chức năng	Ứng dụng cũ	Ứng dụng mới
Thuật toán lấy liên	- Dùng thuật toán dựa vào đuôi	- Dùng thuật toán cờ trạng

kết trong file HTML	file. - Lấy các liên kết cùng thư mục vớI liên kết ban đầu (internal link)	thái. - Dùng thuật toán dựa vào đuôi file. - Lấy các liên kết cùng thư mục, cùng site & khác site vớI URL ban đầu. - Hỗ trợ thêm chức năng user defined.
Số StartURL của mỗI project	MỗI project chỉ có 1 StartURL	MỗI project có nhiều StartURL.
Download	Giới hạn kích thước cho mọI kiểu file giống nhau.	Các kiểu file khác nhau có thể có kích thước khác nhau.
Cập nhật project	Cập nhật lại toàn bộ các liên kết trong file HTML của URL ban đầu.	Hỗ trợ nhiều tuỳ chọn.
Resume	- Bỏ xót URL. - Xử lý trùng lặp.	- Không sót. - Không trùng lắp.
Lập lịch	Hỗ trợ lập lịch tự động.	Không hỗ trợ lập lịch.

Bảng 7.6: Bảng tóm tắt so sánh những chức năng chính giữa ứng dụng cũ và mới

Chương 8: LẬP CHỈ MỤC

1. Tính trọng số của từ:

Sau khi tách từ là giai đoạn tính trọng số các từ để xác định mục từ có nghĩa đại diện cho nội dung tài liệu. Như đã trình bày trong phần I, có rất nhiều cách tính trọng số của mục từ. Ở đây, ta chọn công thức:

$$W=n_{ik}/n_k$$

Trong đó:

n_{ik}: số lần xuất hiện của mục từ k trong tài liệu i

n_k : số lần xuất hiện của mục từ k trong tất cả các tài liệu được lập chỉ mục

Ngưỡng được sử dụng để loại bỏ các mục có trọng số thấp là ½ giá trị trọng số trung bình của các mục từ xuất hiện trong toàn bộ tài liệu.

Tính title

Do nội dung bên trong title có ý nghĩa quan trọng , nên cách tính trọng số của mục từ xuất hiện trong title đặc biệt hơn trong nội dung văn bản

Có các cách giải quyết như sau :

- Lấy trọng số những mục từ có trong title = trọng số lớn nhất của các từ trong nội dung được lập chỉ mục
- Trọng số gấp 3 lần trọng số bình thường

- Lập chỉ mục thẳng cho từ có trong title .

2. Tập tin nghịch đảo :

Giả sử câu truy vấn của người sử dụng sau khi lập chỉ mục là một tập các mục từ { t1, t2, ..,tn }. Ví dụ : truy vấn "công nghệ phần mêm " sẽ được lập chỉ mục gồm hai từ "công nghệ" và "phần mềm") với giá trị n thường không lớn (2,3,4..)

Yêu cầu của người sử dụng là mong muốn tìm kiếm các tài liệu có chứa tất cả các mục từ t1, t2,..., tn. Như thế ta không cần khảo sát tất cả các vector chỉ mục mà chỉ cần tìm các vector nào có chứa t1, t2, ... , tn.Điều này có thể thực hiện dễ dàng bằng cách lưu các nhóm vector (tài liệu) theo từng mục từ.

t1 : 1, 3, 4

t2 : 1, 2, 4, 5

t3 : 2, 4, 5

Nghĩa là mục từ t1 có trong các tài liệu 1, 3, 4.

t2 có trong các tài liệu 1,2,4,5

t3 có trong các tài liệu 2, 4, 5

Khi đó quá trình tìm kiếm (t1, t3) sẽ được thực hiện theo các bước sau:

1. Tìm tập các tài liệu có chứa t1 , gọi là T1={1,3,4}
2. Tìm tập tài liệu có chứa t3, gọi là T2={2,4,5}
3. Tập các tài liệu có chứa cả t1 và t3 là T=T1∩ T2={4}
4. Tính toán độ tương tự giữa câu truy vấn và các tài liệu có trong tập T

Sử dụng công thức tính độ tương tự :

$$Sim(D, Q) = v_i * w_i \ , i=1..n$$

với t_i là mục từ có trong Q (do $w_i=0$ với mục từ t_i không có trong Q và $w_i=1$ nếu ti có trong Q)

Rõ ràng việc tính độ tương tự chỉ cần tới trọng lượng của các mục từ có trong Q nên để có thể tăng thêm hiệu quả ta sẽ **lưu thêm giá trị trọng lượng** của mục từ trong tập tin nghịch đảo.

t1 : (1, 0.5) (3,0.7) (4,0.2)

t2 : (1,0.4) (2,0.8) (4,0.9) (5, 0.1)

t3 : (2,0.3) (4,0.2) (5,0.5)

Nghĩa là mục từ t1 có trong tài liệu 1 với trọng lượng là 0.5, trong tài liệu 3 với trọng lượng là 0.7 v...v...

Khi đó để tìm kiếm cho câu truy vấn (t1, t3) chỉ cần đọc 2 khối dữ liệu của t1 và t3 là đủ (giảm truy xuất đĩa và giảm thời gian xử lý).

Mô hình tập tin nghịch đảo hiện nay được sử dụng rất rộng rãi trong các hệ thống tìm kiếm thông tin vì với cách tổ chức này vì các dữ liệu cần đọc được lưu trữ liên tục nên giảm việc di chuyển đầu đọc của đĩa cứng, cũng như nếu ta lưu lại vị trí bắt đầu của các mục từ thì có thể truy xuất trực tiếp đến vị trí đó để đọc dữ liệu.

Khó khăn: của việc sử dụng tập tin nghịch đảo là khi cần thêm một tài liệu vào mục từ, giả sử cần thêm tài liệu 6 vào mục từ t1.

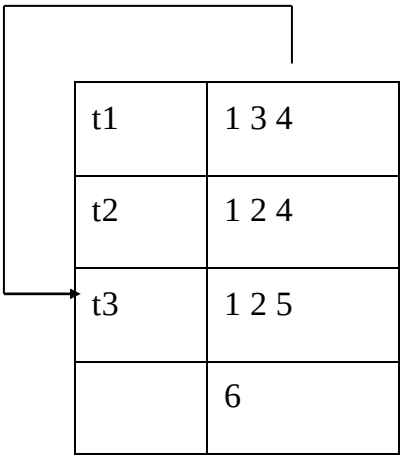
t1 : 1,3,4,6

t2 : 1,2,4,5

t3 : 2,4,5

Với chú ý rằng các khối dữ liệu của t1, t2, t3 được lưu trữ liên tiếp nhau trên đĩa cứng và dung lượng của tập tin nghịch đảo này rất lớn (chứa hàng trăm ngàn mục từ với hàng triệu tài liệu), hơn nữa việc thêm tài liệu này rất thường xuyên (lập chỉ mục cho các Web site mới , cập nhật lại các Web site có thay đổi) cho nên không thể sử dụng phương pháp chèn bằng cách dời dữ liệu ra sau để tạo khoảng trống chèn tài liệu 6 vào.

Cách giải quyết: cấp phát không gian cho các mục từ **theo trang**, khi một mục từ đã chứa hết trang này thì sẽ cấp phát thêm vào cuối tập tin và có một link chỉ đến trang cuối này.



Phương pháp này mặc dù lãng phí không gian cho các trang chưa dùng đến, giả sử có 100.000 mục từ, trang dung lượng là 1K, **dung lượng đĩa lãng phí** lớn nhất là 100.000 K (100 M) **và phải di chuyển đầu đọc** nhiều nhưng **giải quyết được vấn đề thêm tài liệu** cũng như dễ dàng đọc được dữ liệu cần thiết cho một mục từ nào đó (đọc theo các link). Có thể điều chỉnh giữa dung lượng lãng phí và việc phải di chuyển đầu đọc (tính bằng số trang cấp phát cho một mục từ) bằng cách tăng hoặc giảm dung

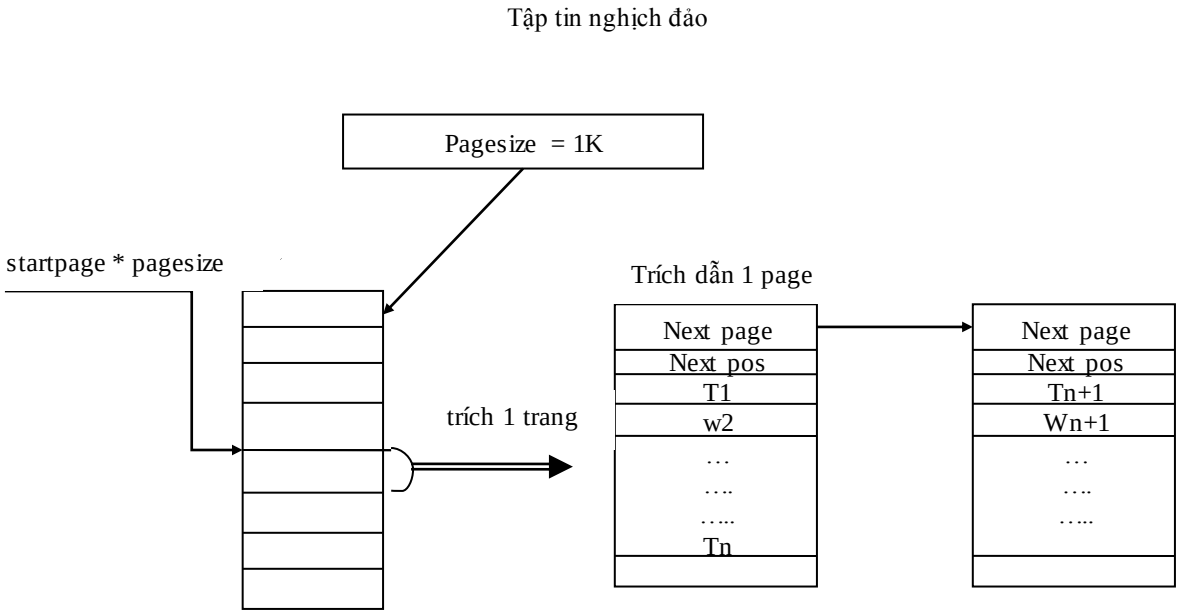
lượng cấp phát cho một trang. Nếu tăng dung lượng cấp phát cho một trang thì sẽ giảm việc di chuyển đầu đọc và ngược lại.

Hệ thống đã sử dụng mô hình tập tin nghịch đảo với việc cấp phát theo trang như đã trình bày trên , **dung lượng trang được chọn là 1K.**

Tập tin nghịch đảo lưu trữ danh sách các tài liệu ứng với từng mục từ để cho phép hệ thống nhanh chóng có được danh sách các tài liệu có chứa một mục từ nào đó có dạng sau:

Mục từ	Tài liệu, trọng lượng
t1	(2,w1), (3,w2),(4,w3)
t2	(3,w4),(4,w5),(5,w6)
t3	(2,w7),(4,w8)
t4	(1,w9)

Bảng trên có nghĩa là mục từ t1 có các tài liệu 2,3,4 với trọng lượng tương ứng là w1,w2,w3.



Hình 8.1 Tập tin nghịch đảo

Một mục từ có thể có nhiều trang. Do kích thước của page là cố định pagesize = 1024B ~ 1K & chứa tối đa 1024/8 - 1 = 127 tài liệu trên 1 trang, 8 = 4byte luu docID , 4 byte luu trọng số cho nên tạo 1 chuỗi các trang chứa mục từ, 8 byte đầu của trang lưu vị trí trang tiếp theo(nếu có) và vị trí trống tiếp theo trong trang .

Vị trí	Chiều dài	Tên trường	ý nghĩa
0	4	NextPage	Vị trí trống tiếp theo chưa được sử dụng trong trang này, chỉ có ý nghĩa khi đây là trang cuối

4	4	NextPos	Trang tiếp theo (nếu có) của mục từ sở hữu trang này
8	4	DocID1	$DocID_i$: định danh tài liệu có chứa mục từ sở hữu trang này $Weight_i$: trọng số của mục từ trong từng tài liệu tương ứng $DocID_i$
12	4	Weight1	
16	4	DocID2	
20	4	Weight2	
24	4	DocID3	
28	4	Weight3	
……	……..	……..	
1016	4	DocID127	
1020	4	Weight127	

Bảng 8.1: Cấu trúc của một trang cấp cho từng mục từ trong tập tin nghịch đảo

Như vậy, có thể đọc toàn bộ danh sách các tài liệu có chứa một mục từ bằng cách đọc toàn bộ các trang được liên kết theo con trỏ NextPage. Vị trí đầu tiên chứa trang thuộc quyền sở hữu của mục từ đó được xác định như sau:

Vị trí đầu tiên = startpage*kích thước 1 page (ở đây là 1024 byte)

Các thao tác chính trong tập tin nghịch đảo gồm :

- **Thêm một tài liệu vào một mục từ**: khi một tài liệu được lập chỉ mục, nếu tài liệu này có chứa một mục từ t nào đó thì tài liệu này được thêm vào danh

sách các tài liệu ứng với mục từ t trong tập tin nghịch đảo. Tài liệu được thêm vào vị trí trống đầu tiên trong trang cuối của mục từ t.

- **Đọc danh sách các tài liệu của một mục từ**: kết quả của thao tác này được trả về theo luồng (stream) dưới dạng ($docID_1$, $weight_1$, $docID_2$, $weight_2$, ... , $docID_n$, $weight_n$) nghĩa là có thể đọc kết quả trả về theo từng tài liệu , xử lý xong tài liệu này mới đọc tài liệu tiếp theo.

Sau khi lấy được luồng danh sách các tài liệu của từng mục từ , nó lựa xem các danh sách đạt yêu cầu (chứa tất cả các mục từ yêu cầu).

Việc xử lý dữ liệu theo luồng là một **ưu điểm** lớn của hệ thống này vì giải quyết được vấn đề bộ nhớ hạn chế khi phải xử lý trên khối lượng dữ liệu lớn. Điều này cũng cho thấy hệ thống này vẫn có thể đáp ứng được khi tăng khối lượng tài liệu phải xử lý hoặc tăng số yêu cầu phải xử lý đồng thời.

File nghịch đảo được truy cập thường xuyên khi xử lý yêu cầu tìm kiếm và khi lập chỉ mục. Do đó, thao tác đọc và cập nhật file nghịch đảo chiếm nhiều thời gian nhất trong tổng số thời gian cần thiết để hoàn tất một yêu cầu tìm kiếm. Vì dung lượng file nghịch đảo thay đổi và có thể trở nên quá lớn khi số tài liệu được lập chỉ mục tăng lên nên không thể lưu toàn bộ file nghịch đảo vào bộ nhớ do đó để tăng tốc độ tìm kiếm chúng tôi cấp phát **một vùng nhớ đóng vai trò bộ đệm** cho file này. Bộ đệm được chia thành các trang với dung lượng bằng dung lượng trang được cấp phát cho từng mục từ (1K). Khi có yêu cầu truy xuất một trang trong file nghịch đảo , trang cần sẽ được nạp lên bộ đệm nếu chưa có trong bộ đệm và tồn tại ở đó để có thể sử dụng cho những lần truy xuất sau (không phải đọc lại từ đĩa).

3. Từ điển chỉ mục

Từ điển chỉ mục chứa danh sách các mục từ. Từ điển chỉ mục xây dựng sẵn gồm 1100.000 từ gồm cả tiếng Anh và tiếng Việt . Trong quá trình lập chỉ mục , từ mới nào chưa có sẽ được thêm vào tự điển . Do đó số lượng từ trong từ điển đã lên hơn 150.000 từ , từ tăng thêm chủ yếu là từ tiếng Anh

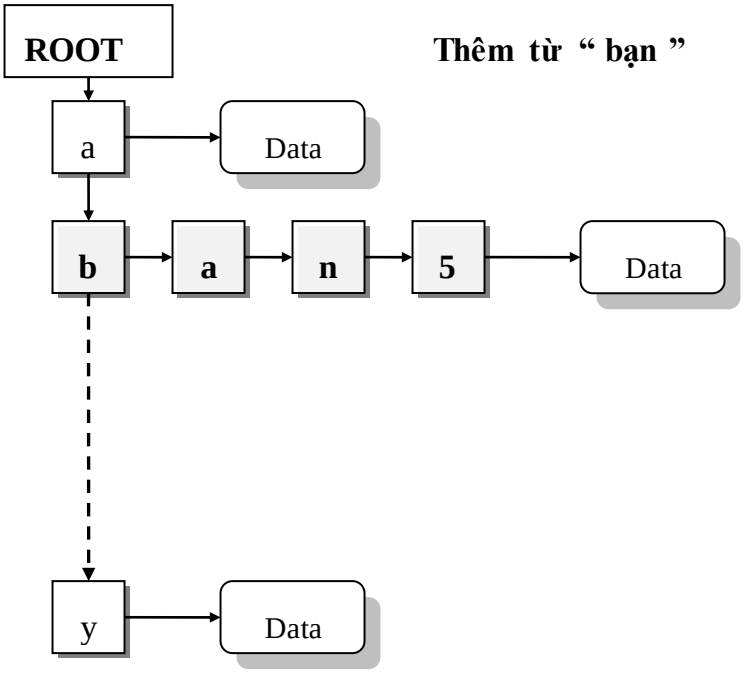
Số lượng mục từ trong từ điển chỉ mục lớn và thao tác tìm kiếm được thực hiện thường xuyên nên từ điển phải tổ chức sao cho việc tìm kiếm một mục từ được thực hiện nhanh chóng.

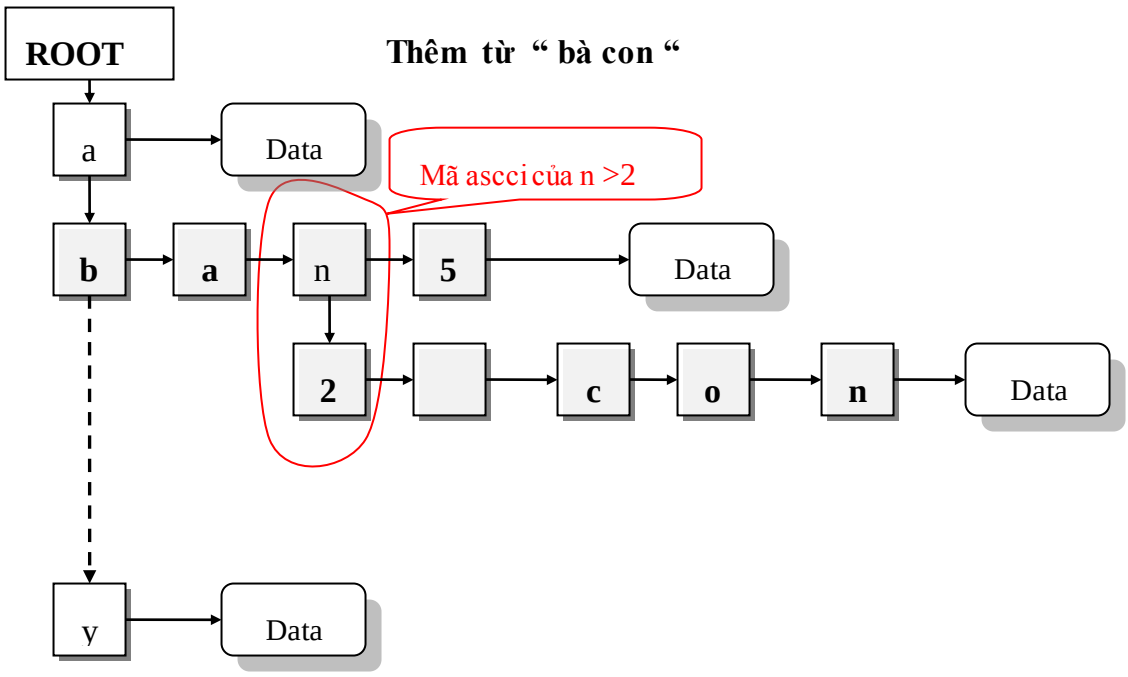
Chúng ta có thể tổ chức từ điển theo danh sách tuyến tính được sắp xếp của các mục từ và thực hiện giải thuật tìm kiếm **nhị phân** tuy nhiên gặp phải trở ngại là khi thêm một mục từ vào đòi hỏi phải **sắp xếp lại** từ điển, điều này gây khó khăn cho việc quản lí từ điển .

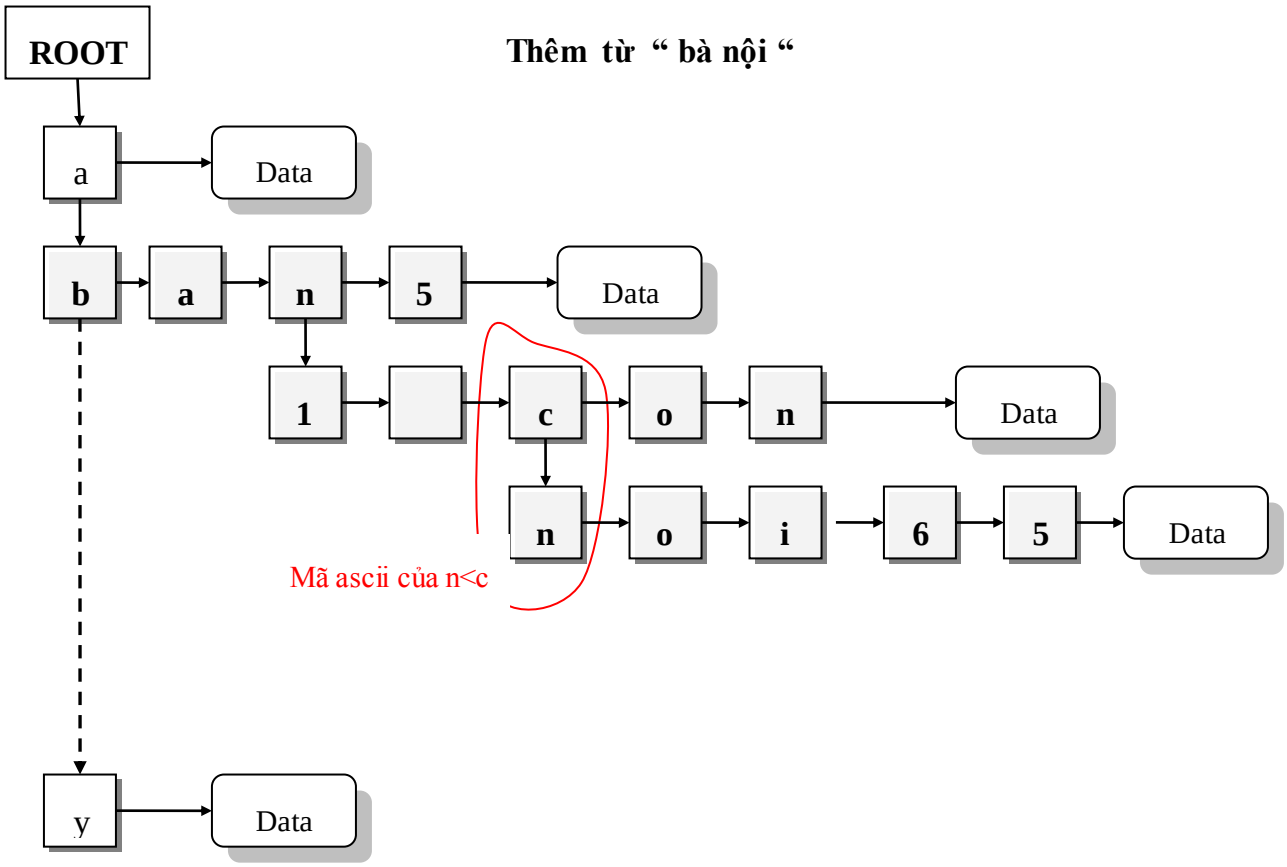
Hệ thống tổ chức từ điển dưới dạng cây **n-phân** biến thể thành **cây nhị phân để dễ dàng cho việc cài đặt**

Dưới đây là mô hình cây từ điển n-phân chứa các mục từ "bạn", "bà con", "bà nội":

Hình 8.2 Cây từ điển n-phân




ROOT
Thêm từ " bà con "
a
Data
Mã ascci của n >2
b
a
n
5
Data
2
c
o
n
Data
y
Data



Mỗi mục từ trong từ điển có một **cấu trúc dữ liệu Info** kèm theo, được gắn vào ký tự cuối cùng của mục từ. Cấu trúc Info gồm các trường sau:

```
Struct Info{
    int n;          //số lần xuất hiện của mục từ này trong danh sách các
                    trang Web mà hệ thống đã lập chỉ mục
    int nDoc;       //số tài liệu chứa mục từ này
```

```
int signal;        //xác định thuộc tính của mục từ này 0:tiếng Anh;
                   1: tiếng Việt; 2: stop-word
int startPage;     //trang bắt đầu trong danh sách các trang trong file chỉ
                   mục thuộc về mục từ này.
int endPage;       //trang kết thúc trong danh sách các trang trong file chỉ
                   mục thuộc về mục từ này.
}
```

Thuộc tính endPage được đưa vào nhằm làm tăng tốc độ lập chỉ mục. Với endPage, ta có thể truy xuất đến trang cuối cùng nhanh nhất khi cần thêm tài liệu vào file nghịch đảo, không cần phải duyệt tuần tự từ đầu danh sách các trang thuộc về mục từ đó.

Biến cờ signal có các giá trị như sau:

- Stopword : signal =1
- Từ mới : signal =2
- Tiếng Anh : signal = 3
- Tiếng Việt : signal = 4

Trong cấu trúc cây từ điển, dấu được chuyển về cuối để tiện cho việc tìm kiếm không dấu hoặc bỏ dấu không đúng kiểu, đồng thời giải quyết được tình trạng bỏ dấu khác biệt vị trí trong tiếng Việt. Ví dụ : Đối với từ Cộng Sản => Cong65 san3

Các thao tác chính trên tự điển chỉ mục gồm có:

- Thêm một mục từ
- Xoá một mục từ

- Cập nhật thông tin một mục từ
- Xem thông tin về một mục từ

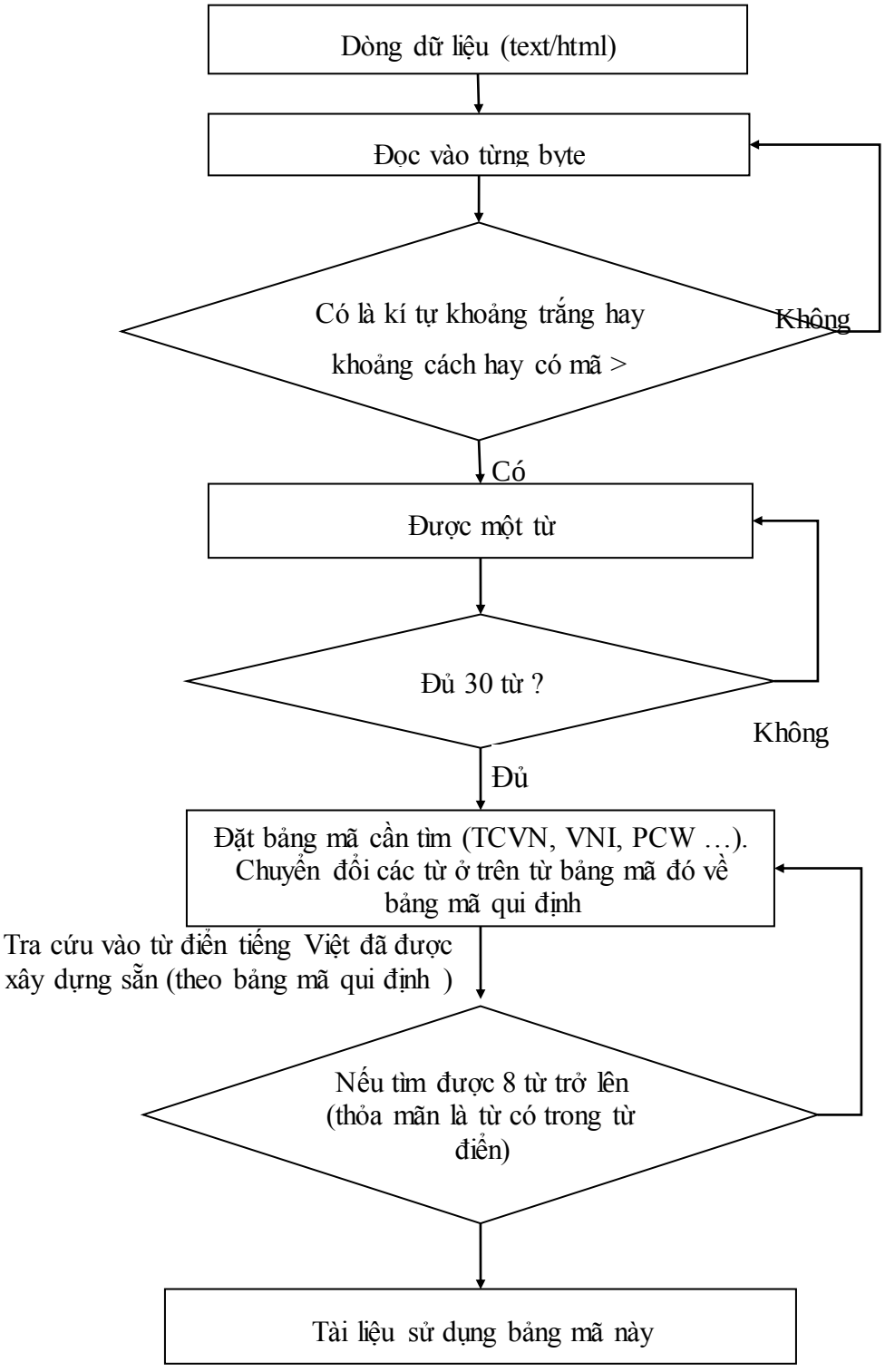
4. Quá trình stemming

Trong quá trình lập chỉ mục Tiếng Anh , Stemming là quá trình lượt bỏ các suffix (phần hậu tố / tiếp vĩ ngữ) của các từ . Việc nằm làm tăng giá trị recall của chương trình, làm cấu trúc cây từ điển chính xác và gọn nhẹ hơn , đương nhiên hiệu quả truy vấn cũng cao hơn .

Ví dụ : studies , studying , studied là các biến thể khác nhau của từ gốc study , nếu không có giai đọan stemming này thì tất cả các từ này đều được lập chỉ mục và bổ sung vào cây từ điển nếu nó chưa có . Rõ ràng điều này là khuyết điểm lớn của chương trình.

Có nhiều thuật tóan phổ biến cho việc loại bỏ phần đuôi của một từ tiếng Anh , thông thường đều dựa vào danh sách các hậu tố để đối chiếu .

Chương trình của chúng em lược bỏ các hậu tố sau : _s , _ing, _ed ...



Hình 8.3 Lưu đồ nhận dạng bảng mã

- Hệ thống ngoài xử lý được các bảng mã thông thường như TCVN3 , VNI , PCW , VIRQ còn xử lý được văn bản dùng bảng mã Unicode . Như chúng ta đã biết bảng mã unicode ngày nay trở thành chuẩn chung của mọi dạng bảng mã và hầu như được sử dụng hầu hết trong các trang web . Do đó xử lý được bảng mã Unicode là vấn đề hết sức quan trọng , là giá trị của chương trình.
- Unicode là 1 loại bảng mã rất đặc biệt , ta tìm hiểu sơ lược về loại mã này :

  Font Unicode có 2 dạng :

  . UTF8 (tổ hợp) : 1byte , 2 byte , 3 byte

  . UCS2 (dựng sẵn) : 2 byte , 4 byte – thông thường sử dụng 2 byte

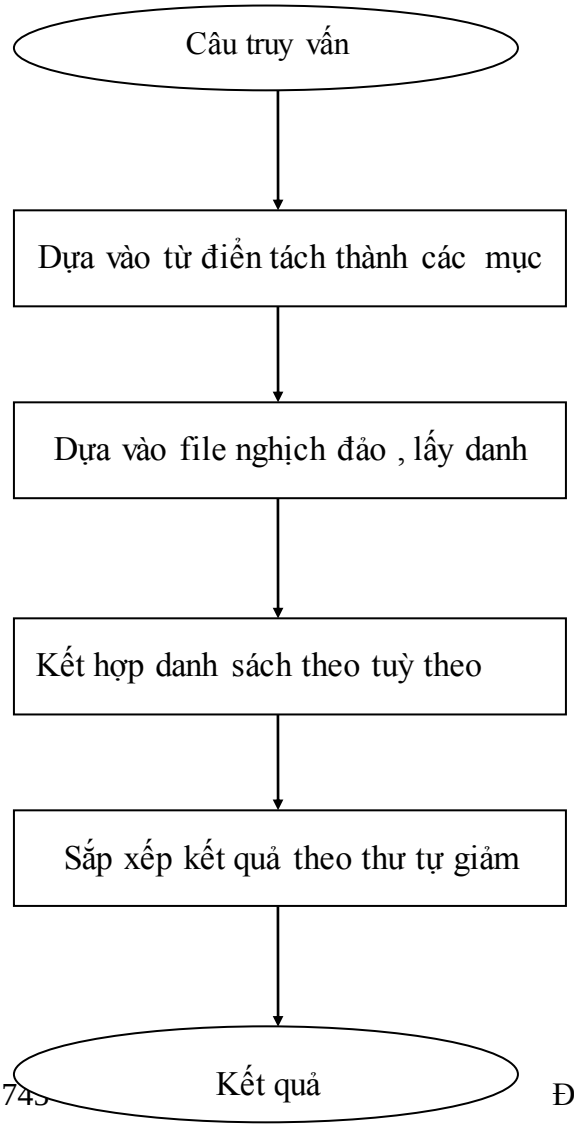
Do cấu trúc 2 dạng trên khác nhau nên cách xử lý khác nhau.

Chương 9: TÌM KIẾM THÔNG TIN

Hầu hết các Search engine hỗ trợ 2 tuỳ chọn là tìm cơ bản và nâng cao. Quy trình tìm kiếm cơ bản gần như giống nhau ở từng hệ thống. Đó là tiếp nhận câu hỏi, xử lý toán tử và trả về kết quả được mô tả qua lưu đồ dưới đây.

Nhằm mục đích minh hoạ, ứng dụng chỉ hỗ trợ :

- Các toán tử : AND (mặc định) , OR.
- Hệ thống dấu chấm câu : " "(tìm cụm từ)



Hình 9.1 Lưu đồ xử lý câu truy vấn

Khi muốn tìm thông tin, người dùng tương tác với hệ thống thông qua giao diện web. Bộ tìm kiếm thông tin sau khi tiếp nhận câu truy vấn sẽ dựa vào từ điển để tách câu hỏi thành các mục từ có nghĩa. Thuật toán tách mục từ là thuật toán được dùng trong bộ lập chỉ mục nhằm đảm bảo sự tương thích giữa tập mục từ của câu truy vấn và cơ sở dữ liệu chỉ mục.

Ứng dụng dựa vào file nghịch đảo lấy danh sách tài liệu tương ứng với từng mục từ. Một lần nữa lọc lại danh sách này tuỳ theo phép toán được chọn. Sắp xếp kết quả thu được và trả về cho người dùng.

Chương 10: CÁC MODULE ,PACKAGE, LỚP CHÍNH CỦA CHƯƠNG TRÌNH

1. Các module, package của chương trình

STT	Module	Ý nghĩa
1	DBController	**X**ử lý việc kết nối ODBC giữa database Oracle & Java
2	ProcessDoc	**X**ử lý các trang web đã được download về , như lập chỉ mục , đưa vào file nghịch đảo ..... Sau khi xử lý xong , lưu vào database các thông số cần thiết , xóa các file đó
3	Query	**Đ**ối tượng trung gian giữa module *ProcessDoc* và module *SE*. Nhận câu truy vấn từ *SE*, yêu cầu *ProcessDoc* phân tích câu này thành các từ có nghĩa. Khi *ProcessDoc* trả về tập các tài liệu có chứa những từ khoá cần tìm, *Query* sẽ hợp các tài liệu này lại tuỳ theo toán tử được sử dụng để trả về danh sách các URL có tổng trọng số các từ khoá giảm dần.
4	SE	**G**iao diện người dùng : tiếp nhận câu hỏi từ user, nhờ *Query* xử lý và hiển thị kết quả tìm được.
5	Webcopy	**X**ử lý việc lấy thông tin /download các trang web

		(text/html , asp , php...) từ internet của URL input, dò tìm các liên kết mới, xử lý thông tin thu được để tạo CSDL chỉ mục.
6	WebcopyGUI	**G**iao diện người quản trị : hỗ trợ người quản trị trong việc quản lý hoạt động của một webrobot.

2. Các lớp đối tượng chính trong từng module

2.1 Module DBController

STT	File	Ý nghĩa
1	ConnectionPool.java	**H**oạt động dựa trên lớp *ResourcePool.java***,** phụ trách việc đóng, mở, duy trì kết nối giữa ứng dụng và CSDL.
2	DatabaseObject.java	**L**ớp ảo tương tác với CSDL để thực hiện các thao tác như lấy thông tin, thêm, xoá, sửa,…
3	DriverUtilities.java	**C**ung cấp thông tin cần thiết về các driver khác nhau để kết nối CSDL.
4	ResourcePool.java	**D**uy trì, phân phát và giải phóng tài nguyên hệ thống.

2.2 Module ProcessDoc

STT	File	Ý nghĩa
1	DicTree.java	**Q**uản lý về cây từ điển
2	DocInfo.java	**K**hởi tạo thông tin ban đầu về tài liệu như n , signal , trọng lượng
3	DocObject.java	**Đ**ối tượng điều khiển việc lấy thông tin và đánh dấu các URL đã được lập chỉ mục trong CSDL.
4	DocTree.java	**X**ây dựng cây từ điển các mục từ cho tài liệu văn bản
5	HtmlInformation.java	**T**hông tin về file html :url , trích dẫn , tiêu đề , ngày cập nhật
6	HtmlStreamFilter.java	**B**ộ lọc các tag đặc biệt của file html
7	Info.java	**I**nterface của cây từ điển từ và cây từ điển tài liệu
8	InverseFile.java	**T**ập tin nghịch đảo
9	Manager.java	**T**rình quản lý chung cho chương trình như các tham số , các file input , trạng thái , từ điển chỉ mục .....
10	Node.java	**C**ác thao tác trên node của cay từ điển

11	NodeInfo.java	**K**hởi tạo ban đầu về thông tin về 1 node (n , nDoc , signal , startPage, endPage)
12	Paicedemo.java	**H**ỗ trợ cho việc xử lý hậu tố
13	ProcessDicTree.java	**X**ử lý thao tác theo cấu trúc cây từ điển
14	ProcessFileDownloaded.java	**X**ứ lý các file download về (lập chỉ mục)
15	ProcessWord.java	**C**ác thao tác xử lý trên từ như tính trọng số ....
16	SpecialChar.java	**X**ử lý các ký tự đặc biệt (dùng bảng băm) quá trình này bao gộp xử lý sơ khởi cho phone Unicode UCS2
17	Stemmer.java	**L**ọc bỏ hậu tố của từ tiếng Anh
18	Sentence.java	
19	Utils.java	**N**hận dạng bảng mã ...

2.3 Module Query

STT	File	Ý nghĩa
1	JoinStream.java	**H**ợp các tập tài liêu tuỳ theo toán tử được chọn.
2	Query.java	**S**ử dụng từ điển phân tích một câu truy vấn thành các từ có nghĩa, xử lý những từ này tuỳ theo chúng thuộc loại từ nào (tiếng Việt, tiếng

		Anh, stop word)
3	StreamInverseFile.java	**T**ập các tài liệu lấy được dựa vào file nghịch.

2.4 Module SE

STT	File	Ý nghĩa
1	SearchGUI.java	**T**ương tác với người dùng thông qua giao diện web với những xử lý như : kiểm tra các ràng buộc nhập liệu đối với một câu truy vấn, hiển thị kết quả tìm được,…
2	Search.java	**Đ**ối tượng trung gian giữa module *Query* và lớp *SearchGUI.java*

2.5 Module Webcopy

STT	File	Ý nghĩa
1	FlexVector.java	**G**iống như đối tượng *Vector* của Java nhưng có một số thay đổi nhằm giúp cho việc điều khiển các phần tử tốt hơn.
2	ProjectObj	**Đ**ối tượng trung gian điều khiển những xử lý giữa cấu trúc *ProjectInfo* và các module, package, đối tượng, cấu trúc khác.
3	Queue.java	**D**ựa trên đối tượng *FlexVector* để truy xuất các phần

		tử đầu hàng đợi.
4	StartUrlObject	**Đ**ối tượng trung gian điều khiển những xử lý giữa cấu trúc *StartUrlInfo* và các module, package, đối tượng hoặc cấu trúc khác.
5	Spider.java	**N**hận 1 URL cần xử lý từ đối tượng *WebRobot.java*. Các xử lý bao gồm : yêu cầu thông tin từ WebServer, cung cấp các liên kết mới cho WebRobot, phân tích tài liệu để tạo CSDL chỉ mục.
6	SysProjects.java	**Đ**ối tượng điều khiển danh sách project của ứng dụng
7	Timer.java	**Đ**ối tượng điều khiển việc lưu thông tin của project hiện hành một cách định kỳ.
8	WebRobot.java	**Đ**iều khiển hoạt động của các Spider.
9	UrlObject.java	**Đ**ối tượng trung gian điều khiển những xử lý giữa cấu trúc *UrlInfo* và các module, package, đối tượng hoặc cấu trúc khác.
10	Utils.java	**C**ung cấp một số tiện ích cho module *Webcopy*

2.6 Module WebcopyGUI

STT	File	Ý nghĩa
1	MainClass.java	**X**ử lý các chức năng có trong *MainFrame.java*

2	MainFrame.java	**G**iao diện người quản trị.
3	PropertyProjectDlg.java	**T**hông qua *PropertyProjectDlg*, quản trị cung cấp một số thông tin cần thiết về một project như : tên project là gì, chu kỳ tự động lưu project, bao nhiêu spider hoạt động đồng thời, bao nhiêu kết nối CSDL được dành sẵn cho project,….
4	StartingUrlDlg.java	**T**hông qua *StartingUrlDlg*, quản trị cung cấp một số thông tin cần thiết về một URL ban đầu như : địa chỉ, account, password của trang web sẽ truy xuất, lần theo URL này đến mấy cấp,…
5	TreeInfo.java	**C**ây project mà mỗi nút là một URL ban đầu.
6	TableInfo.java	**D**anh sách những liên mới có được từ việc phân tích các URL ban đầu.

Phần 3 : KẾT QUẢ, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả thử nghiệm

Hệ thống search engine thử nghiệm được cài đặt trên server có cấu hình máy Pentium IV, CPU 1.5 GHz, bộ nhớ RAM 256MB, đĩa cứng 120 GB. Từ điển xử lý khoảng 150000 từ bao gồm cả từ tiếng Việt và tiếng Anh. Hỗ trợ quản trị quản lý webrobot và bộ lập chỉ mục **thông qua ứng dụng (application)** của Jbuilder, phục vụ nhu cầu tìm kiếm thông tin của người dùng thông qua **giao diện web**. Dữ liệu đươc thu thập chủ yếu trên mạng cục bộ (localhost). Dữ liệu mẫu gồm 8272 tài liệu, khoảng 145MB, (290MB trên đĩa).

Các website được đưa vào thử nghiệm:

- ✓ Azit Nexin
- ✓ Codeguru
- ✓ Covan
- ✓ Tự học tiếng Anh
- ✓ Su tích
- ✓ Tam quốc bình giảng
- ✓ Thơ Việt Nam
- ✓ Thuyền trưởng Blad
- ✓ Truyện cười
- ✓ Truyện ngắn

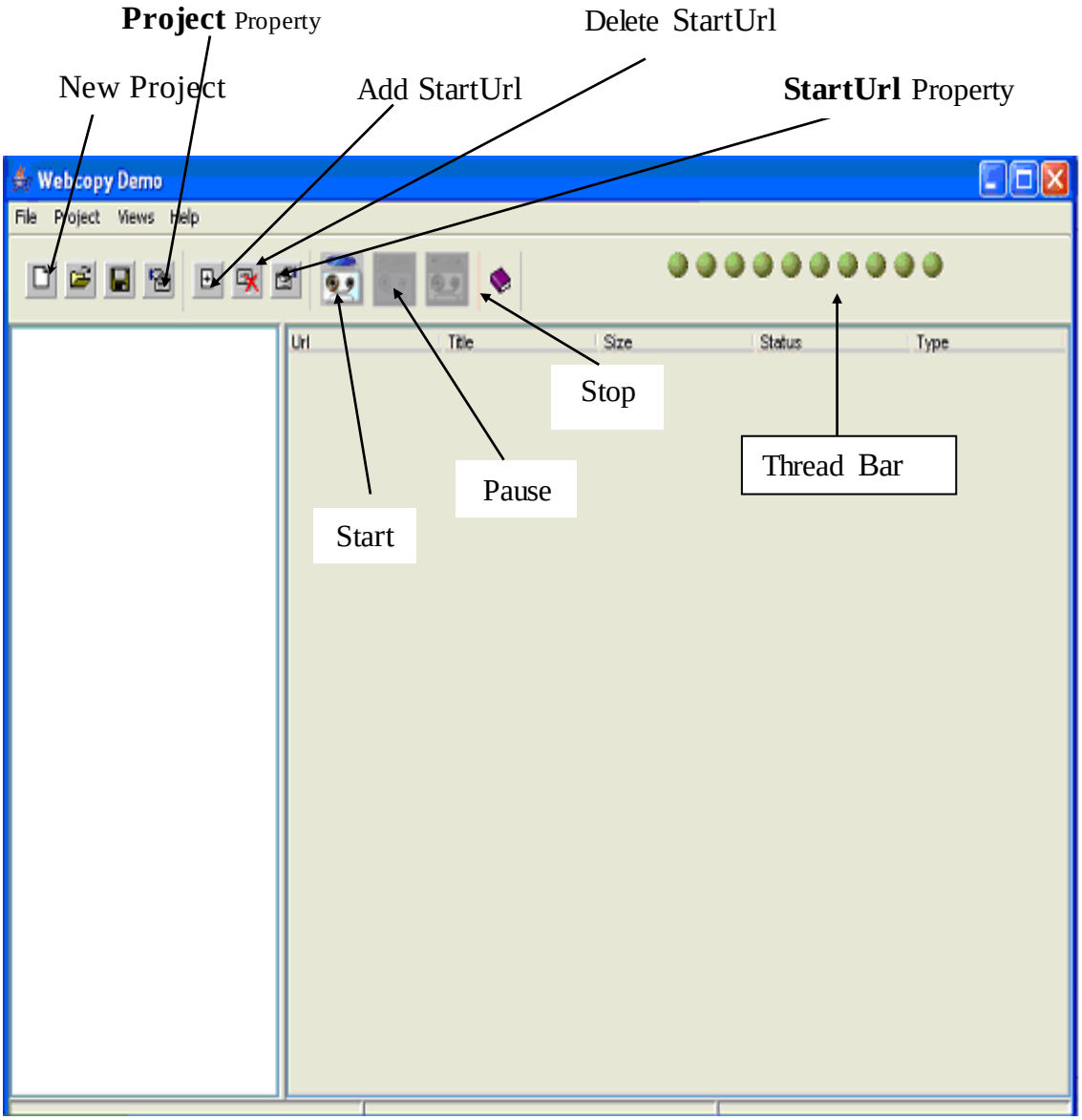
- ✓ Truyện Nguyễn Nhật Ánh
- ✓ Tuitor Java
- ✓ Tự Lực Văn Đoàn
- ✓ Unix Operating System

Kết quả lập chỉ mục: tạo ra tập tin nghịch đảo: file inverse.dat 4475KB

2. Hoạt động của chương trình

2.1 Giao diện quản trị

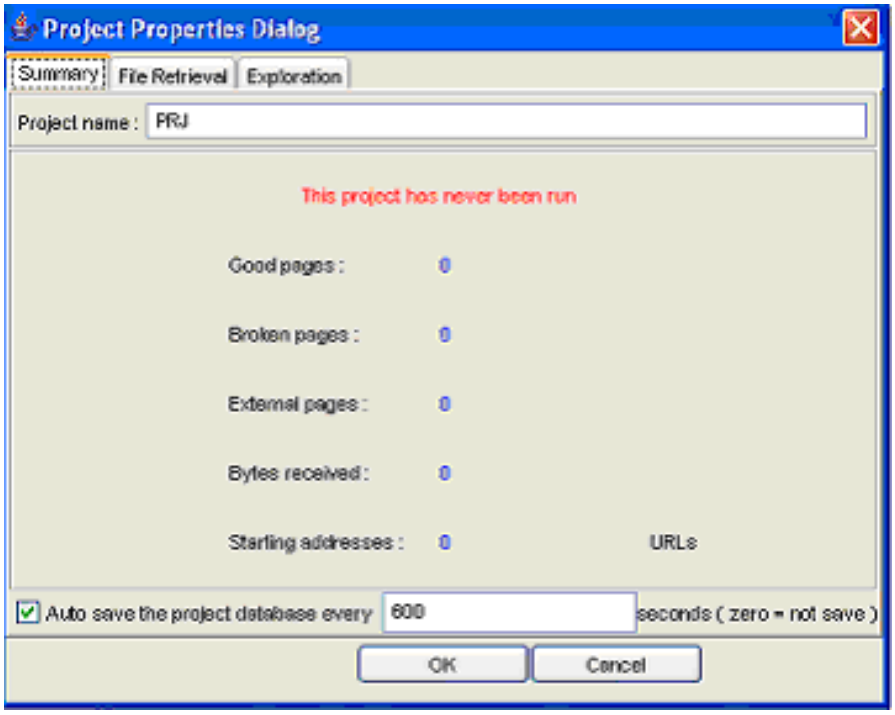
2.1.1 Giao diện chính của quản trị



Hình 10.1 Giao diện chính của quản trị

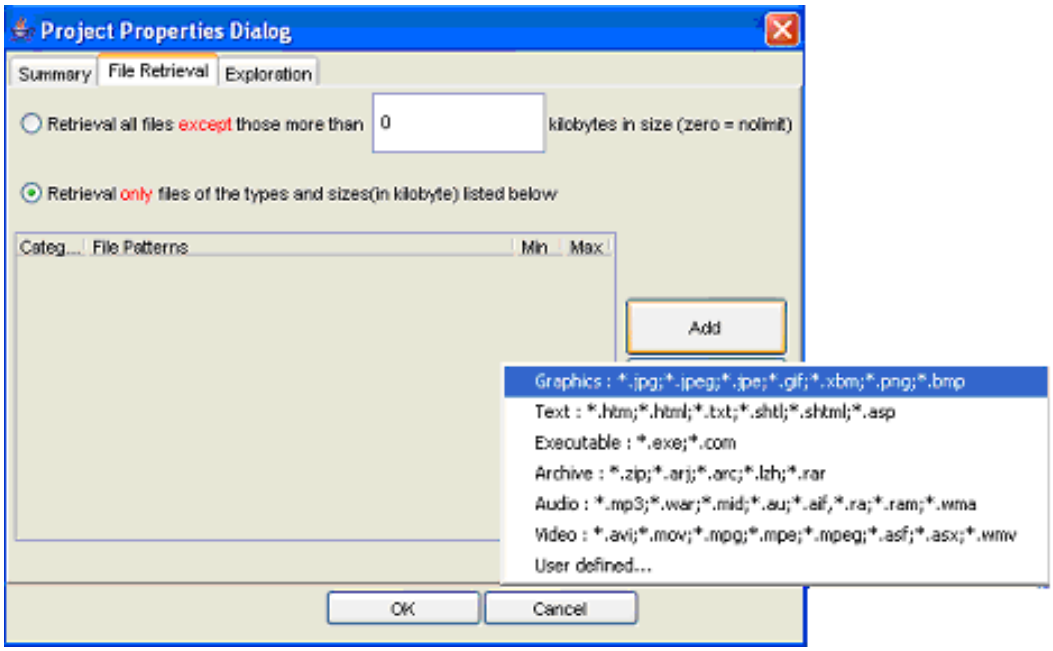
2.1.2 Tạo mới project

- Chọn File/New hoặc nhấn nút New trên thanh công cụ.



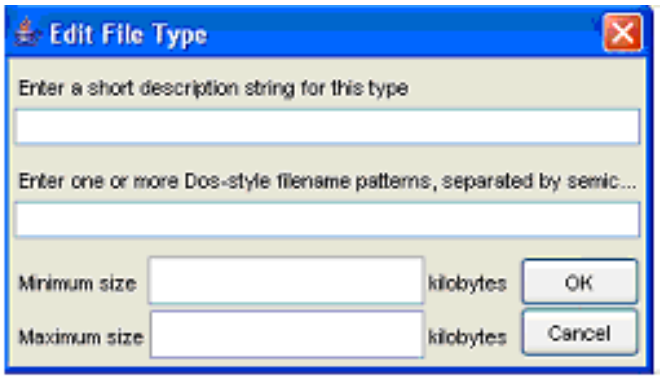
Hình 10.2Màn hình thể hiện một số thông tin chung về project

- Nhập tên project mới trong textbox Project Name

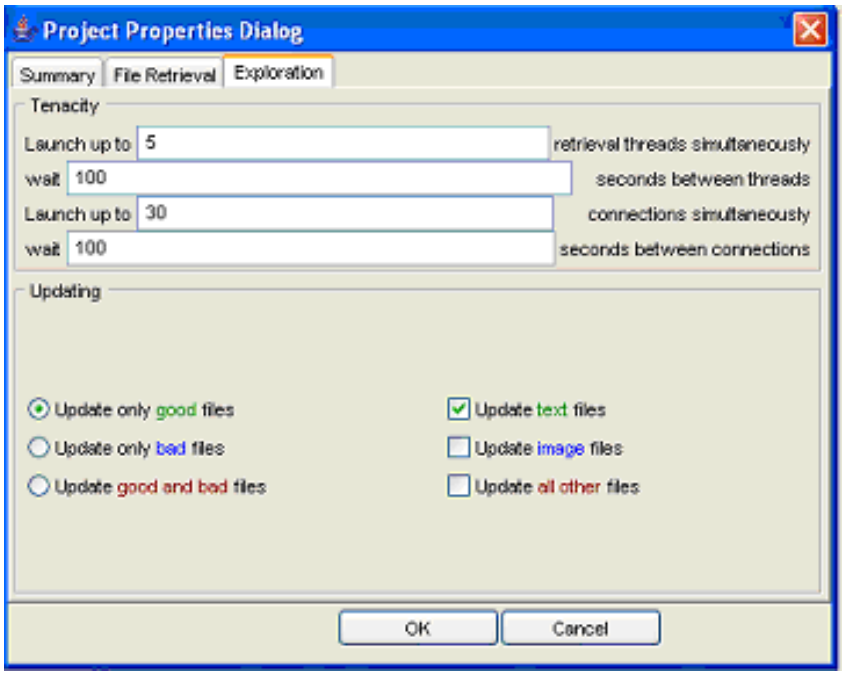


Hình 10.3Các tuỳ chọn thu thập dữ liệu của project

- Xác định các tuỳ chọn thu thập dữ liệu :
- Lấy mọi file [không | có] giới hạn kích thước.
- Chỉ lấy về các file có đuôi file và [không | có] giới hạn kích thước như liệt kê.
- Sửa chữa dạng file hiện có hoặc định nghĩa thêm dạng file mới bằng dialog EditFileType



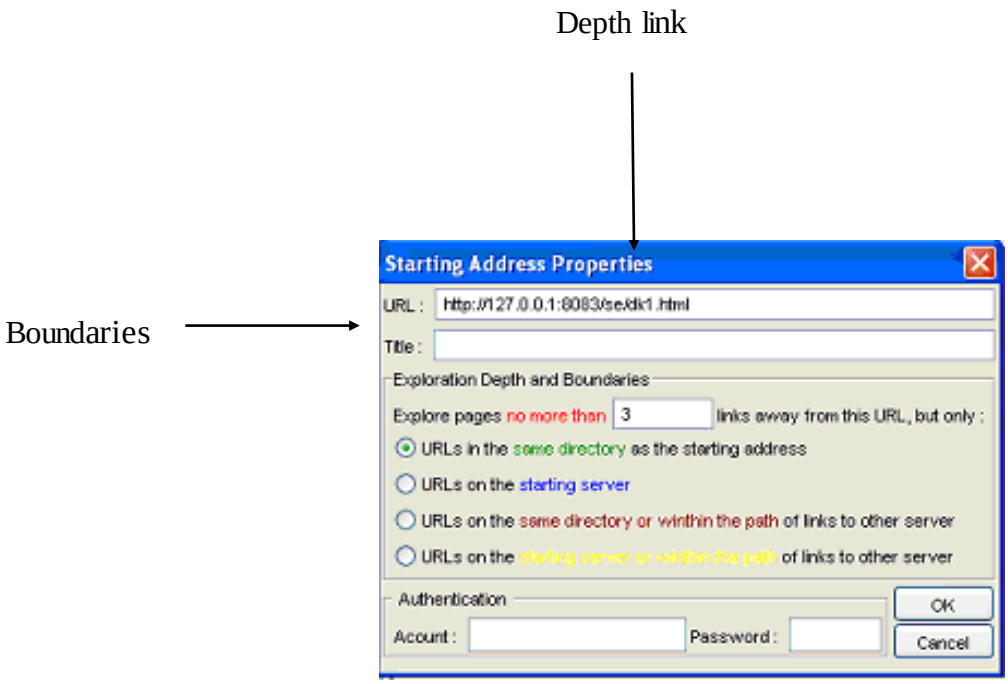
Hình 10.4Màn hình sửa chữa thông tin hoặc thêm mới một dạng file



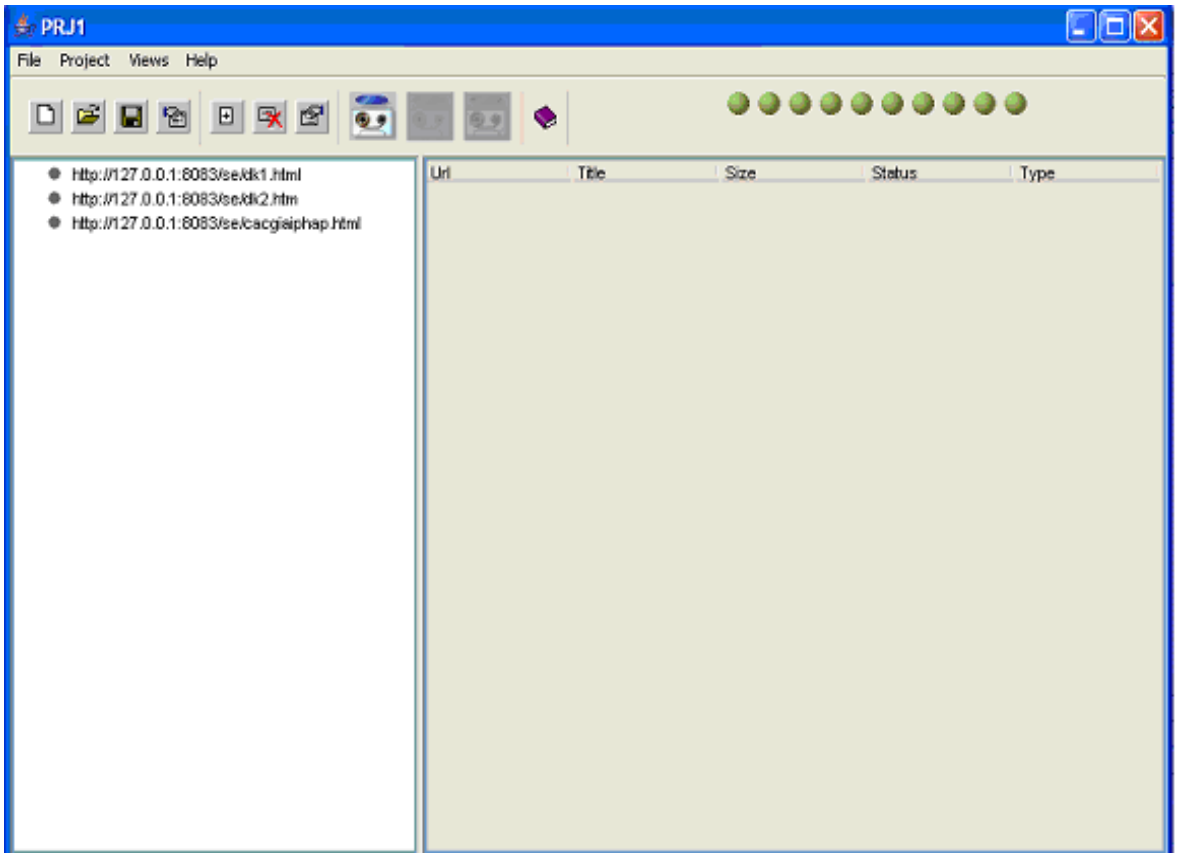
- Xác định số spider, thời gian đợi giữa các spider, số kết nối tạo sẵn đến CSDL, thời gian đợi nhận kết nối khi hệ thống bận trong khung Tenacity.
- Chọn các dạng file cần xử lý lại khi cập nhật thông tin cho một StartUrl trong khung Updating

2.1.3 Tạo mới một StartUrl :

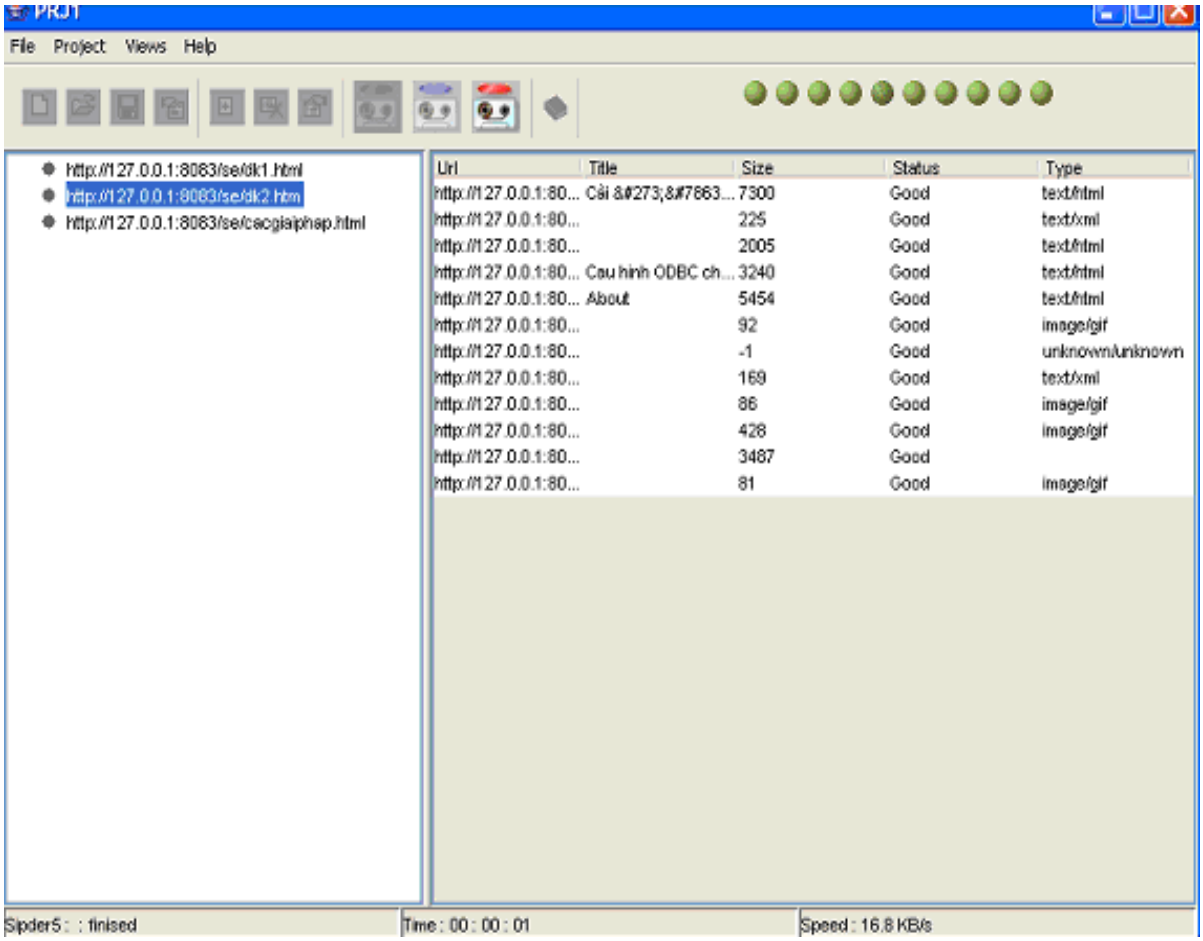
- Nhập địa chỉ URL vào textbox URL
- Nhập chuỗi mô tả về URL này trong textbox Title.
- Giới hạn phạm vi thu thập thông tin về StartUrl bằng cách định độ sâu liên kết và chọn kiểu ràng buộc đối với StartUrl.



Hình 10.5Màn hình chứa thông tin của một StartUrl

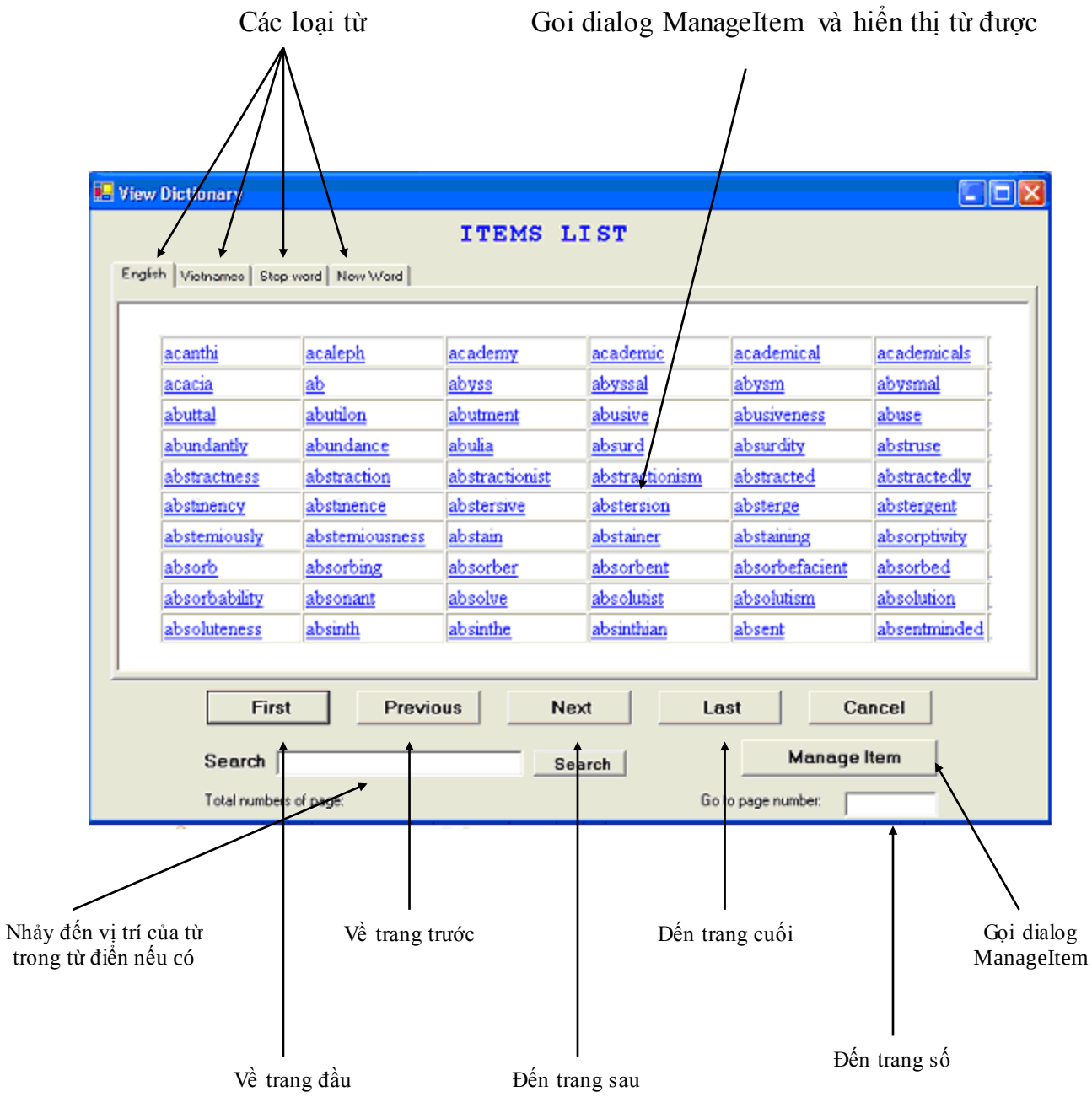


Hình 10.6Màn hình sau khi thêm một số StartUrl



Hình 10.7Màn hình thể hiện trạng thái đang xử lý StartUrl thứ 2

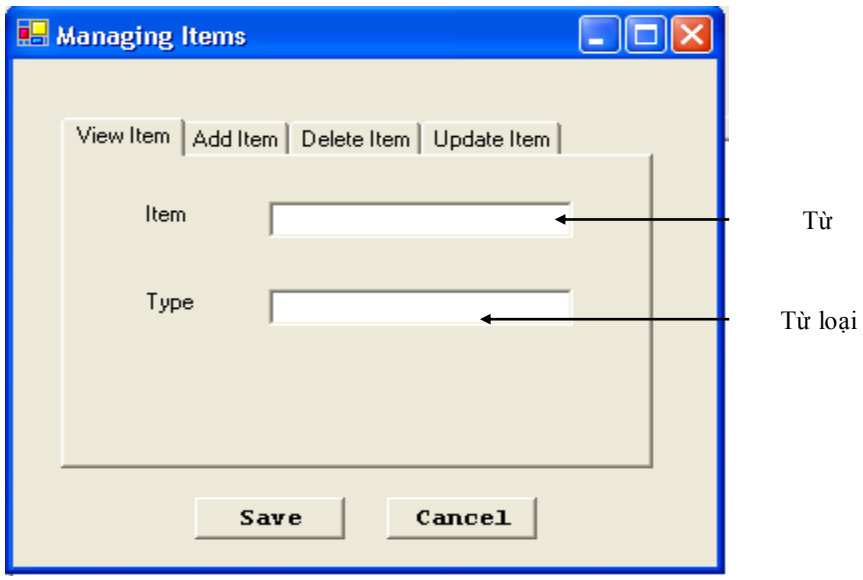
2.1.4 Xem từ điển chỉ mục



Hình 10.8 Màn hình xem từ điển chỉ mục

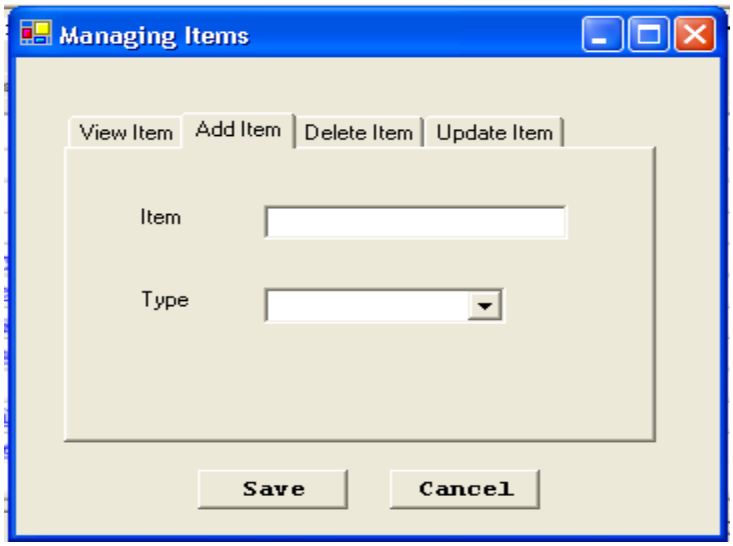
2.1.5 Quản lý mục từ

2.1.5.a xem một mục từ



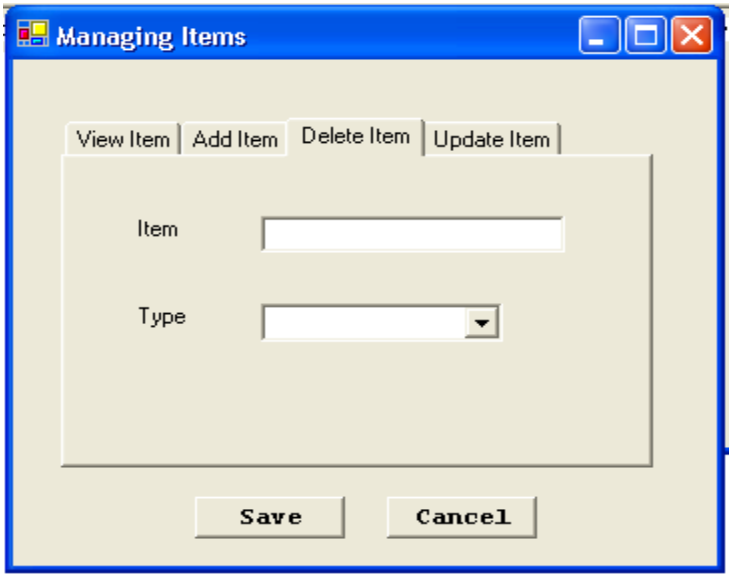
Hình 10.9Màn hình xem thông tin của một từ trong từ điển chỉ mục

2.1.5.b Thêm mục từ



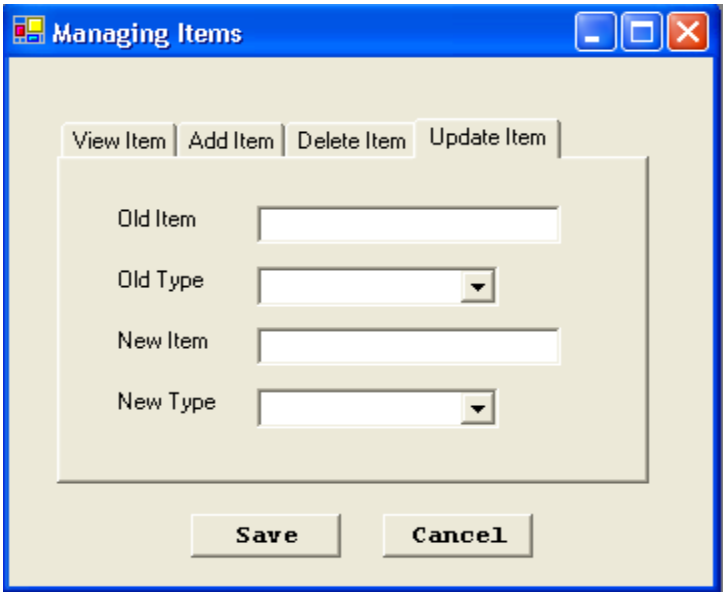
Hình 10.10Màn hình thêm một từ mới vào từ điển chỉ mục

2.1.5.c Xoá mục từ



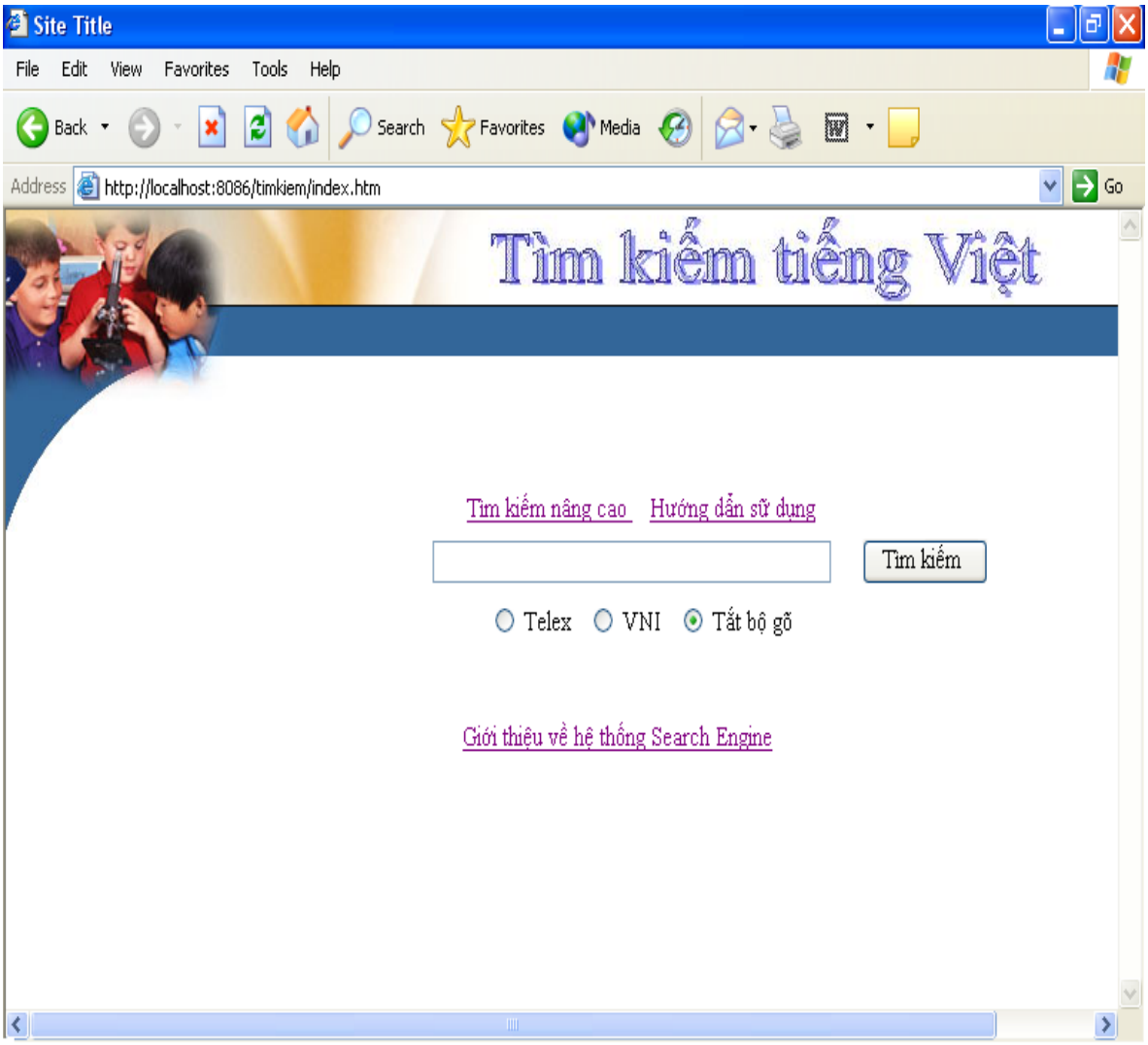
Hình 10.11Màn hình xóa một từ khỏi từ điển chỉ mục

2.1.5.d Cập nhật mục từ

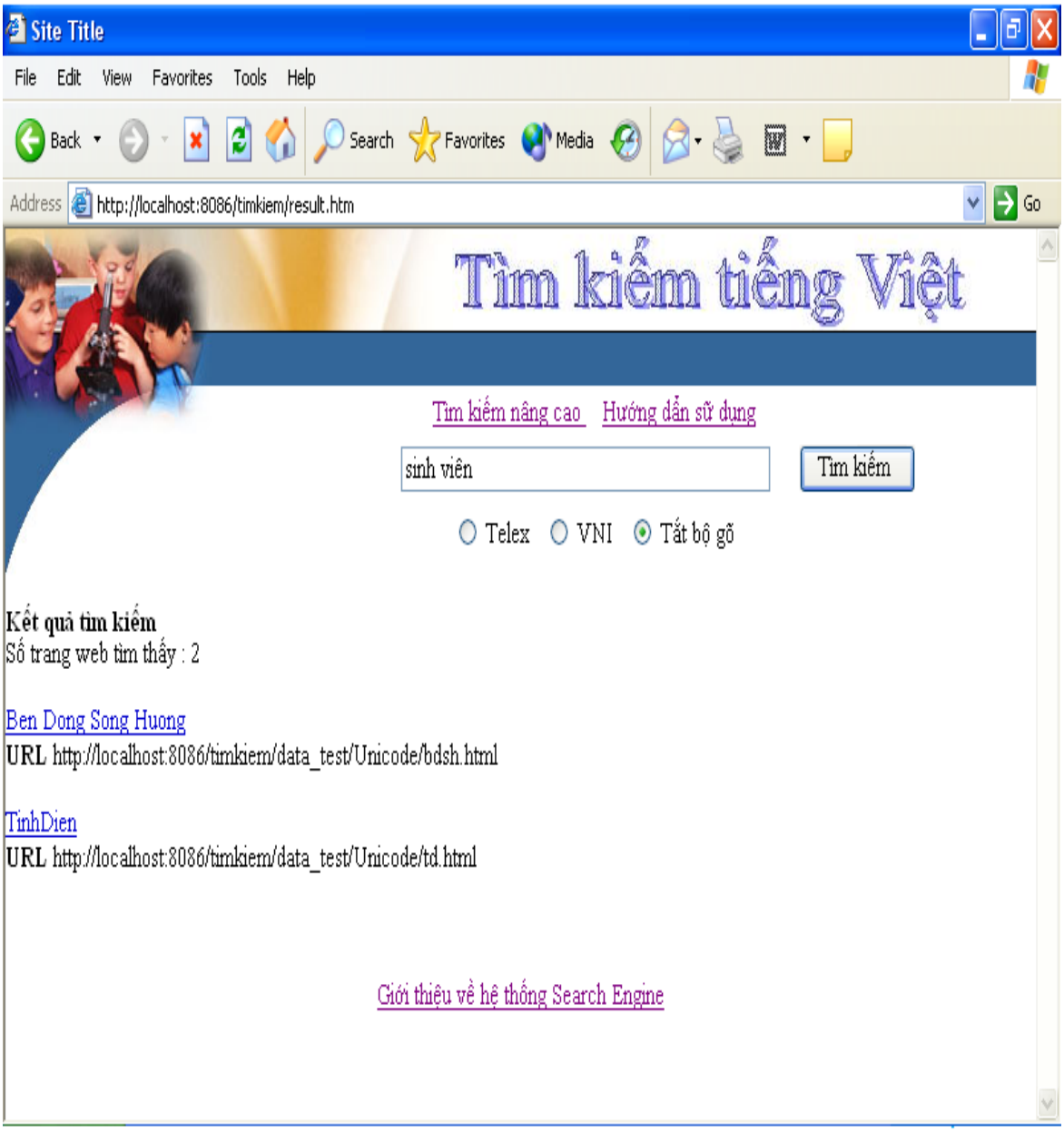


Hình 10.12 Màn hình cập nhật mục từ trong từ điển chỉ mục

2.2 Giao diện tìm kiếm



Hình 10.13Giao diện tìm kiếm thông tin của người dùng



Hình 10.14Màn hình kết quả

3. Đánh giá

3.1 Ưu điểm

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 về nguyên lý vận hành của một hệ thống search engine.
- ✓ Tìm hiểu các phương thức và chiến lược trong việc thiết kế từng module cụ thể cho hệ thống.
- ✓ Tìm hiểu các vấn đề đặc trưng của một hệ thống thu thập thông tin hoạt động trên môi trường mạng. Đề xuất một vài giải pháp xử lý những khó khăn của webrobot.
- ✓ Tìm hiểu các vấn đề đặc trưng của một hệ thống search engine tiếng Việt. Đề xuất một vài giải pháp đơn giản để xử lý những vấn đề khó khăn của tiếng Việt.
- ✓ Tìm hiểu hoạt động, thống kê một số đặc trưng và cách sử dụng của một số search engine thông dụng trên thế giới và Việt Nam.
- ✓ Tìm hiểu cơ bản về Semantic Search Engine.
- ✓ Xây dựng ứng dụng thử nghiệm cho một hệ thống search engine tiếng Việt với những kết quả đạt được như sau:
  - × Xây dựng công cụ đảm nhận việc thu thập các trang web một cách tự động với nhiều tiến trình đồng hành và nhiều tuỳ chọn trong tuỳ chọn trong qúa trình xử lý.

- Xây dựng công cụ lập chỉ tự động cho các từ tiếng Anh, tiếng Việt có dấu và không dấu.
- Hỗ trợ việc cập nhật, thêm, xoá, sửa từ mới vào từ điển.
- Xử lý hậu tố trong quá trình lập chỉ mục tiếng Anh.
- Xử lý được bỏ dấu không đồng nhất và Tiếng Việt không dấu
- Thời gian xử lý yêu cầu tìm kiếm khá nhanh và kết quả tương đối phù hợp.
- Giao diện đẹp, thân thiện, dễ sử dụng.

3.2 Khuyết điểm

Tuy nhiên do ứng dụng chỉ mang tính chất minh hoạ nên còn một số hạn chế cần phải cải tiến :

- Chưa có thời gian thử nghiệm ứng dụng trong môi trường mạng.
- Chưa tóm tắt được nội dung Website trả về

4. Hướng phát triển

4.1 Đối với từng module :

✓ Bộ thu thập thông tin

- Hỗ trợ nhiều hệ quản trị CSDL khác nhau.
- Lập lịch download các project một cách tự động.

- Xác định được các font trong file css mà những trang HTML tham chiếu tới.
- Phân tích cả những trang HTML không ở dạng text để lấy thông tin cho bộ lập chỉ mục.

✓ Bộ lập chỉ mục

- Hoàn chỉnh phần xử lý font chữ
- Xử lý thêm nhiều hậu tố khác
- Sử dụng các cách tổ chức, lưu trữ và xử lý dữ liệu như khác: bảng băm…
- Lập lịch cho việc lập chỉ mục các file được download về một cách tự động.

✓ Bộ tìm kiếm thông tin

- Hỗ trợ thêm nhiều toán tử và các tuỳ chọn tìm kiếm.
- Cải tiến kết quả tìm kiếm dựa vào kỹ thuật gom nhóm trên nhật ký người sử dụng (user log) hoặc dùng các thư mục web.

4.2 Đối với toàn luận văn:

✓ Cho phép ứng dụng chạy trên môi trường Web.

✓ Tăng tính hiệu quả, tăng tốc độ tìm kiếm, tăng tính ổn định và tính bảo mật của chương trình.

✓ Tóm tắt được nội dung các Website trả về.

✓ Hỗ trợ nhiều hơn việc tìm kiếm nâng cao: theo tiêu đề, theo ngày cập nhật, theo kiểu file….

DANH SÁCH CÁC BẢNG

DANH SÁCH CÁC HÌNH VẼ

TÀI LIỆU THAM KHẢO

I. Sách, ebook:

[I.1] Gerard Salton, Michael J.McGill, **Introduction to Modern Information Retrieval**

[I.2] C.J. van Rijsbergen , Department of Computing Science University of Glasgow, **Information Retrieval**

II. Luận văn, luận án

[II.1] Huỳnh Thụy Bảo Trân. Luận án thạc sĩ khoa học. **Nghiên cứu một số mô hình và xây dựng thử nghiệm một search engine Tiếng Việt**. Người hướng dẫn khoa học : GS.TS.Hoàng Văn Kiếm.

[II.2] Đoàn Hữu Quang Vinh. Luận văn cử nhân tin học. **Xây dựng công cụ hỗ trợ quá trình tiền xử lý cho hệ thống search engine**. GVHD : Huỳnh Thụy Bảo Trân.

[II.3] Bùi Ngọc Tuấn Anh, Trần Nguyễn Hoàng Phương. Luận văn cử nhân tin học. **Nghiên cứu một số thuật toán tra cứu thông tin trên Internet và cài đặt thử nghiệm.** GVHD: Hồ Bảo Quốc.

[II.4] Nguyễn Hải Quyền, Lương Thị Hoàng Thuý. Luận văn cử nhân tin học. **Tạo từ khoá cho văn bản tiếng Việt.** GVHD: Chu Tất Bích San.

III. Bài báo

[III.1] Dong Thi Bich Thuy, Ho Bao Quoc, Marie-France Bruandet, Jean-Pierre Chevallet, **An approach to Vietnamese Information Retrival**

IV. Website

[IV.1] http://citeseer.nj.nec.com

[IV.2] Conceptual Graph Home Page. http://www.cs.uah.edu/~delugach/CG

[IV.3] CYC ontology. http://www.cyc.com

[IV.4] *Search Engine Glossary*. http://www.cadenza.org/search_engine_terms

[IV.5] W3C SemanticWeb Activity. http://www.w3.org/2001/sw

[IV.6] WordNet . ftp://clarity.princeton.edu/pub/wordnet/ .Princeton University

[IV.7] http://www.robotstxt.org/wc/thread-or-treat.html

[IV.8] http://infopeople.org/search/chart.html

[IV.9] http://infopeople.org/search/guide.html

[IV.10] http://www.vinaseek.com

[IV.11] http://www.panvietnam.com

[IV.12] http://www.netnam.vn

[IV.13 http://monash.com