

Nhóm Nghiên cứu Đề tài

Đỗ Phúc	Tiến sĩ	Tin học	ĐH Công nghệ Thông tin, ĐHQG
Đỗ Hoàng Cường	Thạc sĩ	Tin học	Khoa CNTT, ĐHKHTN, ĐHQG
Nguyễn Tri Tuấn	Thạc sĩ	Tin học	Selab, ĐHKHTN, ĐHQG
Huỳnh Thụy Bảo Trân	Thạc sĩ	Tin học	Khoa CNTT, ĐHKHTN, ĐHQG
Nguyễn Văn Khiết	Thạc sĩ	Tin học	Khoa CNTT, ĐHKHTN, ĐHQG
Nguyễn Việt Hoàng	Cao học	Tin học	Khoa CNTT, ĐHKHTN, ĐHQG
Nguyễn Việt Thành	Cao học	Tin học	Khoa CNTT, ĐHKHTN, ĐHQG
Phạm Phú Hội	Cao học	Tin học	ĐH Công nghệ Thông tin, ĐHQG
Dương Ngọc Long Nam	Cao học	Tin học	Selab, ĐHKHTN, ĐHQG
Nguyễn Phước Thanh Hải	Cao học	Tin học	Selab, ĐHKHTN, ĐHQG

Nội dung

MỞ ĐẦU.....	5
1 PHẦN I:	6
TÌM HIỂU VÀ SO SÁNH MỘT SỐ S.E THÔNG DỤNG HIỆN NAY	6
1.1 MỘT SỐ S. E NƯỚC NGOÀI THÔNG DỤNG HIỆN NAY (xem Bảng Tổng hợp chi tiết trong Phụ lục 1, 2,3).....	6
1.1.1 GOOGLE	6
1.1.2 LYCOS	9
1.1.3 ALTA VISTA	10
1.2 MỘT SỐ S. E TIẾNG VIỆT THÔNG DỤNG HIỆN NAY (xem Bảng tổng hợp chi tiết trong Phụ lục 4).	12
1.2.1 NETNAM	12
1.2.2 VINASEEK	16
1.3 NHẬN XÉT – SO SÁNH VỀ MỘT SỐ S.E.	17
1.3.1 SO SÁNH.	17
1.3.2 NHẬN XÉT.	19
2 PHẦN 2:.....	23
XÂY DỰNG TỪ ĐIỂN NGỮ NGHĨA THUẬT NGỮ TIN HỌC.....	23
2.1 TÌM KIẾM THEO NGỮ NGHĨA.....	23
2.2 BIỂU DIỄN NGỮ NGHĨA	24
2.2.1 ĐỒNG HIỆN (CO-OCCURRENCE).....	24
2.2.2 HỆ THỐNG QUAN HỆ ĐỒNG NGHĨA ĐƠN GIẢN	25
2.3 ONTOLOGY.....	42
2.3.1 XÂY DỰNG ONTOLOGY	42
2.3.2 TRAO ĐỔI ONTOLOGY	44
2.3.3 XÂY DỰNG ONTOLOGY TỪ VĂN BẢN	45
2.3.4 XÂY DỰNG ONTOLOGY CHUYÊN NGÀNH TIN HỌC	51

2.3.5	BIỂU DIỄN ONTOLOGY TRONG CƠ SỞ DỮ LIỆU	55
2.4	BIỂU DIỄN CẤU TRÚC PHÂN CẤP CỦA ONTOLOGY TRONG CƠ SỞ DỮ LIỆU QUAN HỆ.....	62
2.4.1	CÁC NHƯỢC ĐIỂM CỦA CÁCH BIỂU DIỄN BẰNG CON TRỎ.	62
2.4.2	BIỂU DIỄN CẤU TRÚC CÂY TRONG ORACLE	63
2.4.3	NHẬN XÉT	71
2.5.	KẾT LUẬN.....	72
3	PHẦN III:.....	73
	THIẾT KẾ HỆ THỐNG S.E VÀ KẾT QUẢ THỬ NGHIỆM.....	73
3.1	THIẾT KẾ HỆ THỐNG.....	73
3.1.1	Đặt tả Hệ thống:.....	73
3.1.2	Thiết kế các Chức năng của Hệ thống.	73
3.1.3	Thuật giải nhận dạng bảng mã.....	83
3.2	CÀI ĐẶT HỆ THỐNG.	86
3.2.1	Tổ chức Các Giao diệnModule WebRobot.	86
3.3	Kết quả thử nghiệm.	95
4.	KẾT LUẬN.....	100
	PHỤ LỤC.....	101
	PHỤ LỤC 1. BẢNG TÓM TẮT ĐẶC TRƯNG CỦA MỘT SỐ S.E NƯỚC NGOÀI.....	101
	PHỤ LỤC 2. BẢNG TÓM TẮT ĐẶC TRƯNG MỘT SỐ META-S E NƯỚC NGOÀI.....	103
	PHỤ LỤC 3. BẢNG TÓM TẮT MỘT SỐ HỆ THỐNG DANH MỤC (SUBJECT DIRECTORIES).....	104
	PHỤ LỤC 4. BẢNG TÓM TẮT ĐẶC TRƯNG CỦA MỘT SỐ S.E TRONG NƯỚC.....	105
	PHỤ LỤC 5. QUAN HỆ GIỮA ĐỘ CHÍNH XÁC & ĐỘ GỌI LẠI.....	106
	PHỤ LỤC 6. THỐNG KÊ VỀ PHÂN HẠNG CỦA CÁC DOMAIN	107
	PHỤ LỤC 7. SƠ ĐỒ QUAN HỆ S.E	110

PHỤ LỤC 8: CÁC MÃ NGŨ NGHĨA CỦA LDOCE	111
PHỤ LỤC 9. TỔNG QUAN VỀ CÔNG NGHỆ ORACLE TEXT ĐỂ PHÁT TRIỂN S.E.	112
PHỤ LỤC 10. SƠ LƯỢC VỀ THƯ VIỆN VNCONVERT:	116
TÀI LIỆU THAM KHẢO.	118
CÁC TRANG WEB.....	119

MỞ ĐẦU

Hiện nay, InterNET đã trở thành một Siêu Xa lộ Thông tin, cung cấp thông tin cho mọi người, ở mọi nơi, trong mọi ngành, mọi lĩnh vực. Hiện nay trên thế giới có rất nhiều SEARCH ENGINE chẳng hạn như GOOGLE (xem [2], [3], [5]), YAHOO, ALLTHEWEB, ALTA VISTA (xem [4]), ... có khả năng tìm kiếm trên nhiều ngôn ngữ khác nhau, nhưng với Tiếng VIỆT vẫn có hạn chế. Và trong nước cũng có vài SEARCH ENGINE chẳng hạn như NETNAM (xem [7]), VINASEEK (xem [8]),...]). Mặc dù đã có rất nhiều SEARCH ENGINE, nhưng vẫn rất cần thiết có một sự nghiên cứu đầy đủ để phát triển một Hệ thống SEARCH ENGINE trên tiếng VIỆT có chú ý đến từ khóa là TỪ GHÉP và NGỮ NGHĨA trong lĩnh vực Công nghệ thông tin (CNTT). Trên cơ sở này, có thể phát triển một Hệ thống SEARCH ENGINE tiếng VIỆT tổng quát cho mọi lĩnh vực.

Thời gian thực hiện Đề tài là 18 tháng từ tháng 01/2003 đến 07/2004.

Bản báo cáo này nhằm trình bày một số kết quả bước đầu:

- ✚ PHẦN I. Thu thập và nghiên cứu tính năng của một số SEARCH ENGINE thông dụng. So sánh và đánh giá các SEARCH ENGINE (S.E) này.
- ✚ PHẦN II. Xây dựng Từ điển ngữ nghĩa Thuật ngữ Tin học.
- ✚ PHẦN III. Thiết kế Hệ thống & kết quả thử nghiệm.

1 PHẦN I:

TÌM HIỂU VÀ SO SÁNH MỘT SỐ S.E THÔNG DỤNG HIỆN NAY

Phần này nhằm tìm hiểu phương thức hoạt động và tóm tắt các đặc trưng chính của một số search engine tiếng Anh, tiếng Việt thông dụng hiện nay. Đưa ra những so sánh về sự giống nhau, khác nhau và những nhận xét về xu hướng hoạt động, xử lý thông tin của chúng. Đồng thời đánh giá hiệu năng hoạt động và thống kê vài số liệu xử lý của một số S.E cụ thể.

1.1 MỘT SỐ S. E NƯỚC NGOÀI THÔNG DỤNG HIỆN NAY (xem Bảng Tổng hợp chi tiết trong Phụ lục 1, 2,3).

1.1.1 GOOGLE

Hiện nay, GOOGLE là một trong các S.E được ưa chuộng nhất. Để đạt được kết quả tìm kiếm với độ chính xác cao thì cần phải nhắc đến hai đặc trưng quan trọng của Google, đó là việc sử dụng cấu trúc của các siêu liên kết để tính độ phổ biến (pageRank) (phân hạng) cho mỗi trang web. Đặc trưng thứ hai là tận dụng lại những siêu liên kết để cải tiến kết quả tìm kiếm.

GOOGLE được cài đặt bằng C hay C++, có thể hoạt động trên cả Solaris và Linux Việc dò tìm các trang web thực hiện bởi các bộ dò tìm (web crawler) được đặt phân tán. Một Máy chủ (Server) sẽ đảm nhận việc gửi danh sách các URL cần tìm đến cho các bộ dò tìm. Các trang web tìm về sẽ được lưu trữ vào kho của các server dưới dạng nén. Khi phân tích một URL mới, mỗi trang web sẽ được gán một số hiệu nhận dạng, gọi là DocID. Việc lập chỉ mục thực hiện bởi bộ lập chỉ mục (Indexer) và bộ sắp xếp (Sorter). Bộ lập chỉ mục thực hiện các chức năng như đọc kho dữ liệu, giải nén và phân tích các tài liệu. Mỗi tài liệu được chuyển đổi thành tập tần số xuất hiện của các từ, gọi là các hit.

Các hit ghi nhận từ, vị trí trong tài liệu, kích thước font xấp xỉ, và chữ hoa hay chữ thường. Bộ chỉ mục phân phối các hit này vào trong một tập các barrels (thùng), tạo một chỉ mục thuận đã sắp xếp theo từng phần. Ngoài ra, bộ chỉ mục còn phân tích tất cả liên kết trong mỗi trang web và lưu thông tin quan trọng về chúng trong một anchor file. Tập tin này chứa đủ thông tin để xác định liên kết này từ đâu, chỉ đến đâu và chứa đoạn văn bản liên kết. Trình phân giải URL đọc tập tin các neo tạm thời (anchor) và chuyển các URL tương đối thành các URL tuyệt đối và trả về các docID. Đặt văn bản neo vào chỉ mục forward có liên quan đến docID mà neo chỉ đến và tạo một cơ sở dữ liệu tương ứng giữa các liên kết với các docID. Cơ sở dữ liệu này được dùng để tính các PageRank cho tất cả các tài liệu.

Bộ sắp xếp lấy các barrel, đã được sắp xếp cục bộ, và sắp xếp lại chúng theo docID để sinh ra một chỉ mục nghịch đảo. Công việc này được thực hiện ngay tại chỗ nên không mất nhiều bộ đệm. Bộ sắp xếp cũng đồng thời sinh ra một danh sách WordID và bù lại cho chỉ mục nghịch đảo. Một chương trình gọi là DumpLexicon lấy danh sách này và từ vựng (lexicon) được sinh bởi bộ lập chỉ mục và tạo một từ vựng mới được dùng cho bộ tìm kiếm (searcher). Bộ tìm kiếm được chạy bởi một web server và sử dụng từ vựng đã được DumpLexicon xây dựng cùng với chỉ mục nghịch đảo và các PageRank để trả lời các truy vấn.

Tốc độ tìm kiếm của Google phụ thuộc vào hai yếu tố: hiệu quả của thuật toán tìm kiếm và sự liên kết xử lý của hàng ngàn hàng ngàn máy tính cấp thấp để tạo nên một S.E siêu tốc.

Google sắp thứ tự các kết quả một cách tự động nhờ vào hơn 100 bộ xử lý, sử dụng thuật toán tính độ phổ biến PageRank.

Phần mềm quan trọng nhất là PageRank, một hệ thống phân loại các trang web được phát triển bởi Larry Page và Sergey Brin ở đại học Stanford. PageRank sử dụng cấu trúc liên kết của các trang web như một giá trị chỉ báo ban đầu cho trang riêng lẻ đó. Thực chất, Google xem các liên kết từ trang A đến trang B như một lá phiếu từ trang A cho trang B. Google còn xem xét một khối lượng lớn các lá phiếu khác, hay phân tích liên kết trong các trang nhận

được để thu thập lá phiếu. Việc thu thập các lá phiếu nhằm xác định trọng số hay độ quan trọng của trang web. Những site chất lượng cao sẽ nhận được độ phổ biến cao, đây chính là giá trị được xem xét đến trong quá trình tìm kiếm. Dĩ nhiên, một trang quan trọng sẽ không có giá trị nếu nó không phù hợp với câu truy vấn. Google kết hợp pagerank với một kỹ thuật so khớp từ khoá tinh vi để tìm ra các trang mà nó vừa quan trọng lại vừa phù hợp với nội dung tìm kiếm. Để tìm được kết quả phù hợp nhất cho câu truy vấn Google không chỉ dựa trên số lần từ tìm kiếm xuất hiện mà còn xem xét đến nội dung của trang và nội dung của các trang liên kết đến nó.

Hệ thống chỉ mục của Google được cập nhật hàng tháng. Mỗi khi cơ sở dữ liệu các trang web cập nhật thì có những thay đổi: thêm site mới, mất site cũ và phân hạng của một số site có thể thay đổi. Sự phân hạng ban đầu của một site có thể bị ảnh hưởng bởi sự phân hạng lại của các site khác. Không một ai có can thiệp để nâng kết quả phân hạng cho một site, những kết quả trả về đều được xác định hoàn toàn tự động.

Mặc dù chức năng tìm kiếm trên Yahoo được hỗ trợ bởi Google, nhưng cách xử lý các truy vấn của hai site này không hoàn toàn giống nhau. Vì vậy kết quả của cả hai cũng không thể nào giống nhau một cách hoàn toàn. Điều này không phải là lỗi của một S.E nào cả mà chỉ đơn thuần phản ánh sự khác nhau trong tuần suất mà mỗi site dùng để cập nhật thông tin hay số lượng các trang thông tin mà hệ thống đã xử lý. Thuật toán tìm kiếm cơ bản của hai hệ thống giống nhau hoàn toàn. Tính năng bộ nhớ đệm (lưu trữ tạm thời nội dung của trang web để tăng tốc độ truy cập hoặc tìm kiếm) của GOOGLE, được giới thiệu vào năm 1997, là một tính năng độc đáo so với các công cụ tìm kiếm khác, nhưng không giống các site lưu trữ trên web lưu trữ lại bản sao của các trang web. Tính năng này cho phép mọi người truy cập vào một bản sao của hầu như bất kỳ website nào, ở dạng mà lần cuối cùng Google phân loại và lập chỉ mục. Có thể trang web cache này được truy cập có tuổi đời chỉ vài phút hoặc vài tháng, điều này tùy thuộc vào lần cuối cùng mà Google tìm đến lập chỉ mục. Không như những dự án lưu trữ web khác, tính năng cache của

Google không cố gắng tạo ra một bản sao lưu trữ cố định của trang web mà thực hiện tìm kiếm liên tục các đường link chết để xóa bỏ, khi nào trang web không còn tồn tại thì công cụ tìm kiếm sẽ thanh lọc các cache có liên quan đến link đó trong thời gian sớm nhất có thể. Tuy nhiên tính năng cache này cũng làm cho Google phải đụng chạm đến vấn đề bản quyền vì người tìm kiếm đôi khi có thể xem được các thông tin, bài viết chỉ dành riêng cho các thuê bao có đăng ký.

Hiện nay GOOGLE đã xử lý hơn 8 tỷ trang tài liệu, đang thử nghiệm một phiên bản mới tại đại chỉ <http://www.scholar.google.com/>

Tuy nhiên, GOOGLE vẫn còn hạn chế trong tìm kiếm tiếng Việt

1.1.2 LYCOS

Thế giới của Lycos là gia đình nhện Lycosidae, nó liên tục duyệt các trang web để tìm thông tin. Kết quả tìm kiếm sau đó được trộn vào catalog theo chu kỳ hàng tuần. Lycos giúp người dùng tìm các tài liệu Web chứa các từ khóa đặc biệt do người dùng cung cấp. Lycos nhanh chóng trở nên rất phổ biến đối với những người dùng Web có nhu cầu tìm kiếm toàn bộ nội dung (full-content) trong không gian các tài liệu.

Lycos định nghĩa không gian Web là bất kỳ tài liệu nào trong các không gian HTTP, FTP, Gopher. Lycos có thể lấy các tài liệu mà nó chưa từng tìm kiếm bằng cách dùng text trong tài liệu mẹ như là một mô tả cho các kết nối chưa được khám phá (anchor text). Tuy nhiên, Lycos không tìm kiếm và index các không gian ảo vô hạn, hay biến đổi. Do đó, Lycos bỏ qua các không gian sau: các CSDL WAIS, Usenet news, không gian Mailto, các dịch vụ Telnet, không gian tập tin cục bộ.

Nhằm giảm lượng thông tin cần lưu trữ, từ những tài liệu thu được Lycos chỉ lưu các thông tin sau: tựa đề, heading và sub-heading, 100 từ quan trọng nhất, 20 dòng đầu tiên, kích thước tính theo bytes, số từ. Lựa chọn 100 từ quan trọng, được thực hiện theo thuật toán định lượng, dựa trên việc xem xét vị trí và tần số của từ. Các từ được cho điểm theo mức độ nhúng sâu vào tài liệu.

Do đó, các từ xuất hiện trong tựa đề và đoạn đầu tiên sẽ được tính điểm cao hơn.

Lycos sử dụng phương pháp thống kê để lướt qua các server trong không gian Web, nhằm tránh làm quá tải một server với hàng loạt các yêu cầu và cũng cho phép Lycos tăng độ ưu tiên đối với các Url nhiều thông tin hơn. Các bước cơ bản của thuật toán như sau:

1. Khi một tìm thấy một Url, Lycos quét qua nội dung của nó, tìm các tham chiếu đến các Url mới và đưa vào một hàng đợi nội bộ.
2. Để chọn Url kế tiếp, Lycos lựa ngẫu nhiên một tham chiếu trong hàng đợi trên theo độ ưu tiên.

Lycos thường tìm kiếm các tài liệu phổ biến, đó là các tài liệu có nhiều kết nối, Lycos cũng ưu tiên cho các Url ngắn gọn, chính là các thư mục ở mức cao nhất (top-level) và các tài liệu gần gốc hơn.

1.1.3 ALTA VISTA

Vào cuối năm 2002, Alta Vista đã thực hiện nâng cấp hệ thống tìm kiếm và hiện nay trang web này đã có hơn 65 triệu lượt người truy cập mỗi tháng. Hiện nay Alta Vista có 250 nhân viên và công cụ tìm kiếm này được thể hiện với 25 thứ tiếng.

Alta Vista là một S.E rất mạnh về tìm kiếm theo từ khóa. Cho phép tìm kiếm theo nhiều cụm từ bằng cách đặt những cụm từ cần tìm vào trong hai dấu nháy kép. Ví dụ: "search engine" or "information retrieval". Ngoài ra, Alta Vista còn cung cấp nhiều lựa chọn để cải tiến việc tìm kiếm. Giống như những S.E khác, Alta Vista cũng tổ chức dữ liệu thành từng nhánh thư mục, như: tin tức, du lịch, thể thao, sức khỏe. Bên cạnh đó, AltaVista còn có những tính năng đặc biệt, ví dụ như người dùng nhập vào một truy vấn, bên cạnh kết quả tìm được, AltaVista còn đưa ra một số câu hỏi liên quan đến vấn đề tìm kiếm đề gợi ý. Chẳng hạn, nếu tìm mục "dog"(con chó), AltaVista sẽ đưa ra câu hỏi "Hot dog (xúc xích nóng) làm như thế nào?" cùng với nút Answer để kết nối tới các site liên quan.

Trên biểu mẫu tìm kiếm cơ bản của AltaVista, người dùng có thể chỉ định kết quả khai báo bằng một trong 25 thứ tiếng; tính năng này chỉ có trong các biểu mẫu tìm kiếm nâng cao đối với các site khác. Ngoài ra, Alta Vista còn hỗ trợ nhiều tiện ích, đặc biệt là công cụ Babelfish(babelfish.altavista.com) cho phép dịch từng câu hay cả trang web giữa các tiếng Anh, Pháp, Ý, Tây Ban Nha ...

Alta Vista có những web crawler thường xuyên đi dò và lấy về những dữ liệu text, sau đó chuyển cho bộ lập chỉ mục. Crawler chính tên là Scooter, và nó có thêm những hệ thống con đảm nhận việc kiểm tra và duy trì các kết quả trong hệ thống index hiện hành, như là kiểm tra các siêu liên kết nào không hoạt động (dead link), đã di chuyển sang nơi khác hay không còn tồn tại, để có những xử lý thích hợp như sẽ loại những trang này khỏi hệ thống chỉ mục. Scooter phát đi cùng một lúc hàng ngàn các tiến trình. Trong 24 giờ một ngày, 7 ngày một tuần, scooter và các hệ thống con của nó truy cập đến hàng ngàn trang web trong cùng một thời điểm, như hàng ngàn người mù bắt lấy các dữ liệu text, kéo về hệ thống và chuyển cho hệ thống lập chỉ mục và đến ngày hôm sau thì những dữ liệu đó đã được lập chỉ mục. Trong lúc duyệt những trang web thì tất cả các siêu liên kết tìm thấy trong đó sẽ được đưa vào một danh sách để duyệt vào lần kế tiếp. Trong một ngày thường Scooter và những hệ thống con của nó sẽ duyệt qua trên 10 triệu trang web.

Hoạt động của Alta Vista không giống như những S.E khác. Không chỉ quan tâm đến dữ liệu metatag (những câu lệnh đặc biệt được nhúng vào trong header của trang web) mà nó còn quan tâm đến tất cả mọi từ trong trang web. Chúng ta thường nghĩ rằng những gì có thứ tự cũng tốt hơn những gì không được sắp thứ tự, nhưng điều này thì không đúng đối với Alta Vista, nó thực hiện lập chỉ mục trên toàn bộ văn bản (full-text indexing). Và một quan niệm chung cho rằng: nếu có quá nhiều dữ liệu và cần phải tìm kiếm, rút trích thông tin trong đó thì chỉ có cách duy nhất là quản lý bằng một hệ quản trị cơ sở dữ liệu. Có nghĩa là cần phải xác định các trường dữ liệu, phân loại các thông tin Như vậy, có rất nhiều việc phải thực hiện khi xác lập hệ thống và bảo trì nó.

Đối với Alta Vista thì ngược lại, dữ liệu không phân hạng và cũng không cần bảo trì. Tất cả các tập tin đều không có cấu trúc và cũng không có thứ tự.

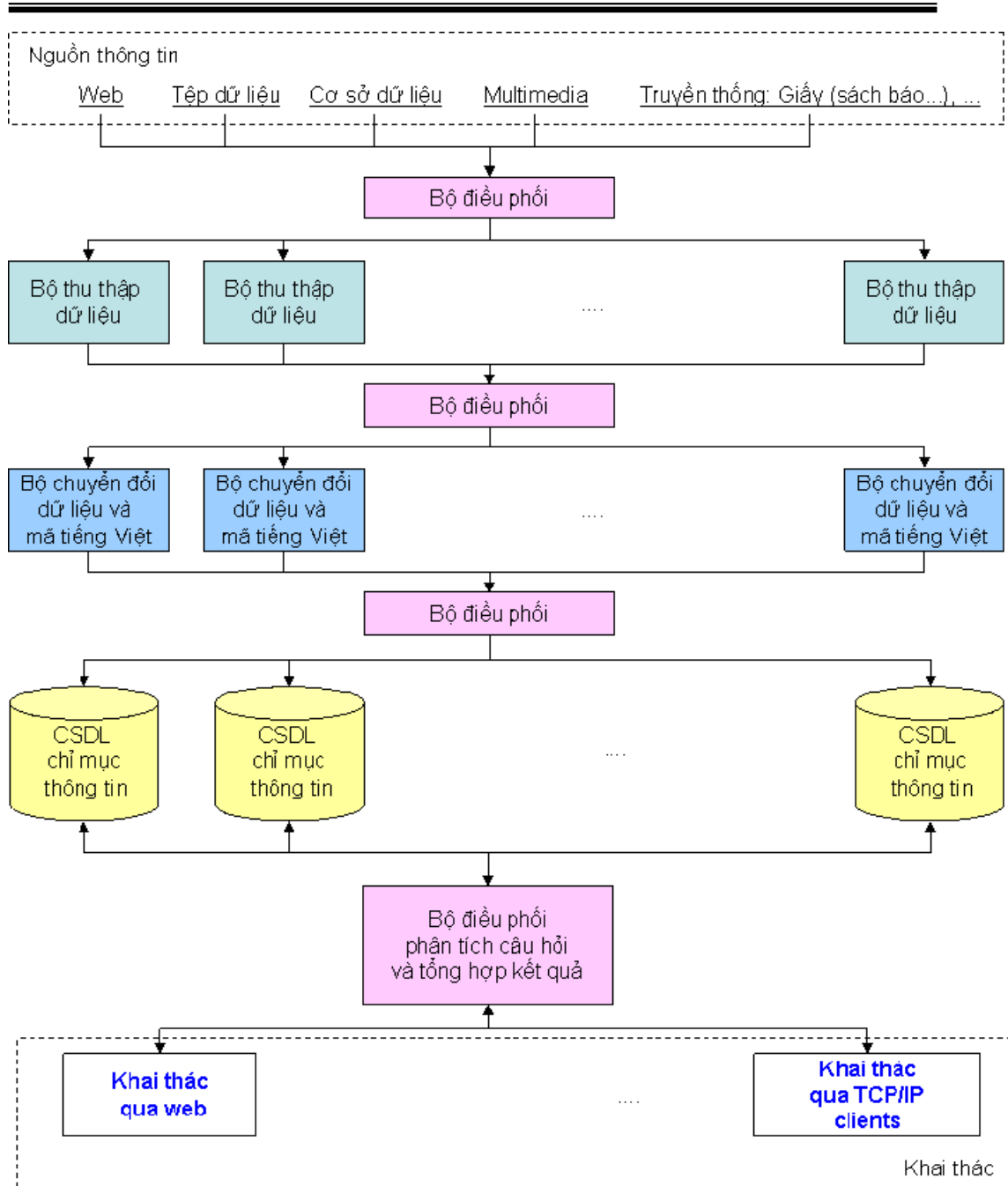
1.2 MỘT SỐ S. E TIẾNG VIỆT THÔNG DỤNG HIỆN NAY (xem Bảng tổng hợp chi tiết trong Phụ lục 4).

1.2.1 NETNAM

NetNam là một trong số ít ỏi các S.E dành cho tiếng Việt. NetNam chú trọng đến việc thiết kế hệ thống phù hợp với điều kiện cơ sở vật chất, hạ tầng của Việt Nam Vì thế nên hệ thống được thiết kế theo kiến trúc xử lý song song, với các khối chức năng được minh hoạ như Hình1. Kiến trúc này cho phép hệ thống có thể hoạt động phân tán từ một đến hàng trăm máy tính, cho phép các máy tính cỡ nhỏ thay thế cho các hệ máy chủ cao cấp. Khi nhu cầu phục vụ tăng lên thì chỉ cần thêm máy tính vào hệ thống mà không cần bổ sung thêm bất cứ thành phần nào. Về mặt vật lý các máy tính trong hệ thống có thể kết nối với nhau bằng hệ thống mạng Ethernet 10/100/1000Mbps. Hệ thống cho phép thay đổi nóng (hotswap) một hoặc vài máy tính khi hệ thống gặp sự cố mà không ảnh hưởng gì đến hoạt động tổng thể.

Hệ thống chia thành ba tầng chính theo như Hình1: 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 chia thành nhiều đơn vị độc lập, hoạt động theo kiểu chia sẻ tính toán hoặc dự trữ. Đơn vị khai thác dữ liệu được tích hợp cùng với phần lập chỉ mục cơ sở dữ liệu, cho phép người dùng sử dụng giao thức TCP/IP khai thác trên bất cứ hệ thống nào (Windows, Unix...)



Hình I.1. Sơ đồ hệ thống S.E của NetNam

Bằng việc chia hệ thống thành các khối chức năng phối hợp nhau thông qua Bộ điều phối, hệ thống có thể phân tán xử lý trên nhiều máy tính nhỏ. Nhờ đó mà lượng dữ liệu hệ thống có thể xử lý lên rất cao.

Phương thức lập chỉ mục của S.E NetNam

NetNam lập chỉ mục tất cả các từ trong tài liệu, 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. Khi tìm kiếm có thể dùng thẻ Meta để tăng cơ hội tìm kiếm; đưa ra các miêu tả riêng để hiển thị kết quả tìm kiếm.

Cú pháp tìm kiếm của S.E NetNam

Từ và cụm từ: S.E NetNam định nghĩa một từ như một chuỗi những chữ cái và con số được tách rời nhau.

Phần mềm tìm kiếm sẽ chỉ ra tất cả các từ mà nó tìm được trong một trang tài liệu web mà không quan tâm đến việc từ đó có trong từ điển hay đánh vần sai hay không.

Tìm kiếm cụm từ: Để tìm được một cụm từ, dùng ngoặc kép ở đầu và cuối cụm từ. Cụm từ đảm bảo rằng S.E NetNam 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 S.E NetNam sẽ bỏ qua mọi chấm câu trừ trường hợp 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, 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 do người dùng gõ vào.

- + Một yêu cầu bằng chữ thường sẽ có kết quả tìm kiếm không theo dạng chữ gõ vào. Ví dụ, nếu gõ chữ yết kiêu vào ô yêu cầu, S.E NetNam 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 quý vị điền **Yết Kiêu** vào ô yêu cầu, S.E NetNam 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**.

Sử dụng từ khoá để lọc các tìm kiếm

Cả giao diện của search engine đơn giản và nâng cao đều hỗ trợ việc sử dụng các từ khoá để hạn chế 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, 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 chỉ định
- + 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"

Các từ khoá có thể sử dụng trong việc tìm kiếm của NetNam: anchor:link; applet:class; domain:domainname; host:name; image:filename; link:URLtex; title: cụm từ; url: cụm từ

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 khoá 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

Thẻ title sẽ tìm kiếm nội dung tiêu đề của một tài liệu. Từ khoá tiêu đề sẽ giới hạn việc 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.

1.2.2 VINASEEK

VinaSeek là một S.E cho các web site tiếng Việt của Công ty Công nghệ Tin học Tinh Vân, cho phép tìm kiếm và hiển thị theo bất kỳ bảng mã nào. Cùng với khả năng xử lý tiếng Việt, VinaSeek còn có đầy đủ các tính năng của một công cụ tìm kiếm trên Internet như tính chính xác, đầy đủ, tính cập nhật cũng như tốc độ tìm kiếm. Các web site khác có thể dùng VinaSeek làm công

cụ tìm kiếm riêng cho mình. Chu kỳ tạo chỉ mục của VinaSeek là 5 ngày, thời gian tìm kiếm trung bình là 0.3 giây.

Hiện nay VinaSeek đổi tên thành UniVIS và đã được đóng gói nhằm mục tiêu phục vụ các hệ thống dữ liệu sử dụng tiếng Việt. UniVIS là hạt nhân của dịch vụ VinaSeek, nên có toàn bộ những tính năng ưu việt của dịch vụ VinaSeek. UniVIS có khả năng tạo chỉ mục cho hàng triệu văn bản các loại (HTML, XML, MS Word, PDF, RTF...) và các cơ sở dữ liệu lớn trên Oracle, MS SQL và DB2. Đặc biệt, UniVIS còn có khả năng tùy biến giao diện, dễ dàng cài đặt và quản trị. Quản trị mạng sẽ mất không đến 30 phút để cài đặt và cấu hình uniVIS tạo chỉ mục và tìm kiếm được mọi văn bản trên các website đã cài uniVIS.

1.3 NHẬN XÉT – SO SÁNH VỀ MỘT SỐ S.E.

1.3.1 SO SÁNH.

1.3.1.1 GIỐNG NHAU

Các S.E đều dùng một quy trình gồm ba giai đoạn: thu thập thông tin, tạo chỉ mục trên thông tin, tìm kiếm trên chỉ mục và tìm kiếm, sắp xếp kết quả. Nhưng mỗi search engine có giải pháp xử lý khác nhau nên có thể cho kết quả khác nhau.

Hiện nay ngày càng nhiều các S.E kết hợp dịch vụ thư mục web vào trong web site của họ. Những thư mục này tương tác với search engine chính (primary search engine) theo nhiều cách khác nhau. Ví dụ: như Excite, Terra Lycos, Alta Vista... không chỉ là một search engine. Đặc điểm chính của chúng có thể 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, gửi thư điện tử,

Trong việc phân tích từ khóa và tính độ phổ biến cũng có nhiều trường hợp đặc biệt cần xem xét, ví dụ như trong trường hợp chuỗi cần xử lý và tìm kiếm là “to be or not to be”, những S.E không tốt sẽ cho rằng chuỗi trên toàn là

những từ thông dụng không quan trọng để tính toán, và quá phổ biến. Để giải quyết những trường hợp như trên thì các S.E cung cấp giải pháp là dùng hai dấu nháy đôi để chứa chuỗi cần tìm, bắt buộc S.E tìm kiếm mọi cụm từ trong hai nháy kép.

Hiện nay các S.E cung cấp cơ chế tự động thêm toán tử “AND” vào giữa hai từ truy vấn. Kết quả tìm kiếm sẽ là những tài liệu phù hợp với toàn cụm từ tìm kiếm và sau đó là những kết quả phù hợp với từng từ trong cụm từ.

1.3.1.2 KHÁC NHAU

Yahoo lập chỉ mục tốt nhất. S.E dùng phần mềm con nhện này bò khắp nơi trên mạng, nhắm đến nhiều site khác nhau và theo mọi siêu liên kết trên từng trang để tạo chỉ mục. Chất lượng các chỉ mục thay đổi tùy theo chúng có thường xuyên được cập nhật hay không, bao lâu thì các trang web đã bị xóa khỏi site cũng bị xóa khỏi chỉ mục đó. Kết quả truy tìm có đúng là thứ ta cần hay không cũng còn tùy bởi lập chỉ mục bằng con nhện có thể đưa vào những metatag do các webmaster thêm vào, tiêu đề, từ khóa ngữ đoạn lấy từ các trang đó. Những yếu tố này đều có thể dẫn tới kết quả sai lạc, đặc biệt là do nhiều Webmaster lạm dụng chúng để dôn thông tin về web site của họ. Chính vì vậy mà yahoo, với diễn đàn site được tạo bởi con người và khả năng truy tìm mạnh theo từ khóa, thường tìm ra đúng những thứ người dùng tìm hơn.

Một điểm khác biệt lớn giữa các S.E là việc sắp xếp lại các kết quả tìm kiếm được. Các S.E sau khi tìm được những kết quả sẽ thực hiện tác vụ lọc bớt những kết quả trùng hay những kết quả có độ chính xác kém. Sắp xếp các kết quả này theo một trật tự nào đó, như theo độ chính xác của tài liệu.... Mỗi S.E có một cơ sở dữ liệu khác nhau và chiến lược xử lý kết quả khác nhau nên kết quả trả về cho người sử dụng cũng rất khác nhau.

1.3.2 NHẬN XÉT.

Mục tiêu của người dùng khi tìm kiếm là:

- ✚ Tìm ra tất cả các thông tin có liên quan: gọi là Perfect recall (độ gọi lại cao nhất), sao cho chúng không bị quá tải.
- ✚ Không nhận bất kỳ tài liệu nào không có liên quan: gọi là High Precision (độ chính xác cao nhất)

Hai độ đo trên mâu thuẫn với nhau. Perfect Recall có thể cho kết quả tìm kiếm là tất cả những gì có trên web. Nhưng còn precision thì là tối thiểu. Một trình duyệt phải dùng những phương thức nào đó để cực đại hoá độ chính xác của các kết quả trả về (bằng cách phân hạng kết quả) (Xem Chi tiết trong Phụ lục 5, 6,7)

Hầu hết các S.E lập chỉ mục “bằng tay” đều mang lại kết quả tốt hơn so với lập chỉ mục tự động. Nhìn chung, độ đo quan trọng nhất để đánh giá hiệu quả hoạt động của một S.E là chất lượng của kết quả tìm kiếm. Các kết quả hợp lý là các trang chất lượng cao, không có các liên kết bị gãy. Chi tiết xem Bảng sau:

Bảng I.1. Một Thí dụ về Kết quả tìm kiếm của Google


Query: bill clinton

<http://www.whitehouse.gov/>

100.00%  (no date) (0K)


<http://www.whitehouse.gov/>

[Office of the President](#)

99.67%  (Dec 23 1996) (2K)

http://www.whitehouse.gov/WH/EOP/OP/html/OP_Home.html

[Welcome To The White House](#)

99.98%  (Nov 09 1997) (5K)

<http://www.whitehouse.gov/WH/Welcome.html>

[Send Electronic Mail to the President](#)

99.86%  (Jul 14 1997) (5K)

http://www.whitehouse.gov/WH/Mail/html/Mail_President.html

<mailto:president@whitehouse.gov>

99.98% 

<mailto:President@whitehouse.gov>


99.27% 

[The "Unofficial" Bill Clinton](#)

94.06%  (Nov 11 1997) (14K)


<http://zpub.com/un/un-bc.html>

[Bill Clinton Meets The Shrinks](#)

86.27%  (Jun 29 1997) (63K)

<http://zpub.com/un/un-bc9.html>

[President Bill Clinton - The Dark Side](#)

97.27%  (Nov 10 1997) (15K)

<http://www.realchange.org/clinton.htm>

[\\$3 Bill Clinton](#)

94.73%  (no date) (4K)

<http://www.gatewy.net/~tjohnson/clinton1.html>

Ngoài chất lượng tìm kiếm, một khía cạnh của yêu cầu lưu trữ cần quan tâm là phải sử dụng hiệu quả bộ nhớ. Bảng 2. trình bày một số thống kê và một số yêu cầu lưu trữ của Google.

Bảng 2. Thống kê về dung lượng lưu trữ	
Tổng dung lượng các trang web tìm được	147.8 GB
Kho dữ liệu nén	53.5 GB
Chỉ mục nghịch đảo có thứ tự	4.1 GB
Chỉ mục nghịch đảo ban đầu	37.2 GB
Từ điển	293 MB
Dữ liệu neo (anchor) tạm thời	6.6 GB
Document Index Incl. Variable Width Data	9.7 GB
Cơ sở dữ liệu các liên kết	3.9 GB
Tổng dung lượng không kể kho lưu trữ	55.2 GB
Tổng dung lượng kể cả kho lưu trữ	108.7 GB

Điều quan trọng nhất của một S.E là hiệu quả dò tìm và lập chỉ mục. Các thông tin này có thể lưu giữ đến một hạn (date) và các thay đổi chủ yếu đến hệ thống có thể được kiểm tra một cách tương đối nhanh chóng. Trong Google, hoạt động chính là dò tìm, lập chỉ mục và sắp xếp. Thật khó để biết bao lâu thì dò tìm hoàn thực hiện hoàn tất, vì nếu đĩa bị đầy, hay các sự cố khác thì hệ thống sẽ bị ngừng hoạt động. Trong 9 ngày, lấy được 26 triệu trang web (gồm cả lỗi). Tuy nhiên, nếu hệ thống hoạt động êm xuôi thì nó chạy nhanh hơn và download khoảng 11 triệu trang chỉ trong 63 giờ, trung bình chỉ hơn 4 triệu trang mỗi ngày hay 48,5 trang mỗi giây. Google có thể chạy bộ lập chỉ mục và bộ dò tìm đồng thời. Bộ lập chỉ mục có thể chạy nhanh hơn các bộ dò tìm, điều này có được là do bộ lập chỉ mục có đủ thời gian để tối ưu và không bị tình trạng thất cổ chai. Các tối ưu này nhờ việc cập nhật rất lớn cho

chỉ mục tài liệu và việc thay thế các cấu trúc dữ liệu quan trọng trên đĩa cục bộ. Bộ lập chỉ mục thực hiện khoảng 54 trang trên mỗi giây. Các bộ sắp xếp có thể thực hiện hoàn tất đồng thời; sử dụng 4 máy, thực hiện xử lý sắp xếp mất khoảng 24 giờ.

Bảng 0. Phân tích số lượng các trang Web	
Các trang web tìm được	24 million
Các URL tìm thấy	76.5 million
Các địa chỉ mail tìm thấy	1.7 million
Số lượng các lỗi 404's	1.6 million

Phiên bản hiện nay của Google trả lời hầu hết các truy vấn từ 1 đến 10 giây. Thời gian này hầu như bị chi phối bởi vào/ra đĩa trên NFS (vì các đĩa được trải trên nhiều máy). Ngoài ra, Google không có bất kỳ sự tối ưu về cache truy vấn, phân nhỏ lập chỉ mục trên các thuật ngữ chung, và các tối ưu hoá chung khác. Để nâng cao tốc độ của Google người ta đang xem xét việc phân tán phần cứng và phần mềm và cải tiến thuật toán. Mục đích cuối cùng là có thể đáp ứng hàng trăm các truy vấn khác nhau trong một giây. Bảng 4. nói lên thời gian truy vấn trên phiên bản hiện nay của Google.

Bảng 4. Thống kê thời gian tìm kiếm				
	<i>1.3.2.1.1.1.1 Initial Query</i>		Same Query Repeated (IO mostly cached)	
Query	CPU Time(s)	Total Time(s)	CPU Time(s)	Total Time(s)
al gore	0.09	2.13	0.06	0.06
vice president	1.77	3.84	1.66	1.80
hard disks	0.25	4.86	0.20	0.24
search engine	1.31	9.63	1.16	1.16

2 PHẦN 2:

XÂY DỰNG TỪ ĐIỂN NGỮ NGHĨA THUẬT NGỮ TIN HỌC

2.1 TÌM KIẾM THEO NGỮ NGHĨA

Tìm kiếm theo ngữ nghĩa là tìm đúng theo nghĩa mình mong muốn trong số những nghĩa của từ mình muốn truy vấn.

Ví dụ:

với từ khóa tìm kiếm là: “cò” (theo nghĩa: con cò) thì kết quả tìm kiếm có thể là: “Miền Tây Nam bộ có một số vườn cò rất lớn.”.

Tuy nhiên không phải lúc nào từ “cò” cũng có nghĩa con cò cho nên những trường hợp sau sẽ không là kết quả của quá trình tìm kiếm trên:

“Khẩu súng đã cướp cò khi anh ấy sửa chữa.”

“Những tay cò mỗi có rất nhiều mảnh khoé trong làm ăn kinh tế.”

Bên cạnh đó tìm kiếm theo ngữ nghĩa còn là tìm những từ có ngữ nghĩa liên quan chứ không đơn thuần là tìm chính xác nghĩa. Trong một số trường hợp tìm đúng nghĩa của từ sẽ có kết quả hạn chế và không có tính ứng dụng cao.

Ví dụ:

Sau đây là một kết quả có thể có của quá trình tìm kiếm trên: “Sếu cổ đỏ là một loài chim quý”.

Vì lý do sếu là một từ có cùng nguồn gốc với cò (theo nghĩa con cò).

Biểu diễn ngữ nghĩa có thể xem như một bài toán con của biểu diễn tri thức. Trong những phần sau, chúng tôi đề cập đến các dạng quan hệ ngữ nghĩa khác nhau (2), cũng như cách chúng được tổ chức thành hệ thống trong các hệ biểu diễn ngữ nghĩa hiện có (3), phần (4) trình bày về WordNet, một từ điển ngữ nghĩa hoàn chỉnh nhất hiện nay, phần (5) trình bày sơ lược về ontology, lý

thuyết chung cho các hệ thống biểu diễn ngữ nghĩa. Tiếp theo là các chi tiết kỹ thuật của quá trình thực hiện đề án và báo cáo kết quả của đề án (6).

2.2 BIỂU DIỄN NGỮ NGHĨA

2.2.1 ĐỒNG HIỆN (CO-OCCURRENCE)

Trong văn bản, sự xuất hiện của các từ đều có quan hệ mật thiết với nhau theo một ngữ nghĩa nào đó nhằm để diễn tả một ngữ cảnh xác định. Do đó có những từ luôn đi cùng với nhau (đồng hiện) và mang một nghĩa xác định và ngược lại.

Ví dụ:

trong văn bản có chứa từ “plant”, “factory”, “worker” thì nói chung từ “plant” có nghĩa là nhà máy nhưng nếu văn bản có chứa các từ “plant”, “tree”, “orange” thì khi đó từ “plant” có nghĩa là thực vật.

Việc xác định các quan hệ đồng hiện này dựa trên việc thống kê trên một tập ngữ liệu lớn nhằm bao quát được các ngữ cảnh khác nhau của các từ để đảm bảo các quan hệ đồng hiện này luôn đúng trong mọi trường hợp.

Đây là hệ thống quan hệ được phát sinh qua phân tích ngữ liệu.

- ▶ network ----- network protocol
- ▶ network ----- node
- ▶ LAN server ----- central mass storage
- ▶ LAN server ----- network server
- ▶ LAN server ----- server
- ▶ LAN server ----- workstation
- ▶ License ----- Copyright
- ▶ License ----- Portions Copyright
- ▶ License ----- software licence

2.2.2 HỆ THỐNG QUAN HỆ ĐỒNG NGHĨA ĐƠN GIẢN

Từ điển LDOCE và LLOCE (Longman Dictionary of Contemporary English và Longman Lexicon of Contemporary English) đã được sử dụng rộng rãi để rút trích từ vựng cho xử lý ngôn ngữ tự nhiên và được sử dụng như là một dạng từ điển máy tính có thể đọc được (machine-readable dictionary – MRD). Tổ chức và tạo dựng chúng dựa trên phương pháp truyền thống để tạo ra từ điển. Nhưng một số đặc điểm đã làm cho chúng đặc biệt phù hợp cho việc tìm kiếm từ vựng cho xử lý ngôn ngữ tự nhiên.

LDOCE

LDOCE(Longman Dictionary of Contemporary English) là một từ điển mà máy có thể đọc được có kích thước trung bình khoảng 45.000 mục từ và 75.000 nghĩa. Các mục từ được phân biệt dựa trên nguồn gốc của từ và từ loại của chúng mà mỗi mục từ có thể có một hoặc nhiều mục nghĩa. Nghĩa của từ được phân biệt dựa trên từ loại của chúng.

LDOCE được tổ chức theo ngữ nghĩa ở dạng phân cấp. Gồm 32 mã ngữ nghĩa khác nhau được sử dụng trong LDOCE: Một sự phân biệt được tạo ra giữa 19 mã cơ bản và 13 mã nối kết của những mã căn bản đó.

A (animal): thú vật

B(female animal): thú vật giống cái

C(concrete): cụ thể

D(male animal): thú vật giống đực

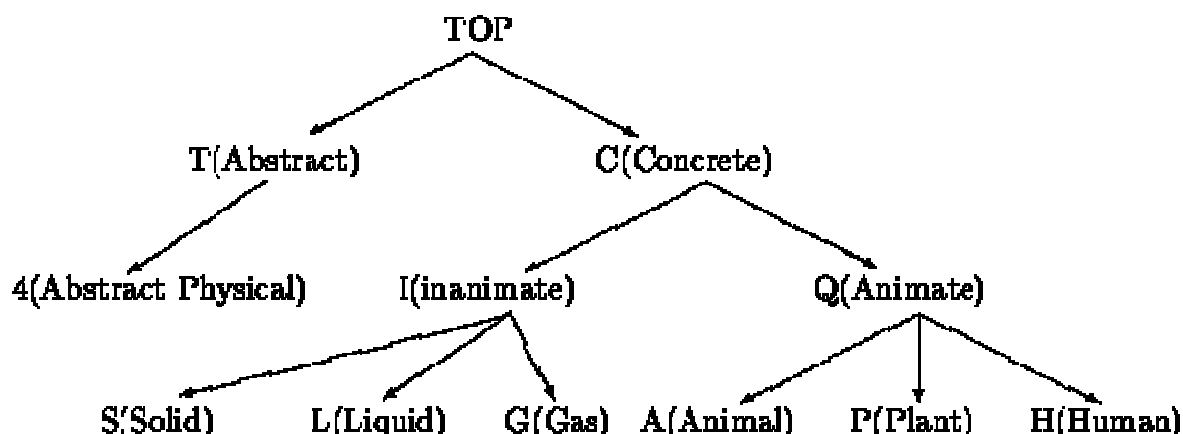
E (chất rắn hay chất lỏng (không phải thể khí))

F (female human): phái nữ

H (human): con người

.....

Những mã cơ bản này được tổ chức thành cây phân cấp:



Hình II.1: cây phân cấp của LDOCE

Hầu hết các nghĩa của danh từ có một mã ngữ nghĩa. Đối với một số danh từ có nhiều mã thì các mã này là cơ sở để phân loại nghĩa. Tuy nhiên đối với một số động từ và tính từ, những mã này cho biết giới hạn sử dụng của đối số.

LLOCE

LLOCE (Longman Lexicon of Contemporary English) là một từ điển LDOCE và được tổ chức lại dựa trên nguyên tắc ngữ nghĩa. Sự phân lớp trong LLOCE được phân thành 3 cấp theo mức độ cụ thể của khái niệm tăng dần: 14 phân lớp → 127 nhóm → 2441 tập hợp. Các tập hợp gồm các từ có liên quan mà không cần phải đồng nghĩa. Mỗi quan hệ liên quan ở đây được xét trên mặt ngữ nghĩa.

Ví dụ:

<MAJOR: A> *Life and living things*

|
|

<GROUP: A50-61> *Animals/Mammals*

|
|

<SET: A53> *The cat and similar animals: cat, leopard, lion, tiger,...*

Mỗi mục trong từ điển được gắn kết với một tập hợp các mã, ví dụ như sau:

```
<SET: A53> nouns The cat and similar animals
-----
cat 1 a small domestic [=> A36] animal ...
    2 any animal of a group ...
...
panther [Wn1] 1 a leopard ...
           2 AmE cougar.
...

<SET: A53> nouns The dog and similar animals
-----
dog a domestic animal with a coat of hair ...
```

Một phần từ điển LLOCE

*A1#tồn tại,sống,hiện có,hiện hữu,tạo ra,sáng tạo,tạo nên,tạo
thành,làm sống động,cổ vũ,tạo sự sống,làm sinh động,làm sôi
nổi,làm phấn khởi,đầy sức sống,có sinh khí,nhộn nhịp,náo
nhiệt,sôi nổi,hoạt hình,làm nảy sinh ra,tích cực,ủng hộ,động
viên,làm vui lên,làm hăng lên,làm náo nhiệt,làm nhộn nhịp,
A10#thủ tiêu,giết chết,cho về chầu,khử,khử đi,giết,hủy hoại,kết
liễu,hạ,đánh quỵ,làm chết,diệt,tiêu diệt,
A100#cá thuộc họ cá trể,cá nheo,cá êfin,cá vược,cá tuyết,cá
moruy,cs tuyết to đầu,cá chình,lươn,cá trích,cá bẹ,cá thu,cá
rutilut,cá dầy,cá đuối,cá cơm,cá cơm biển,cá chim,cá bơn vĩ,cá
chép,cá đối,cá phèn,cá chó,cá dưa răng nhọn,cá bơn sao,cá
bơn,cá hồi,cá sac-đin,cá bơn cát,cá biển nhỏ màu xám bạc,cá hét,
A101#giống cá họ cá mập,cá đuối,cá nhám,cá mập,
A102#cua,ốc mượn hồn,tôm ký cư,tôm hùm,tôm sông,tôm,tôm
càng,tôm pandan,tôm he,
A103#hàu,sò,traí,vẹm,ốc buxin,ốc tù và,ốc hương,ốc mút,bạch
tuộc,mực phủ,mực ống,
A104#sứa,sao biển,nhím biển,bọt biển,miếng bọt biển,san hô,
A11#chết người,chỉ tử,gây chết người,như chết rồi,tai hại,gây
chết,giết chết người,nguy đến tính mạng,phải chết,làm chết
người,như chết,đã chết,trí mạng,gây tai họa,*

A110#ong,ong bắp cày,bọ cánh cứng,bọ sừng
hươu,gián,muối,kiến,châu chấu,châu chấu voi,bướm đêm,sâu
bướm,ngài,chuồn chuồn,ruồi,ruồi nhà,xâu tai,bọ râu tai,bọ
chết,ong nghệ,ong vò vẽ,đế,ve sầu,ve,mòng,ruồi trâu,rệp rừng,sâu
cây,rận,chấy,rệp,

A111#trứng,ấu trùng,sâu bướm,nhộng,sâu,ruồi,ruồi
nhà,bướm,giòi,

A112#nhện,bọ cạp,

A113#giun đất,ốc sên,sên trần,giun,sâu,trùng,đĩa,

A12#bắt tử,bắt tận,vô hạn,bắt diệt,sự bắt tử,trường sinh bắt tử,sự
bất hủ,sự lưu danh,đặc biệt,vô cùng,hết sức,rộng lớn,bất hủ,sống
mãi,mãi mãi,không chết,vĩnh cửu,sống mãi đời đời,không thể tiêu
diệt được,

A120#đầu,cổ,mắt,tai,cổ họng,

A121#sừng,gạc,bờm,mào,chỏm lông mào,mào gà,tóc
mai,râu,ria,xúc tu,râu sờ,tua,anten,lông,

A122#mũi,miệng,mồm,mỗm,vòi,vòi voi,rọ mồm,mặt,đầu,mỏ
chim,mỏ,diều,hàm dưới,càng,vòi con voi,mang,mang cá,yếm,cổ
họng,

A123#răng,răng nanh,răng nọc,ngà,ngà voi,

A124#chân,cẳng,bàn chân,ngón chân,móng guốc,chân có
vú,gang bàn chân,vuốt,càng,móng,màng da chân,giác,xương
ống chân,

A125#cánh,đuôi,đuôi chồn,đuôi cáo,vây cá,chân chèo,

A126#da,bộ da lông,da sống,da động vật,tám da sống,bộ lông
mao,lông tóc,tóc,lông,bộ lông tóc,lông cứng,lông tóc cứng,bộ da
lông con vật,bộ lông cừu,lông cừu,lông vũ,bộ lông,bộ lông
chim,vỏ,bao,mai,vẩy,vẩy,lông gai,ngạnh,gai,ống lông,lông gai
cứng,túi,ống,màng bọc,

A127#vết,đốm,chấm,đốm tròn,sọc,vằn,dấu,viền,điểm,lỗ thở,đường
khía,

A128#bầu vú,đầu vú,núm vú,vú,

.....

B1#thể xác,thân thể,thân xác,vật thể,thể hình,vóc người,dáng
người,tâm vóc,khổ người,thể trạng,thể chất,thân hình,ngoại
hình,thân,tạng người,

B10#đầu,vòm họng,vòm miệng,khẩu cái,lợi,răng,miệng,môi,lưỡi
gà,lưỡi,họng,cuống họng,trái cổ,tóc,lông,thái dương,lông
mày,lông mi,sống mũi,lỗ tai,lỗ mũi,hàm,quai hàm,đỉnh
đầu,trán,mắt,mũi,má,cằm,cổ,tay,cánh tay,vai,nách,cánh tay
trên,cơ hai đầu,nhượng tay,chỗ tay gấp,khuỷu tay,cẳng tay,cổ
tay,nắm tay,chân cẳng,mông đít,đùi,bắp đùi,đầu gối,bắp
chân,cẳng chân,mắt cá chân,gót chân,gáy,thân,ngực,vú,núm
vú,đầu vú,dạ dày,bụng,rốn,sườn,eo,hông,háng,cơ quan sinh dục,bộ
lông,cái kẹp,hầu,cổ họng,mỏ,đũa,

B100#béo,mập,mập mạp,béo lẳn,tròn trĩnh,phúng phính,béo
tròn,mũm mĩm,múp mịch,múp,múp máp,bệu nhũn,nhảo,nhẽo
nhèo,béo phì,béo phệ,phệ,quá béo,quá nặng cân,mập tròn,giết thịt
được rồi,béo phì,phục phịch,phình phính,mềm,nhũn,bụ bẫm,đầy
đặn,quá nặng,

B101#mảnh khảnh,gầy,gầy còm,chắc người,thon thả,không
béo,mảnh dẻ,nhỏ bé,nhẹ cân,gầy nhom,da bọc xương,gầy mòn,hóc
hác,rất gầy,không có nhiều thịt,mảnh mai,thon,mỏng mảnh,yết
ớt,không to dày,mỏng manh,dễ vỡ,hấp dẫn,

B102#tăng cân,mập ra,béo ra,làm béo ra,vỗ béo,tròn ra,lên
cân,nặng lên,

B103#sút cân,gầy đi,nhẹ cân đi,ăn kiêng,trở nên thon nhỏ,giảm
cân,thon gọn đi nhiều,gầy mòn,ốm đi,sút cân,sút cân dần dần,trở
nên mảnh khảnh,bớt nặng đi,

B104#đầy đà,béo tốt,múp máp,trông đầy đà,có bộ ngực to,có vú
to,béo,bệ vệ,

B105#chắc nịch,vạm vỡ,to khỏe,mập,chắc,chắc mập,béo lùn,lùn,

B106#gầy nhom,cao lêu nghêu,có cẳng dài,xương xẩu,khắc khiêu,

B11#sọ,hộp sọ,đầu lâu,xương hàm dưới,xương bả vai,xương cánh
tay,xương sống,xương cột sống,xương quay,xương trụ,xương cổ
tay,xương đốt ngón tay,xương đốt ngón chân,xương bánh
chè,xương chày,xương cổ chân,xương quay xanh,xương ức,đốt
xương sống,khung chậu,xương cụt,xương lòng bàn tay,xương
đùi,xương mác,xương bàn chân,bộ xương,mô hình bộ xương,hình
đầu lâu,hình hộp sọ,xương sườn,khớp xương,cột sống,

B110#khỏe mạnh,mạnh khỏe,sung sức,khỏe,khỏe hơn,dễ chịu
hơn,khoẻ mạnh,tốt,không làm sao,cường tráng,tốt cho sức khỏe,có
lợi cho sức khỏe,được,mạnh mẽ,tráng kiện,không bệnh tật,dư sức,

B111#đau yếu, ốm, bệnh, không được khỏe, hay ốm đau, có thể không
có lợi cho sức khỏe, không lành mạnh, có vẻ ốm yếu, không
khỏe, không đủ sức khỏe, sức khỏe tồi, ốm yếu, đau, luôn đau
yếu, thường xuyên ốm đau, đau ốm, bị bệnh, hơi bị đau, yếu, không
khỏe mạnh, khó ở, ốm đau, kiệt sức, có liên quan đến bệnh, cho thấy
là có bệnh, choáng váng, hơi mệt, hay mất trí, cảm thấy không
khoẻ, mệt rã rời, không khoẻ, khó chịu, sắp chết, suy nhược, làm sa
sút, làm suy nhược, làm kiệt sức, làm mệt lử, mệt lử, bệnh tật,
B112#sức khỏe, tình trạng sức khỏe, trạng thái khỏe mạnh, tình
trạng sung sức, trạng thái khoẻ tốt, hạnh phúc,
B113#sự đau yếu, trình trạng đau ốm, loại bệnh, bệnh, sức khỏe
kém, ốm yếu, rối loạn, bệnh tật, các bệnh nói chung, bệnh hoạn, tình
trạng rối loạn, sự ốm đau bệnh tật, sự khó ở, se mình, sự ốm yếu tàn
tật, sự suy nhược thần kinh, sự tàn tật, điều cản trở, điều bất lợi, có
vấn đề, khó ở, tình trạng, không khoẻ, khó chịu, hơi mệt, cảm thấy
không khoẻ, cảm thấy chán nản,
B114#đau đớn bởi, bị ốm, mắc bệnh, bị ốm đột xuất, bệnh, ốm, có
triệu chứng ốm, khó chịu trong người, bắt đầu ốm, cảm thấy ốm,
B115#ngất, choáng, bất bình tĩnh, ngất đi, ngủ say, lịm đi,
B116#sự lên cơn bệnh, lên cơn bệnh, cơn bệnh bất ngờ, cơn bệnh, cú
sốc, sốc, đột quỵ,
B117#hỗn hển, thở gấp, vừa nói, vừa thở hỗn hển, hành động thở hỗn
hển, tiếng nói thở hỗn hển, đập thình thình, hơi thở phò, hành động
thở phì phò, tiếng thở phì phò, khịt mũi, hành động khịt mũi, tiếng
khịt mũi, ho, hắt hơi, hành động hắt hơi, tiếng hắt hơi, khụt khịt, người
ngủ, đánh hơi, sụt sịt, nấc cục, hành động nấc cục, tiếng nấc cục, ợ, sự
ợ, phun, huyết sáo, động tác huyết sáo, khò khè, hành động thở khò
khè, tiếng thở khò khè, thở dài, sự thở dài, tiếng thở dài, ngáy, sự
ngáy, tiếng ngáy, đánh rắm, sự đánh rắm, tiếng đánh rắm, địt, vừa
ho, ho mà khạc ra, hít, khịt khịt, sổ mũi,
B118#nghẹt thở, tức thở, nghẹn, làm ngạt thở, chết ngạt, làm chết
ngạt, gây ngạt, quá trình gây ngạt, sự gây ngạt, thắt cổ, bóp nghẹt,
B119#nôn, mửa, buồn nôn, nôn mửa, nôn ra, mửa ra, nôn khan, ọe, ọe
ra, ói, ói ra, thổ ra, tống, phun ra, phụt ra, ứa, ợ ra,
B120#sự buồn nôn, trạng thái gây nôn, cảm thấy buồn nôn, chứng
say sóng, chứng say xe, buồn nôn khi đi máy bay,

B121#sự đau,rất đau đớn,đau,đau nhức,cơn đau nhói,cơn đau
thót,sự đau nhói,sự còn cào,ngồi đốt,kim đốt,chất châm ngứa,chỗ
đốt,chỗ châm,

B122#gây đau,làm đau,đau nhức,sốt ruột,làm cho đau,thấy
đau,làm đau nhói,bó chặt,làm tức,

B123#đau đớn,làm đau,đau,không đau,hơi đau khi sờ vào,dễ
đau,nhạy cảm,buồn phiền,làm đau đớn,hành hạ,tính khốc liệt,dữ
đội,ngghiêm trọng,nặng,đau cấp tính,đau buốt,thình lình,đột
ngột,nhức nhói,buốt,nhói,sự đau đớn,đau âm ỉ,đau nhói,đau mãn
tính,ngghiện,đánh như tử,đau buồn,ngao ngán,bạc đãi,ngược
đãi,làm khổ,đầy đọa,đau khổ,làm đau khổ,thảm thía,

B124#sưng lên,sưng,phồng,căng,tấy,mưng,viêm,sưng tấy,nhiễm
trùng,mưng mủ,bị mưng mủ,làm giộp lên,bị giộp,bị phỏng,

B125#sự sưng lên,sự bị sưng,phồng,viêm,sự đau,viêm khớp,

B126#vết lóm đóm,vết,đóm,sự phồng ra,chỗ sưng,mụn nhọt,chỗ
sần,mụn mủ,có nhiều mụn,nhọt,nốt đậu mùa,mụn cóc,chỗ bỏng
giộp,chỗ giộp,cước,nốt viêm tấy ở kẽ ngón,chỗ sưng tấy,sưng
tấy,vết chai,chai chân,chỗ sưng u lên,chỗ u lên,chỗ sưng vù
lên,dấu,lỗ thở,

B127#chỗ vết thương,lở loét,chỗ loét,loét,vết loét,khỏi u,u,ung
thư,sinh ra ung thư,u bứu,bứu,u nang,mủ,áp xe,bệnh thối
hoại,hoại thư,

B128#vảy,sẹo đậu mùa,rỗ,vết rỗ,có các vết rỗ,mặt rỗ,vết
nhơ,tật,bớt,vết chàm,vết bớt,vết sẹo,để lại sẹo,có sẹo,sẹo rỗ,

B129#vết thương,làm bị thương,gây chấn thương,thương tật,làm
tàn tật,bị thương thành tật,đánh thâm tím,cào xé thịt làm bị
thương,đối xử thô bạo,đối xử khó chịu,làm tổn thương,chỗ bị
thương,việc bị thương,làm biến dạng,méo mó,xấu đi,biến dạng
đi,sự làm xấu đi,bị làm xấu đi,làm què,tàn tật,làm què quặt,việc bị
tàn tật,bị tàn tật,đánh đập tàn tẽ,phá hỏng,làm mất khả năng hoạt
động,loại ra khỏi vòng chiến đấu,làm hại,

B130#chém,mổ,cắt,chặt,chỗ cắt,chỗ bị cắt,cắn,vết cắn,khía,cắt,cắt
nắc,làm đứt,khía vào,vết cắt,vết chém,vết rạch,gây ra các vết
cắt,gây ra các vết chém,rạch,cắt khía,vết cứa,gây ra các vết
cứa,cào,làm xước da,vết cào,quào,cào xé,lôi,đâm bằng dao,nhát
đâm,đâm,đường rạch,cắt đứt,kết liễu đột ngột,làm chết đột

ngột,cắn đứt ra,đón,chặt đi,gặt,sự cắt,sự chém,sự
rach,xé,gọt,thái,vết khắc,vết nạo,vết cưa,cưa,vết xước,
B131#đập,đánh,đánh đập,đánh mạnh,cú đòn,cú đánh,đòn,vết thâm
tím,vết bầm,gây ra vết bầm,đánh vỡ đầu,truy lũng,đuổi bắt,tuyển
mộ,quất,hành hạ,đụng tới,sờ tới,hành hung,đắm tới tấp,lật đổ,đánh
gục,giáng đòn,đuổi,va đụng,đắm,vỗ,cú,
B132#làm gãy,chỗ bị gãy,rạn,bị gãy,chỗ gãy xương,bong gân,làm
bong gân,sự bong gân,làm căng ra,làm căng thẳng,gắng quá
mức,sự căng thẳng quá mức,kéo giãn,làm to ra,
B133#đui,mù,người mù,làm cho mù,làm chói mắt,cận thị,viễn
thị,điếc,lãng tai,nặng tai,những người điếc,câm,không thể nói
rõ,hoàn toàn không thể nói,im lặng,không nói,người bị làm cho
câm,què,khập khiễng,bị đui,lòa,nhìn xa thấy rộng,không làm tiếng
động,một cách im lặng,
B134#già yếu,lụ khụ,hom hem,yếu đuối,ốm yếu,suy yếu vì tuổi
già,lão suy,nhút nhát,có chân đất sét,dễ bị lật đổ,ở thế không
vững,hèn nhát,
B135#nhiễm trùng,nhiễm bệnh,toả ra gây ô nhiễm,sự nhiễm
trùng,sự nhiễm bệnh,bệnh truyền nhiễm,sự lây nhiễm,truyền cho
người khác,bệnh lây,sự lây,làm nhiễm bệnh,làm bẩn,làm ô uế,làm
nhiễm,sự nhiễm,sự làm bẩn,sự làm nhiễm bệnh,vật làm lây
nhiễm,chất gây nhiễm bệnh,ô nhiễm,gây ô nhiễm,sự ô nhiễm,quá
trình ô nhiễm,sự gây ô nhiễm,vật làm ô nhiễm,tác nhân gây ô
nhiễm,làm nhơ,
B136#lây nhiễm,bị nhiễm khuẩn,gây ra nhiễm khuẩn,truyền
nhiễm,hay lây,lây,miễn dịch,khả năng miễn dịch,
B137#sự bộc phát,sự bùng nổ,bệch dịch,bệch dịch hạch,dịch,tai
họa,bệnh dịch hạch,thuộc về một bệnh dịch,giống như một bệnh
dịch,nhiễm khuẩn cao,
B140#cảm lạnh,bệch cúm,cúm,bệnh cúm,chứng chảy,viêm xổ,viêm
xổ mũi,chứng ho,ón lạnh,rét run,bệnh sốt,sốt,
B141#viêm phổi,bệnh viêm cuống phổi,viêm phế quản,bệnh
lao,lao,bệnh lao phổi,lao phổi,bệnh hen,suyễn,hen,bị hen,liên quan
đến hen,người bị hen,người bị suyễn,
B142#triệu chứng sốt,hơi sốt,có sốt,thấy sốt,sốt,lên cơn sốt,bị
sốt,về sốt,do sốt gây ra,bị đổ bưng lên,đổ ửng,nóng,nóng sốt,

2.2.3 HỆ THỐNG PHÂN CẤP NGŨ NGHĨA WORDNET

2.2.3.1 GIỚI THIỆU WORDNET

WordNet là một cơ sở dữ liệu tri thức từ vựng học được thiết kế dựa trên những lý thuyết về ngôn ngữ tâm lý theo cách liên tưởng từ ngữ của con người. WordNet được tổ chức dựa theo các quan hệ ngữ nghĩa bởi vì một quan hệ ngữ nghĩa là một quan hệ giữa các nghĩa và các nghĩa có thể được đại diện bởi nhiều synset. Và chúng ta có thể xem những quan hệ ngữ nghĩa như là những con trỏ giữa các synset. Đó là đặc tính của quan hệ ngữ nghĩa và chúng có tác động qua lại với nhau: Nếu có một quan hệ ngữ nghĩa R giữa nghĩa (x, x', ...) và (y, y', ...) thì cũng có một quan hệ R' giữa (y, y', ...) và (x, x', ...).

Một từ bất kỳ có thể có nhiều nghĩa (word meaning) và khi đó mỗi nghĩa của nó sẽ thuộc vào những tập đồng nghĩa khác nhau. Ngược lại mỗi tập đồng nghĩa lại có thể chứa một hoặc nhiều hơn một từ khác nhau. Xét ví dụ sau.

Ví dụ:

Khi tìm từ **letter** trong WordNet ta sẽ được kết quả như sau

- *the noun letter has 4 senses :*
 - i. *letter, missive : a written message addressed to a person or organization; "wrote an indignant letter to the editor".*
 - ii. *letter, letter of the alphabet, alphabetic character : the conventional characters of the alphabet used to represent speech; "his grandmother taught him his letters".*
 - iii. *letter : a strictly literal interpretation (as distinct from the intention); "he followed instructions to the letter"; "he obeyed the letter of the law".*
 - iv. *letter, varsity letter : an award earned by participation in a school sport; "he won letters in three sports".*
- (Trong WordNet danh từ **letter** có 4 nghĩa thuộc vào 4 tập đồng nghĩa)
 - i) *Tập đồng nghĩa thứ nhất gồm: letter, missive với nghĩa tiếng Việt tương ứng là "lá thư", "thư tín".*

- ii) *Tập đồng nghĩa thứ hai gồm: letter, letter of the alphabet, alphabetic character với nghĩa tiếng Việt tương ứng “kỳ tự”, “chữ” hay “chữ cái”.*
- iii) *Tập thứ ba chỉ gồm một từ: letter với nghĩa tiếng Việt là “nghĩa chặt hẹp”, “nghĩa mặt chữ”.*
- iv) *Tập cuối cùng gồm hai từ: letter, varsity letter với nghĩa tiếng Việt tương ứng là “huy hiệu”, “danh hiệu” tặng cho những sinh viên có thành tích thể thao đặc biệt ở trường*

2.2.3.2 CÁC LOẠI QUAN HỆ TRONG WORDNET

Như vừa trình bày ở trên, các từ trong WordNet được sắp xếp vào thành các tập đồng nghĩa. Và giữa các tập đồng nghĩa này có thể mang các mối quan hệ ngữ nghĩa với nhau. Phần sau đây sẽ trình bày các quan hệ được xây dựng bên trong WordNet.

2.2.3.2.1 QUAN HỆ ĐỒNG NGHĨA (SYNONYMY)

Các tập đồng nghĩa được gọi là có quan hệ đồng nghĩa với nhau khi chúng có thể thay thế cho nhau trong một số ngữ cảnh nào đó. Vì thế WordNet đã được chia thành nhóm danh từ (noun), động từ (verb), tính từ (adjective), và trạng từ (adverb). Và những mối quan hệ đồng nghĩa chỉ tồn tại giữa các tập đồng nghĩa ở cùng dạng từ loại. Điều này cũng thật dễ hiểu, bởi vì các danh từ sẽ diễn tả những khái niệm thuộc về danh từ, động từ thì diễn tả những khái niệm chỉ hành động, còn tính từ và trạng từ thì giúp ta có thể diễn tả mức độ của những khái niệm trên.

2.2.3.2.2 QUAN HỆ TRÁI NGHĨA (ANTONYMY)

Một từ trái nghĩa của từ x thông thường sẽ là not-x, nhưng không phải lúc nào cũng đúng như vậy. Chẳng hạn, ta có từ rich (giàu) và poor (nghèo) là hai từ trái nghĩa, nhưng ta không thể nói rằng một người không giàu là một người nghèo.

Quan hệ trái nghĩa là một quan hệ giữa các từ với nhau chứ không phải là quan hệ giữa các nghĩa của từ với nhau.

Ví dụ: nghĩa của hai cụm từ {raise, ascend} và {fall, descend} có hai ý trái ngược nhau. {raise, ascend} có nghĩa là “tăng lên” và {fall, descend} có nghĩa là “giảm xuống”. Nhưng chỉ có {raise, fall} và {ascend, descend} là những cặp trái nghĩa với nhau và không có quan hệ trái nghĩa giữa raise và descend hay giữa fall và ascend.

2.2.3.2.3 QUAN HỆ CẤP BẬC (HYPONYM / HYPERNYM)

Ngược với quan hệ đồng nghĩa và trái nghĩa là các quan hệ giữa các từ với nhau, quan hệ cấp bậc là quan hệ giữa các nghĩa của từ.

Ví dụ:

maple(cây thích) là một hyponym của tree(cây).

tree(cây) là một hyponym của plant(thực vật).

Có thể hiểu hyponym/hypernym(nghĩa con/nghĩa cha) là một loại quan hệ theo kiểu ISA(là một). Một ý niệm tương ứng với synset {x, x, ...} được gọi là một hyponym của ý niệm tương ứng với synset {y, y, ...} khi chúng ta có thể nói x là một (một dạng của) y - an x is (a kind of) y.

Một nghĩa con nghĩa con (hyponym) kế thừa tất cả những tính chất của nghĩa cha đồng thời bổ sung thêm những thuộc tính mới phân biệt với những nghĩa con khác.

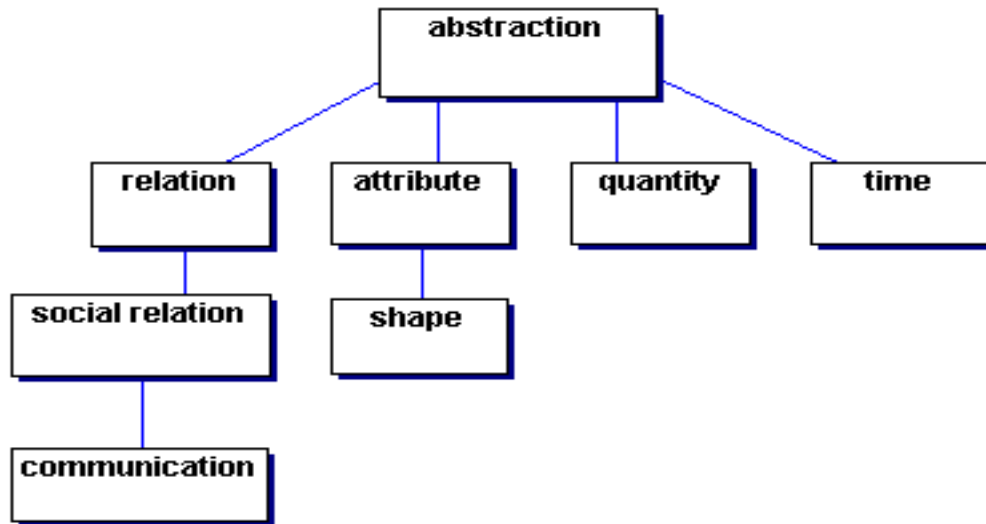
Ví dụ: *maple (cây thích) kế thừa, sẽ có tất cả những thuộc tính của nghĩa cha (hypernym) của nó là tree (cây). Nhưng có thêm những thuộc tính mới để phân biệt với các loại cây khác, chẳng hạn như: độ cứng của gỗ, hình dạng của lá, nhựa của nó được dùng để làm nước xirô (một loại thức uống ngọt có hương trái cây).*

Bằng cách định nghĩa quan hệ cấp bậc như thế, toàn bộ các danh từ trong WordNet tạo nên một hệ thống cấu trúc ngữ nghĩa phân cấp được gọi là hệ thống cây ngữ nghĩa kế thừa. WordNet định nghĩa 25 synset cấp cao nhất

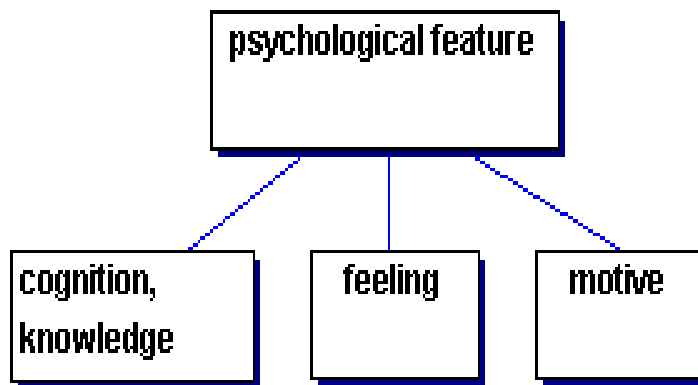
(Top level) được gọi là các nút gốc. Các nút gốc quan hệ với nhau theo các sơ đồ dưới đây:

Bảng II. 1: Nhóm các danh từ gốc của WordNet

Danh từ	Ý nghĩa
<i>Act, action, activity</i>	Hành động, hoạt động
<i>Animal, fauna</i>	Động vật, súc vật
<i>Artifact</i>	Nhân tạo
<i>Attribute, property</i>	Thuộc tính, tính chất
<i>Body, corpus</i>	Cơ thể, ngữ liệu
<i>Cognition, knowledge</i>	Nhận thức, tri thức
<i>Communication</i>	Giao tiếp
<i>Event, happening</i>	Sự kiện
<i>Feeling, emotion</i>	Cảm giác, cảm xúc
<i>Food</i>	Thức ăn
<i>Group, collection</i>	Nhóm, tập hợp
<i>Location, place</i>	Nơi chốn, địa điểm
<i>Motive</i>	Vận động, chuyển động
<i>Natural object</i>	Vật thể tự nhiên
<i>Natural phenomenon</i>	Hiện tượng tự nhiên
<i>Person, human being</i>	Con người, loài người
<i>Plant, flora</i>	Cây cối, thực vật
<i>Possession</i>	Sở hữu
<i>Process</i>	Quá trình, quy trình
<i>Quantity, amount</i>	Chất lượng, số lượng
<i>Relation</i>	Quan hệ
<i>Shape</i>	Hình dạng
<i>Sate, condition</i>	Trạng thái, điều kiện
<i>Substance</i>	Vật liệu, chất liệu
<i>Time</i>	Thời gian

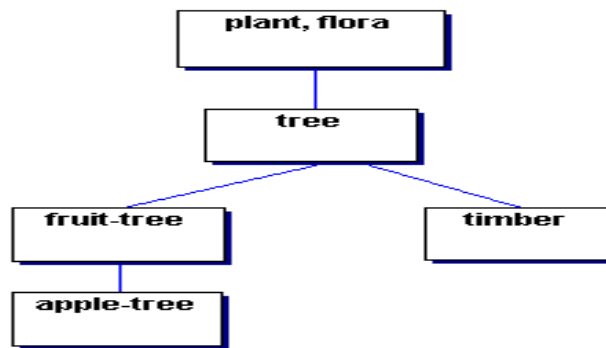


Hình II. 2. Sơ đồ của WordNet đối với các danh từ trừu tượng



Hình II. 3: Sơ đồ của WordNet đối với các danh từ chỉ các đặc điểm tâm lý

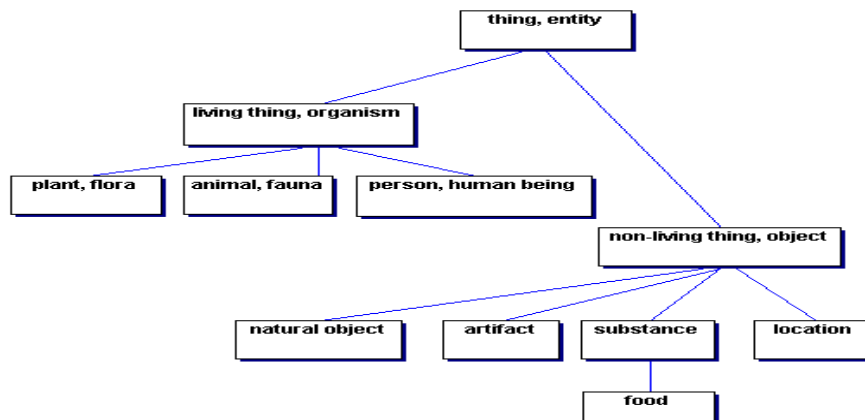
Những sơ đồ trên chỉ trình bày những ý niệm gốc, synset của WordNet. Mỗi nhóm có thể là hypernym của một hay nhiều nhóm ngữ nghĩa, synset con. Ví dụ: Plant, flora (thực vật, cây cối) có cây ăn quả (fruit-tree) và cây lấy gỗ (timber), trong nhóm cây ăn quả lại có cây táo (apple tree), v.v...



Hình II.5: Thí dụ một sơ đồ ngữ nghĩa của plants, flora

2.2.3.2.4 QUAN HỆ BỘ PHẬN VÀ TOÀN THỂ (MERONYMY)

Một loại quan hệ ngữ nghĩa khác được định nghĩa trong WordNet đó là quan hệ bộ phận và toàn thể (part-whole relation) còn gọi là meronym/holonym. Một ý niệm tương ứng với synset {x, x, ...} được gọi là một meronym của ý niệm tương ứng với synset {y, y, ...} khi chúng ta có thể nói “một y có một x” (a y has an x) hoặc “một x là một phần của y” (an x is a part of a y).



Hình II.5: Cây phân cấp ngữ nghĩa của nhánh thing, entity

2.2.3.3 KHOẢNG CÁCH NGŨ NGHĨA

Một số công thức để xác định mức độ tương đồng về ngữ nghĩa của các từ cũng như khái niệm của hệ thống.

2.2.3.3.1 Độ tương đồng về ngữ nghĩa giữa hai từ

$$\text{dist}(w_1, w_2) = \min_{\substack{c_{1i} \in w_1 \\ c_{2j} \in w_2}} \sum_{c_k \in \text{path}(c_{1i}, c_{2j})} \frac{1}{\text{depth}(c_k)}$$

Trong đó:

$\text{dist}(w_1, w_2)$ là độ tương đồng về ngữ nghĩa giữa 2 từ w_1, w_2 .

$\text{path}(c_{1i}, c_{2j})$ là đường đi từ c_{1i} đến c_{2j} trên cây biểu diễn ngữ nghĩa.

$\text{depth}(c_k)$ là độ sâu của c_k trên cây so với vị trí gốc.

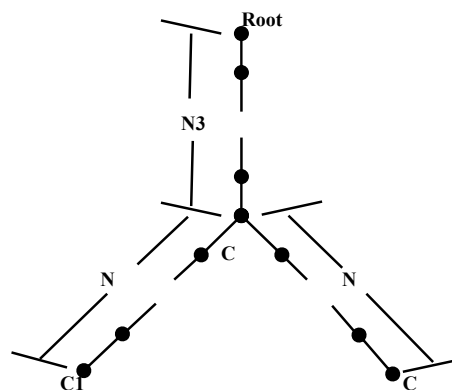
2.2.3.3.2 Độ tương đồng về ngữ nghĩa giữa hai khái niệm

$$\text{ConSim}(C_1, C_2) = \frac{2 * N_3}{N_1 + N_2 + 2 * N_3}$$

Trong đó:

$\text{ConSim}(C_1, C_2)$: là mức độ tương đồng về ngữ nghĩa giữa 2 khái niệm.

C_3 : là nốt chung gần nhất của C_1 và C_2 trên cây.



N_1, N_2, N là các khoảng cách giữa các vị trí (như trên hình vẽ).

2.2.3.3.3 Các độ đo khác

Budanitsky (1999) đã tổng kết và phân loại các phương pháp đo khoảng cách ngữ nghĩa. Các phương pháp này rất khác nhau, từ việc đơn giản là đếm các cạnh đến xác định các đặc trưng của cấu trúc mạng dựa trên hướng của các cạnh. Các phương pháp phân tích này có một số phương pháp cạnh tranh khác như sử dụng thống kê hay máy học. Một số phương pháp lại cũng đã được giới thiệu.

❖ **Hirst & St-Onge**

Ý tưởng là hai khái niệm gần nhau về ngữ nghĩa nếu như các synset của hai khái niệm ấy được nối với nhau bằng một đường đi không quá dài và không đổi hướng quá thường xuyên. Cụ thể độ đo như sau:

$$\text{relHS}(c_1, c_2) = C \text{ path length } k \times d$$

với c_1, c_2 là các synset; d là số lần chuyển hướng trên đường đi từ c_1 đến c_2 ; C và k là các hằng số.

❖ **Leacock-Chodorow**

Độ đo này cũng dựa trên chiều dài của đường đi ngắn nhất từ c_1 đến c_2 , tuy nhiên chỉ tập trung vào quan hệ IS-A mà thôi, và chia chiều dài đường đi cho độ sâu D của hệ thống cây phân cấp.

$$\text{simLC}(c_1, c_2) = -\log(\text{len}(c_1, c_2) / 2D)$$

❖ **Resnik**

Định hướng bởi ý tưởng rằng mức tương tự nhau của hai khái niệm có thể được đánh giá bằng mức độ chia sẻ thông tin giữa chúng. Đây là phương pháp sử dụng kết hợp ontology và corpus. Resnik định nghĩa độ tương tự giữa hai khái niệm là “hàm lượng thông tin” của cha chung gần nhất của chúng:

$$\text{simR}(c_1, c_2) = -\log p(\text{lso}(c_1, c_2))$$

với $p(c)$ là xác suất xuất hiện của một synset c trong một tập ngữ liệu nào đó, lso (lowest super-ordinate) là hàm xác định cha chung gần nhất của hai synset.

❖ **Jiang-Conrath**

Phương pháp này cũng sử dụng khái niệm “hàm lượng thông tin” nhưng ở dạng xác suất có điều kiện: xác suất bắt gặp một synset con khi đã có một synset cha.

$$\text{distJC}(c_1, c_2) = 2 \times \frac{\log(p(\text{lso}(c_1, c_2)))}{\log(p(c_1)) + \log(p(c_2))}$$

❖ **Lin (1998)**

Độ đo này lấy từ lý thuyết của ông ta về tính tương tự giữa hai đối tượng

bất kỳ. Cũng gần
$$simL(c_1, c_2) = 2 \times \frac{\log(p(ISO(c_1, c_2)))}{\log(p(c_1)) + \log(p(c_2))}$$
 giống như distJC:

Năm độ đo trình bày trên được đem so sánh với sự đánh giá của con người về mức độ gần nghĩa. Sai biệt giữa các độ đo của cột 2 nằm trong khoảng 0.1 và đều nằm bên dưới con số 0.88 (0.88 là đánh giá của Resnik về giới hạn của các phương pháp lượng giá bằng máy tính). Hơn nữa, sai biệt giữa các độ đo giảm đi phân nửa khi dùng tập dữ liệu lớn hơn (R&G). Thực ra là: các độ đo “phản ứng” khác nhau khi ta tăng kích thước dữ liệu thử: relHS, simLC, và simR trở nên tốt hơn, trong khi distJC và simL thì xấu đi.

Dĩ nhiên là sử dụng thẩm định của chuyên gia người để đánh giá các độ đo là trường hợp lý tưởng. Thực tế thì tập dữ liệu thử thường nhỏ, vì tạo ra tập dữ liệu thử lớn cho chuyên gia người là công việc mất nhiều công sức. Hơn thế nữa, vấn đề nằm ở chính phương pháp luận của cách tiếp cận này: chuyên gia người thường đánh giá dựa trên nghĩa trội hơn của mỗi từ, hay đánh giá dựa trên một quan hệ ưu tiên nào đó, trong khi điều chúng ta cần là quan hệ giữa tất cả các khái niệm mà mỗi từ đại diện.

2.2.3.3.4 MỘT SỐ ĐÁNH GIÁ

Hệ thống quan hệ đồng hiện có tính khả thi cao do có thể thực hiện được một cách tự động bằng máy tính với các mô hình xác suất thống kê, nhưng không biểu diễn được những quan hệ phân cấp trong ngôn ngữ. Hệ thống LDOCE, LLOCE có biểu diễn quan hệ phân cấp ngữ nghĩa nhưng ở chỉ mức đơn giản. Hệ thống WordNet biểu diễn đầy đủ các quan hệ ngữ nghĩa, đã được xây dựng rất tốt cho tiếng Anh, là cơ sở để xây dựng WordNet tiếng Việt.

Giải pháp cho hệ thống: kết hợp quan hệ đồng hiện và quan hệ phân cấp WordNet tiếng Việt.

2.3 ONTOLOGY

Trong vài năm gần đây, xuất hiện một lĩnh vực nghiên cứu mới là ontology. Có thể kể ra đây một số nguyên nhân đã thôi thúc việc nghiên cứu về ontology: vấn đề biểu diễn tri thức của trí tuệ nhân tạo (đặc biệt là biểu diễn quan hệ ngữ nghĩa), vấn đề sắp xếp và tìm kiếm các tài liệu tương tự nhau (đặc biệt là bài toán tìm kiếm trên mạng), vấn đề tìm hình thức biểu diễn mới cho cơ sở dữ liệu (sự ra đời của cơ sở dữ liệu lai giữa quan hệ và hướng đối tượng)... Tất cả những vấn đề trên đã dẫn đến việc ra đời lĩnh vực ontology mà mục tiêu trọng tâm là: phân loại các phạm trù, các khái niệm của tri thức, và biểu diễn mối liên hệ giữa các phạm trù đó với nhau.

2.3.1 XÂY DỰNG ONTOLOGY.

Theo cách dùng thông dụng trong AI, ontology hàm chỉ một quá trình xây dựng, và tạo thành bởi một tập các từ vựng, và dùng để mô tả một thực thể nào đó; cộng với những giả định tường minh về nghĩa hàm chỉ của các từ trong tập từ vựng. Tập các giả định này thường là một dạng lý thuyết lô-gích bậc nhất (first-order logic), còn tập từ vựng thường là các vị từ (predicate) một ngôi hay hai ngôi; và chúng được gọi tên tương ứng là: khái niệm và quan hệ. Trong trường hợp đơn giản nhất, ontology được mô tả như một cấu trúc phân cấp các khái niệm liên hệ với nhau bởi các quan hệ; trong trường hợp phức tạp hơn, các tiên đề thích hợp được thêm vào để diễn tả quan hệ giữa các khái niệm cũng như ràng buộc các diễn dịch có thể có.

Gruber (1995) đưa ra các tiêu chuẩn thiết kế một ontology:

- a. **Tính rõ ràng:** ontology phải hiệu quả trong các tiến trình giao tiếp, nghĩa là ngữ nghĩa của các khái niệm phải rõ ràng và mang tính khách quan. Khi có thể, nên đưa ra một định nghĩa hoàn chỉnh (một mệnh đề với các điều kiện cần và đủ) hơn là đưa ra một định nghĩa một phần (chỉ đưa ra các điều kiện cần).
- b. **Tính mạch lạc:** ontology phải mạch lạc nghĩa là phải thừa nhận các suy luận đúng từ các định nghĩa. Nếu một câu được suy luận từ các tiên đề mâu thuẫn với một định nghĩa thì ontology đó là không mạch lạc (nhất quán).
- c. **Tính có thể mở rộng:** cung cấp khả năng định nghĩa các thuật ngữ mới từ tập từ vựng có sẵn mà không phải xem lại định nghĩa của các từ vựng đã có.
- d. **Tối thiểu hóa các mã hóa:** để cho phép chọn lựa nhiều tùy chọn mã hóa khác nhau.
- e. **Tối thiểu hóa các “cam kết” (commitment):** ontology cần khẳng định về thế giới thực nó mô hình càng ít càng tốt, để cho những người sử dụng ontology quyền tự do được chuyên biệt hóa ontology.

Công việc xây dựng ontology thực tế trông đợi nhiều vào các hỗ trợ từ các khía cạnh hình thức và triết học của ontology. Trong phần này, chúng ta sẽ đúc kết một danh sách các mục mà khi thực hành, chúng ta cần được hỗ trợ giải quyết:

- ▶ Vị thế của ontology so với các dạng tài nguyên khác trong một hệ thống, hay trong một ứng dụng.
- ▶ Sự lựa chọn các khái niệm cần biểu diễn
- ▶ Sự lựa chọn các nội dung cần được gán cho mỗi khái niệm, và
- ▶ Sự đánh giá chất lượng ontology sử dụng cả hai mô hình hộp trắng và hộp đen.

Trong một số ứng dụng, ontology được dùng như là nguồn tri thức duy nhất (như là trong ứng dụng dịch máy sử dụng cơ sở tri thức), ontology được sử dụng như là:

- ▶ Nguồn hỗ trợ ngôn ngữ giải thích các nghĩa của các từ vựng được ghi nhận trong bộ từ vựng của một ngôn ngữ nào đó.
- ▶ Kết cấu mang nghĩa cho một ngôn ngữ biểu diễn ngữ nghĩa.
- ▶ Cung cấp các tri thức dạng heuristic cho các tài nguyên tri thức động như: bộ phân tích hay sản sinh ngữ nghĩa.

Điều mà người xây dựng ontology cần lưu tâm là việc chọn những khái niệm và việc biểu diễn chúng. Một ontology tốt sẽ có độ bao quát cần thiết cũng như độ đồng chất hợp lý. Độ bao quát phụ thuộc vào lĩnh vực và ứng dụng cụ thể trong lĩnh vực đó, và việc mà ontology hình thức có thể làm là giúp xác định cách tổ chức cấu trúc phân cấp kinh tế nhất, hay là cách xác định các nút nào không phải là lá? Ontology hình thức không những cần đặt ra các tính chất mà một ontology cần phải có mà còn phải đặt ra các tiêu chuẩn trong quá trình thiết kế và các tiêu chí về độ sâu và độ rộng của ontology.

2.3.2 TRAO ĐỔI ONTOLOGY

Vấn đề quan trọng tiếp theo là xu hướng chia xẻ và tái sử dụng các ontology. Thực ra vấn đề này đã được bao hàm trong tiêu chí e. nêu trên. Dù vậy vẫn còn hai khoản phải cân nhắc. Thứ nhất là sự lưỡng phân, biết đến trong ngôn ngữ học tính toán và ngôn ngữ học mô tả, trong tình huống chỉ có một lĩnh vực cần mô hình hóa và trong tình huống có nhiều lĩnh vực cần mô hình hóa cùng lúc. Khi thiết kế ontology chỉ cho một lĩnh vực, chúng ta đạt đến sự chính xác dễ dàng hơn vì bản chất hạn chế của lĩnh vực đó. Điều đó cũng có nghĩa là: càng chi tiết bao nhiêu thì ontology càng khó khả chuyển đổi với các lĩnh vực khác bấy nhiêu. Một điều quan trọng nữa là việc phát triển các công cụ hình thức để có thể trao đổi giữa các ontology với nhau được. Gruber (1993) đã định nghĩa một công cụ như vậy: Ontolingua, là công cụ nổi tiếng nhất dùng để dịch từ ontology này sang ontology khác. Ontolingua sử dụng KIF (Định dạng trao đổi tri thức - Knowledge Interchange Format) (KIF được thiết kế bởi Genesereth and Fikes (1992)):

“KIF trong ý đồ là một ngôn ngữ để xuất bản và giao tiếp tri thức. Nó được thiết kế để người đọc cảm thấy rõ ràng các nội dung thuộc về mức nhận thức luận, nhưng không hỗ trợ việc suy luận tự động ở mức đó. KIF được thiết kế theo kịp với những thành tựu mới nhất của biểu diễn tri thức, nhưng nó không phải là một hệ thống hoạt động biểu diễn tri thức.”

2.3.3 XÂY DỰNG ONTOLOGY TỪ VĂN BẢN.

2.3.3.1 GIỚI THIỆU

Phần này tóm tắt một số kinh nghiệm xây dựng ontology của một số nhóm nghiên cứu của Pháp. Các nhóm nghiên cứu này làm việc chủ yếu trên văn bản. Họ đã xây dựng được một số nguyên tắc chung cũng như đã trình bày một số phương pháp để tiến hành quy trình xây dựng ontology.

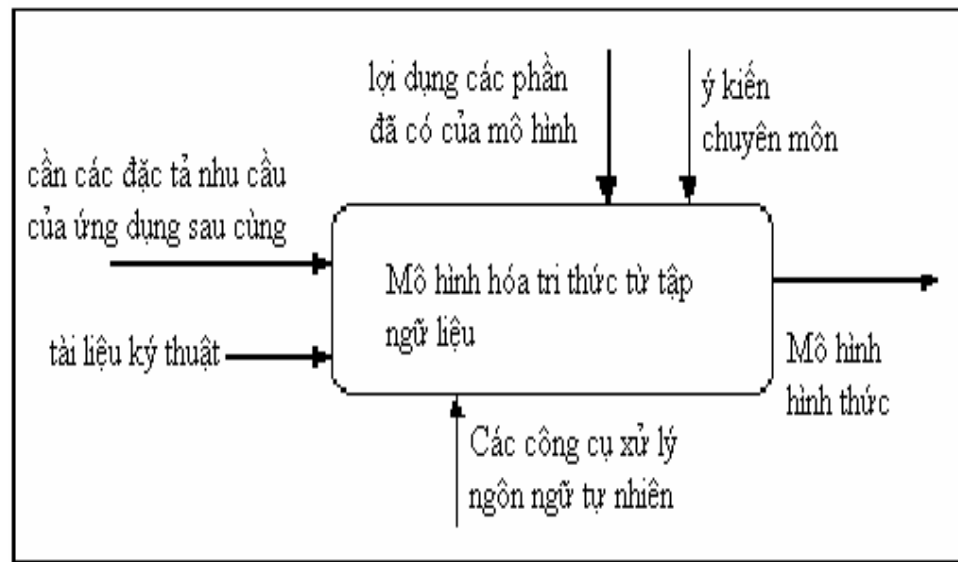
Một số nguyên tắc chung của nhóm nghiên cứu TIA (Pháp):

- ▶ Khởi đầu từ văn bản để đi rút trích tri thức: văn bản tập trung nhiều tri thức, kinh nghiệm của các chuyên gia trong các lĩnh vực. Các chuyên gia thường cho rằng văn bản thường cho một cái nhìn "già dặn" về một lĩnh vực nào đó ("già dặn" hơn so với các dạng dữ liệu khác). Tuy vậy điều này không có nghĩa văn bản là nguồn tri thức duy nhất.
- ▶ Luôn giữ mối liên kết từ mô hình xây dựng được đến văn bản nguồn ban đầu: các liên kết đến văn bản thực ra chính là định nghĩa của khái niệm trong mô hình và luôn có thể được dùng để cải tiến mô hình (mô hình ở đây là cách nói chung chung cho "ontology"). Các liên kết này còn được dùng để giải thích mô hình và bảo trì mô hình.
- ▶ Phân tích văn bản bằng cách sử dụng các công cụ xử lý ngôn ngữ tự nhiên và dựa trên các kết quả nghiên cứu ngôn ngữ học: nguyên tắc này rất rõ ràng, người ta hy vọng bằng các phân tích ngôn ngữ học: phân tích hình thái học (morphology), phân tích từ vựng học (lexical), phân tích cú pháp (syntactic)...có thể dẫn đến phân tích ngữ nghĩa của văn bản.

Các bài báo được tóm tắt trong bài này đều ít nhiều tuân thủ các nguyên tắc trên. Do đó phương pháp được dùng có xu hướng nặng về nghiên cứu ngôn ngữ và sử dụng các công cụ xử lý ngôn ngữ tự nhiên. Tuy nhiên các hướng tiếp cận khác cũng được đề cập tới.

2.3.3.2 MÔ HÌNH TỔNG QUÁT

Sau khi đề xuất các nguyên tắc nói trên, chúng ta có được mô hình chung nhất cho việc xây dựng ontology. Đây là mô hình rất chung, có thể áp dụng cho các ngôn ngữ khác nhau. Ở mô hình này chưa đề cập đến các phương pháp và chi tiết kỹ thuật. Đây là lựa chọn cụ thể cho từng bài toán xây dựng ontology khác nhau:



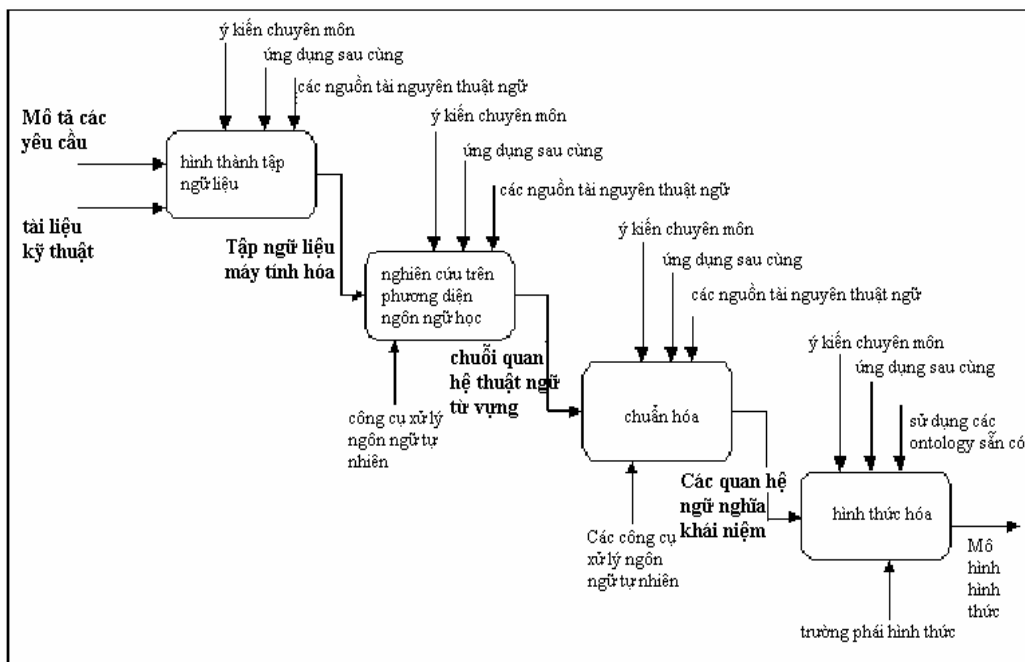
Hình II. 7: Mô hình xây dựng ontology tổng quát

Từ hình trên, có thể một số điều cần lưu ý khi xúc tiến tiến trình xây dựng mô hình ontology:

- ▶ Yêu cầu của ứng dụng cụ thể của ontology. Đây gần như là kim chỉ nam trong suốt quá trình xây dựng ontology: mục tiêu của bài toán sau cùng là gì.
- ▶ Các loại tài liệu kỹ thuật nào được sử dụng đến. ("Tài liệu kỹ thuật" ở đây hiểu là các văn bản đầu vào cho quá trình xây dựng ontology, đôi khi dùng từ corpus cũng để chỉ khái niệm đó).
- ▶ Các thành phần khác nhau của mô hình đã có để có thể tái sử dụng (điều này có ý nghĩa đặc biệt vì quá trình xây dựng ontology là một quá trình học lặp đi lặp lại)
- ▶ Các ý kiến và lựa chọn của chuyên gia trong từng bước xây dựng (ý kiến của chuyên gia.khi tiến hành học có giám sát)
- ▶ Các công cụ xử lý ngôn ngữ tự nhiên nào có thể dùng được.

2.3.3.3 MÔ HÌNH CHI TIẾT

Từ mô hình tổng quát trên, người ta vạch ra một mô hình tương đối chi tiết hơn như sau:



Hình II.8. Mô hình xây dựng ontology chi tiết

Mô hình hình trên đã tóm tắt các công đoạn chính của quá trình xây dựng ontology. Đầu tiên là giai đoạn hình thành tập ngữ liệu (corpus), sau đó tiến hành phân tích ngôn ngữ học trên tập ngữ liệu đó để rút ra các term và quan hệ giữa chúng (term là viết tắt của terminology, là thành phần chính để hình thành các khái niệm của ontology, ngoài ra term cũng có nghĩa là một đơn vị ngôn ngữ học cấu trúc; một từ, một từ kép, một ngữ (phase) hay cả một câu cũng đều có thể xem như là một term - một đơn vị tùy theo đối tượng chúng ta thao tác với trong từng giai đoạn là gì; vì vậy ở đây chúng tôi dùng nguyên từ term để chỉ cùng lúc cả hai ý nghĩa trên). Tiến hành phân tích hình thái, từ vựng và cú pháp để rút ra được các term và quan hệ giữa chúng. Giai đoạn này phát hiện các term và quan hệ là một bước lại gần các khái niệm và quan hệ ngữ nghĩa của ontology, kết quả của giai đoạn này là một mạng ngữ nghĩa. Sang giai đoạn

chuẩn hoá, mạng ngữ nghĩa ban đầu được chuẩn hoá nhiều lần lặp đi lặp lại và cuối cùng được hình thức hoá để có được ontology. Cụ thể các công đoạn như sau:

- ▶ I: Chuẩn bị tập ngữ liệu (corpus): cần có một chuyên gia để chọn ra trong các tài liệu kỹ thuật các văn bản cần thiết để hình thành corpus. Corpus phải rộng khắp lĩnh vực mà chúng ta muốn tạo ontology cho nó, đồng thời cũng phải đồng chất để bảo đảm "hàm lượng" vừa phải của các lĩnh vực con, các khái niệm con của lĩnh vực lớn ban đầu.

Có điểm cần lưu ý là có thể sử dụng các tài liệu dạng bán cấu trúc trong corpus, ví dụ như các từ điển. Trong các từ điển, các khái niệm đã được sắp xếp và định nghĩa của chúng cũng đã được cung cấp. Vì vậy có thể lợi dụng chúng cho việc xây dựng ontology.

- ▶ II: Phân tích ngôn ngữ học (linguistic analysis): mục tiêu của công đoạn này là rút trích các term và quan hệ từ vựng (lexical) giữa chúng. Kết quả của công đoạn này tương đối thô và cần phải được tinh chỉnh thêm.
- ▶ III: Chuẩn hoá (normalization): công đoạn này tiến hành kết hợp giữa tự động hoá và ý kiến của chuyên gia. Các term được thay thế bằng nhãn khái niệm (concept label) và các quan hệ dần dần được chuyển thành quan hệ ngữ nghĩa. Công đoạn này và công đoạn trên là hai công đoạn được lặp đi lặp lại xen kẽ nhau để thu được một mạng ngữ nghĩa sau cùng. Chuẩn hoá bao gồm hai công đoạn con:

1: công đoạn 1: vẫn mang tính ngôn ngữ học: tinh chỉnh các kết quả của giai đoạn 1. Trong các term và quan hệ đã được xác định, chuyên gia phải chọn ra term và quan hệ nào sẽ được đưa vào mô hình. Ở công đoạn này định nghĩa của các term cũng phải được chuẩn bị để phục vụ cho việc hình thành các khái niệm ở mức cao hơn.

2: công đoạn 2: các term được chuyển thành khái niệm sử dụng nhãn (label). Các quan hệ được chọn lọc và tổng quát hoá thành quan hệ ngữ nghĩa. Một mạng ngữ nghĩa được hình thành trong đó quan hệ phân cấp được chú trọng. Tuy nhiên các dạng quan hệ khác cũng được chú ý. Điều này hoàn toàn phụ thuộc vào mục đích xây dựng ontology là gì.

Như vậy giai đoạn II và III được xen kẽ để hình thành các mức cao thấp khác nhau của mô hình.

- ▶ IV: Công đoạn này sử dụng một ngôn ngữ hình thức nào đây (thường là logic mô tả - discription logic) để chuyển mạng ngữ nghĩa thành mạng hình thức. Giai đoạn này cũng làm chặt chẽ hoá thêm mô hình bằng cách đặt ra các khái niệm mới, các khái niệm trung gian cùng với việc chỉnh sửa lại các liên hệ.

2.3.3.4 CÔNG CỤ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Các công cụ xử lý ngôn ngữ tự nhiên này tiến hành một số phân tích sau đây:

- ▶ phân đoạn văn bản (chunking) tìm ra biên của các đoạn, câu, ngữ, từ.
- ▶ phân tích từ vựng (lexical): tìm ra liên hệ giữa các từ cụm từ.
 - * phân tích hình thái (morphology) để từ các từ tìm ra từ gốc của chúng. Các dạng số nhiều hay động từ phân ngôi được gom về làm một. Các tiếp đầu ngữ cũng như tiếp vĩ ngữ (tiền tố hay hậu tố) cũng được phân tích để tìm ra các liên hệ giữa các từ với nhau.
 - * phân tích từ loại (POS-part of speech): gán nhãn từ loại cho các từ, thao tác này có ích rất nhiều cho các phân tích mức cao hơn.
- ▶ phân tích cú pháp (syntactic) tìm ra liên hệ về cú pháp (theo một ngữ pháp nào đó) giữa các term. Công đoạn này phụ thuộc vào các công đoạn trên đây.

Từ góc độ xây dựng ontology, có thể phân loại các công cụ như sau:

- ▶ Công cụ rút trích thuật ngữ: dùng các phân tích ngôn ngữ tự nhiên hay là các công cụ thống kê để rút trích ra các term cần thiết.
- ▶ Công cụ rút trích quan hệ: sử dụng nhiều phương pháp khác nhau, một số là thống kê, một số là dựa trên luật (rule-based). Nhưng cơ bản là phát hiện các mẫu luật phổ biến trong corpus và các con số liên quan.

Hai dạng công cụ trên có thể được phối hợp theo nhiều cách khác nhau. Có thể đi tìm term trước, sau đó mới đi tìm quan hệ giữa chúng. Cũng có thể đi tìm quan hệ trước, rồi chắt lọc trong các quan hệ đó các term quan trọng.

Xây dựng ontology từ dưới lên (bottom-up)

Dùng các công cụ rút trích thuật ngữ, chúng ta tạo được một danh sách các

term. Các term này còn được gọi là CP (conceptual primitive), các đơn vị cơ bản của quá trình mô hình hoá. Sau đó sẽ sử dụng chuyên gia người để chọn lọc các CP này. Mỗi CP được định nghĩa bằng ngôn ngữ tự nhiên và có các văn bản liên kết với nó. Các văn bản này lại được sử dụng để rút trích ra các CP mới. Các CP mới thuộc 1 trong 3 loại sau:

- ▶ CP diễn tả quan hệ giữa các khái niệm ở mức cao
- ▶ CP đã có trong danh sách CP trước
- ▶ CP chỉ có trong danh sách CP mới này

Hai trường hợp sau cùng thường chỉ ra các CP ở mức cao hơn. Trường hợp đầu tiên chỉ ra quan hệ giữa chúng. Như vậy ta có được một quá trình lặp đi lặp lại: tìm các CP cao hơn các CP trước, xác định quan hệ giữa chúng và dần dần tinh chỉnh như vậy để có được một ontology.

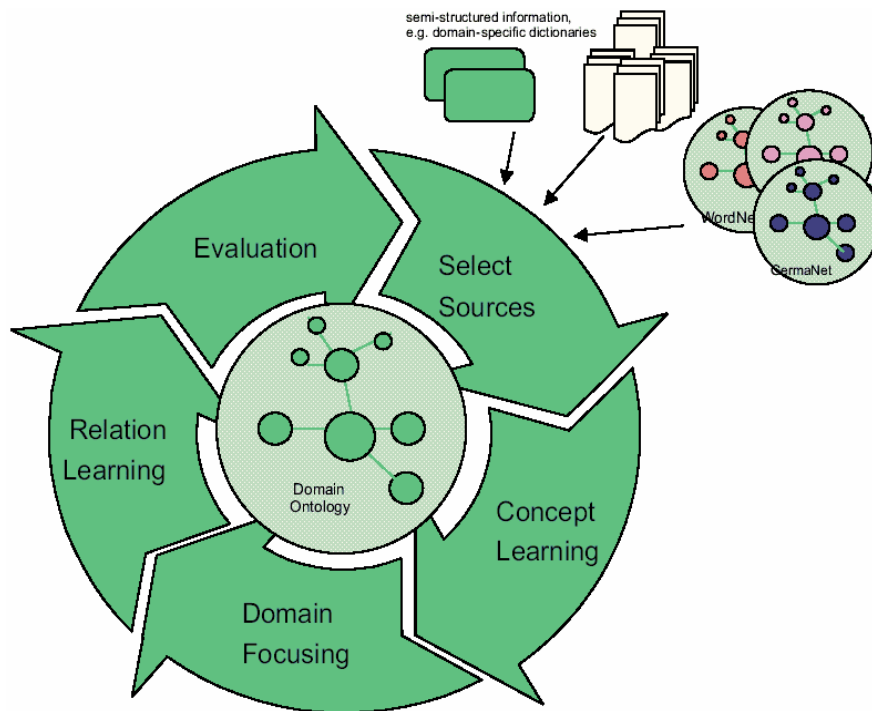
Xây dựng ontology từ trên xuống (top-down)

Phương pháp này khác biệt ở chỗ có sử dụng một ontology lõi. Thường ontology được chọn là của lĩnh vực tổng quát hơn lĩnh vực ta đang xây dựng ontology cho nó (ví dụ như luật pháp là tổng quát hơn của luật y tế).

Sau khi chọn được ontology lõi, tiến hành học để kết nạp thêm các khái niệm mới vào ontology như phương pháp trên. Sau đó tiến hành tỉa cành để thu được ontology sau cùng.

Phương pháp học ontology

Việc tạo ontology về cơ bản có thể xem như một quá trình học có giám sát. Máy móc tự động rút trích ra các term và quan hệ giữa chúng và chuyên gia người thì chọn lựa trong các term và quan hệ ấy các yếu tố thích hợp cho mô hình. Quá trình lặp đi lặp lại như hình sau đây:



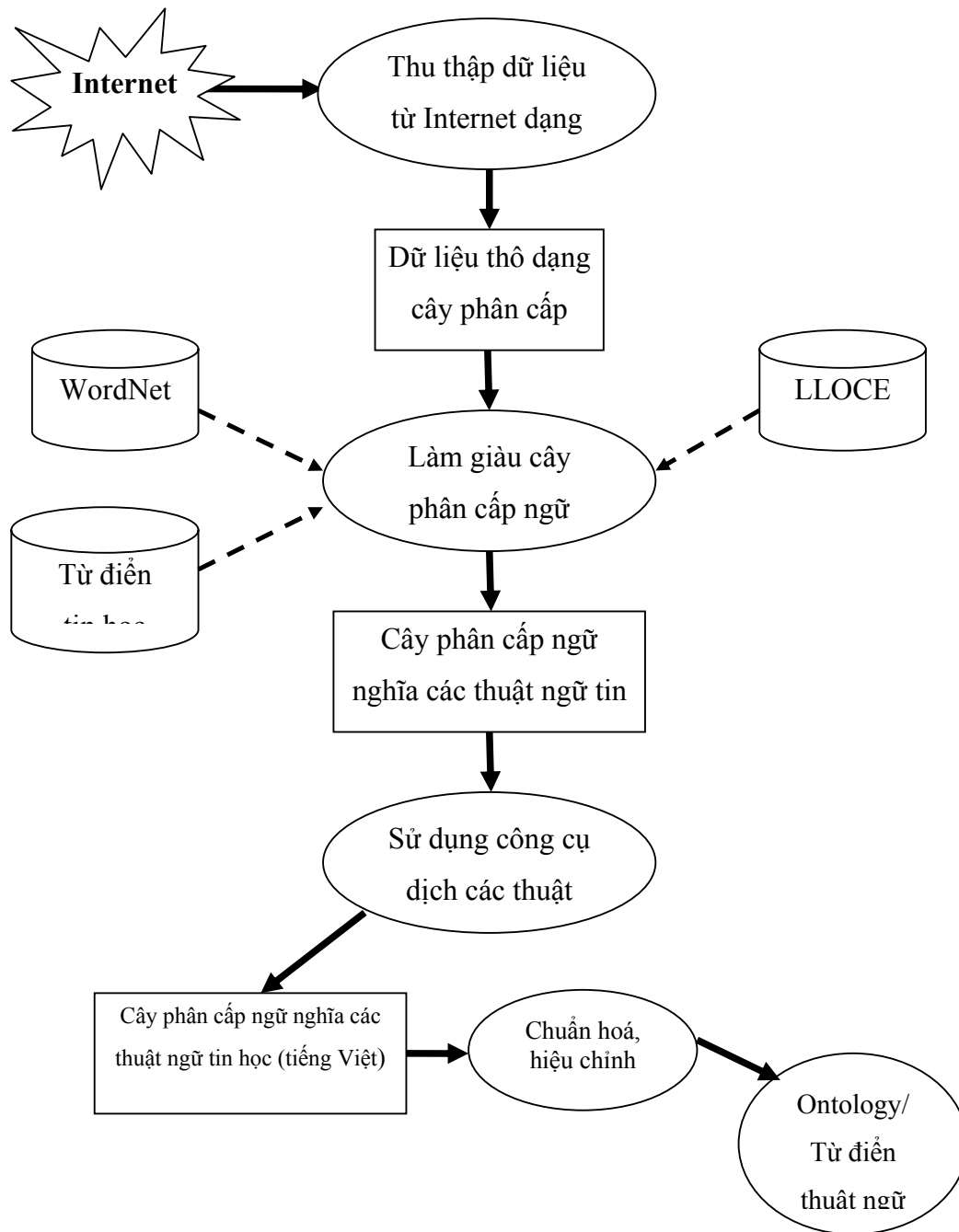
Hình II.9. Quá trình “học” ontology

Để ý dữ liệu đầu vào có thể có nhiều dạng khác nhau: văn bản, ontology sẵn có, các loại tài liệu bán cấu trúc như từ điển.

Sau khi chọn lọc các term để biến chúng thành các khái niệm, sự lựa chọn của chuyên gia người là cần thiết. Sau đó đến công đoạn học các quan hệ và đánh giá các quan hệ này. Quá trình tiếp diễn bắt đầu từ các vị trí cục bộ và càng ngày các đi lên các lớp trên cao của ontology.

2.3.4 XÂY DỰNG ONTOLOGY CHUYÊN NGÀNH TIN HỌC

Trong phần này chúng tôi sẽ trình bày mô hình và các bước để xây dựng ontology chuyên ngành tin học. Mô hình xây dựng ontotogy cụ thể như sau:



2.3.4.1 THU THẬP DỮ LIỆU

Hiện tại có rất nhiều trang Web trên thế giới cung cấp sẵn các ontology chuyên ngành tin học. Một số trang Web cho phép chúng ta xem ontology trực tuyến (Online). Do đó, mục đích của bước này là thu thập các ontology từ nhiều nguồn khác nhau. Các ontology được tổ chức dưới dạng cây phân cấp. Hình sau đây là cây phân cấp các thuật ngữ tin học được lấy từ trang web: www.yahoo.com



2.3.4.2 LÀM GIÀU DỮ LIỆU

Sau khi thu thập dữ liệu thô nhiều nguồn khác nhau trên Internet, kết quả có được là dữ liệu thô. Ở bước này, chúng ta tích hợp có chọn lọc các dữ liệu thu được đó thành nguồn dữ liệu mới đầy đủ hơn. Ngoài ra, dựa vào WordNet, từ điển LLOCE, từ điển tin học,... để làm giàu nguồn dữ liệu có được. Trong quá trình tích hợp các nguồn dữ liệu, mỗi nút trong cây phân cấp sẽ được gán một tần số (tần số tương quan đến các nút khác trong cùng một nhánh và đến nút cha). Việc chọn mục từ để bổ sung vào cây phân cấp chủ yếu dựa vào tần số này để quyết định có nên bổ sung vào hay không.

2.3.4.3 TẠO ONTOLOGY TIẾNG VIỆT

Để tạo được cây ontology tiếng Việt, ta sử dụng một số công cụ dịch tự động để dịch các thuật ngữ trong cây phân cấp đã được thu thập ở các bước trên. Sau khi dịch tự động xong, chúng ta hiệu chỉnh và dịch các thuật ngữ còn sót lại mà các công cụ chưa thể dịch được.

2.3.4.4 CHUẨN HOÁ ONTOLOGY

Sau khi có được ontology các thuật ngữ tin học bằng tiếng Việt, việc chuẩn hoá và hiệu chỉnh ontology đó là cần thiết. Việc chỉnh sửa được thực hiện dưới sự giám sát của con người và một số chuyên gia ngôn ngữ học và các chuyên gia tin học.

Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"

đơn thể	cấu trúc	đối tượng	biến	hàm	lớp đối tượng
đơn vị điều hợp số liệu	thiết bị tiếp nối số liệu	bộ làm thích ứng dữ liệu			
dòng dữ liệu	bộ dữ liệu	luồng dữ liệu			
đóng gói	phát triển	qui trình phần mềm	công nghệ phần mềm		
dòng lệnh	mã nguồn	lập trình	ngôn ngữ lập trình		
dòng lệnh	mã nguồn	lập trình	ngôn ngữ lập trình		
dự án	triển khai	phần mềm	chiến lược	phân phối	
dữ liệu	cơ sở dữ liệu	tập tin	thư mục	bộ nhớ	sao chép
dữ liệu	cấu trúc	đối tượng	biến	thiết kế	chương trình
dữ liệu	đối tượng	cấu trúc	biến thành phần	hàm thành	lớp đối tượng
dữ liệu	cơ sở dữ liệu	hệ quản trị cơ sở dữ liệu	giao tác	truy vấn	cập nhật
dữ liệu	cơ sở dữ liệu	tập tin	thư mục	bộ nhớ	sao chép
đường	tuyến				
đường dẫn	tập tin	thư mục	ổ đĩa	tương đối	tuyệt đối
đường truyền	băng thông	gói tin	tuyến	luồng	
ép kiểu	biến số	kiểu dữ liệu	lệnh gán	đổi kiểu	
Ethernet	Token Ring	ATM	Wifi		
ghi chú	chú thích	mã nguồn	giải thích	làm rõ	giúp dễ hiểu
giá	giá trị	định giá	chi phí		

Hình trích ngang ontology các thuật ngữ tin học

2.3.5 BIỂU DIỄN ONTOLOGY TRONG CƠ SỞ DỮ LIỆU (CSDL).

2.3.5.1 MỘT SỐ PHƯƠNG PHÁP BIỂU DIỄN.

2.3.5.1.1 RDF.

RDF được phát triển bởi W3C cho các siêu dữ liệu (metadata) cho các ứng dụng Web, và sử dụng XML làm cú pháp trao đổi dữ liệu. RDF được phát triển với mục đích tiện lợi hóa các tác nhân tự động (autonomous agents), và do đó cải tiến các dịch vụ web như máy tìm kiếm, các thư mục dịch vụ...

Cấu trúc của RDF gồm có 3 phần:

- ❑ Chủ thể (subject) ("*This article*").
- ❑ Mệnh đề (predicate) ("*is authored by*").
- ❑ Khách thể (object) ("*Uche Ogbuji*")

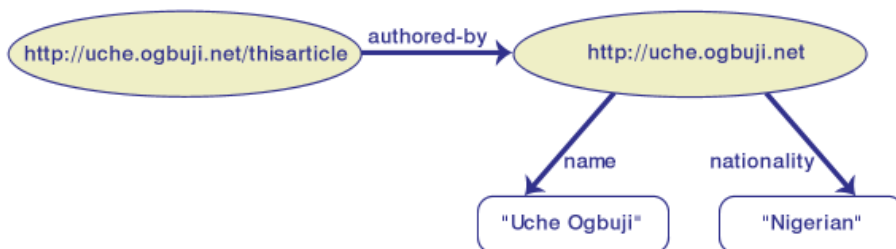
Đây là cách phân tích phổ biến của một phát biểu như vậy, cho dù là phân tích theo kiểu của ngữ pháp hay của logic hình thức. RDF thực ra là thành quả của quá trình nghiên cứu lâu dài của hai lĩnh vực: logic hình thức và ngữ pháp để mô tả tài nguyên (*resources*), nhưng hạng mục nào có thể truy cập được qua Web. Trong RDF, tài nguyên được xác định bằng URIs (Uniform Resource

Identifiers), và URL là một tập con của URI. Chủ thể của một phát biểu RDF phải là một tài nguyên, do đó phát biểu trên có thể được minh họa như sau:



Hình 1. Phát biểu RDF

Hình sau minh họa những phát biểu RDF được kết nối lại trong một sơ đồ (và được gọi là một mô hình). Và RDF chỉ là sự mở rộng như vậy: một đồ thị có hướng bao gồm các phát biểu mô tả tài nguyên Web. Nhìn có vẻ như RDF quá đơn giản để có thể thành một công nghệ quan trọng, nhưng sức mạnh của RDF nằm ở tính đơn giản của nó. Khoa học máy tính đã làm việc lâu dài với đồ thị để biểu diễn thông tin, và RDF cho phép các phát biểu đơn giản có thể được kết hợp lại với nhau để các tác nhân máy áp dụng các thuật toán duyệt đồ thị để xử lý dữ liệu. Một phát biểu đôi khi còn được gọi là một bộ ba (vì bao gồm 3 phần chính như đã trình bày). Các cơ sở dữ liệu các bộ ba như vậy đã chứng tỏ khả năng xử lý trên dữ liệu lớn hàng triệu bộ ba cũng vì tính đơn giản của dạng thông tin này. Và khả năng xử lý lớn đó được hy vọng là giúp các công nghệ khác xử lý được khối lượng thông tin khổng lồ của Web.



Hình 2. Mô hình RDF

Tuy nhiên trong thực tế, thường không khả thi khi trao đổi hay nhúng các mô tả RDF như vậy với HTML. Và người ta đã dùng XML để biểu diễn RDF. Hình sau cho chúng ta thấy một bản “tuần tự hóa” của RDF trong XML.


```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://schemas.uche.ogbuji.net/rdfexample/">
  <rdf:Description about="http://uche.ogbuji.net/thisarticle">
    <authored-by>
      <rdf:Description ID="uche.ogbuji.net">
        <name>Uche Ogbuji</name>
        <nationality>Nigerian</nationality>
      </rdf:Description>
    </authored-by>
  </rdf:Description>
</rdf:RDF>
```

Để ý việc dùng namespace của XML trong hình trên, RDF phụ thuộc vào namespace của XML để làm rõ các tên, các phần tử, và thuộc tính phải được định nghĩa rõ trong namespace.

2.3.5.1.2 RQL.

RQL là ngôn ngữ truy vấn RDF, là một ngôn ngữ có kiểu, định nghĩa những phép truy vấn và phép lặp cơ bản. Các phần sau minh họa một số ví dụ về truy vấn meta-schema, schema và truy vấn dữ liệu. Từ các phép truy vấn và lặp cơ bản, chúng ta có thể hình thành các truy vấn phức tạp hơn. RQL hỗ trợ các biểu thức đường dẫn tổng quát *generalized path expressions*, một dạng biến số thay cho nút và cạnh (của đồ thị cần tìm kiếm).

Hình dưới đây là RDF-schema của Cultural Portal. Phần trên của hình là các lớp của meta-schema RDFS: *Class* và *Property*, cũng như những siêu lớp do người dùng định nghĩa (*RealWorldObject*, *WebResource*, và *SchemaProperty*). Ngoài ra nó còn chứa 2 thuộc tính meta-schema là *related* (liên kết các lớp), và *maxCardinality* (một thuộc tính có kiểu nguyên). Phần giữa của hình gồm 2 schema, một dành cho các chuyên gia của bảo tàng và một dành cho quản trị viên của portal. Phần dưới của hình bao gồm một số mô tả tài nguyên của một số website bảo tàng trên mạng.

Để duyệt các cấu trúc cây trong schema, RQL cung cấp hai hàm *subClassOf* (đệ quy) và *subClassOf[^]* (trực tiếp), tương tự có các hàm *superClassOf*, *superClassOf[^]*.

subClassOf(Artist)
subClassOf[^](Artist)
subPropertyOf(creates)
subPropertyOf[^](creates)

Tương tự, các hàm *superPropertyOf*, *superPropertyOf[^]* trả về các tính chất cha. RQL cũng cung cấp các hàm tìm nút lá, cũng như nút cha chung gần nhất của hai nút (nút có thể là lớp, tính chất, siêu lớp).

leafclass(Artist)
leafproperty(creates)
nca(Painter, Sculptor)

RQL còn có các hàm định nghĩa trước (không có tham số) trả về nút gốc và lá của một hệ thống phân cấp.

Topclass
Leafclass
Topproperty
leafproperty

Với một thuộc tính nào đó, chúng ta có thể tìm định nghĩa bằng các hàm sau:

domain(creates)
range(creates)

RQL hỗ trợ câu lệnh *select – from – where* để truy vấn trên toàn bộ tập hợp. Cùng với sử dụng các biểu thức đường dẫn, để duyệt trên toàn bộ đồ thị ở độ sâu bất kỳ. Câu lệnh truy vấn sau tìm xem lớp nào có thể là *domain* và *range* của thuộc tính *creates*.

<i>SELECT \$C1, \$C2</i> <i>FROM {\$C1}creates{\$C2}</i>

Kết quả trả về như sau

class Artist	class Artifact
class Artist	class Sculpture
class Artist	class Painting
class Sculptor	class Artifact
class Sculptor	class Sculpture
class Sculptor	class Painting
class Painter	class Artifact
class Painter	class Sculpture

class Painter	class Painting
class Flemish	class Artifact
class Flemish	class Sculpture
class Flemish	class Painting
class Cubist	class Artifact
class Cubist	class Sculpture
class Cubist	class Painting

Câu lệnh truy vấn: tìm tất cả các thuộc tính định nghĩa trên lớp *Painter* và tất cả các lớp cha của nó có thể được viết như sau:

```
SELECT @P, range(@P)
FROM {;Painter}@P
```

hay

```
SELECT P, range(P)
FROM DProperty{P}
WHERE domain(P) >= Painter
```

Và các lệnh có thể được kết hợp để có được các lệnh phức tạp hơn:

```
SELECT X, SELECT $C, (SELECT @P, Y
FROM {W; ^$C} ^@P {Y}
WHERE namespace($C) != ns1 and namespace(@P) != ns1)
FROM ^$C {X}
FROM Resource {X}
USING NAMESPACE
ns1=&http://139.91.183.30:9090/RDF/VRP/Examples/demo/admin.rdf#
```

Hay các lệnh truy vấn lồng nhau:

```
SELECT C, count(SELECT @P FROM {;C}@P)
FROM Class{C}
WHERE C!= Resource and count(SELECT @P FROM {;C}@P)=
max( SELECT count(SELECT @Q FROM {;D}@Q)
FROM Class{D}
WHERE D != Resource )
```

2.3.5.2 KẾT LUẬN

RDF đã được dùng để kết hợp với cơ sở dữ liệu truyền thống tạo nên những hệ thống được kiểm soát tốt nhưng vẫn mang tính tiến hóa cao, và nó đã làm giảm khá nhiều chi phí duy tu, bảo, trì, phát triển cho các portal, máy tìm kiếm, các hệ thống làm chỉ mục...

Tuy vậy, RDF không phải là một công nghệ hoàn hảo. Các thực hiện của nó có thể là khó khăn trên một số mặt, và đặc tả RDF Schema duy nhất hiện có cũng chỉ vừa mới hoàn tất. RDF có hai đặc điểm nổi bật: được thiết kế để làm việc với XML, và nó đơn giản đủ để các trường hợp khó khăn nhất vẫn có thể xử lý được. Hiện tại, đã có nhiều công cụ RDF được phát triển, và cho phép chúng ta nghiên cứu các ưu điểm của RDF trong các hệ thống đóng.

Ontology là dạng tri thức biểu diễn dưới dạng mạng, và với những gì trình bày trên đây, RDF chính là ngôn ngữ thích hợp nhất để biểu diễn ontology. Tuy nhiên những nghiên cứu này còn dừng lại trên mức giấy tờ và ứng dụng đơn lẻ, chưa thể đưa ra thành một ứng dụng Web ở phạm vi lớn. Tuy nhiên, tương lai của điều đó không phải là xa, những công nghệ về XML và RDF càng ngày càng có nhiều ứng dụng và càng trở nên hiện thực trên WWW

Phần sau trong bản báo cáo sẽ trình bày biểu diễn ontology trong một cơ sở dữ liệu truyền thống.

2.4 BIỂU DIỄN CẤU TRÚC PHÂN CẤP CỦA ONTOLOGY TRONG CƠ SỞ DỮ LIỆU QUAN HỆ

Ontology được tổ chức thành một hệ thống phân cấp, trong trường hợp đơn giản, nó là một cây. Cấu trúc cây có thể được biểu diễn trong cơ sở dữ liệu quan hệ theo nhiều cách khác nhau. Nếu độ sâu tối đa của cây là cố định, thì chỉ cần dùng đến các kỹ thuật cơ bản: chuẩn hóa bảng với khóa và khóa ngoại. Tuy nhiên khi độ sâu của cây lớn thì cách biểu diễn này trở nên “cồng kềnh”. Cách tiếp cận này càng không thể làm việc khi độ sâu của cây không biết trước hay thay đổi trong quá trình sử dụng cơ sở dữ liệu.

Phần này đề cập cách biểu diễn cấu trúc cây theo hai mô hình: tập lồng nhau (nested sets), con trỏ - pointer (hay cũng có thể gọi là danh sách kề - adjacent list), và một cách biểu diễn tham khảo sử dụng định danh phả hệ (genealogical identifier). Phần cuối của báo cáo trình bày các hỗ trợ riêng của Oracle cho dạng cấu trúc dữ liệu này.

2.4.1 CÁC NHƯỢC ĐIỂM CỦA CÁCH BIỂU DIỄN BẰNG CON TRỎ.

Trong cơ sở dữ liệu quan hệ, các quan hệ cũng phải được biểu diễn tường minh như là dữ liệu. Cách thường thấy để biểu diễn cây là dùng con trỏ: có một cột là nút con, một cột là nút cha, và bảng là biểu diễn cạnh của cây (cũng có tài liệu gọi cách biểu diễn này là biểu diễn dạng danh sách kề - adjacent list). Ví dụ:

<i>CREATE TABLE Personnel</i> (<i>emp CHAR(20) PRIMARY KEY,</i> <i>boss CHAR(20) REFERENCES Personnel(emp),</i> <i>salary DECIMAL(6,2) NOT NULL</i>);	Emp	Boss	salary
	'Jerry'	NULL	1000.00
	'Bert'	'Jerry'	900.00
	'Chuck'	'Jerry'	900.00
	'Donna'	'Chuck'	800.00
	'Eddie'	'Chuck'	700.00
	'Fred'	'Chuck'	600.00

Khóa chính là **emp**, nhưng cột **boss** phụ thuộc vào **emp**, do đó ta gặp vấn đề chuẩn hóa ở đây. Ràng buộc **REFERENCES** là để cho không có “sếp” nào không phải là một nhân viên.

Nếu vì lý do nào đó ‘Jerry’ đổi thành một người khác thì ta phải cập nhật tất cả các dòng có liên quan. Một nhược điểm khác là khi ta muốn tìm “sếp” của các nhân viên, câu truy vấn buộc phải kết bảng:

```
SELECT B1.emp, 'bosses', E1.emp  
FROM Personnel AS B1, Personnel AS E1  
WHERE B1.emp = E1.boss;
```

Nếu ta muốn tìm “sếp” của “sếp” của một nhân viên nào đó, lúc đó ta phải dùng thao tác kết bảng phức tạp hơn như sau:

```
SELECT B1.emp, 'bosses', E2.emp  
FROM Personnel AS B1, Personnel AS E1, Personnel AS E2  
WHERE B1.emp = E1.boss AND E1.emp = E2.boss;
```

Và nếu muốn truy vấn sâu hơn thì phải kết bảng nhiều hơn. Thao tác kết bảng đòi hỏi chi phí lớn, hơn nữa ta thường không biết trước độ sâu của cây, nên không thể mở rộng câu truy vấn như trên mãi được.

Tuy nhiên vấn đề thực sự với mô hình này là các thao tác duyệt cây (ví dụ như tính tổng lương của mọi người trong công ty). Ta thường phải dùng đến các ngôn ngữ thủ tục (procedural) phía client (client hiểu trong khung cảnh server là hệ quản trị cơ sở dữ liệu) để hoàn tất những thao tác này.

2.4.2 BIỂU DIỄN CẤU TRÚC CÂY TRONG ORACLE

Thoạt nhìn thì cơ sở dữ liệu quan hệ không phải là một công cụ tốt để biểu diễn và thao tác trên cây. Bài viết này trình bày các điểm sau:

- ▶ Một dòng trong cơ sở dữ liệu quan hệ có thể được xem như là một đối tượng.
- ▶ Con trỏ từ một đối tượng này đến đối tượng khác có thể được biểu diễn bằng một trường số trong bảng dữ liệu.
- ▶ Minh họa phần mở rộng cho cây (tree extension) của Oracle (*CONNECT BY ... PRIOR*).

Một ví dụ điển hình về cây là sơ đồ tổ chức cán bộ của một cơ quan.

```
create table employee_boss  
(  
    employee_id integer primary key  
    boss_id          references employee_boss  
    name            varchar(100)  
);  
insert into employee_boss values (1, NULL, "Big Boss");  
insert into employee_boss values (2, 1, "Marketing");  
insert into employee_boss values (3, 1, "Sales");  
insert into employee_boss values (4, 3, "Joe Sales");  
insert into employee_boss values (5, 4, "Bill Sales");  
insert into employee_boss values (6, 1, "Engineer");  
insert into employee_boss values (7, 6, "Jane");  
insert into employee_boss values (8, 6, "Bob");
```

Số boss_id thực chất là con trỏ đến một dòng khác trong bảng employee_boss. Nếu cần hiển thị cả sơ đồ tổ chức (chỉ với SQL chuẩn), ta cần viết chương trình bằng các ngôn ngữ client (C, Java, Perl...) như sau:

- ▶ truy vấn cơ sở dữ liệu để tìm nhân viên có boss_id là NULL (tức là "sếp").
- ▶ tìm các nhân viên trực tiếp của sếp này.
- ▶ lặp lại để tìm hết toàn bộ cây.

Với câu lệnh "connect by" của Oracle, có thể lấy tất cả dòng ra trong một lúc:

```
select name, employee_id, boss_id  
from employee_boss  
connect by prior employee_id = boss_id;
```


NAME	EMPLOYEE_ID	BOSS_ID
Big Boss	1	
Marketing	2	1
Sales	3	1
Joe Sales	4	3
Bill Sales	5	4
Engineering	6	1
Jane	7	6
Bob	8	6
Marketing	2	1
Sales	3	1
Bill Sales	4	3
Joe Sales	5	4
Bill Sales	4	3
Joe Sales	5	4
Engineering	6	1
Jane	7	6
Bob	8	6
Jane	7	6
Bob	8	6
Bill Sales	5	4

Với câu lệnh như trên, Oracle in ra tất cả các cây con của cơ sở dữ liệu.

Bây giờ ta thêm vào mệnh đề "**start with**":

```
select name, employee_id, boss_id
from employee_boss
connect by prior employee_id = boss_id
start with employee_id in
(
    select employee_id
    from employee_boss
    where boss_id is NULL
);
```

NAME	EMPLOYEE_ID	BOSS_ID
Big Boss	1	
Marketing	2	1
Sales	3	1
Joe Sales	4	3
Bill Sales	5	3
Engineering	6	1
Jane	7	6
Bob	8	6

Ở đây, ta dùng một câu truy vấn con trong mệnh đề "start with" để tìm ra "sếp" lớn nhất. Để đơn giản trình bày, chúng ta quy ước là "Big Boss" có employee_id bằng 1. Oracle cung cấp một cột giả là "level" như sau (chỉ có tác dụng khi câu truy vấn có sử dụng "**connect by**"):

```
select name, employee_id, boss_id, level
from employee_boss
connect by prior employee_id = boss_id
start with employee_id = 1;
```

NAME	EMPLOYEE_ID	BOSS_ID	LEVEL
Big Boss	1		1
Marketing	2	1	2
Sales	3	1	2
Joe Sales	4	3	3
Bill Sales	5	4	4
Engineering	6	1	2
Jane	7	6	3
Bob	8	6	3

Cột giả "level" có thể được dùng để trình bày dữ liệu như sau:

```
column padded_name format a30
select
    lpad(' ', (level - 1) * 2) || name as padded_name,
    employee_id,
    boss_id,
    level
from employee_boss
connect by prior employee_id = boss_id
start with employee_id = 1;
```

PADDED_NAME	EMPLOYEE_ID	BOSS_ID	LEVEL
Big Boss	1		1
Marketing	2	1	2
Sales	3	1	2
Joe Sales	4	3	3
Bill Sales	5	4	4
Engineering	6	1	2
Jane	7	6	3
Bob	8	6	3

Có thể dùng mệnh đề "**where**" để giới hạn kết xuất:

```
column padded_name format a30
select
    lpad(' ', (level - 1) * 2) || name AS padded_name,
    employee_id,
    boss_id,
    level
from employee_boss
where level <= 3
connect by prior employee_id = boss_id
start with employee_id = 1;
```

<i>PADDED_NAME</i>	<i>EMPLOYEE_ID</i>	<i>BOSS_ID</i>	<i>LEVEL</i>
<i>Big Boss</i>	<i>1</i>		<i>1</i>
<i>Marketing</i>	<i>2</i>	<i>1</i>	<i>2</i>
<i>Sales</i>	<i>3</i>	<i>1</i>	<i>2</i>
<i>Joe Sales</i>	<i>4</i>	<i>3</i>	<i>3</i>
<i>Engineering</i>	<i>6</i>	<i>1</i>	<i>2</i>
<i>Jane</i>	<i>7</i>	<i>6</i>	<i>3</i>
<i>Bob</i>	<i>8</i>	<i>6</i>	<i>3</i>

Giả sử chúng ta muốn các nhân viên ở cùng một mức được sắp xếp theo thứ tự ABC. Tuy nhiên "**order by**" không làm được việc đó khi dùng chung với "**connect by**":

```
column padded_name format a30
select
    lpad(' ', (level - 1) * 2) || name as padded_name,
    employee_id,
    boss_id,
    level
from employee_boss
where level <= 3
connect by prior employee_id = boss_id
start with employee_id = 1
order by level, name;
```

<i>PADDED_NAME</i>	<i>EMPLOYEE_ID</i>	<i>BOSS_ID</i>	<i>LEVEL</i>
<i>Big Boss</i>	<i>1</i>		<i>1</i>
<i>Engineering</i>	<i>6</i>	<i>1</i>	<i>2</i>
<i>Marketing</i>	<i>2</i>	<i>1</i>	<i>2</i>
<i>Sales</i>	<i>3</i>	<i>1</i>	<i>2</i>
<i>Bob</i>	<i>8</i>	<i>6</i>	<i>3</i>
<i>Jane</i>	<i>7</i>	<i>6</i>	<i>3</i>
<i>Jane Sales</i>	<i>4</i>	<i>3</i>	<i>3</i>
<i>Bill Sales</i>	<i>5</i>	<i>4</i>	<i>4</i>

SQL là một ngôn ngữ hướng tập hợp (set-oriented), khi dùng mệnh đề "**connect by**" thì thứ tự của kết xuất đã có ý nghĩa, và không hữu ích lắm khi chúng ta dùng thêm "**order by**".

"join" không làm việc với "connect by"

Giả sử chúng ta muốn in một danh sách các nhân viên và sếp trực tiếp của họ, chúng ta sẽ gặp lỗi như sau:

```
select
    lpad(' ', (level - 1) * 2 || cs1.name as padded_name, cs2.name as
supervisor_name
from employee_boss cs1, employee_boss cs2,
where cs1.boss_id = cs2.employee_id(+)
connect by prior cs1.employee_id = cs1.boss_id
start with cs1.employee_id = 1;
```

ERROR at line 4:

ORA-01437: cannot have join with CONNECT BY

Chúng ta có thể xử lý trường hợp này bằng cách tạo view như sau:

```
create or replace view connected_slaves
as
select
    lpad(' ', (level - 1) * 2 || name as padded_name,
employee_id,
boss_id,
level as the_level
from employee_boss
connect by prior employee_id = boss_id
start with employee_id = 1;
select * from connected_slaves
```

<i>PADDED_NAME</i>	<i>EMPLOYEE_ID</i>	<i>BOSS_ID</i>	<i>LEVEL</i>
<i>Big Boss</i>	<i>1</i>		<i>1</i>
<i>Marketing</i>		<i>1</i>	<i>2</i>
<i>Sales</i>	<i>3</i>	<i>1</i>	<i>2</i>
<i>Joe Sales</i>	<i>4</i>	<i>3</i>	<i>3</i>
<i>Bill Sales</i>	<i>5</i>	<i>4</i>	<i>4</i>
<i>Engineering</i>	<i>6</i>	<i>1</i>	<i>2</i>
<i>Jane</i>	<i>7</i>	<i>6</i>	<i>3</i>
<i>Bob</i>	<i>8</i>	<i>6</i>	<i>3</i>

Bây giờ chúng ta đã có thể sử dụng "**JOIN**"

Để ý rằng chúng ta đã sử dụng **outer join** để kết quả không loại trừ "Big Boss". Thay vì sử dụng **view** và **join**, chúng ta có thể thêm một câu lệnh truy vấn con vào sau danh sách lựa chọn như sau:

```
select
    lpad(' ', (level - 1) * 2) || name as padded_name,
    (
        select name
        from employee_boss cs2
        where cs2.employee_id = cs1.boss_id
    ) as supervisor_name
from employee_boss cs1
connect by prior employee_id = boss_id
start with employee_id = 1;
```

Luật tổng quát trong Oracle là có thể thêm một câu truy vấn con chỉ trả về một dòng kết quả vào bất kỳ nơi đâu trong danh sách lựa chọn.

Giả sử chúng ta làm một ứng dụng web và có những thông tin phải trình bày cho sếp của sếp thay vì cho những người phụ tá hay ngang hàng. Câu lệnh sau kiểm tra "Marketing" có quyền giám sát "Jane" và "Bob" hay không:

```
select count(*)
from employee_boss
where employee_id = 7 and level > 1
start with employee_id = 2
```

connect by prior employee_id = boss_id;

Rõ ràng kết quả trả về là không. Đề ý là chúng ta bắt đầu với nút "Marketing" và chỉ định "level > 1" để không phải kết luận là có ai giám sát "Marketing". Bây giờ kiểm tra xem "Big Boss" có quyền giám sát với "Jane" hay không:

```
select count(*)  
from employee_boss  
where employee_id = 7 and level > 1  
start with employee_id = 1  
connect by prior employee_id = boss_id;
```

Cho dù "Big Boss" không phải là sếp trực tiếp của "Jane", nhưng bắt Oracle tìm cây con đã chỉ ra mối liên hệ là có. Câu truy vấn này thường rất quan trọng, nên thay vì để nó trong một trang web, chúng ta có thể tập trung vào một hàm PL/SQL.

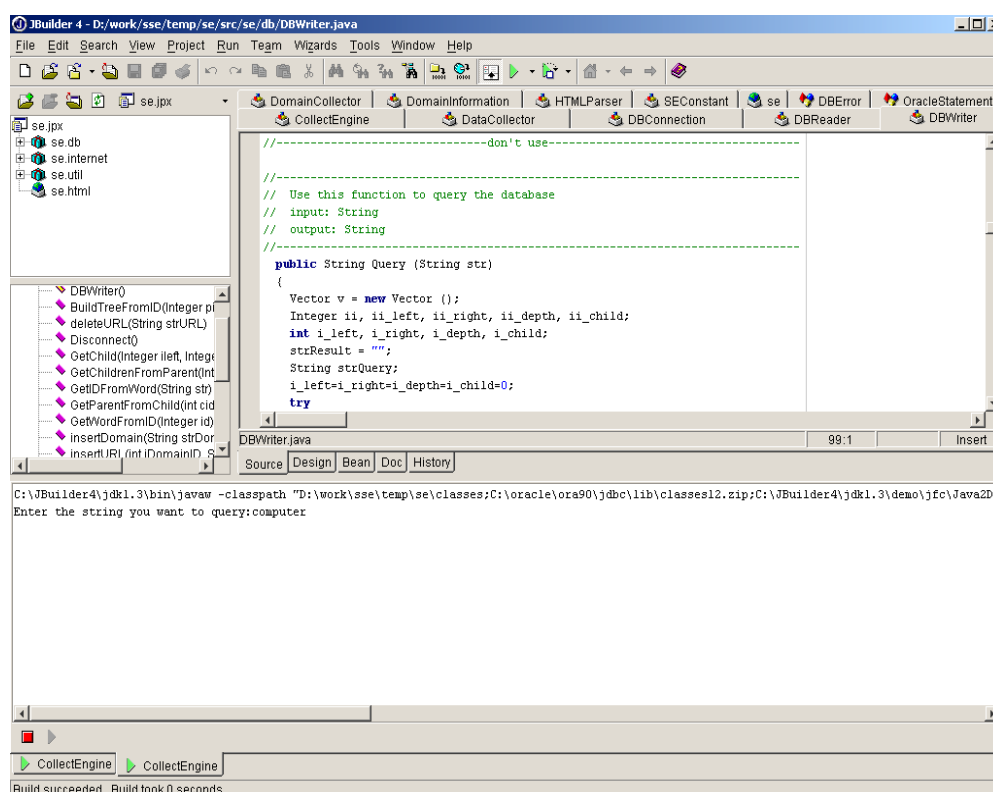
2.4.3 NHẬN XÉT

Như ở trên đã trình bày các cách biểu diễn cây trong cơ sở dữ liệu quan hệ. Cho dù các CSDL quan hệ lớn hiện tại như Oracle hay SQL-Server đều có các hỗ trợ cho dạng cấu trúc dữ liệu cây tổng quát, nhưng việc sử dụng các dạng biểu diễn riêng cho từng trường hợp cụ thể là cần thiết. Mô hình con trỏ (hay danh sách kề) đơn giản, tuy nhiên các thao tác tìm kiếm lại chậm. Mô hình tập lồng nhau có tốc độ tìm kiếm nhanh, tuy nhiên thêm/xóa gặp nhiều vấn đề. Dạng biểu diễn dùng định danh phủ hệ được đưa ra thảo luận vì nó kế thừa được ưu điểm của cả hai mô hình trên, tuy nhiên việc cài đặt cụ thể lại khá phức tạp. Chúng tôi chọn mô hình tập lồng nhau cho việc biểu diễn ontology vì đây là dạng dữ liệu ít khi phải thay đổi hay cập nhật, trong khi đòi hỏi rất nhiều ở khâu tìm kiếm.

2.5 KẾT LUẬN

Ontology được lưu trong cơ sở dữ liệu Oracle và được tối ưu hóa cho phép tìm kiếm. Vì vậy, truy vấn cơ sở dữ liệu là rất nhanh, nhưng bù lại việc cập nhật ontology có chi phí lớn. Tuy nhiên cập nhật ontology là việc không thường xuyên làm vì ontology là tri thức ít thay đổi.

Chương trình truy vấn được viết bằng Jbuilder, giao diện đơn giản dạng console như sau:



Một số vấn đề gặp phải của đề tài:

- ▶ Nghiên cứu các phương pháp và kỹ thuật xây dựng ontology phù hợp cho tiếng Việt.
- ▶ Việc dịch các thuật ngữ tiếng Anh: đây là hạn chế lớn nhất vì rất nhiều thuật ngữ không có dạng tương đương trong tiếng Việt, hay có mà không được dùng thống nhất.
- ▶ Vấn đề dịch còn có một số lỗi.
- ▶ Vấn đề biểu diễn dữ liệu trong CSDL còn có thể được cải tiến thêm.




3 PHẦN III:

THIẾT KẾ HỆ THỐNG S.E VÀ KẾT QUẢ THỬ NGHIỆM.

3.1 THIẾT KẾ HỆ THỐNG.

3.1.1 Đặt tả Hệ thống:

Hệ thống gồm 3 modul

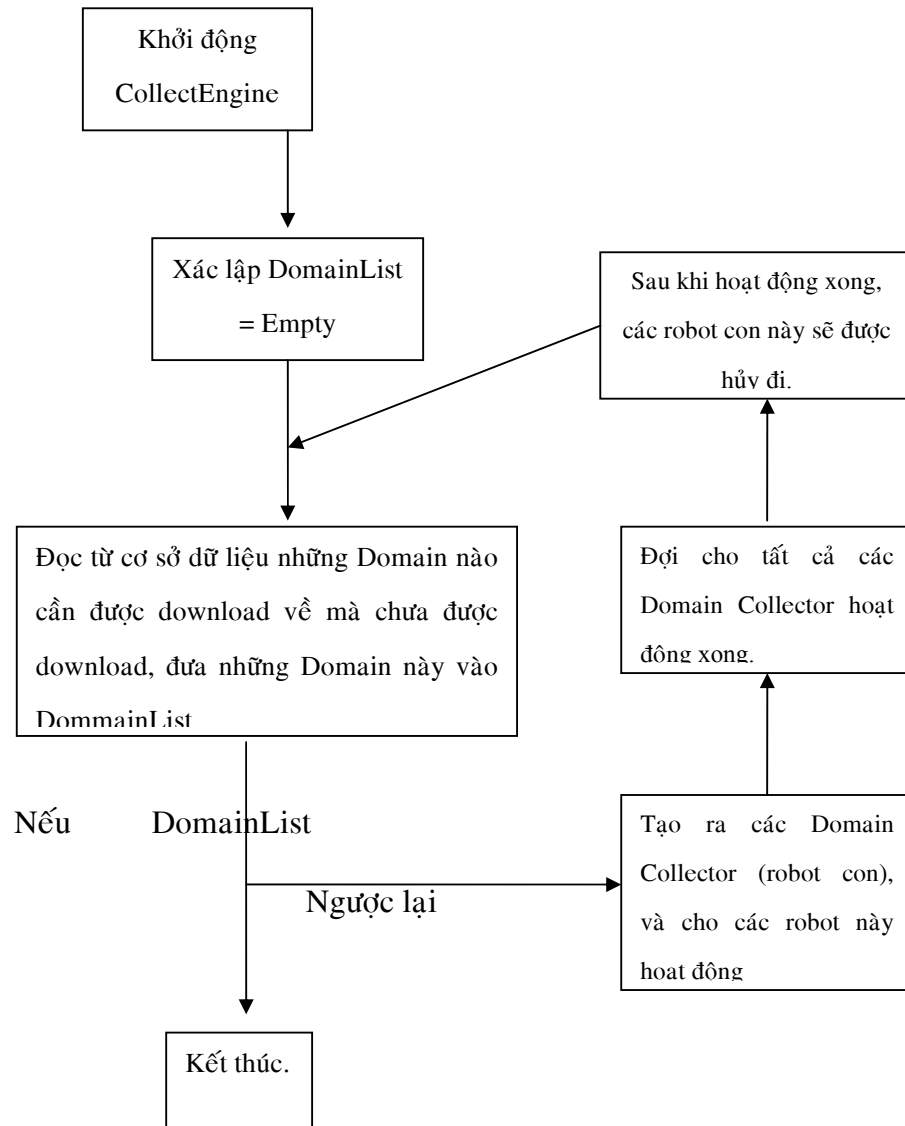
-  Module Web Robot. Chức năng chính là thu thập dữ liệu từ các trang Web trên các Website được viết bằng tiếng Việt.
-  Module ConvertFile. Công việc chính của module này là thống nhất lại toàn bộ dữ liệu trước khi chuyển cho Oracle TEXT thực hiện việc lập Chỉ mục (Index). Việc thống nhất này bao gồm:
 - Chuyển toàn bộ các định dạng (PDF, DOC, EXCEL...) sang dạng HTML.
 - Chuyển mã của trang (TCVN3, VNI, Unicode...) sang bảng mã Windows-CP1258 (Unicode tổ hợp).
 - Lưu dữ liệu vào trong CSDL Oracle để index.
-  Modul Truy vấn

3.1.2 Thiết kế các Chức năng của Hệ thống.

3.1.2.1 Module Web Robot. Gồm các thành phần sau:

3.1.2.1.1 CollectEngine.

Đây là thành phần điều khiển quá trình thu thập thông tin từ Internet. Nó sẽ tạo ra các DomainCollector và quản lý việc download. Khi CollectEngine hoàn tất công việc cũng là lúc module này kết thúc. Sơ đồ hoạt động của hệ thống CollectEngine theo Sơ đồ sau:



Mô tả công việc đọc DomainList từ cơ sở dữ liệu (từ sơ đồ trên):

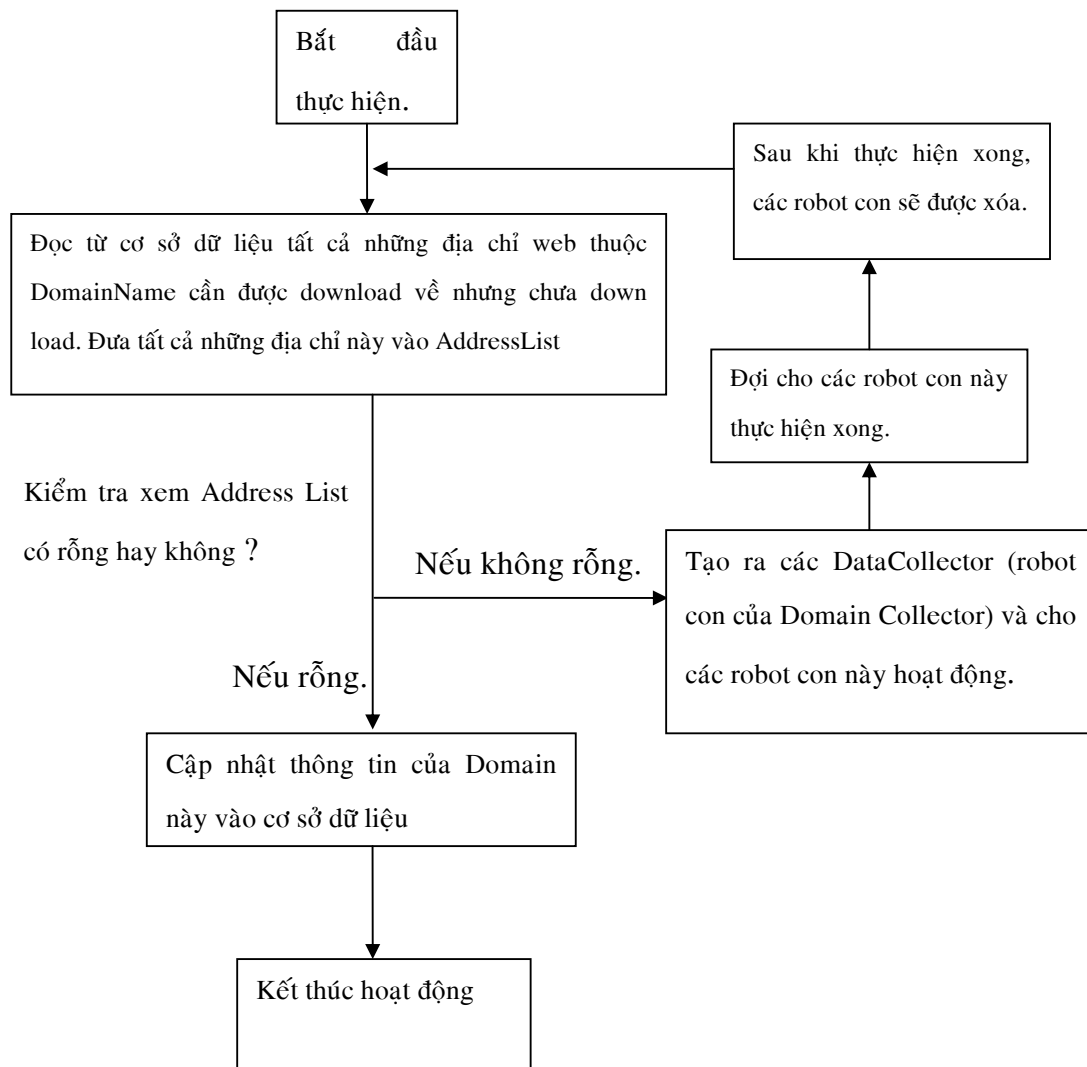
3.1.2.1.2 DomainCollector.

Có nhiệm vụ thu thập tất cả các trang thông tin bên trong một domain về.

- + Đầu vào : tên domain mà hệ thống cần thu thập về.
- + Hoạt động : DomainCollector lấy danh sách tất cả các trang web đã được đánh dấu thuộc về domain nhưng chưa lấy về. Kế tiếp, DomainCollector sẽ tạo ra các DataCollector và cho các DataCollector này thực hiện việc lấy các trang WEB về.
- + Đầu ra : domain được tải về.

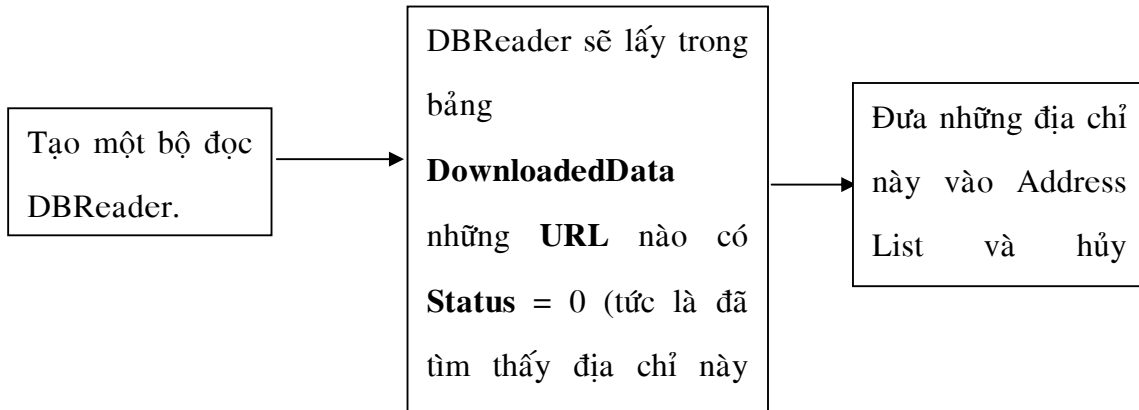
Ngoài ra, DomainCollector còn lưu trữ một Collect Engine (là robot cha của nó) để thực hiện việc giao tiếp.

Sơ đồ hoạt động của DomainCollector:



Mô tả công việc tải toàn bộ domain có tên DomainName về.

Mô tả công việc đọc AddressList từ cơ sở dữ liệu :



3.1.2.1.3 DataCollector.

Bộ thu thập thông tin (các trang WEB). Mục tiêu của DataCollector là thu nhận thông tin từ Internet về, sau đó sẽ lưu trữ chúng xuống cơ sở dữ liệu, đồng thời cũng ghi những dữ liệu này xuống hệ thống file.

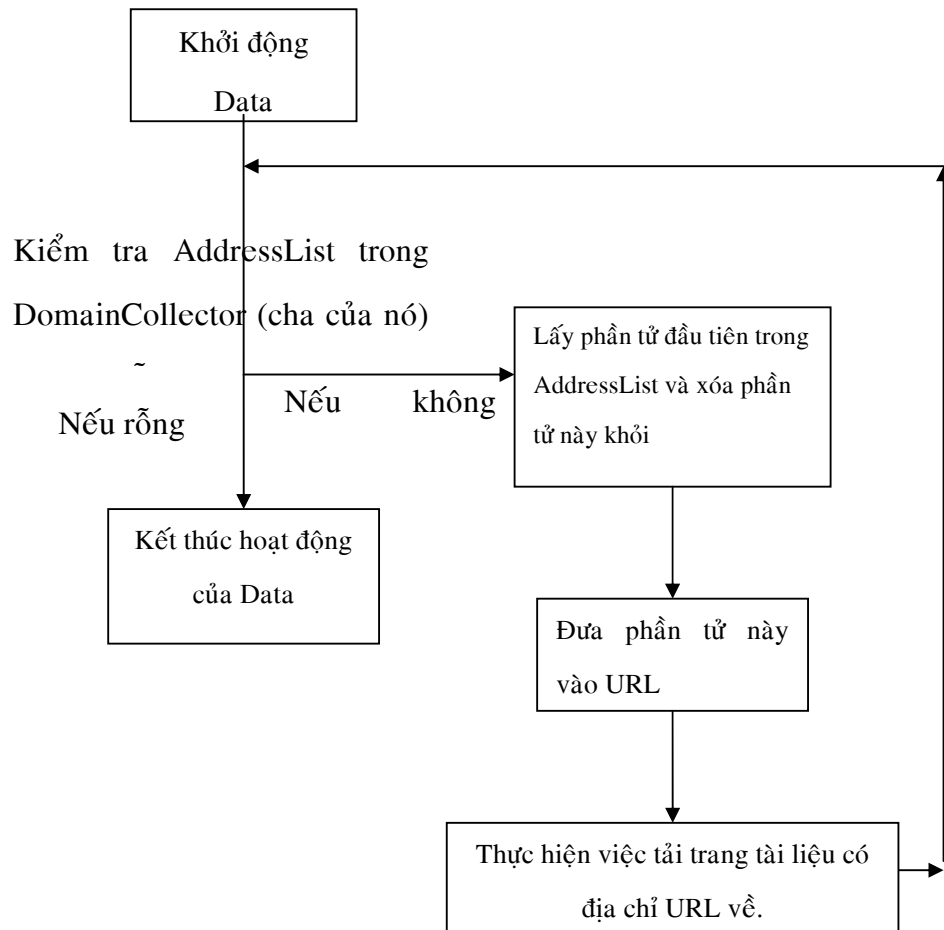
- + Đầu vào : địa chỉ trang WEB cần thu thập.
- + Hoạt động : Đầu tiên, DataCollector sẽ tạo một kết nối đến domain của trang WEB, sau đó kiểm tra tình trạng của kết nối này. Nếu kết nối ở tình trạng tốt, DataCollector sẽ lấy các thông số về trang web như : ContentLength, LastModified và ContentType. Nếu ContentType thuộc một trong các loại sau: application/pdf, application/vnd.ms-excel, application/msword, application/msexcel, application/ms-powerpoint, application/postscript, text/rtf, text/html thì mới lấy về. Sau khi lấy về, thì ghi trang web này xuống hệ thống Tập tin.

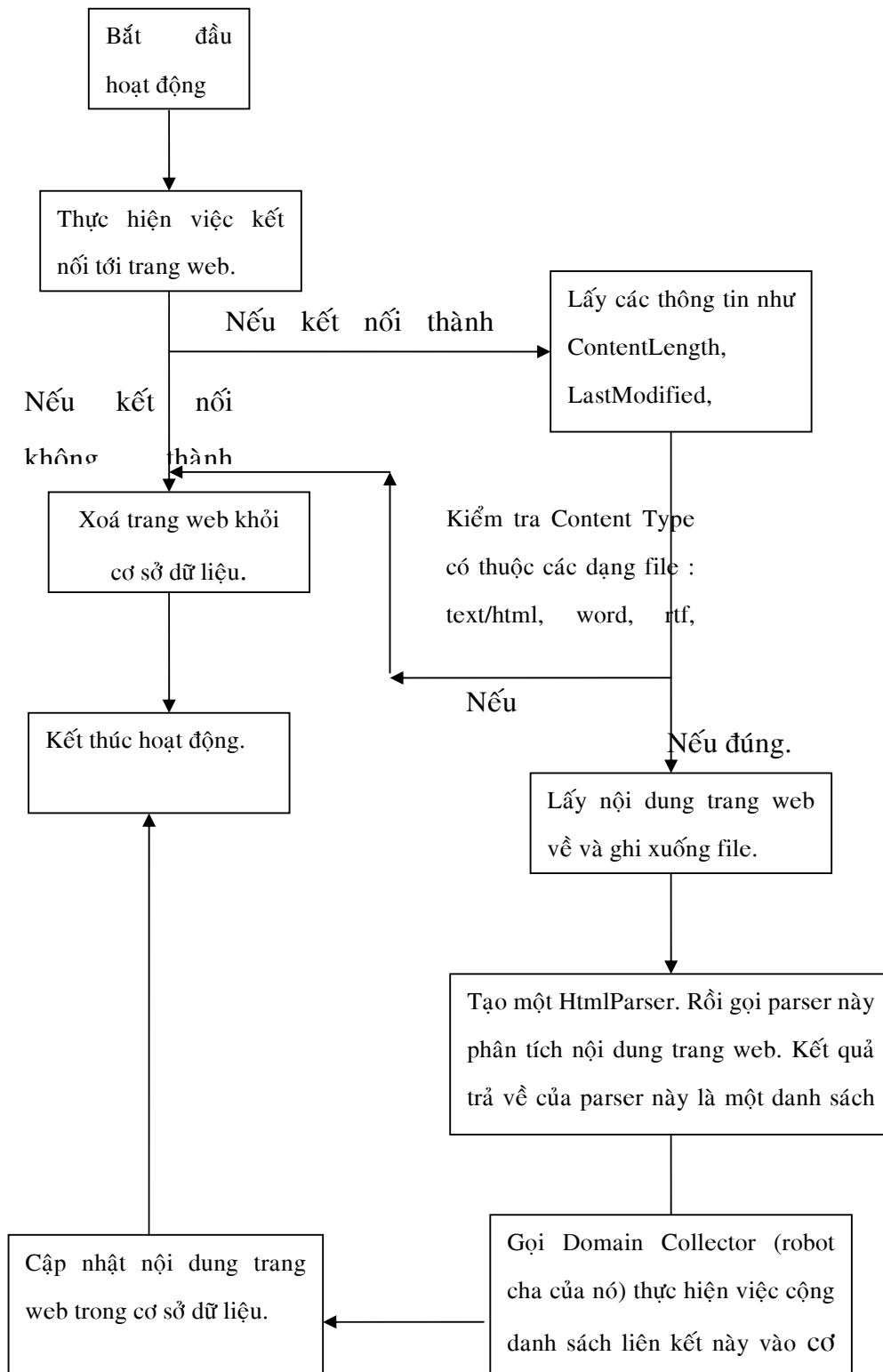
+ Đầu ra : Trang WEB được lấy về.
DataCollector bao gồm các thành phần sau:

- + URL : địa chỉ trang web mà DataCollector cần lấy về.
- + DataBuffer : là vùng đệm chứa nội dung của trang web khi được download về.
- + ContentLength : độ lớn của trang web.
- + LastModified : ngày cập nhật cuối cùng của trang web.
- + Location : vị trí trên file system mà nội dung trang web sẽ được ghi xuống.

DomainCollector : là robot cha của nó, dùng để thực hiện việc giao tiếp trong khi hoạt động.

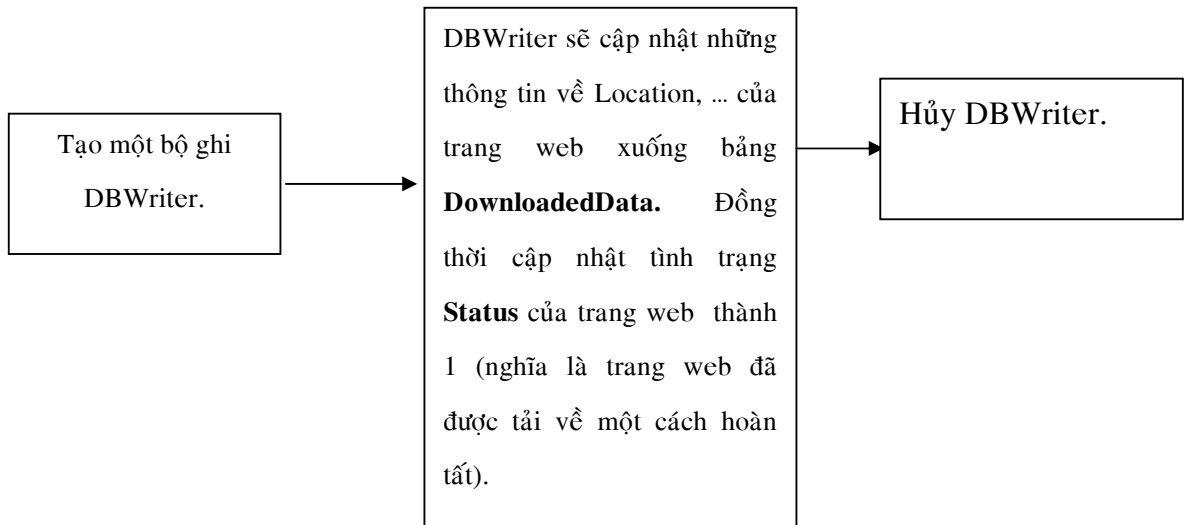
Sơ đồ hoạt động của DataCollector:



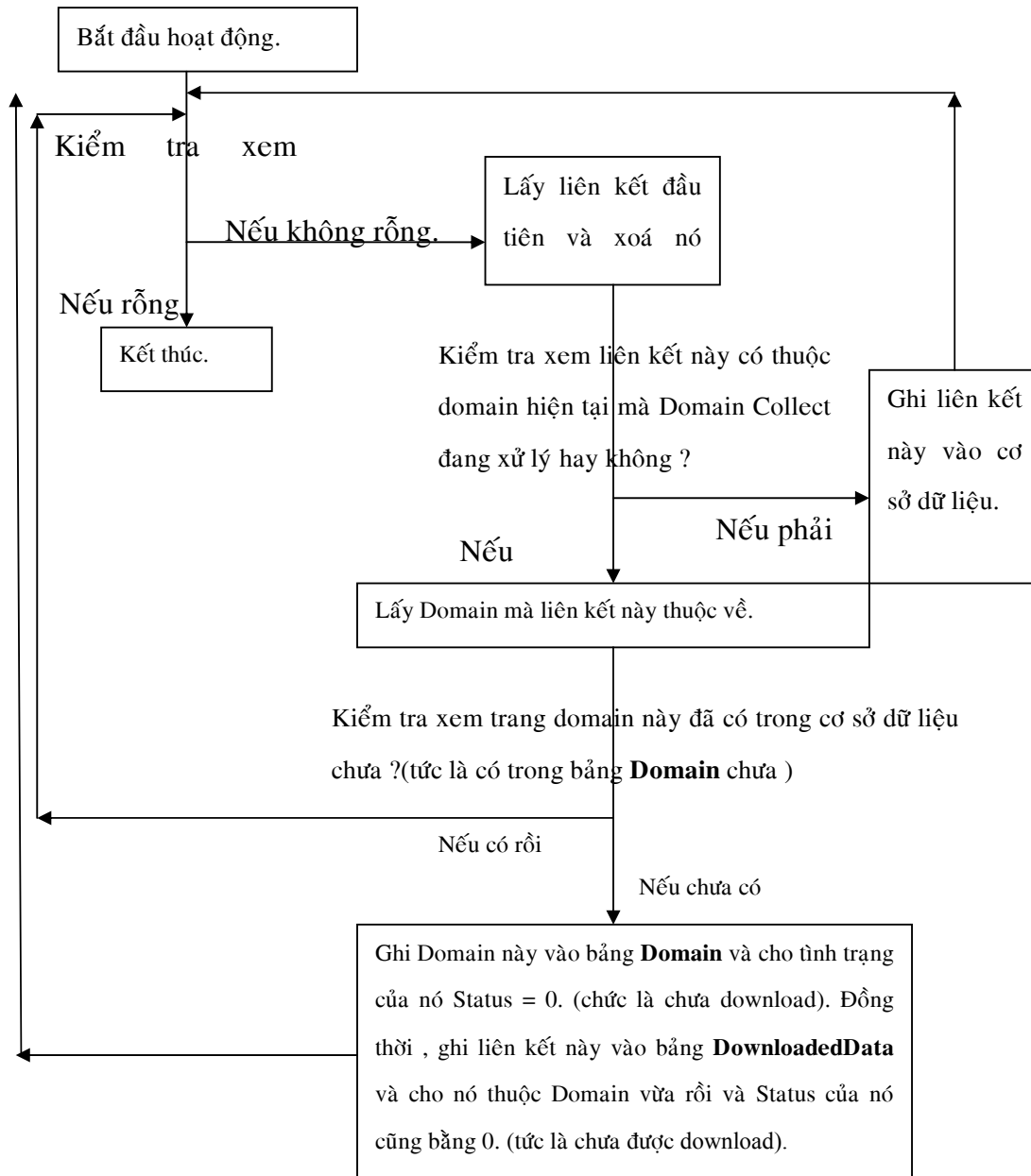


Mô tả công việc tải trang tài liệu về.

Mô tả việc Cập nhật nội dung trang web :

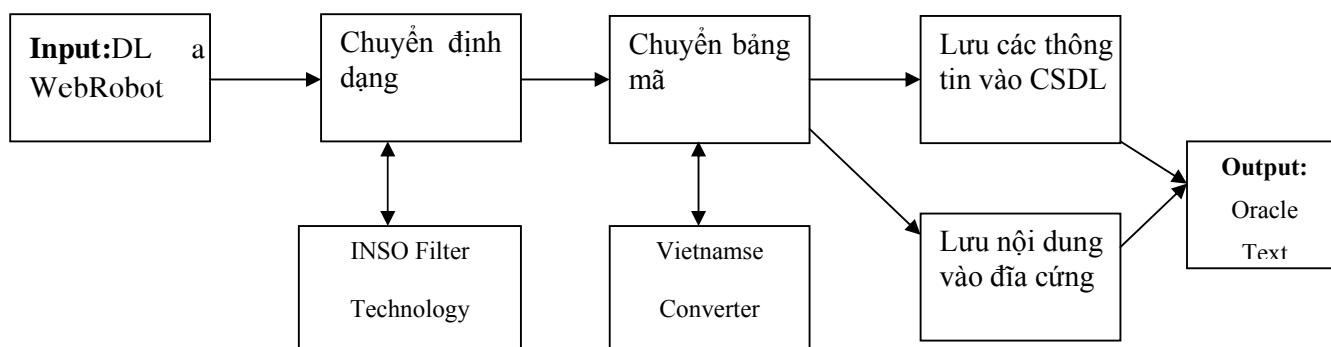


Mô tả công việc Domain Collector cộng danh sách các liên kết vào cơ sở dữ liệu:



3.1.2.2 Module ConvertFile

Toàn bộ công việc của module này được mô tả qua sơ đồ sau:



Công đoạn chuyển định dạng.

Dữ liệu lấy về có thể ở nhiều định dạng, điều này có thể gây khó khăn trong việc index và chuyển mã. Do đó, mục đích của công đoạn này là chuyển tất cả các định dạng về dạng HTML. Sử dụng công nghệ “INSO filter” để lọc tài liệu.

Công đoạn chuyển mã.

Vấn đề ngôn ngữ luôn là vấn đề khó khăn trong việc phát triển các ứng dụng. Riêng đối với tiếng Việt điều này lại càng phức tạp. Tính cho đến nay, tiếng Việt có khoảng 9 bảng mã, bao gồm : TCVN3, VIQR, VNI Windows, VISCII, VPS, BK HMC1, BK HCM2, VietwareX, Vietware F. Cho đến khi Unicode ra đời, thống nhất bảng mã của các ngôn ngữ. Tuy vậy, các tài liệu hiện nay lại ở rất nhiều bảng mã khác nhau. Điều này gây ra rất nhiều khó khăn trong việc index tài liệu. Chính vì vậy, trước khi index, các tài liệu phải được chuyển về một bảng mã chung để có thể index được. Do Oracle hỗ trợ Unicode tổ hợp, và bản thân unicode tổ hợp có rất nhiều lợi điểm như dễ dàng xử lý, kích thước lưu trữ tài liệu nhỏ... nên chúng tôi quyết định sử dụng unicode tổ hợp trong việc lưu trữ và xử lý ngôn ngữ.

Khó khăn lớn nhất trong vấn đề chuyển bảng mã chính là làm sao nhận được bảng mã của tài liệu. Một tài liệu được lưu trữ dưới dạng VNI có thể rất giống với VietwareX, hay một tài liệu lưu trữ dưới dạng TCVN3 lại rất giống với VPS. Một vấn đề kế tiếp chính là chuyển bảng mã sang Windows-1258. Sau khi thu thập các bảng mã và khảo sát để tìm ra những điểm khác biệt, chúng tôi đề ra một thuật giải Heuristic để giải quyết vấn đề này. Theo thống kê của chúng tôi, thuật giải này chỉ đúng với khoảng trên 95% trong việc chuyển đổi, tuy nhiên theo chúng tôi, sai số 5% có thể chấp nhận được.

3.1.3 Thuật giải nhận dạng bảng mã

- **Các đặc trưng của bảng mã:**

Đặc trưng đầu tiên dùng để nhận biết văn bản là unicode. Một văn bản là unicode nếu 2 byte đầu của văn bản là FFFEh hoặc FEFFh.

Đặc trưng thứ hai của bảng mã là số byte lưu trữ ký tự trong bảng mã. Trong một bảng mã, một ký tự có thể được lưu 1 byte, 2 bytes, 3 bytes, hay có khi là 4 bytes, hoặc biến động. Sau đây là bảng thống kê các bảng mã và số byte lưu trữ ký tự :

Bảng mã	Số byte
TCVN3	1 byte
VPS	1 byte
VISCII	1 byte
BK HCM1	1 byte
Vietware-F	1 byte
VNI Windows	2 bytes
BK HCM2	2 bytes
Vietware-X	2 bytes
Windows-1258	2 bytes
UTF-8	Biến động

Đặc trưng thứ ba, quan trọng nhất, để nhận biết bảng mã chính là sự ánh xạ khác nhau của ký tự. Ví dụ : chữ “Â” trong bảng mã TCVN3 có giá trị 162, còn trong bảng mã VPS có giá trị là 194. Đa số các ký tự trong các bảng mã khác nhau sẽ có ánh xạ khác nhau, tuy nhiên có một số ký tự khác nhau của các bảng mã khác nhau lại có cùng một giá trị ánh xạ (ví dụ : chữ ‘oả’ có giá trị trong bảng mã unicode là E16F, và mã này sẽ trùng với chữ ‘ố’ trong bảng mã VNI Windows...) . Số lượng những ký tự như vậy, tuy không nhiều nhưng sẽ gây ra sự nhập nhằng trong việc nhận dạng bảng mã. Để khử sự nhập nhằng này, chúng tôi sử dụng một Heuristic sau.

- ***Heuristic để khử nhập nhằng bảng mã:***

“Khi có sự nhập nhằng về bảng mã của ký tự, ký tự sẽ thuộc về bảng mã của ký tự trước đó”.

Chúng ta có thể thấy rằng, có rất ít văn bản sử dụng nhiều bảng mã cùng một lúc. Do đó, heuristic này sẽ cho phép chọn những bảng mã có nhiều ký tự thuộc về bảng mã đó với I vọng rằng : ký tự này sẽ thuộc về bảng mã giống như những ký tự trước đó. Tất nhiên với heuristic này, độ chính xác sẽ không thể là 100%, nhưng do hầu hết các văn bản đều sử dụng một bảng mã trong toàn bộ văn bản, nên độ chính xác, theo thống kê của chúng tôi là khoảng >95%.

- ***Thuật giải nhận dạng bảng mã:***

B1 : Với mỗi bảng mã, làm công việc từ B2, B5.

B2 : Đọc một ký tự C ứng với bảng mã.

B3 : Nếu đây C là ký tự Western thì qua B6.

B4 : Nếu C là ký tự hợp lệ và số byte đọc được ≥ 1 , qua bước 6.

B5 : Nếu phát hiện sự nhập nhằng, lấy bảng mã trước rồi ánh xạ C.

B6 : Lưu lại bảng mã phát hiện được.

B7 : Đưa ký tự C vào kết quả. Quay lại bước 1.

- ***Thực hiện việc chuyển mã:***

Sau khi đã xác định được bảng mã, chúng ta đã có thể thực hiện việc chuyển mã. Công việc chuyển đổi bảng mã bao gồm chuyển đổi các ký tự và chuyển tag META, tag FONT của HTML. Công việc chuyển tag đòi hỏi phải có 1 parser đơn giản.

Ghép nối hai công đoạn chuyển định dạng và chuyển bảng mã.

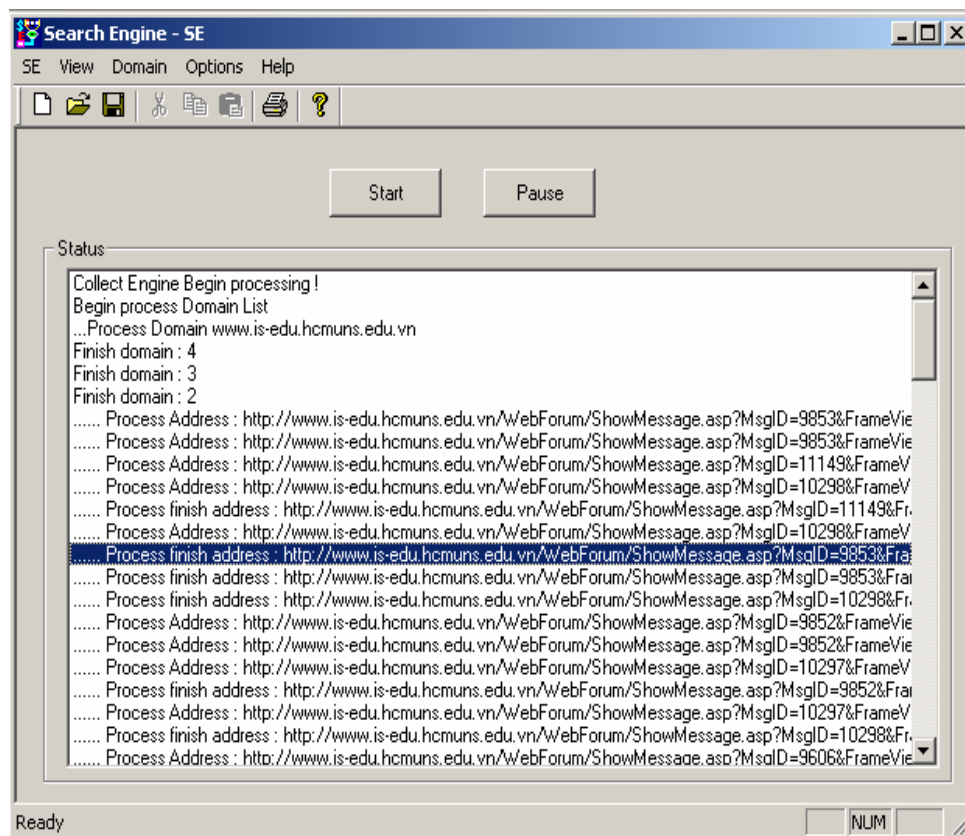
ở module robot, thông tin được lưu trữ trong bảng DOWNLOADEDATA. Bảng này có thuộc tính STATUS cho biết trang này đã được chuẩn (chuyển định dạng, chuyển bảng mã) hay chưa. Do đó module này chỉ xử lý những tài liệu có STATUS = 0, sau đó đưa dữ liệu này vào bảng INDEXEDDATA để chuẩn bị thực hiện việc index, sau đó cập nhật lại thuộc tính STATUS trong bảng DOWNLOADEDATA.

3.2 CÀI ĐẶT HỆ THỐNG.

3.2.1 Tổ chức Các Giao diện Module WebRobot.

Gồm các màn hình sau :

✚ Màn hình chính :

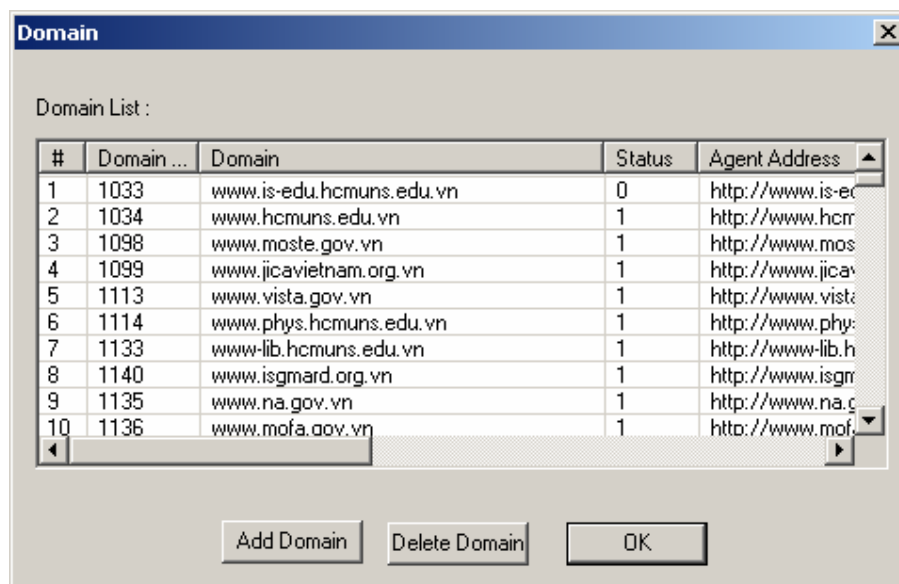


Nút **Start** để bắt đầu (Start) chạy hoặc chạy tiếp (Resume).

Nút **Pause** để tạm ngưng hoạt động của WebRobot.

Chọn **Domain** | **Domain Monitor**, sẽ xuất hiện màn hình giám sát các domain hiện thời :

✚ Các Màn hình tạo, thao tác trên Domain List, thu thập thông tin:

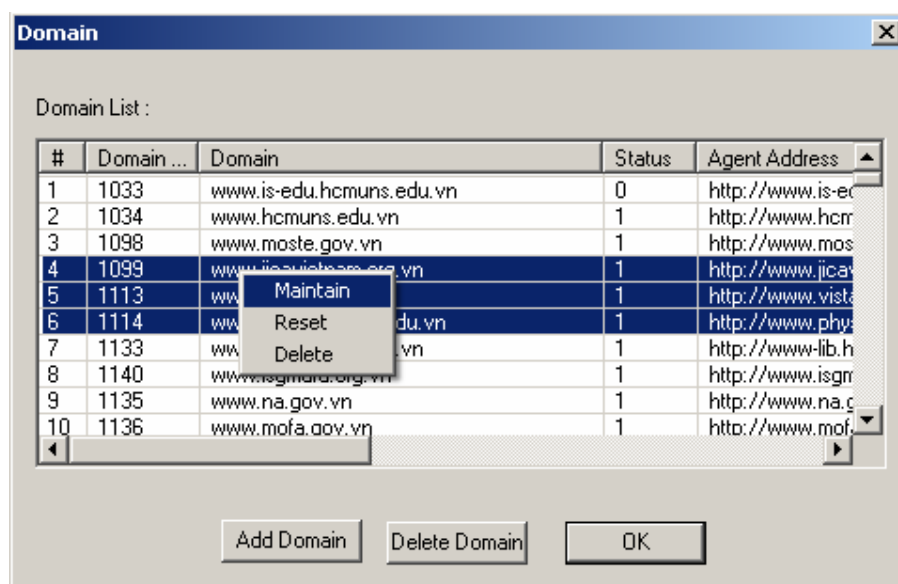


Nút **Add Domain** để thêm một Domain mới vào danh sách các domain mà web robot đã tìm thấy.

Nút **Delete Domain** để xóa Domain đó ra khỏi Domain mà WebRobot đã tìm thấy, tức là WebRobot không còn nhận biết đến sự tồn tại của Domain này.

Nút **OK** để thoát khỏi màn hình giám sát các Domain.

Nếu ta chọn các Domain bên trong danh sách, và nhấn chuột phải, một context menu sẽ hiện lên :

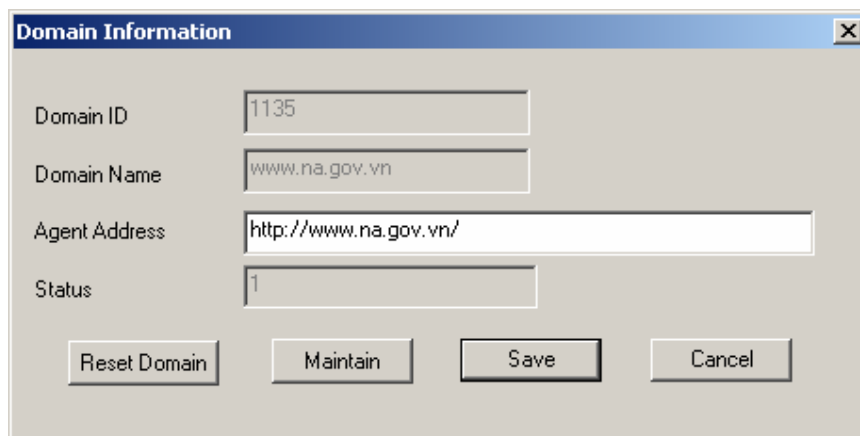


Chọn mục chọn **Maintain** của Menu : tức là ta cho WebRobot bảo trì lại những Domain này nhằm mục đích thu thập thêm các thông tin mới của Domain này.

Chọn mục chọn **Reset** của Menu : tức là ta cho WebRobot xóa hết tất cả những thông tin đã được tải về và thực hiện lại quá trình tải các trang web của domain này này.

Chọn mục chọn **Delete** của Menu: tức là ta cho WebRobot xóa Domain này khỏi danh sách các Domain mà WebRobot nhận biết.

Khi ta click đúp vào một domain trong danh sách các Domain trên màn hình Domain Monitor, sẽ xuất hiện màn hình sau :



Trong màn hình này cho ta nhập các thông tin về Domain mà mình đã chọn ở bước trước.

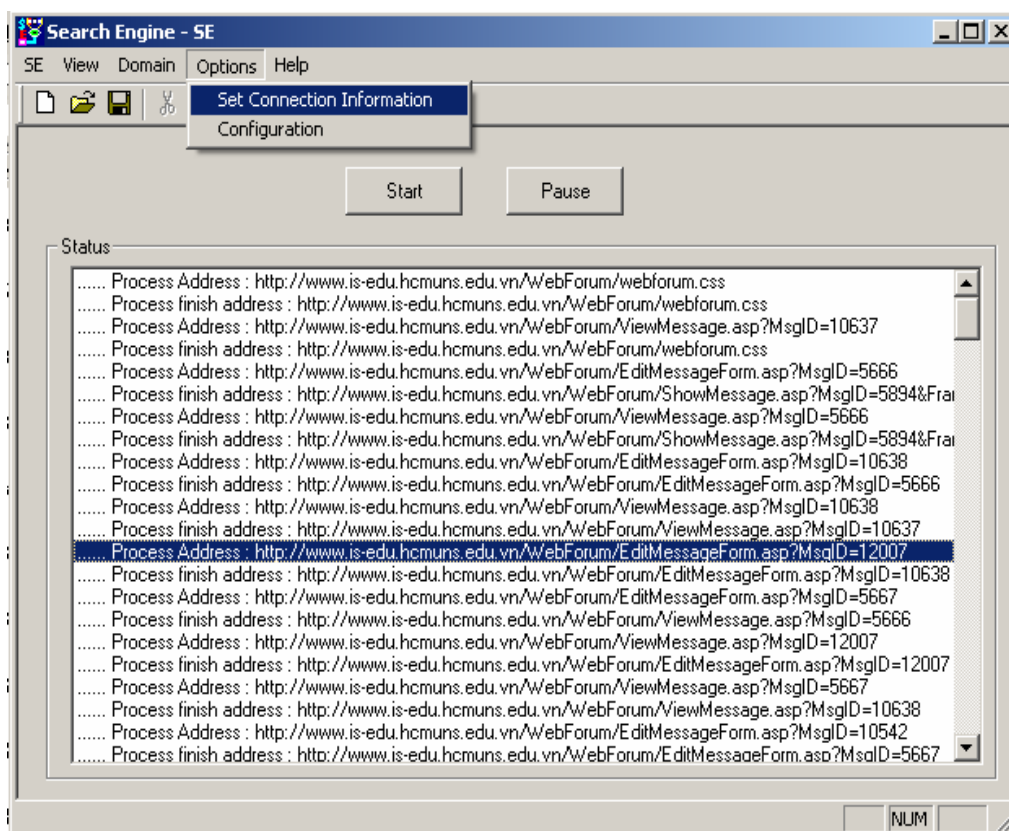
Nếu ta nhấn Save, tức là ta chấp nhận những giá trị mà ta vừa nhập vào màn hình.

Nếu ta nhấn Cancel, tức là ta không chấp nhận những giá trị mà ta vừa nhập vào màn hình.

Nếu ta nhấn Reset Domain, tức là ta cho WebRobot xóa hết tất cả những thông tin đã được tải về và thực hiện lại quá trình tải các trang web của domain này.

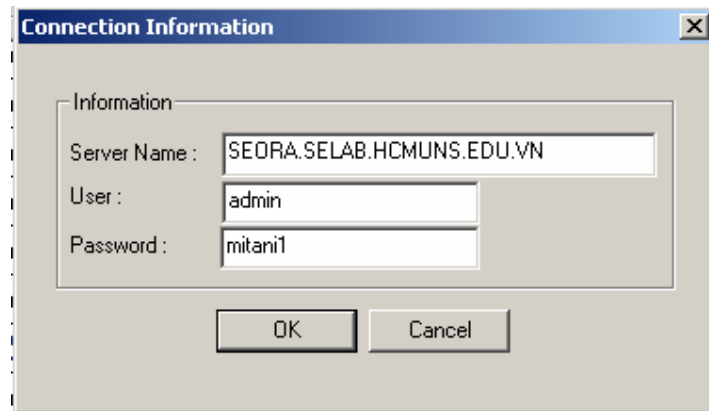
Nếu ta nhấn Maintain, tức là ta cho WebRobot bảo trì lại những Domain này nhằm mục đích thu thập thêm các thông tin mới của Domain này.

Trở lại màn hình chính, nếu ta chọn **Options|Set Connection Configuration :**

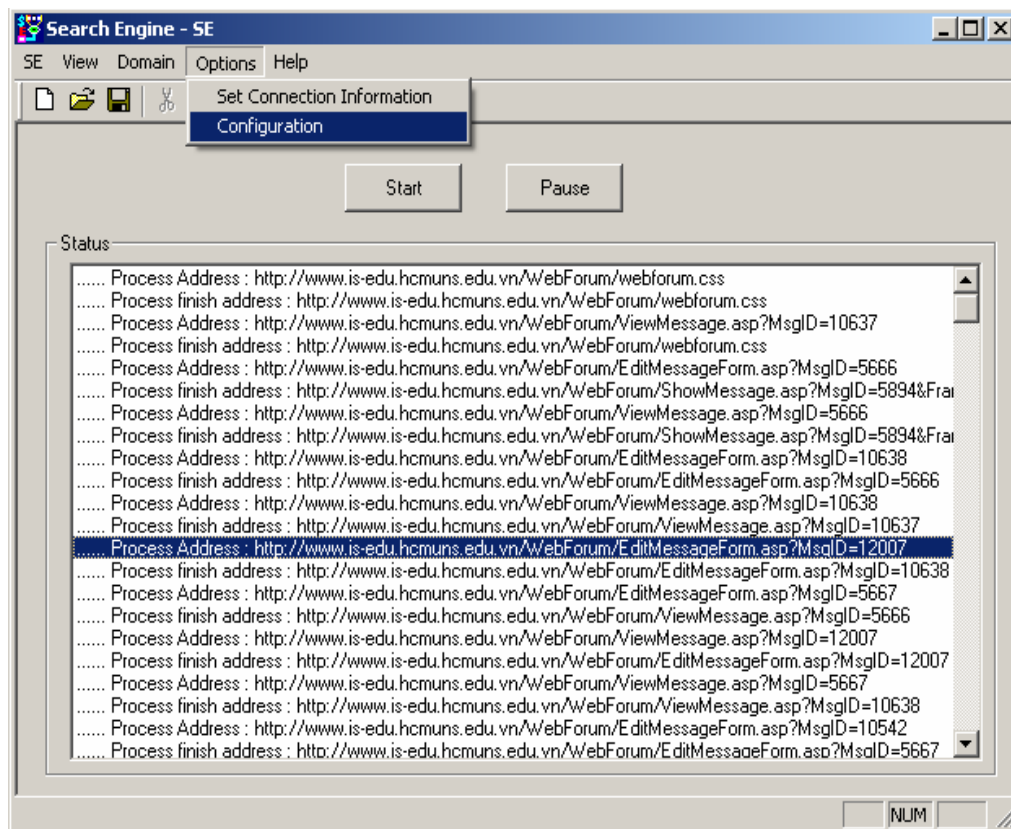


thì màn hình sau sẽ hiện lên cho ta thay đổi các thông số kết nối CSDL:

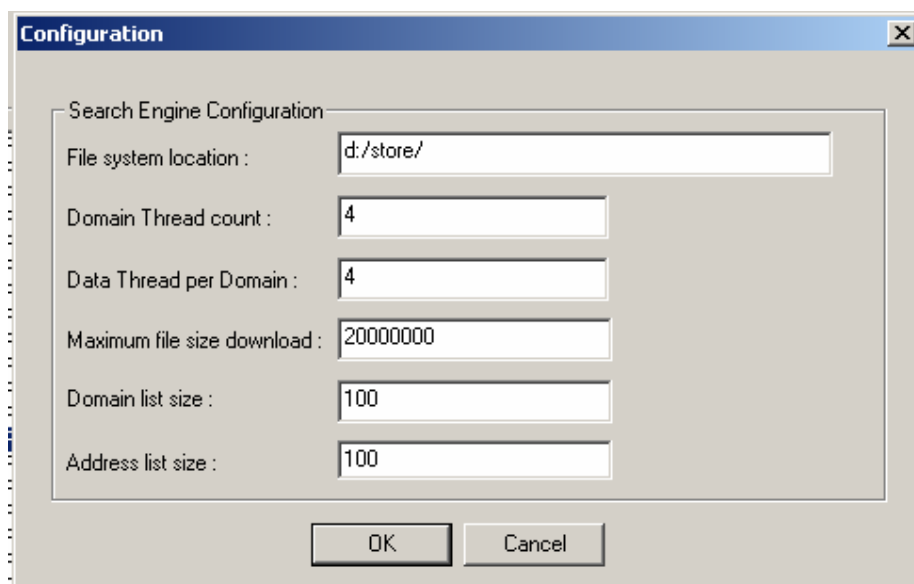
Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc
lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"



✚ Thay đổi cấu hình Hệ thống ta chọn **Options | Configuration** :



thì màn hình sau sẽ hiện lên :

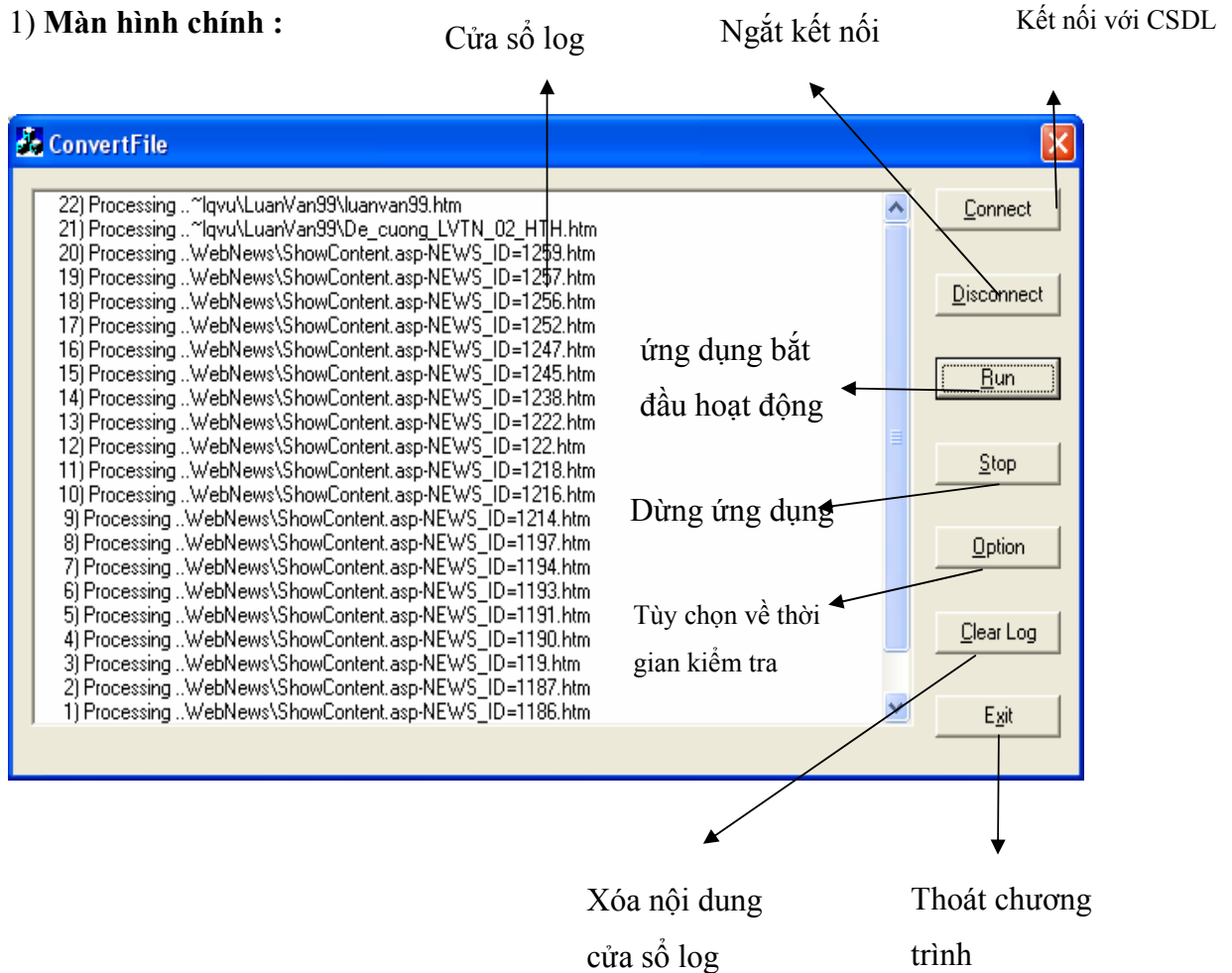


cho ta thay đổi các thông số về hoạt động của WebRobot:

- File system Location : vị trí lưu các trang web được tải về.
- Domain Thread count : số DomainCollector cùng lúc hoạt động.
- Data Thread per Domain : Số DataCollector cùng hoạt động tìm kiếm một Domain.
- Maximum File size download : kích thước tối đa của trang web mà ta tải về.
- Domain List size: kích thước hàng đợi Domain List khi webrobot hoạt động.
- Address List size: kích thước hàng đợi Address List khi webrobot hoạt động.

3.2.1.1 Module ConvertFile:

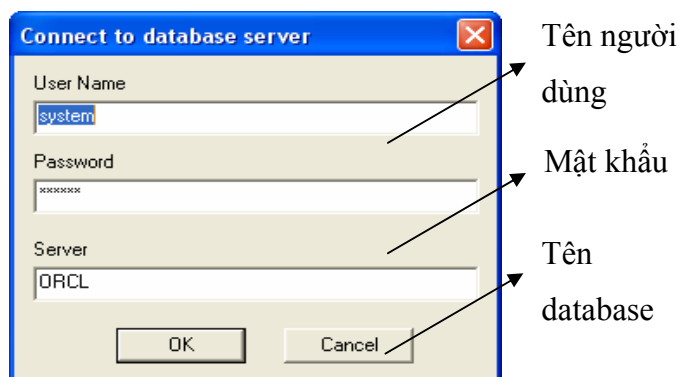
1) Màn hình chính :



Để thực hiện việc chuyển đổi tập tin (file), phải thực hiện các thao tác sau

- Nhấn nút **Connect**: kết nối với CSDL (xem phần hướng dẫn “Màn hình kết nối CSDL”.
- Sau khi thực hiện việc kết nối, hiệu chỉnh tùy chọn cho thích hợp (xem phần hướng dẫn “Màn hình tùy chọn thời gian”
- Nhấn nút **Run** để bắt đầu việc chuyển đổi tập tin.
- Trong quá trình chuyển đổi, trạng thái của chương trình được xuất ra trong cửa sổ nhật ký. Nhấn nút **Clear log** để xóa nhật ký.
- Trong quá trình chuyển đổi, có thể nhấn nút **Stop** để dừng việc chuyển đổi, sau đó có thể nhấn nút **Run** để tiếp tục việc chuyển đổi.
- Sau khi thực hiện xong việc chuyển đổi, có thể nhấn nút **Disconnect** để chấm dứt kết nối với CSDL.

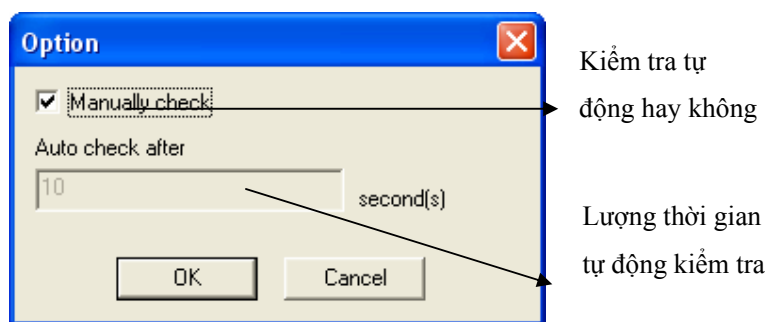
2) Màn hình kết nối với cơ sở dữ liệu:



Để thực hiện việc kết nối với cơ sở dữ liệu, phải thực hiện các thao tác sau:

- Nhấn nút Connect trong màn hình chính (xem phần hướng dẫn “Màn hình chính”)
- Trong màn hình kết nối, nhập tên tài khoản (User) và mật khẩu (Password).
- Nhập tên của máy chủ muốn kết nối (server).
- Nhấn nút **OK** để thực hiện kết nối, nhấn nút **Cancel** để đóng màn hình.

3) Màn hình tùy chọn thời gian:



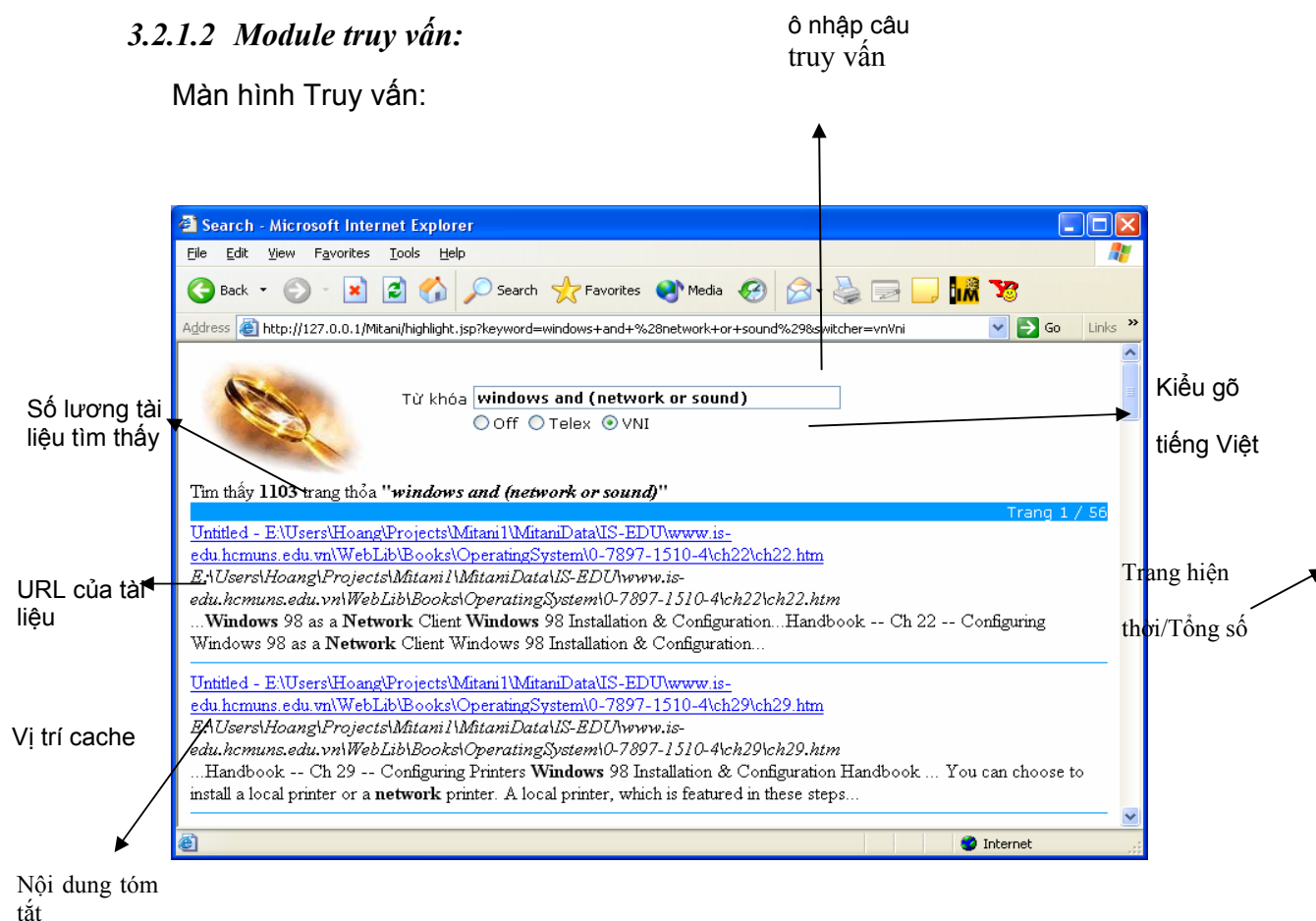
Để hiệu chỉnh tùy chọn, có thể làm các thao tác sau:

- Nhấn vào hộp kiểm (checkbox) “Manually check” để xác định chế độ kiểm tra tự động hay thủ công (manual).
- Check vào “Manually check” có nghĩa là thực hiện chế độ kiểm tra thủ công, mỗi lần kiểm tra phải nhấn nút “Start” trong màn hình chính.

- Không check vào “Manually check” có nghĩa là thực hiện chế độ kiểm tra tự động, nhập số giây giữa mỗi lần kiểm tra vào ô “Auto check after”. Lưu ý đơn vị tính bằng giây.

3.2.1.2 Module truy vấn:

Màn hình Truy vấn:



Để thực hiện việc truy vấn, có thể làm các thao tác sau:

- Nhập từ khóa muốn tìm vào ô “Từ khóa”, rồi nhấn phím ENTER. VD : “windows”.
- Lưu ý có thể nhập các toán tử and, or hoặc nháy kép để tìm chính xác. VD : “windows and (network or sound)”
- Bên dưới màn hình chính là kết quả với địa chỉ và phần tóm tắt kèm
- Nhấn vào các địa chỉ để xem tài liệu trên Internet.

3.3 Kết quả thử nghiệm.

Hệ thống S.E này đã được thử nghiệm, kết quả bước đầu như sau:

- **Module WEBROBOT:** Việc thu thập và phân tích các trang Web đã được thử nghiệm trên hơn 500 miền khác nhau. Tổng số trang Web tải về trên 80000 trang. Tỷ lệ thành công của WebRobot vào khoảng 96%.
- **Module ConvertFile:**
 - Số lượng tài liệu chuyển đổi : 22495 tài liệu (HTML, Tài liệu Word (.doc), PDF)
 - Tổng thời gian : 30 phút
 - Số lượng lỗi : 90
- **Module Truy vấn:**
 - Hỗ trợ bộ gõ tích hợp sẵn hay bộ gõ rời.
 - Hỗ trợ truy vấn local tại máy chủ hay trực tiếp qua các URL
 - Khi truy vấn module này tìm luôn nhưng trang có từ gần nghĩa.

Để tìm kiếm ta gõ URL vào trình duyệt, hiện module truy vấn đã thử nghiệm trên 3 trình duyệt và đều chạy tốt. Ba trình duyệt đó là:

 Internet Explorer

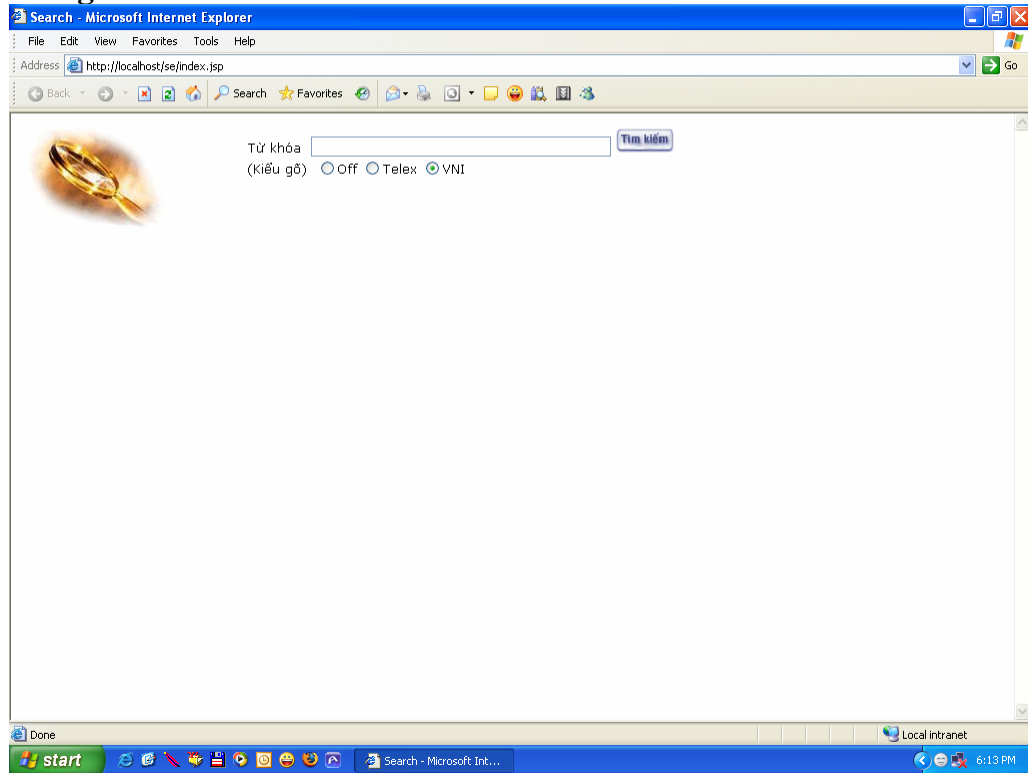
 FireFox

 Avant

Sau đây là giao diện của trang tìm kiếm :

Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"

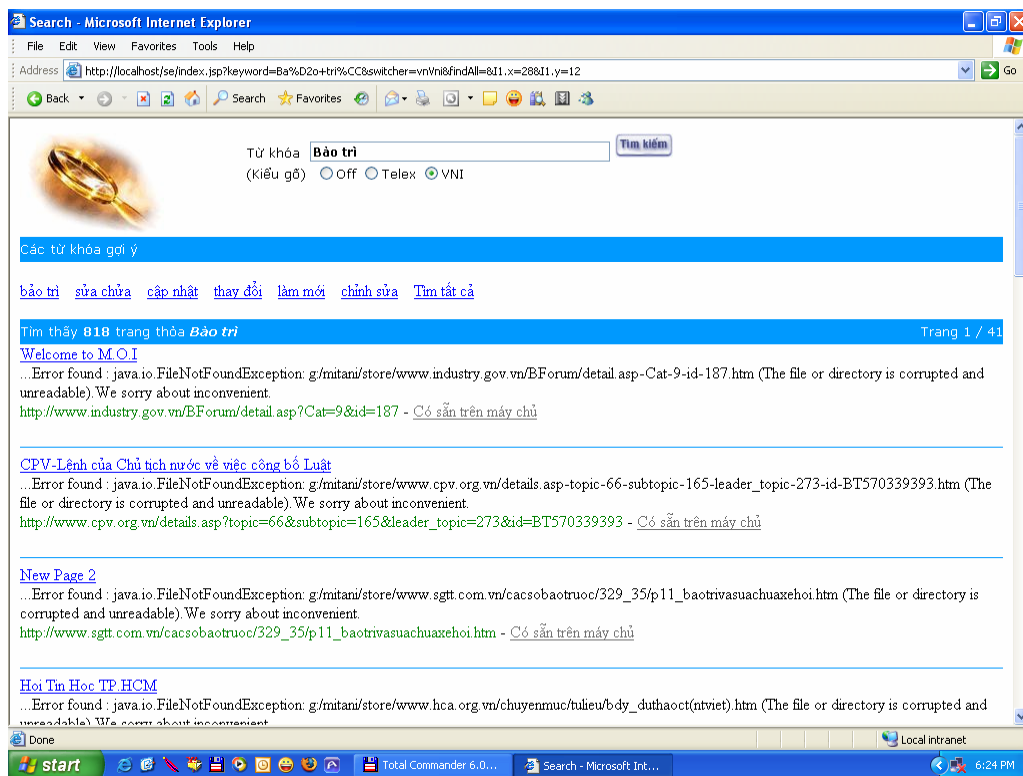
Trang tìm kiếm



Muốn tìm kiếm các trang có chứa từ khóa bảo trì ta gõ vào ô **Từ khóa** chữ **"Bảo trì"** và nhấn vào nút **tìm kiếm** hay nhấn **enter**, kết quả như sau:

Ghi chú: Trang tìm kiếm hỗ trợ cả bộ gõ tích hợp sẵn hay bộ gõ rời.

Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"



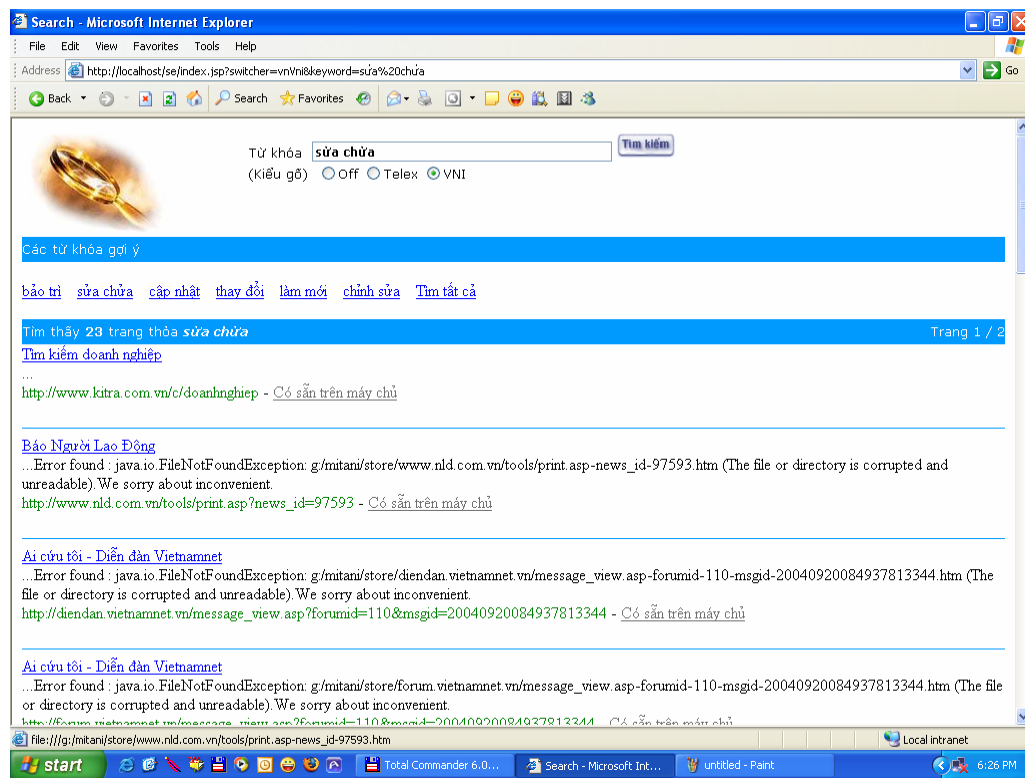
Trang trên liệt kê kết quả tìm kiếm là các trang có chứa từ khóa cần tìm kiếm "Bảo trì" mà module WebRobot đã thu thập về và các từ gần nghĩa với từ "Bảo trì" như **sửa chữa, cập nhật, thay đổi, làm mới, chỉnh sửa** và cuối cùng là **"tìm tất cả"**.

Click vào hyperlink để xem các trang kết quả thông qua URL trực tiếp đến server chứa web gốc hoặc click vào link **"Cố sẵn trên máy chủ"** để xem offline trang tìm kiếm mà module WebRobot đã thu thập về.

Có 818 trang có chứa từ khóa bảo trì.

Có 23 trang có chứa từ khóa sửa chữa.

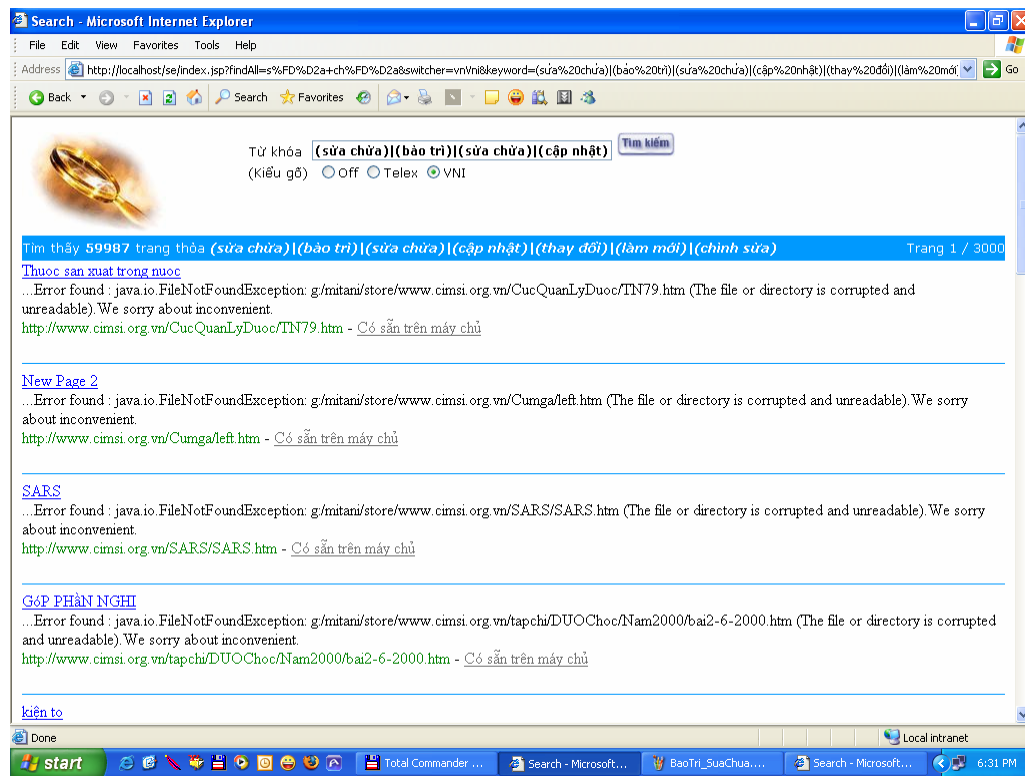
Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"



Trang sau là kết quả sau khi click vào hyperlink "tìm tất cả".

Có 59.987 trang chứa các từ khóa **bảo trì, sửa chữa, cập nhật, thay đổi, làm mới, chỉnh sửa**.

Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"



KẾT LUẬN.

So với mục tiêu ban đầu đề ra, nhóm nghiên cứu đề tài đã thực hiện hoàn chỉnh nội dung nghiên cứu, xây dựng được một từ điển ngữ nghĩa về lĩnh vực công nghệ thông tin, thu thập từ Internet gần 80 Gigabytes dữ liệu, và đã thiết kế được một Search Engine hỗ trợ và đáp ứng nhu cầu tìm kiếm thông tin từ InterNET trong lĩnh vực công nghệ thông tin, theo từ khóa tiếng Việt có hỗ trợ thêm các từ gần nghĩa.

So với các Search Engine tiếng Việt khác đã có, kể cả GOOGLE (hiện nay cũng đã hỗ trợ tiếng Việt, nhưng chỉ cho từ khóa đơn, và đối với cụm từ, (coi như là từ đơn và có sử dụng toán tử OR), Search Engine mà đề tài xây dựng có vượt trội hơn là tìm thêm những từ gần nghĩa, và Dữ liệu có thể được thu thập từ Internet (24/24 giờ), được lưu trữ trên Server tại chỗ, và Index định kỳ, nên việc tìm kiếm thông tin có khả năng sẽ nhanh hơn.

Tuy đây chỉ là kết quả bước đầu, nhưng đã đáp ứng nhu cầu tìm kiếm thông tin và có thể phát triển, mở rộng cho nhiều lĩnh vực ứng dụng khác.

PHỤ LỤC

PHỤ LỤC 1. BẢNG TÓM TẮT ĐẶC TRƯNG CỦA MỘT SỐ S.E NƯỚC NGOÀI.

(Trích từ tài liệu thống kê của Infopeople Project, cập nhật ngày 29/01/2003)

S.E	Cơ sở dữ liệu	Toán tử Logic	Khả năng tìm kiếm	Hỗ trợ
Google www.google.com Hỗ trợ tìm kiếm nâng cao. Hệ thống danh mục theo đề mục, mở (Subject , Open Directory)	Toàn bộ văn bản của các trang Web, PDF, MS.Office, PostScript, Lotus WordPerfect, ...	AND (mặc định) O.	Dùng * để rút gọn (thay thế từ trong cụm từ). Dùng " " tìm cụm từ Tìm theo vùng Tìm các trang liên quan	Hỗ trợ kiểm tra chính tả. Tìm hình ảnh. Tìm tóm tắt. Tìm kiếm theo nhiều ngôn ngữ khác nhau.
AllTheWeb www.allTheWeb.com Hỗ trợ tìm kiếm nâng cao. Không có Hệ thống danh mục theo đề mục	Toàn bộ văn bản của các trang Web, PDF, MS.Office, PostScript, Lotus WordPerfect, ...	AND (mặc định) OR, ANDNOT, RANK	Không rút gọn, Dùng " " tìm cụm từ Tìm theo vùng.	Tìm kiếm mở rộng: hình ảnh và Video.

Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"

<p><u>AltaVista</u> <u>www.altavista.com</u> Hỗ trợ tìm kiếm nâng cao (rất tốt) Hệ thống danh mục theo đề mục, mở <u>Subject, Open Directory</u>)</p>	<p>Toàn bộ văn bản của các trang Web.</p>	<p>AND (mặc định) OR.</p>	<p>Dùng “ ” tìm cụm từ . Dùng * để rút gọn. Phân biệt chữ Hoa, chữ thường.</p>	<p>Tự động xác định cụm từ và kiểm tra chính tả. Tìm hình ảnh, âm thanh, Video và tin tức. Tìm kiếm theo nhiều ngôn ngữ khác nhau.</p>
<p><u>Teoma</u> <u>teoma.com</u> Hỗ trợ tìm kiếm nâng cao. Phân loại dựa trên # tiêu đề cụ thể của các trang liên kết</p>	<p>Toàn bộ văn bản của các trang Web.</p>	<p>AND (mặc định) Dùng “” để tìm các từ thông dụng.</p>	<p>Không rút gọn Dùng “ ” tìm cụm từ . Tìm kiếm theo các giới hạn như ngày, ngôn ngữ, vị trí, độ sâu liên kết của trang WEB.</p>	<p>Có thể gom nhóm các kết quả.</p>

PHỤ LỤC 2. BẢNG TÓM TẮT ĐẶC TRƯNG MỘT SỐ META-S E NƯỚC NGOÀI

Meta-S. E	Cơ sở dữ liệu	Toán tử Logic	Khả năng tìm kiếm	Hỗ trợ
Ixquick ixquick.com	Searches AltaVista, Ask Jeeves/Teoma, MSN, Yahoo & more.	Dùng tất cả các toán tử tìm kiếm mà các công cụ tìm kiếm sử dụng.	Hỗ trợ tìm kiếm tin tức, file MP3, file ảnh	Đặc trưng: tập hợp và phân hạng các kết quả, hạn chế sự trùng lặp thông tin
Vivisimo vivisimo.com	Searches AltaVista, MSN, Lycos, BBC, & more. (Select in Advanced Search).	AND (mặc định), OR, - dùng để loại trừ.	Hỗ trợ tìm kiếm theo chủ đề, như: tin tức, kinh doanh, kỹ thuật, thể thao.	Thực hiện gom nhóm các kết quả. Hoạt động hiệu quả trong tìm kiếm các sự kiện thời sự, phổ biến.
Ask Jeeves www.ask.com Also has Ask Jeeves for Kids . www.ajkids.com	Đáp ứng hàng triệu các yêu cầu từ các site ước lượng.	Chỉ dùng ngôn ngữ tự nhiên, không dùng các toán tử.	Dùng dấu nháy kép để tìm kiếm cụm từ trong Teoma.	Hoạt động hiệu quả đối với các yêu cầu đơn giản.

PHỤ LỤC 3. BẢNG TÓM TẮT MỘT SỐ HỆ THỐNG DANH MỤC (SUBJECT DIRECTORIES)

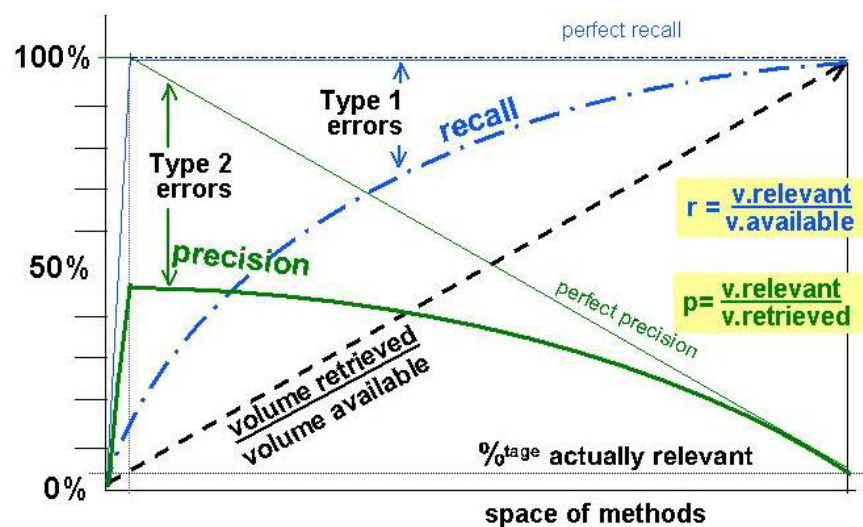
Hệ thống danh mục	Cơ sở dữ liệu	Toán tử Logic	Khả năng tìm kiếm	Hỗ trợ
<u>Librarians' Index to the Internet</u> <u>lii.org</u> <u>Advanced Search.</u>	Tài nguyên hữu ích cho người dùng thư viện, thủ tục thực hiện ước lượng và đánh giá. (khoảng 10K).	AND (mặc định), OR, NOT, và ().	Dùng nháy kép để tìm cụm từ. Tự động Stems automatically (can turn off) Dùng * để rút gọn Tìm kiếm nâng cao theo: chủ đề, tiêu đề, mô tả, URL, tác giả, nhà xuất bản.	Entries Categorized: Directories Databases Specific Resources Tìm các trang theo chủ đề, tiêu đề, ...
<u>Yahoo</u> <u>www.yahoo.com</u> Mặc định sẽ tìm kiếm bằng Google khi tìm trên Yahoo không có kết quả. <u>Advanced Search.</u>	Xem xét các trang web (khoảng 2M)	+ yêu cầu.	Dùng nháy kép để tìm cụm từ. Dùng * để rút gọn Tìm kiếm theo lĩnh vực: t: tìm tiêu đề; u: tìm theo URL	Tin tức: hàng ngày Chứng khoán: v.v. Thể thao: tính điểm, v.v. Trình bày nội dung sơ lược cho Maps, Weather Ratings
<u>Encyclopedia Britannica</u> <u>www.britannica.com</u>	Chủ đề & phân vùng riêng Yahoos - Kids (Yahooligans), Japan, SF Bay, v.v..	- loại trừ	Dùng nháy kép để tìm cụm từ. Dùng * để rút gọn.	Hỗ trợ tìm Merriam Webster Dictionary và Thesaurus. Sử dụng chính tả theo kiểu Anh.

PHỤ LỤC 4. BẢNG TÓM TẮT ĐẶC TRƯNG CỦA MỘT SỐ S.E TRONG NƯỚC.

S.E	Cơ sở dữ liệu	Toán tử Logic	Khả năng tìm kiếm	Hỗn hợp
NetNam www.pan.vietnam . Com	Toàn bộ văn bản của các trang Web.	Dùng "" để tìm các từ thông dụng.	Dùng " " tìm cụm từ . Phân biệt chữ Hoa và thường.	Sử dụng từ khóa để lọc các Tìm kiếm.
VinaSeek www .vinaseek. Com	Toàn bộ văn bản của các trang Web.	Dùng "" để tìm các từ thông dụng.	Dùng " " tìm cụm từ . Phân biệt chữ Hoa và thường.	Sử dụng từ khóa để lọc các Tìm kiếm.

PHỤ LỤC 5. QUAN HỆ GIỮA ĐỘ CHÍNH XÁC & ĐỘ GỌI LẠI.

Relationships among Precision (p) and Recall (r)



Mật độ thông tin giảm theo chiều hướng về phía bên phải. Còn các phương thức tối ưu nhất ở về phía bên trái.

Các lỗi loại 1 biểu diễn các kết quả không nhận được mà đúng ra phải nhận được theo độ đo recall

Các lỗi loại 2 biểu diễn các kết quả nhận được mà đúng ra là không phải kết quả nếu xét theo độ đo precision

Xét trong không gian web thì những kết quả tìm kiếm có được chỉ là một phần rất nhỏ trong vô số các tài liệu tương thích với nó.

PHỤ LỤC 6. THỐNG KÊ VỀ PHÂN HẠNG CỦA CÁC DOMAIN

Cập nhật ngày 08/7/2003. Các con số trong bảng sau theo mỗi cột thể hiện vị trí phân hạng của các domain dựa trên các từ khoá (tiêu chí đánh giá) bên trái

	Activity	Google Sites	Overture Top Bid	<u>Yahoo Dir</u>	<u>Yahoo Index</u>	<u>Google</u>	<u>MSN</u>	<u>AOL Web</u>	<u>Teoma</u>	<u>Inktomi</u>	<u>dmoz Dir</u>	<u>Lycos</u>	<u>Alta Vista</u>	<u>All The Web</u>	<u>Hot Bot</u>	<u>Ask Jeeves</u>	<u>IWon</u>
<u>website promotion</u>	11051	2150000	-	-	25	26	-	-	-	193	-	-	-	-	193	-	26
<u>S.E optimization</u>	7909	794000	-	-	1	1	56	1	15	4	-	54	-	54	4	15	1
<u>S.E ranking</u>	5092	702000	-	-	1	1	-	1	2	2	-	2	-	2	2	2	1
<u>S.E placement</u>	3414	540000	-	-	18	18	-	-	-	36	-	39	-	39	36	-	18
<u>S.E positioning</u>	2804	390000	-	-	6	7	62	7	1	17	-	5	-	5	17	1	7
<u>S.E marketing</u>	2261	1650000	-	-	10	10	100	10	17	8	-	6	-	6	8	17	10
<u>web site optimization</u>	2304	1110000	-	-	3	3	-	3	39	3	-	3	-	3	3	39	3

Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"

<u>web site</u> <u>ranking</u>	2294	1540000	-	-	6	6	15	7	4	15	-	1	-	1	15	4	6
<u>S.E</u> <u>promotion</u>	1247	939000	-	-	1	1	103	1	-	2	-	34	-	34	2	-	1
<u>keyword</u> <u>ranking</u>	341	396000	-	-	7	7	-	7	1	6	-	-	-	-	6	1	7
<u>S.E opt. tools</u>	256	339000	-	-	1	1	21	1	4	1	-	1	-	1	1	4	1
<u>S.E results</u>	138	2190000	-	-	4	4	12	3	-	8	-	12	-	12	8	-	4
<u>S.E ranking</u> <u>tips</u>	127	141000	-	-	1	1	2	1	6	3	-	1	-	1	3	6	1
<u>S.E advice</u>	126	1100000	-	-	2	2	-	2	1	2	-	1	-	3	2	1	2
<u>S.E</u> <u>relationship</u> <u>chart</u>	124	61300	-	-	1	1	-	1	17	1	1	17	-	17	1	17	1
<u>Bruce Clay</u>	105	400000	-	-	1	1	1	1	1	1	1	1	-	1	1	1	1
<u>bruceclay</u>	77	1880	-	-	1	1	1	1	1	1	1	1	-	1	1	1	1

Đề tài: "Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lĩnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt"

<u>seo code of ethics</u>	73	8040	-	-	1	1	-	1	9	1	-	1	-	2	1	9	1
<u>web promotion advice</u>	71	835000	-	-	1	1	7	1	1	8	-	1	-	1	8	1	1
<i>Top-10 Count:</i>				-	17	17	4	17	11	15	3	12	-	12	15	11	17
<i>Top-20 Count:</i>				-	1	1	2	-	3	2	-	2	-	2	2	3	1
<i>Top-30 Count:</i>				-	1	1	1	-	-	-	-	-	-	-	-	-	1
<i>Top-40 Count:</i>				-	-	-	-	-	1	1	-	2	-	2	1	1	-
<i>Top-50 Count:</i>				-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>Count:</i>				0	19	19	11	17	15	19	3	17	0	17	19	15	19

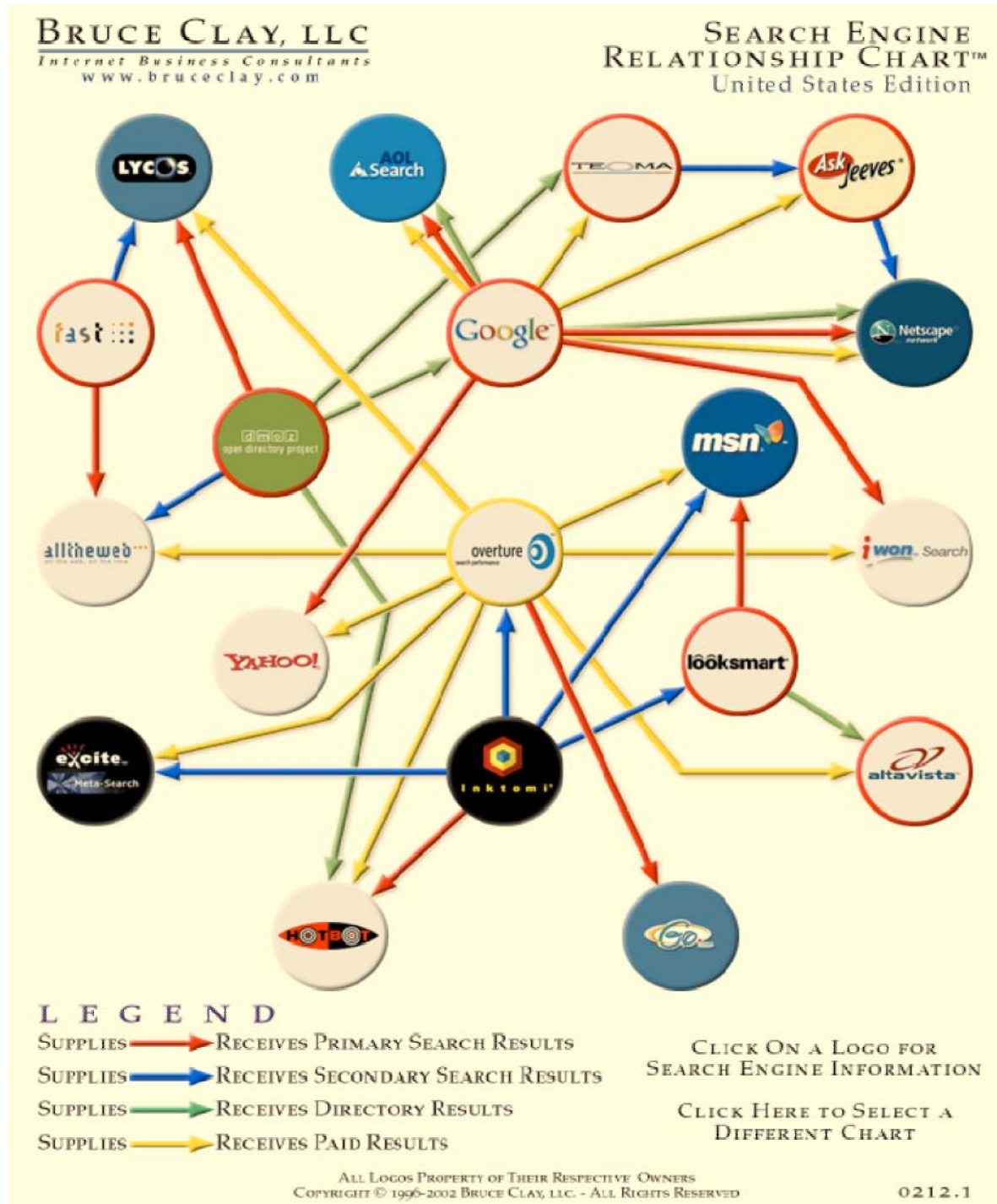
Hoạt động của SE được đánh giá dựa trên các truy vấn hàng ngày cho mỗi từ khoá trên tất cả các SE.

Cơ hội : **266**, tổng số = **1-5: 120 * 1-10: 151 * 11-20: 19 * 21-30: 4 * 31-40: 8 * 41-50: 0**

Tỷ lệ phần trăm = **Top 5: 45.1% * Top 10: 56.8% * Top 20: 63.9% * Top 30: 65.4% * Top 40: 68.4% * Top 50: 68.4%**

SE Traffic Vector™ = 928 (theoretical visitors per day from only the 08/07/2003 rankings)

PHỤ LỤC 7. SƠ ĐỒ QUAN HỆ S.E



PHỤ LỤC 8: CÁC MÃ NGŨ NGHĨA CỦA LDOCE

Ký hiệu	Mô tả
<i>A</i>	Animal
<i>B</i>	Female Animal
<i>C</i>	Concrete
<i>D</i>	Male Animal
<i>E</i>	Solid or Liquid (not gas): S + L
<i>F</i>	Female Human
<i>G</i>	Gas
<i>H</i>	Human
<i>I</i>	Inanimate Concrete
<i>J</i>	Movable Solid
<i>K</i>	Male Animal or Human = D + M
<i>L</i>	Liquid
<i>M</i>	Male Human
<i>N</i>	Not Movable Solid
<i>O</i>	Animal or Human = A + H
<i>P</i>	Plant
<i>Q</i>	Animate
<i>R</i>	Female = B + F
<i>S</i>	Solid
<i>T</i>	Abstract
<i>U</i>	Collective Animal or Human = (Collective + O)
<i>V</i>	Plant or Animal = (P + A)
<i>W</i>	Inanimate Concrete or Abstract = (T + I)
<i>X</i>	Abstract or Human = (T + H)
<i>Y</i>	Abstract or Animate = (T + H)
<i>Z</i>	Unmarked
<i>1</i>	Human or Solid = (H + S)
<i>2</i>	Abstract or Solid = (T + S)
<i>4</i>	Abstract Physical
<i>5</i>	Organic Material
<i>6</i>	Liquid or Abstract = (L + T)
<i>7</i>	Gas or Liquid = (G + L)

PHỤ LỤC 9. TỔNG QUAN VỀ CÔNG NGHỆ ORACLE TEXT ĐỂ PHÁT TRIỂN S.E.

1. Giới thiệu Tổng quát về Oracle TEXT.

Oracle TEXT là công nghệ do Oracle hỗ trợ, ứng dụng vào hệ thống tìm kiếm với phương pháp lập chỉ mục tự động. Oracle Text là một công cụ giúp người dùng có thể xây dựng một Ứng dụng Truy vấn (dạng Tài liệu HTML, XML, tài liệu thuần text, hay tài liệu Microsoft Word) hay Phân loại tài liệu (dạng Tài liệu XML, HTML hay các Tập tin thuần text). Oracle Text cung cấp việc lập chỉ mục, tìm kiếm theo các từ khóa và chủ đề, và các cách trình bày tài liệu được truy vấn. Oracle TEXT hỗ trợ Tìm kiếm theo chủ đề, với Ngôn ngữ sử dụng trong tài liệu là tiếng Anh hay tiếng Pháp (sử dụng toán tử ABOUT). Ngoài ra, có thể sử dụng chức năng này với toán tử ABOUT trong các ngôn ngữ khác ngoài tiếng Anh và Pháp bằng cách sử dụng một cơ sở tri thức của ngôn ngữ đó. Để có thêm các chức năng mở rộng như trình bày văn bản và duy tu từ điển đồng nghĩa, có thể sử dụng các gói Oracle Text PL/SQL.

Để xây dựng một ứng dụng truy vấn, người dùng phải có :

- Một bảng đã có tài liệu
- Một chỉ mục Oracle Text

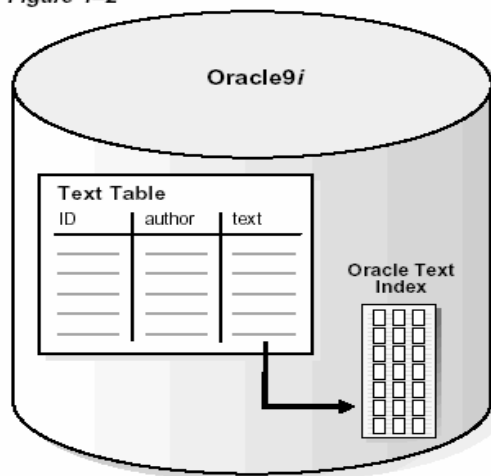
Oracle Text cho phép tạo ra ba loại chỉ mục. Điều này phụ thuộc vào ứng dụng của người dùng và nguồn tài liệu. Người dùng sử dụng câu lệnh CREATE INDEX để tạo ra các loại chỉ mục Oracle Text này, theo Bảng Mô tả sau:

Loại chỉ mục	Loại Ứng dụng	Toán tử truy vấn
CONTEXT	Truy xuất Tài liệu.	CONTAINS
CXTCAT	Truy xuất Tài liệu, với mục đích là để cải tiến hiệu suất câu truy vấn hỗn hợp.	CATSEARCH
CTXRULE	Phân loại tài liệu.	MATCHES

2. Engine lập chỉ mục:

Để truy vấn tập tài liệu của người dùng, người dùng trước hết phải lập chỉ mục cột tài liệu có liên quan của bảng tài liệu. Quá trình lập chỉ mục sẽ phân tích tài liệu thành các token thông thường chính là các từ. Quá trình này sẽ tạo ra loại chỉ mục CONTEXT, loại chỉ mục ghi nhận mỗi token và các tài liệu có chứa token đó. Loại chỉ mục nghịch đảo như vậy cho phép truy vấn trên các từ và các cụm từ.

Figure 1-2



Cấu trúc tổng quát của chỉ mục Oracle Text CONTEXT là một chỉ mục nghịch đảo trong đó mỗi token chứa danh sách các tài liệu (các hàng) có chứa token đó. Ví dụ, sau một thao tác lập chỉ mục ban đầu, từ 'CAT' có thể có một mục từ như sau : CAT DOC1 DOC3 DOC5

Điều này có nghĩa là từ CAT có trong hàng chứa tài liệu một, ba và năm.

Hình trên mô tả một bảng dữ liệu trong Oracle9i cùng với chỉ mục Oracle Text của nó.

Quá trình lập chỉ mục của Oracle Text :

Người dùng khởi tạo quá trình lập chỉ mục với câu lệnh CREATE INDEX. Mục đích là tạo ra một chỉ mục Oracle Text gồm các token theo các tham số và các tham chiếu người dùng đã chỉ định.

Các tham chiếu và tham số đó có thể là:

Đối tượng	Mô tả
Datastore	Tài liệu của người dùng được lưu trữ như thế nào ?
Filter (bộ lọc)	Tài liệu có thể chuyển đổi tài liệu thành text thuần bằng cách nào ?
Lexer (bộ từ vựng)	Ngôn ngữ nào được dùng để lập chỉ mục ?
Wordlist	Có mở rộng cho các câu truy vấn mờ và câu truy vấn từ liên quan ?
Storage	Dữ liệu lập chỉ mục được lưu trữ như thế nào ?
Stoplist	Những từ và những chủ đề nào không được lập chỉ mục ?
Nhóm thành phần	Các phần của tài liệu được định nghĩa như thế nào ?

Ví dụ câu lệnh sau tạo ra một chỉ mục CONTEXT có tên myindex trên cột text của bảng docs

```
CREATE INDEX myindex ON docs(text) INDEXTYPE IS  
CTXSYS.CONTEXT;
```

Sử dụng các mệnh đề tham số trong câu lệnh CREATE INDEX, người dùng có thể cá nhân hóa chỉ mục CONTEXT. Trong mệnh đề tham số người dùng có thể chỉ ra nơi dữ liệu được lưu trữ, chỉ ra cách người dùng muốn lọc tài liệu để lập chỉ mục, và chỉ ra các phần của tài liệu có được tạo ra hay không.

Ví dụ : Để lập chỉ mục tập hợp các file HTML đã được nạp vào trong cột dữ liệu có tên htmlfile, người dùng phát ra câu lệnh CREATE INDEX chỉ ra tham số nơi lưu trữ, bộ lọc và các nhóm thành phần của tài liệu như sau :

```
CREATE INDEX myindex ON doc(htmlfile) INDEXTYPE IS  
ctxsys.context PARAMETERS ('datastore ctxsys.default_datastore filter  
ctxsys.null_filter section group ctxsys.html_section_group');
```

3. Duy tu các chỉ mục :

Duy tu các chỉ mục cần thiết sau khi ứng dụng của người dùng thêm vào, cập nhật hay xóa các tài liệu trong bảng tài liệu. Nếu bảng tài liệu của người dùng là tĩnh nghĩa là không có cập nhật, thêm vào hay xóa tài liệu sau lần lập chỉ mục đầu tiên, người dùng không cần duy tu chỉ mục.

Tuy nhiên nếu người dùng thực hiện các thao tác DML (thêm vào, cập nhật, xóa) trên bảng dữ liệu, người dùng phải cập nhật chỉ mục của họ. Người dùng có thể đồng bộ chỉ mục bằng tay với CTX_DDL.SYNC_INDEX.

Ví dụ sau đồng bộ chỉ mục myindex với 2 mb bộ nhớ :

```
begin  
  
    ctx_ddl.sync_index('myindex', '2M');  
  
end;
```

Nếu người dùng đồng bộ chỉ mục thường xuyên, họ có thể xem xét việc tối ưu hóa chỉ mục để giảm sự phân mảnh và xóa những dữ liệu cũ.

4. Giới thiệu công nghệ “INSO filter” của Oracle Text:

Oracle Text sử dụng công nghệ lọc tài liệu do tập đoàn Chicago cấp phép. Công nghệ này cho phép bạn index hầu hết tất cả các loại tài liệu.

Công nghệ này cũng cho phép chuyển đổi những tài liệu sang dạng HTML thông qua gói CTX_DOC. Để thực hiện việc chuyển đổi, công nghệ này sử dụng những thư viện và dữ liệu của tập đoàn Inso và Adobe.

- **Sử dụng công nghệ “INSO filter” để lọc tài liệu:**

Để sử dụng “INSO filter”, chúng ta sử dụng hàm IFILTER của gói CTX_DOC.

Hàm này đòi tham số đầu vào là kiểu nhị phân (BLOB), sau đó lọc dữ liệu và đưa ra dưới dạng văn bản.

Cú pháp :

CTX_DOC.IFILTER(data IN BLOB, text IN OUT NOCOPY CLOB);

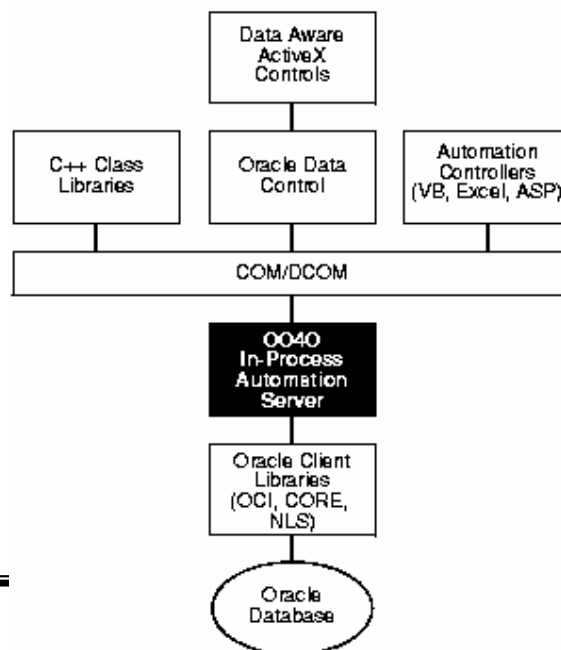
Để gọi hàm này ta sử dụng OO4O để kết nối với Oracle và thực hiện lời gọi hàm.

5. **Giới thiệu OO4O (Oracle Object for OLE):**

OO4O là một sản phẩm được thiết kế cho việc kết nối với CSDL của Oracle với bất kỳ một ngôn ngữ lập trình nào mà có hỗ trợ Microsoft COM Automation và công nghệ ActiveX, bao gồm Visual Basic, Visual C++, ASP,..

OO4O bao gồm những lớp phần mềm sau:

- OO4O “In-Process” Automation Server.
- Oracle Data Control.
- Thư viện OO4O cho C++.



PHỤ LỤC 10. SƠ LƯỢC VỀ THƯ VIỆN VNCONVERT:

Thư viện này được mở rộng từ thư viện mã nguồn mở của Vietnamse Encoding Converter Library. Thư viện này bao gồm các bảng mã, và các lớp đọc ký tự của các bảng mã dùng để chuyển bảng mã. Tuy nhiên, thư viện này không cung cấp việc nhận dạng bảng mã, và chúng tôi đã mở rộng phiên bản này với các hàm nhận dạng bảng mã. Thư viện này bao gồm các lớp sau:

Tên lớp	Ý nghĩa
XXXCharset (XXX sẽ có các giá trị sau: SingleByte – Bảng mã một byte; DoubleByte - Bảng mã 2 byte (VNI) UnicoideCharset - Bảng mã Unicode UnicodeCompCharset – Bảng mã Unicode tổ hợp UnicodeUTF8 - Bảng mã UTF-8)	Lớp đọc, ghi các thông tin ứng với một bảng mã

Thư viện này có các hàm toàn cục sau:

Tên hàm	Ý nghĩa
VnFileConvert	Chuyển đổi file bảng mã với charset biết trước
genDetectAndConvert	Chuyển đổi bảng mã 2 file với charset chưa biết

Sơ lược classes của VnConv:

Unikey hỗ trợ các lớp xử lý bảng mã bao gồm những lớp XXXCharset. Những lớp này đều kế thừa từ VnCharset với 4 hàm quan trọng :

- startInput : chuẩn bị cho việc đọc dữ liệu
- startOutput : chuẩn bị cho việc xuất dữ liệu
- nextInput : là hàm **đọc ký tự kế tiếp ứng với bảng mã** đó. Đây là hàm quan trọng nhất, tất cả những thao tác quan trọng sẽ trong hàm này.
- putChar : là hàm xuất ký tự ứng với bảng mã đó.

Lưu ý : hàm này đòi hỏi tham số VnStdChar. Ký tự này được tính bằng cách cộng tương đối với VnStdCharOffset (=0x10000).

Sơ lược kiểu dữ liệu của VnConv :

VnConv hỗ trợ khá nhiều bảng mã. Các bảng mã này được khai trong các mảng hằng : SingleByteTables, DoubleByteTables... trong file data.h, data.cpp. Trật tự của mảng này như sau :

- Từ trái sang phải : theo thứ tự dấu : a, á, à, â
- Từ trên xuống dưới : theo thứ tự a..z
- 4 dòng cuối là các ký tự Western, dùng để hiển thị các ký tự chuẩn trong Windows-1252. **Khi chuyển đổi sẽ không chuyển đổi ký các tự này.**
- Ngoài ra, mảng CharSetIdMap cũng khá quan trọng, đây chính là bảng index các tên gọi bảng mã trong UniKey.

Sơ lược những hàm quan trọng của VN:

- Hàm quan trọng nhất trong VnConv là hàm VnFileConvert(). Hàm này sẽ đọc file và gọi hàm VnFileStreamConvert(). Hàm này tiếp tục gọi hàm VnConvert. Đây chính là hàm quan trọng nhất.
- Hàm VnConvert() sẽ nhận bảng mã input, output, buffer rồi gọi hàm genConvert().

Hàm genConvert() sẽ gọi hàm nextInput() ứng với bảng mã tương ứng rồi gọi hàm putChar() ứng với bảng mã output.

TÀI LIỆU THAM KHẢO.

TIẾNG VIỆT.

- [01]. Nguyễn Thiện Giáp. Từ và nhận diện từ Tiếng Việt. NXB Giáo dục Hà Nội, 1996.
- [02]. Nguyễn Kim Thản . Nghiên cứu ngữ pháp Tiếng Việt. NXB Giáo Dục, 1997.
- [03]. Nguyễn Tài Cẩn. Ngữ Pháp tiếng Việt. NXB Đại học Quốc gia Hà Nội, 1998.
- [04]. Huỳnh Thụy Bảo Trân. Nghiên cứu một số Mô hình Xây dựng Thử nghiệm một EARCH ENGINE Tiếng Việt, Luận văn Thạc sĩ CNTT, ĐHKHTN, ĐHQG-HCM, 2002.
- [05]. Trương Mỹ Dung, Đỗ Hoàng Cường, Xây dựng và Thiết kế một Hệ thống Search Engine hỗ trợ tìm kiếm theo từ khóa tiếng Việt, Tạp chí Phát triển khoa học công nghệ, ĐHQG-HCM, Vol.7, 4&5 2004, p. 83-88.
- [06]. Do Phuc (2006), Research on the application of the frequent sets and association rules to the Vietnamese document semantic classification, Journal of Science and Technology Development, VNU-HCM, Vol 9, n# 2, pp 23-32, 2006.

TIẾNG ANH.

- [07]. Lenat D, R.V Guha, Build Large Knowledge-based Systems.CVC Project Addison Wesley,1990
- [08]. Uschold Mike, Grunminger Moralee: ONTOLOGIES: Principles, methods and Applications, university of Edinburgh, AIAI, 1996
- [09]. Borgo S, N. Guarini: Using A Large Linguistic Ontology for Internet-based Retrieval of Object-Oriented Components, In Proc of Seke'967. Madrid 1997
- [10]. Kindo T, Yhisida: Adaptive Personal Information Filtering System that Organizes Personal Profiles Automatically, IJCAI, 1997, Japan .
- [11]. Ng H I, "Automatic Induction of Chinese WordNet", 1997
- [12]. Guarino N: Formal Ontology and Information System. In Proc of IOS, 1998, Italy.
- [13]. Farreres X., Rigau G., and Rodniguez H.; "Using WordNet for building WordNets". Proceedings of COLING-ACL 1998. Workshop on Usage of WordNet in Natural Language Processing Systems.
- [14]. Theilmann W.K. Rothermal, Domain Experts for Information Retrieval in the Word WideWeb, In Proc CIA'98, Newyork, 1998.

- [15]. Guarino N, Masolo: OntoSeek: Using Large Linguistic Ontologies for Accessing On-Line Yellow Pages and Product Catalogs, IEEE, May, 1999, pp 70-80.
- [16]. Shari Landes, Claudia Leacock, and Randee I.Tengi (1999). Building semantic concordances. WordNet : an electronic lexical database.
- [17]. Seungwoo Lee and Geunbae Lee. "Unsupervised Noun Sense Disambiguation Using Local Context and Co-occurrence". 2000 Journal of Korean Information Science Society.
- [18]. Wu-Chenggang, Jiao, Shi Zhongzhi, Ontology based Information Retrieval, Chinese Academy of Science, 2000
- [19]. The Anatomy of a large scale search engine. Google 2003.
- [20]. Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. The PageRank Citation ranking: Bringing order to the Web. Manuscript in progress.
- [21]. Truong My Dung, "Report Project Mitani 1: Vietnamese Search Engine", 12/2003, VNU-HCMC
- [22]. Truong Mỹ Dung, Multilingual Ontology-Based Upgrading Extensions Of An Oracle Text-Based Search Engine, Kỹ yếu Hội thảo FAIR 2005.
- [23]. Truong My Dung, Nguyen Dinh Ngoc, "ONTOLOGY OPTIMISATION: PROBLEMATICS & METHODOLOGY, WITH A FIRST STEP OF FORMALISM", Progress in Informatics, No.IEEE, 87-95, 2005.
- [24]. Do Phuc (2006), Document classification using graph model, frequent subgraphs and Galois lattice, In Proceedings (Addendum) of the 4th IEEE (international conference on computer science research, innovation and revision for the future, RIVF'06, pp173-176.

CÁC TRANG WEB.

- [25]. <http://altavista.com>
- [26]. <http://www.bruceclay.com/>
- [27]. <http://www-db.stanford.edu>
- [28]. <http://www.danangpt.vnn.vn>
- [29]. <http://www.excite.com>
- [30]. www.encanto.com
- [31]. <http://www.infopeople.org/search/chart.html>
- [32]. <http://www.google.com/webmaster/4.html>
- [33]. <http://infopeople.org/search>, U.S. Institute of Museum và Library Services của Library Services và Technology Act, State Librarian.
- [34]. <http://www.google.com>
- [35]. <http://www.howsearchenginework.com>
- [36]. <http://www.monash.com>
- [37]. <http://www.panvietnam.com>
- [38]. <http://www.samizdat.com>
- [39]. <http://www.searchenginewatch.com>
- [40]. <http://www.vinaseek.com>
- [41]. <http://www.vietsoftonline.com.vn>
- [42]. www.yahoo.com