PHẦN MỞ ĐẦU

Trong những năm gần đây, trên cơ sở phát triển và ứng dụng công nghệ Internet, khối lượng dữ liệu trên máy tính đã tăng trưởng không ngừng theo cả hai phương diện tạo mới và thu thập. Sự mở rộng các dữ liệu khoa học về địa lý, địa chất, khí tượng do vệ tinh thu thập, sự giới thiệu quảng bá mã vạch đối với hầu hết các sản phẩm thương mại, việc tin học hoá sâu rộng các thương vụ và giao dịch, sự phát triển việc ứng dụng CNTT trong quản lý hành chính nhà nước ... đã phát sinh ra một khối lượng dữ liệu khổng lồ. Mặt khác, trong bối cảnh nền tảng cho một xã hội thông tin, nhu cầu nhận được thông tin một cách nhanh chóng, chính xác cũng như nhu cầu thu nhận được "tri thức" từ khối lượng thông tin khổng lồ nói trên đã trở nên cấp thiết. Bối cảnh đó đã đòi hỏi những phương pháp tiếp cận mới mà trong đó điển hình nhất là các phương pháp thuộc lĩnh vực khai phá dữ liệu và khám phá tri thức trong các cơ sở dữ liệu [7,9]. Sự tăng trưởng hàng năm về số lượng công trình được công bố, về hội thảo khoa học quốc tế liên quan đến việc nghiên cứu, giải quyết từng bước nhiều bài toán điển hình thuộc lĩnh vực này đã thể hiện đầy đủ sự phát triển vượt bậc của lĩnh vực nói trên. Các bài toán biểu diễn dữ liệu, lưu trữ dữ liệu, tìm kiếm dữ liệu, phân lớp dữ liệu, phân cụm dữ liệu ... [2-4,6,8-14] là những bài toán điển hình nhất.

Trong xu thế tăng trưởng không ngừng nguồn dữ liệu, thông qua sự phát triển của công nghệ Web, dạng dữ liệu phi cấu trúc và nửa cấu trúc (điển hình là hệ thống các trang web trên Internet) càng tăng trưởng theo tốc độ nhảy vọt. Đây là dạng dữ liệu gần nhất với con người, mà qua chúng con người mong muốn lưu trữ thông tin, tri thức hoặc chuyển tải nó cho nhiều người khác. Trong những năm gần đây WWW đã trở thành một kênh thông tin quan trọng nhất cho việc phân tán các thông tin về cá nhân, khoa học và thương mại. Một lý do của việc WWW phát triển nhanh chóng là giá cả cho việc tạo và xuất bản các trang web rất rẻ. So sánh với các phương pháp khác như sản xuất tờ rơi hay quảng cáo trên báo và tạp chí thì trang web rẻ hơn rất nhiều và lại được cập nhật thường xuyên hơn đến hàng tỷ người sử dụng, vì vậy mà ngay cả các công ty rất nhỏ cũng có khả năng đưa các sản phẩm và dịch vụ của họ lên WWW. Hơn nữa có rất nhiều

các công ty hoạt động bán hàng trực tuyến trên Internet, vì vậy mà nhu cầu đưa các thông tin lên WWW là hoàn toàn tự nhiên. Nhưng với việc tăng không ngừng các site thì việc tìm ra một trang hay thậm chí một site mà mỗi cá nhân đang cần lại thực sự là một vấn đề ngày càng khó khăn.

Việc nghiên cứu các bài toán liên quan đến hệ thống các dữ liệu dạng này (biểu diễn văn bản, tìm kiếm và phân lớp văn bản) cùng với việc đề xuất những giải pháp đối với các bài toán đó luôn là những vấn đề khoa học và công nghệ thời sự [1-4,6,8-14]. Chẳng hạn, vấn đề phát hiện ra một website mới thực sự thú vị cho người sử dụng là một vấn đề chưa được quan tâm đúng mức. Các hệ tìm kiếm trên Internet hiện nay như Yahoo, Altavista, Google... là những hệ triển khai để giải quyết bài toán tìm kiếm và được sử dụng khá phổ biến hiện nay. Tuy nhiên vẫn còn có các vấn đề chưa thoả mãn được nhu cầu thực tế của người sử dụng. Đó là khi sử dụng dịch vụ tìm kiếm trên các site này thì chỉ có thể tìm được các trang thông tin theo những điều kiện tìm kiếm hết sức giản đơn. Thêm vào đó, có rất nhiều trường hợp mục từ là không trọn vẹn và đôi khi quá hạn vì không được cập nhật thường xuyên. Hơn nữa các dịch vụ tìm kiếm này không cung cấp tất cả các lĩnh vực chuyên sâu hơn, nhất là các lĩnh vực hẹp cho một số người sử dụng đặc biệt. Các hệ này cũng chưa cho phép khai thác những thông tin truy nhập của người sử dụng vì vậy không có cơ chế phản hồi thông tin để sử dụng kết quả tìm kiếm trước đây vào lần tìm kiếm tiếp theo. Cơ chế này là cần thiết vì làm được như vậy hiệu quả và độ chính xác tìm kiếm chắc chắn được nâng cao. Một vấn đề nữa là các hệ tìm kiếm này thường xử lý các yêu cầu tìm kiếm dưới dạng các từ khoá tìm kiếm. Khi có nhiều hơn một từ khoá thì hệ tìm kiếm xử lý các từ khoá này theo cùng một cách thức mà không có cơ chế cho phép người sử dụng xác định độ quan trọng khác nhau cho các từ khoá tìm kiếm. Cũng như vậy, các hệ tìm kiếm điển hình hiện nay chưa quan tâm đến vấn đề đồng nghĩa và đa nghĩa của từ khóa, vì vậy trong quá trình tìm kiếm có thể đã bỏ qua rất nhiều các kết quả tìm kiếm. Nhiều nghiên cứu liên quan đã đề xuất một số phương pháp biểu diễn văn bản cho phép thi hành được những khía cạnh đã đề cập trên đây [2-4,8-14].

Từ việc tìm hiểu và phân tích ưu, nhược điểm của các phương pháp tiếp cận khác nhau, dựa trên ý tưởng nâng cao hiệu quả tìm kiếm, luận văn đề cập việc sử dụng mô hình vector biểu diễn trang web trong các máy tìm kiếm để cho phép dễ dàng bổ sung trọng số cho các từ khoá tìm kiếm và tăng cường được ngữ nghĩa nội dung văn bản vào quá trình tìm kiếm.

Với mục tiêu đề xuất một phương pháp biểu diễn vector cho các trang web trong các máy tìm kiếm để nâng cao hiệu quả tìm kiếm, nội dung của luận văn được định hướng vào các vấn đề sau:

- Giới thiệu, phân tích và đánh giá một số phương pháp biểu diễn trang web điển hình,

- Trên cơ sở một số phương pháp biểu diễn văn bản trang web theo mô hình vector, luận văn nghiên cứu việc cải tiến các phương pháp biểu diễn đó để nhận được một phương pháp mới biểu diễn trang web,

- Nghiên cứu, đề xuất việc bổ sung thêm biểu diễn vector cho trang web trong các máy tìm kiếm theo phương pháp mới, đồng thời bổ sung chức năng tìm kiếm trang Web "theo nội dung" cho hệ tìm kiếm Vietseek.

Luận văn bao gồm Phần mở đầu, ba chương nội dung và Phần kết luận mà nội dung các chương được trình bày như dưới đây.

Chương 1 với tiêu đề là ***Tổng quan về web-mining*** giới thiệu sơ bộ những nội dung tổng quan nhất về cơ sở dữ liệu Fulltext, cơ sở dữ liệu Hypertext, cơ sở dữ liệu trang web và phương pháp biểu diễn vector. Trong chương này cách tiếp cận theo website được trình bày khá chi tiết về cả khía cạnh biểu diễn website lẫn giải pháp cho bài toán tìm kiếm theo website. Luận văn còn đề xuất một thuật toán xây dựng cây website theo cách tiếp cận này.

Tiêu đề của chương 2 là ***Một số phương pháp biểu diễn dữ liệu web và giải pháp kết hợp***. Nội dung của chương này xem xét và đánh giá một số phương pháp biểu diễn trang web điển hình. Đầu tiên luận văn giới thiệu về biểu diễn trang web trong các máy tìm kiếm, sau đó luận văn giới thiệu cách tiếp cận theo mô hình vector để biểu diễn

trang web và một đề xuất về một cách biểu diễn trang web. Phần cuối cùng của chương này trình bày đề xuất của luận văn bổ sung cách biểu diễn mới cho trang web vào máy tìm kiếm và sơ bộ về thuật toán tìm kiếm theo nội dung.

Chương 3 ***Máy tìm kiếm VietSeek và thử nghiệm thuật toán tìm kiếm theo nội dung*** giới thiệu chi tiết về máy tìm kiếm VietSeek, thiết kế lôgic về dữ liệu theo biểu diễn vector và thuật toán tìm kiếm theo nội dung trên cơ sở do luận văn đề xuất.

Phần kết luận tổng hợp những kết quả nghiên cứu chính của luận văn, chỉ ra một số hạn chế chưa hoàn thiện cài đặt thực sự. Đồng thời luận văn cũng đề xuất một số hướng nghiên cứu cụ thể tiếp theo của tác giả luận văn.

Lời cảm ơn

Em xin bày tỏ lòng kính trọng và biết ơn sâu sắc tới Thầy giáo Tiến sĩ Hà Quang Thuỵ, người đã tận tình hướng dẫn luận văn cho em.

Em xin cảm ơn các Thầy Cô trong khoa Công nghệ, Đại học Quốc Gia Hà Nội, và nhóm Xemina chuyên môn "Data Mining và KDD" thuộc bộ môn Các Hệ thống Thông tin, khoa Công nghệ, những người đã giúp đỡ cho em trong suốt quá trình học tập và nghiên cứu, đặc biệt là các bạn Bùi Quang Minh và Đoàn Sơn.

Em xin bày tỏ lòng biết ơn sâu sắc tới gia đình, các đồng nghiệp ở Viện Công nghệ Thông tin, Đại học Quốc gia Hà Nội, và các bạn bè đã giúp đỡ và động viên em trong suốt quá trình học tập, nghiên cứu và làm việc.

Hà Nội ngày 15/04/2003
Học viên

Phạm Thị Thanh Nam

BẢNG CHÚ GIẢI MỘT SỐ CỤM TỪ VIẾT TẮT

CSDL: Cơ sở dữ liệu (DataBase)

CNTT: Công nghệ thông tin (Information Technology)

kNN: k Nearest Neighbour

KPDL: Khai phá dữ liệu (Data Mining)

KPTTCSDL: Khám phá tri thức trong CSDL (Knowledge Discovery in Databases)

SVM: Support Vector Machine

WWW: Hệ thống trang Web (World Wide Web)

BẢNG CHÚ GIẢI MỘT SỐ THUẬT NGỮ TIẾNG VIỆT

Bayes tự nhiên:	Naive Bayes
k người láng giềng gần nhất:	k Nearest Neighbour
Mạng nơron:	Neural Net
Máy tìm kiếm:	Search engine
Bộ điều khiển tìm duyệt:	Crawl Control
Bộ tìm duyệt:	Crawler
Bộ tạo chỉ mục:	Indexer Module
Bộ phân tích tập:	Collection Analysis Modele
Bộ truy vấn:	Query Engine
Bộ xếp hạng:	Ranking
Bộ phân tích URL:	URLresolver
Chỉ mục cấu trúc:	Structure Index
Chỉ mục liên kết ngược:	Inverted Index
Chỉ mục nội dung:	Text Index
Chỉ mục tiện ích:	Utility Index
Hạng hiển thị:	Rank
Hạng trang web (Hạng):	Page Rank
Kho trang web:	Page Repository
Tải trang:	Download
Máy vector trợ giúp:	Support Vector Machine

Mô hình (không gian) vector:	Vector (Space) Model
Siêu liên kết:	Hyperlink
Siêu văn bản:	Hypertext
Tìm kiếm theo nội dung:	text-based retrieval
Trang web:	web page, HTML page, HTML document

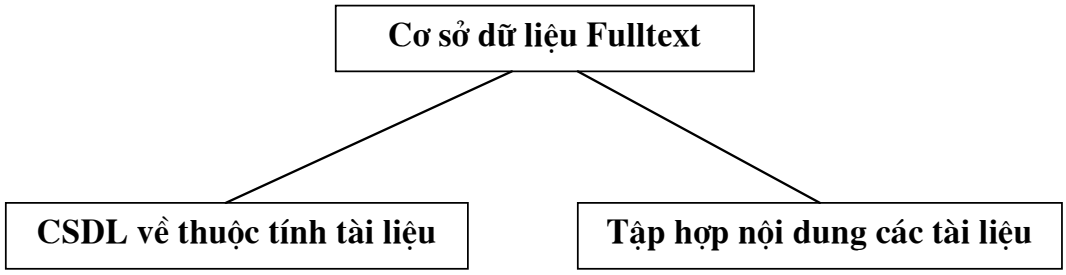
CHƯƠNG I. TỔNG QUAN VỀ WEB-MINING

1.1 Giới thiệu về cơ sở dữ liệu Fulltext và Hypertext

1.1.1 Cơ sở dữ liệu Fulltext

- ***Giới thiệu chung***

Cơ sở dữ liệu Fulltext là cơ sở dữ liệu phi cấu trúc mà dữ liệu chứa trong đó bao gồm các nội dung text và các thuộc tính về tài liệu văn bản với nội dung đó. Dữ liệu trong cơ sở dữ liệu Fulltext thường được tổ chức như một sự kết hợp giữa hai phần: phần cơ sở dữ liệu thông thường quản lý thuộc tính của các tài liệu, và phần tập hợp nội dung các tài liệu được quản lý. Chúng ta có thể hình dung một cơ sở dữ liệu Fulltext được tổ chức như sau:



Hình 1.1 Mô hình tổ chức của cơ sở dữ liệu Fulltext

Trong những trường hợp phổ biến, nội dung tài liệu được lưu giữ gián tiếp trong cơ sở dữ liệu theo nghĩa hệ thống chỉ quản lý các con trỏ (địa chỉ) trỏ tới các địa chỉ chứa nội dung tài liệu (một ví dụ dễ thấy nhất là mạng Internet, các trang web thường lưu giữ các địa chỉ chỉ tới nơi có lưu nội dung các trang thông tin cụ thể mà người sử dụng muốn xem). Còn các con trỏ (địa chỉ) và các thuộc tính khác về nó thì được lưu trực tiếp trong cơ sở dữ liệu bằng hệ quản trị có cấu trúc.

Tuy nhiên, trong một số trường hợp (đặc biệt là đối với các máy tìm kiếm trên Internet như Yahoo, Google, AltaVista ...), để cung cấp nội dung văn bản nhanh chóng, người ta lại tổ chức lưu trữ các văn bản ngay trong hệ thống (dưới dạng vùng cache).

Nội dung của dữ liệu Fulltext (văn bản) không có cấu trúc nội tại, được coi như một là dãy các từ, các dấu ngăn cách. Ngữ nghĩa văn bản dựa trên ý nghĩa các từ mang nghĩa (được gọi là từ khóa - term hoặc keyword) có trong văn bản và cách bố trí các từ khóa trong văn bản đó. Do không có cấu trúc nên bài toán "tổ chức theo cấu trúc hoàn toàn" các từ khóa trong văn bản là không thích hợp do tính chất quá phức tạp khi thực hiện điều đó. Do đó, phổ biến hơn người ta sử dụng các phương pháp biểu diễn ngữ nghĩa văn bản thông qua tập các từ khoá có trong văn bản đó.

Các cơ sở dữ liệu Fulltext hiện nay thường là các tập hợp sách, tạp chí, bài viết được quản lý trong một mạng thư viện điện tử, tập các file và các trang web (là các trang file) được lưu trữ bởi các hệ thống web như hệ thống của Yahoo, Google, AltaVista …

Như đã nói, làm thế nào để hiểu được nội dung của các tài liệu trong cơ sở dữ liệu? Tồn tại các phương pháp biểu diễn được sử dụng như phương pháp tóm tắt, phương pháp vector, mạng logic, lược đồ cú pháp. Nhưng các phương pháp đó chỉ chứa đựng được nội dung sơ sài, tóm tắt của tài liệu. Hơn nữa mỗi một phương pháp lại có các khó khăn riêng, đặc biệt là khi hệ thống cho phép cập nhật thêm dữ liệu. Vì vậy mà việc cải tiến các mô hình biểu diễn này luôn luôn được đặt ra

Cơ sở dữ liệu Fulltext có rất nhiều khía cạnh tiềm năng tốt cho việc khai phá dữ liệu và KDD, với các mục tiêu là tự động trợ giúp người dùng để họ có thể sử dụng hệ thống tài liệu hiệu quả hơn (phân lớp tài liệu, tìm kiếm thông tin và tìm kiếm tài liệu…) và mô hình vector là mô hình tốt hơn cả để trình bày tài liệu Fulltext

Do ngữ nghĩa của các văn bản Fulltext thường được biểu diễn thông qua các từ khoá của nó nên trong quá trình xử lý các dữ liệu Fulltext thường nảy sinh các vấn đề về từ đồng nghĩa và từ đa nghĩa. Như chúng ta đã biết thì trong ngôn ngữ tự nhiên luôn có các từ đồng nghĩa (là trường hợp có nhiều từ viết khác nhau đều chỉ chung một ý

nghĩa giống nhau) và các từ đa nghĩa (là trường hợp một từ nhưng có nhiều nghĩa khác nhau). Trong thực tế giao tiếp chúng ta cũng thường xuyên gặp phải các tình huống hiểu nhầm ý nghĩa muốn diễn đạt của người nói khi gặp phải các từ đồng nghĩa và đa nghĩa. Vì vậy trong xử lý văn bản chắc chắn sẽ không tránh khỏi những khó khăn do vấn đề này gây ra. Do đó chúng ta phải tìm cách khắc phục các vấn đề này. Đã có một số hướng nghiên cứu giải quyết vấn đề từ đồng nghĩa và đa nghĩa được tiến hành [1,4,7] như: liên kết từ đồng nghĩa với từ khoá, dùng trọng số thể hiện độ quan trọng các từ, chuẩn hoá biểu diễn văn bản, biểu diễn ngữ cảnh từ khoá, biểu diễn qua tập mờ...

- ***Mô hình vector với giải pháp vấn đề đa ngôn ngữ và từ đồng nghĩa***

Hiện nay mô hình biểu diễn dữ liệu fulltext điển hình nhất là mô hình. Theo mô hình vector thì hệ thống cơ sở dữ liệu Fulltext quản lý các tài liệu thuộc một phạm vi hoạt động của con người được thể hiện qua một tập từ khoá V (các từ khoá này có mang ý nghĩa của nội dung các tài liệu). Như vậy là tập hợp các từ khoá có trong tài liệu "biểu diễn" nội dung của tài liệu đó.

Áp dụng bài toán tìm kiếm trong cơ sở dữ liệu Fulltext thì quá trình tìm kiếm gồm hai giai đoạn con là: quá trình trình bày câu hỏi (mã hoá câu hỏi) và quá trình xử lý trên các vector. Do số lượng các từ trong câu hỏi thường là nhỏ nên thời gian của quá trình mã hoá câu hỏi thường ngắn. Ngược lại, thời gian cho việc xử lý trên các vector thường khá lớn, và phụ thuộc vào kích thước của các vector và số lượng các phép tính giữa câu hỏi với các vector mã hoá của tài liệu. Trên thực tế thì số lượng lớn nhất các phép toán là $A^* n$, với A là số lượng tài liệu được lưu trữ trong cơ sở dữ liệu và n là số lượng các từ trong câu hỏi được đưa ra. Để giảm số lượng các phép toán trong giai đoạn xử lý trên các vector thì chúng ta có thể xem xét giảm kích thước của vector trình bày tài liệu, và kết quả là thay vì phải mã hóa tất cả các từ khoá xuất hiện trong không gian cơ sở dữ liệu thì ta chỉ cần mã hoá các từ khoá xuất hiện trong tài liệu. Ngoài ra có một cách rất đơn giản có thể tăng độ chính xác tìm kiếm là tách riêng phần tiêu đề của tài liệu ra thành một phần. Thông thường, các tài liệu có phần tiêu đề thể hiện tóm tắt nội dung

của tài liệu, chính vì vậy mà chúng ta có thể tách phần tiêu đề ra khỏi nội dung của tài liệu và biểu diễn nó bằng một vector riêng, độc lập với phần nội dung. Khi đó ngoài việc tìm kiếm theo nội dung chúng ta sẽ đưa thêm lựa chọn tìm kiếm theo tiêu đề. Vì phần tiêu đề bao giờ cũng ngắn hơn phần nội dung rất nhiều nên việc tìm kiếm theo tiêu đề sẽ diễn ra rất nhanh mà lại mang lại cho chúng ta độ chính xác tìm kiếm cao hơn.

Với bài toán tìm kiếm thì vấn đề từ đồng nghĩa như đã nêu ở phần trên cần phải được triển khai nếu không chúng ta sẽ chỉ tìm được các tài liệu chứa các từ có trong câu hỏi, còn các tài liệu có cùng nội dung nhưng có cách thể hiện khác sẽ bị bỏ qua.

Để giải quyết vấn đề này là chúng ta xây dựng một bảng liệt kê danh sách các từ đồng nghĩa thuộc nhiều ngôn ngữ cùng với các hệ số tương quan về mặt ý nghĩa giữa chúng. Và trong một nhóm các từ đồng nghĩa mặc dù cùng biểu đạt một nội dung nhưng vai trò của các từ có thể khác nhau do các lý do sau: với một nội dung cụ thể này thì từ này hay được sử dụng hơn từ kia, còn với một nội dung cụ thể khác thì có thể lại khác [3,9,12]. Việc thống kê và ấn định hệ số cho các từ đồng nghĩa trong một nhóm các từ đồng nghĩa là một việc làm phức tạp và rắc rối, đòi hỏi phải có tri thức về ngữ nghĩa của các từ trong nhiều ngôn ngữ khác nhau. Vì vậy việc này cần nhận được sự phối hợp với các nhà ngôn ngữ học.

1.1.2 Cơ sở dữ liệu Hypertext

Hypertext là thuật ngữ được Theodore Nelson đưa ra lần đầu tiên năm 1965 tại hội thảo của Hội toán học Mỹ ACM lần thứ 20. Theo Nelson thì Hypertext là các tài liệu dạng chữ viết không liên tục. Chúng được phân nhánh và cho phép người đọc có thể chọn cách đọc theo ý muốn của mình, tốt nhất là nên đọc nó trên các màn hình có khả năng tương tác.

Hiểu theo nghĩa thông thường thì Hypertext là một tập các trang chữ viết được kết nối với nhau bởi các liên kết, và nó cho phép người đọc có thể đọc theo các cách khác nhau.

Hypertext cũng có thể bao gồm một tập chữ viết liên tục, và đây cũng chính là dạng phổ biến nhất của chữ viết. Do không bị hạn chế bởi tính liên tục nên trong Hypertext, chúng ta có thể tạo ra các dạng trình bày mới, và nhờ đó mà tài liệu của chúng ta sẽ phản ánh tốt hơn nội dung mà chúng ta đang muốn viết. Và người đọc có thể chọn cho mình một cách đọc phù hợp, ví dụ họ có thể đi sâu vào một vấn đề mà họ thích thú, hoặc có thể tiếp tục mạch suy nghĩ hiện tại của họ theo cách mà từ trước vẫn được coi là không thể.

Theo từ điển của Đại học Oxford (Oxford English Dictionary Additions Series) thì Hypertext được định nghĩa như sau: là loại Text không phải đọc theo dạng liên tục đơn, và nó có thể được đọc theo các thứ tự khác nhau; đặc biệt là Text và ảnh đồ hoạ (Graphic) là các dạng có mối liên kết với nhau theo cách mà người đọc có thể không cần đọc nó một cách liên tục. Ví dụ khi đọc một cuốn sách người đọc không cần đọc lần lượt từ đầu đến cuối mà có thể nhảy cóc đến các đoạn khác nhau để tham khảo các vấn đề có liên quan.

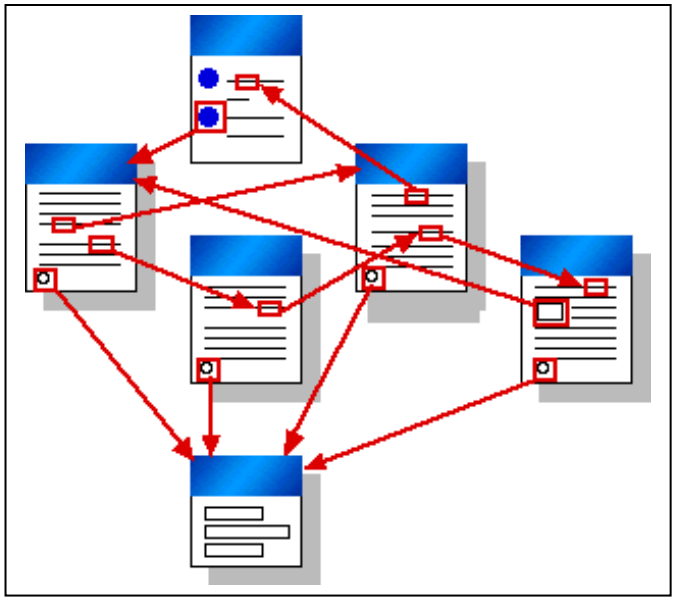
Sáng kiến tạo ra một tập các văn bản cùng với các con trỏ trỏ tới các văn bản khác một cách rõ ràng để liên kết một tập các văn bản có mối quan hệ với nhau là một cách thực sự hay và rất hữu ích để tổ chức thông tin. Với người viết, cách này cho phép họ có thể thoải mái loại bỏ những băn khoăn về thứ tự trình bày những vấn đề có liên quan đến nhau để tập trung vào hoàn thành các vấn đề nhỏ, và sau đó họ có thể sử dụng các kết nối để chỉ ra cho người đọc thấy được các vấn đề nhỏ đó có mối quan hệ với nhau như thế nào. Tại đây, theo một nghĩa nào đó, chúng ta gặp lại tư tưởng mô đun hóa trong thiết kế thuật toán và viết chương trình. Với người đọc, cách này cho phép họ có thể đi tắt trên mạng thông tin và tự quyết định phần thông tin nào có liên quan đến vấn đề họ đang quan tâm để tiếp tục tìm hiểu. So sánh với cách đọc tuyến tính, tức là đọc lần lượt, thì Hypertext đã cung cấp cho chúng ta một giao diện để có thể tiếp xúc với nội dung thông tin hiệu quả hơn rất nhiều.

Theo khía cạnh của thuật toán học máy thì Hypertext đã cung cấp cho chúng ta cơ hội nhìn ra ngoài phạm vi một tài liệu để phân lớp nó. Tất nhiên không phải tất cả các

tài liệu có liên kết đến nó đều có ích cho việc phân lớp, đặc biệt là khi các siêu liên kết có thể chỉ đến rất nhiều loại khác nhau của mối quan hệ giữa các tài liệu. Tuy nhiên chắc chắn vẫn còn tồn tại các tiềm năng mà con người cần tiếp tục nghiên cứu về việc sử dụng các tài liệu liên kết đến một trang để nâng cao độ chính xác phân lớp trang đó.

Tài liệu Hypertext (Hypertext document): một tài liệu Text đơn nằm trong một tập Hypertext. Nếu chúng ta tưởng tượng tập Hypertext như một đồ thị thì một tài liệu Text đơn là một nút trong đó.

Siêu liên kết (Hypertext link): là một sự tham khảo/kết nối từ một tài liệu Hypertext này đến một tài liệu Hypertext khác. Các siêu liên kết đóng vai trò như những đường nối trong đồ thị nói trên. Hình 1.2 cho một ví dụ minh hoạ đơn giản về tài liệu Hypertext.



Hình 1.2. Đồ thị minh hoạ mối quan hệ giữa các tài liệu Hypertext trong một tập tài liệu Hypertext

Hypertext là loại dữ liệu rất phổ biến hiện nay, và cũng là loại dữ liệu có nhu cầu tìm kiếm và phân lớp rất lớn. Nó là loại dữ liệu phổ biến trên mạng thông tin Internet.

Cơ sở dữ liệu trang web (trang web là văn bản Hypertext phổ dụng hiện nay) với tính chất "nửa cấu trúc" do xuất hiện thêm các "thẻ": thẻ cấu trúc (tiêu đề, mở đầu, nội

dung), thẻ nhấn trình bày chữ (đậm, nghiêng...). Nhờ các thẻ này mà chúng ta có thêm một tiêu chuẩn (so với tài liêu Fulltext) để có thể tìm kiếm và phân lớp chúng. Dựa vào các thẻ đã quy định trước chúng ta có thể phân thành các độ ưu tiên khác nhau cho các từ khoá nếu chúng xuất hiện ở các vị trí khác nhau. Ví dụ khi tìm kiếm các tài liệu có nội dung liên quan đến "computer" thì chúng ta đưa vào từ khoá tìm kiếm là "computer". Rõ ràng các tài liệu mà từ "computer" xuất hiện ở phần tiêu đề sẽ có nội dung nói về computer, và sẽ gần với yêu cầu tìm kiếm của chúng ta hơn.

1.1.3 So sánh đặc điểm của dữ liệu Fulltext và dữ liệu trang web

Như đã được trình bày, trang web là một dạng đặc biệt của dữ liệu Full-text. Qua khảo sát sơ bộ tính chất của hai loại dữ liệu này, chúng tôi có một số nhận xét sau đây về đặc điểm giống nhau và khác nhau giữa trang web và một trang Fulltext thông thường. Bảng dưới đây liệt kê ra một số các đặc điểm khác nhau cơ bản như vậy.

STT	Trang web	Văn bản thông thường (Fulltext)
1	Văn bản trang web là "nửa cấu trúc". Trong nội dung có phần tiêu đề, và có các thẻ nhấn mạnh nghĩa của từ hoặc cụm từ.	Văn bản Fulltext là "phi cấu trúc". Trong phần nội dung không có một tiêu chuẩn nào cho phép chúng ta dựa vào để đánh giá.
2	Nội dung của các trang web thường được mô tả ngắn gọn, cô đọng, có các siêu liên kết chỉ đến các web có nội dung liên quan	Nội dung của văn bản Fulltext thường rất chi tiết và đầy đủ.
3	Trong nội dung các trang web có chứa các siêu liên kết cho phép liên kết đến các trang khác có nội dung liên quan	Các trang văn bản thông thường không liên kết được đến nội dung của các trang khác

Bảng 1.1. Đối sánh trang Web và trang Fulltext

1.2 Tổng quan về phương pháp biểu diễn văn bản trong cơ sở dữ liệu trang web

Cùng với sự phát triển nhanh chóng của số lượng các trang web trên mạng máy tính toàn cầu Internet, cũng như số lượng người dùng mạng Internet trong những năm gần đây thì việc xử lý văn bản trang web cũng nhận được mối quan tâm đặc biệt. Do các trang web chỉ là các tài liệu "nửa cấu trúc" nên việc biểu diễn trang web là đặc biệt quan trọng bởi vì việc biểu diễn là bước thực hiện đầu tiên, làm tiền đề cho việc giải quyết rất nhiều bài toán như tìm kiếm, phân lớp, phân cụm văn bản...

Hiện nay có rất nhiều các cách tiếp cận khác nhau trong việc biểu diễn văn bản trong cơ sở dữ liệu trang web. Với mỗi mục đích khác nhau thì mỗi người lại có cách biểu diễn trang web riêng. Có thể kể ra một số cách biểu diễn trang web khác nhau như: Dôna Mladenic [10], Seán Slattery [11] hay Hwanjo Yu, Jiawei Han, Kevin Chen-Chuan [14] coi trang web như văn bản thông thường và chọn mô hình vector biểu diễn; các máy tìm kiếm như Yahoo, Altavista, Google hay Vietseek... không sử dụng mô hình vector mà sử dụng hệ thống từ khóa móc nối song không biểu diễn nội dung văn bản. Một cách tiếp cận khác đang nhận được mối quan tâm của nhiều người hiện nay, đó là cách tiếp cận biểu diễn website, đối tượng quan tâm không là webpage mà là website: Nghĩa là đối tượng tìm kiếm không phải là các trang web đơn nữa mà là cả một website [6].

Sau đây chúng tôi giới thiệu sơ bộ về mỗi cách tiếp cận biểu diễn văn bản trang web cùng một số nhận xét đánh giá của chúng tôi về điểm mạnh và điểm yếu của mỗi cách tiếp cận. Trình bày của chúng tôi tuân theo sự phân loại, loại đầu tiên về các phương pháp biểu diễn trang web đơn và loại thứ hai về các phương pháp biểu diễn website. Vì các phương pháp biểu diễn trang web đơn là đối tượng nghiên cứu của luận văn mà sẽ được khảo sát kỹ lưỡng trong các chương sau của luận văn, nên trong phần dưới đâyluận văn trình bày một cách sơ lược những nội dung này.

1.2.1 Giới thiệu sơ bộ về các phương pháp biểu diễn trang web

- ***Phương pháp biểu diễn trang web trong các máy tìm kiếm***

Trong hầu hết các máy tìm kiếm hiện nay đều không sử dụng mô hình vector để biểu diễn các trang web. Nhằm giải quyết bài toán tìm kiếm theo cụm từ, các máy tìm kiếm hiện nay sử dụng phương pháp biểu diễn văn bản trang web theo xâu các từ khóa xuất hiện trong văn bản đó. Trong một số trường hợp, để phục vụ cho việc tìm kiếm nhanh các văn bản chứa một từ do người dùng đưa vào, từ khóa được coi là đối tượng trung tâm của hệ thống (xem mục 2.1.2).

Lý do không sử dụng mô hình vector để biểu diễn trang web trong các máy tìm kiếm được diễn giải theo các lập luận sau đây. Trong các cơ sở dữ liệu Fulltext truyền thống, các tài liệu có cấu trúc thông tin đồng nhất (về nội dung, ngôn ngữ diễn đạt, định dạng file...), chúng phổ biến là tập các tài liệu trong cùng một lĩnh vực hẹp nào đó, và thường là được kiểm soát tốt. Do đó việc sử dụng mô hình vector để biểu diễn là rất phù hợp. Trong khi đó cơ sở dữ liệu trang web là một cơ sở dữ liệu phức tạp cả về nội dung, kích thước lẫn hình thức trình bày. Những người thiết kế máy tìm kiếm coi rằng hệ thống trang Web là một tập dữ liệu khổng lồ, không đồng nhất và rất khó kiểm soát. Không ai có thể biết chính xác được kích thước của web hiện nay ra sao, và nó sẽ tiếp tục phát triển như thế nào về nội dung lẫn kích thước, vì hầu như mọi người đều có thể xoá, sửa chữa và đưa thêm các trang mới lên Internet bất cứ lúc nào. Web đa dạng cả về nội dung, ngôn ngữ (ngôn ngữ của con người và ngôn ngữ máy) lẫn định dạng file (text, HTML, PDF, images, sounds...) chính vì thế mà việc sử dụng mô hình vector để biểu diễn có thể là không còn phù hợp nữa mà cần phải sử dụng các mô hình biểu diễn khác hoặc phải cải tiến mô hình vector để có thể phù hợp với việc xử lý web. Trong phương án phổ biến hiện nay trong các máy tìm kiếm, người ta chưa sử dụng mô hình vector để biểu diễn trang web.

Các máy tìm kiếm xử lý bài toán tìm kiếm trang web bằng cách kiểm soát nội dung của các trang theo hệ thống các từ khóa và kiểm soát các mối liên kết giữa các trang. Các máy tìm kiếm phân tích các trang để lấy ra các từ khóa xuất hiện trong các

trang đó và lưu trữ để làm cơ sở cho việc tìm kiếm theo nội dung. Trong khi phân tích các từ trong trang web thì các máy tìm kiếm đều ghi lại các thông tin chung nhất về từ như: vị trí xuất hiện trong trang, chữ hoa hay chữ thường... nên có thể sử dụng được các thông tin tiềm ẩn mà người viết các trang web đó muốn diễn đạt. Các máy tìm kiếm còn phân tích được các mối liên kết giữa các trang để phục vụ cho việc xếp hạng các trang làm cơ sở để sắp xếp các trang kết quả khi hiển thị cho người dùng. Chi tiết về cách biểu diễn cũng như xử lý tài liệu web trong các máy tìm kiếm được đề cập đến ở phần 2.1 của luận văn này.

- ***Các phương pháp dựa trên mô hình vector***

Phát triển kết quả của các nghiên cứu trước đây, trong luận văn tiến sĩ năm 2002 của mình, Seán Slattery [11] đã giới thiệu và đề xuất sử dụng mô hình vector biểu diễn văn bản. Trong lĩnh vực xử lý văn bản truyền thống từ trước đến nay thì thông thường vẫn thực hiện các công việc biểu diễn, tìm kiếm, phân lớp ... trên cơ sở coi trang web như là các trang văn bản thông thường và sử dụng mô hình không gian vector để biểu diễn văn bản. Cũng tiến hành việc biểu diễn và xử lý tài liệu web dựa trên cách tiếp cận đó, tuy nhiên Seán Slattery cũng đã có những cải tiến để có thể tận dụng được tính nửa cấu trúc, đặc biệt là khai thác thế mạnh của siêu liên kết trong văn bản. Seán Slattery đã sử dụng các siêu liên kết giữa các trang web để có thể lấy được các thông tin về mối liên hệ giữa nội dung các trang, và dựa vào đó để nâng cao hiệu quả phân lớp và tìm kiếm.

Tuy nhiên, một số phương pháp theo cách thức khai thác yếu tố siêu liên kết lại làm tăng nhanh kích thước vector biểu diễn văn bản trang web và vì vậy một số cải tiến nhằm khắc phục tình huống này đã được đề xuất. Cải tiến các phương pháp biểu diễn của Seán Slattery, chúng tôi cũng đề xuất bổ sung thêm một phương pháp biểu diễn khác.

Một số tác giả khác đưa ra cách cải tiến định hướng vào việc cách liệt kê thêm các từ khóa từ các trang web láng giềng bằng cách chỉ bổ sung các từ khóa xuất hiện trong

đoạn văn bản lân cận với siêu liên kết. Vấn đề này hiện cũng đang được quan tâm nghiên cứu và triển khai.

Ưu điểm của tất cả các phương pháp biểu diễn trên đây là vừa khai thác được thế mạnh của mô hình vector trong biểu diễn văn bản lại vừa đưa thêm được yếu tố liên kết của các trang web theo các siêu liên kết.

Chi tiết theo cách tiếp cận biểu diễn trang web theo mô hình vector, mà trọng tâm là các giải pháp của Seán Slattery bao gồm cách biểu diễn webpage do luận văn đề xuất, được đề cập tại phần 2.2.2 của luận văn.

1.2.2 Cách tiếp cận theo web site

Cách tiếp cận theo website là cách coi đối tượng tìm kiếm là các web site thay cho các trang web trong cách tiếp cận thông thường. Vào những năm 1999-2000, một số tác giả [2,4] đã đề xuất sơ bộ về việc sử dụng website như đối tượng của biểu diễn, phân lớp và tìm kiếm. Phát triển các đề xuất đó, trong công trình nghiên cứu khoa học [6], Martin Ester, Hans-Peter Kriegei, Matthias Schubert đã trình bày giải pháp khá đầy đủ về vấn đề này.

- ***Cơ sở thực tiễn của phương pháp tiếp cận website***

Toàn bộ một website (cấu trúc và nội dung của nó) thường cho thông tin khá trọn vẹn về lĩnh vực hoạt động của một công ty, một cơ quan, một tổ chức ... Tuy nhiên, khi chiết xuất thông tin từ Internet thì hầu hết các phương pháp đã thiết lập đều tập trung vào việc phát hiện ra các trang web độc lập, còn việc phát hiện hoàn toàn các website thì vẫn chưa được quan tâm thỏa đáng, mặc dù vấn đề này rất quan trọng trong nhiều lĩnh vực. Ví dụ trong lĩnh vực thương mại về Công nghệ thông tin, khi mà các sản phẩm và các dịch vụ thay đổi với tốc độ nhanh chóng thì một hệ thống có năng lực đặc biệt trong việc phát hiện các website và cung cấp khả năng để tìm kiếm các website đó sẽ rất có ích. Ngày nay hầu hết các công ty kinh doanh và buôn bán trong tất cả các lĩnh vực đều thiết lập các website giới thiệu về mình trên WWW. Toàn bộ nội dung và cấu trúc của các website thường được thiết kế có mục đích và dựa vào nội dung cung cấp

trên toàn bộ website đó chúng ta có thể biết được họ hoạt động trong lĩnh vực gì ... còn nếu chỉ dựa vào nội dung của các trang web đơn trong các website đó thì khó có thể hình dung và biết chính xác được về chủ để của toàn bộ website. Khi các công ty có nhu cầu cần biết ai là các đối thủ hoạt động trong cùng một lĩnh vực, ai là những người có thể trợ giúp, liên kết hoạt động và ai là khách hàng thì họ có thể dựa vào nội dung của toàn bộ các website để quyết định được điều này.

Một số lý do khác nữa để việc tìm kiếm tập trung vào các website thay vì theo từng trang web đơn là: số lượng các website trên Internet thì ít hơn nhiều so với các trang web đơn, do đó không gian tìm kiếm sẽ giảm đi đáng kể. Và khi khai phá các website thì chính là một bước lọc cho việc tìm kiếm thông tin chi tiết. Ví dụ khi muốn tìm giá vé máy bay thì đầu tiên chúng ta nên tìm kiếm các website của các đại lý du lịch để thu hẹp phạm vi tìm kiếm trước, sau đó mới tiến hành tìm kiếm theo cách tìm kiếm thông thường.

Lý do tiếp theo cho cách tiếp cận websita là độ ổn định của các website cao hơn hẳn các trang đơn. Các site xuất hiện, thay đổi và biến mất với tần số ít hơn hẳn so với các trang đơn, do các trang đơn là các trang được cập nhật thường xuyên hàng ngày. Tất nhiên một số ít các site cũng thay đổi, nhưng trong hầu hết các trường hợp thì các site là rất ít thay đổi.

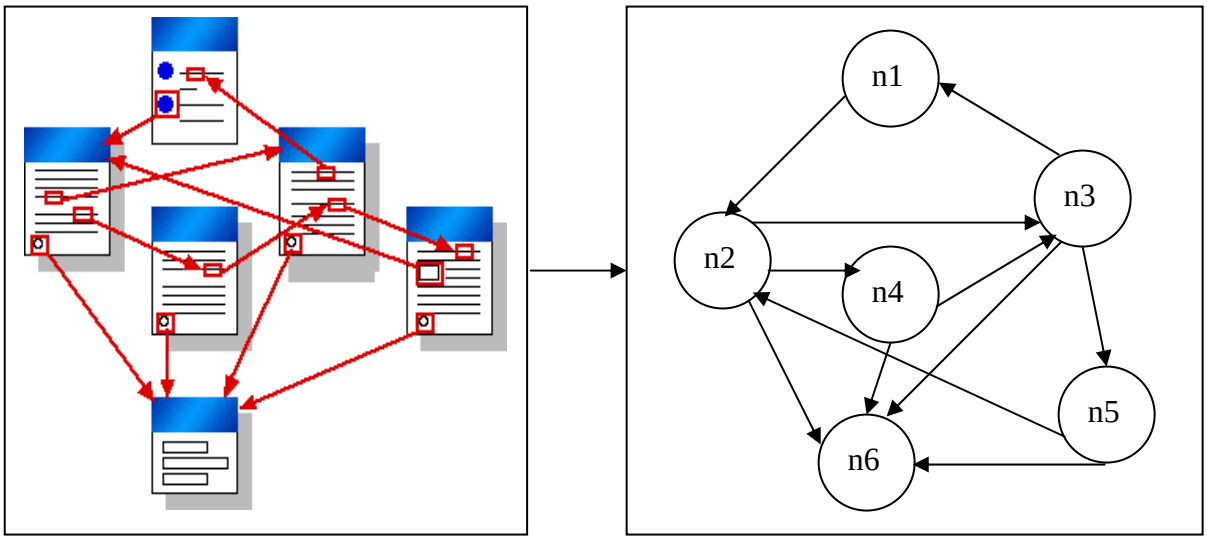
- ***Các vấn đề cần giải quyết***

Việc khai phá hoàn toàn một website có rất nhiều điểm khác biệt so với việc khai phá các trang web đơn. Các site thường có kích thước lớn, được xây dựng nên từ các cấu trúc và kỹ thuật phức tạp. Còn một khía cạnh khác nữa là ngôn ngữ. Rất nhiều các trang chuyên nghiệp được viết ít nhất là song ngữ (có thêm bản tiếng Anh) để tiện lợi cho tất cả mọi người có thể hiểu được tiếng Anh. Không kể các nghiên cứu có tính đến tính chất đa ngôn ngữ [9,12] thì hầu hết các dự án phân lớp các trang web thường chỉ tính đến các tài liệu viết bằng một ngôn ngữ, vì vậy mà có thể sẽ thiếu điều kiện khi muốn xử lý hoàn toàn cả website.

Vấn đề thứ hai xuất hiện là công việc xác định phạm vi của các site. Khi phân lớp các trang đơn thì vấn đề này rất đơn giản vì mỗi trang là một đối tượng cần quan tâm, còn đối với một site thì phức tạp hơn. Một số tác giả đã chọn giải pháp xác định phạm vi của một website bằng cách dựa vào sự phân lớp các trang web thuộc website đó [6].

Một vấn đề nữa là mỗi site không chỉ là một tập các thuật ngữ mà còn là một tập các trang đơn, do đó muốn xử lý chúng thì còn cần phải biểu diễn được cấu trúc của toàn bộ website.

- ***Cách giải quyết***



Hình 1.3. Mô hình biểu diễn cấu trúc một website bằng đồ thị

Martin Ester, Hans-Peter Kriegel and Matthias Schubert [6] đã thực hiện việc phân lớp các website dựa vào việc trình bày mỗi website như một cây, và máy phân lớp sẽ làm việc dựa vào đường đi trong các cây đó. Để biểu diễn cấu trúc của một website, các tác giả đã sử dụng các phương pháp biểu diễn chung của đồ thị.

Một website của một tên miền D là một đồ thị có hướng, ký hiệu là G (N, E). Một nút n $\in$ N biểu diễn một trang web, mà URL bắt đầu với D. Một liên kết giữa n1 và n2 (với n1, n2 $\in$ N) được biểu diễn bằng cạnh có hướng (n1, n2) $\in$ E (hình 1.3).

Như vậy tất cả các trang web trong cùng một miền thì đều là các nút trong đồ thị biểu diễn cho tên miền đó, và các liên kết giữa các trang là các cạnh nối các nút đó.

Định nghĩa đơn giản này thực sự lại giúp chúng ta rất nhiều trong quá trình thực hiện các ứng dụng nhằm mục đích phát hiện ra các site thương mại có kích thước nhỏ và vừa. Hầu hết tất cả các công ty đều thuê tên miền riêng để sử dụng cho mình, do đó khả năng để một website mới khác bắt đầu dưới một tên miền (một website) đang xét là rất ít (nghĩa là dưới một tên miền thì thường là các trang web nằm trong chính website đó chứ ít khi có một website mới bắt đầu). Còn các website trải dài trên một vài tên miền khác nhau thì thường là ít và là các website của các công ty rất lớn, mà các website đó thì hầu hết mọi người đều đã biết, do đó không cần thiết phải quan tâm đến chúng.

Để tải về một website từ Internet có thể áp dụng thuật toán sau đây: bắt đầu từ một trang web có địa chỉ URL là một tên miền trực tiếp, gọi đó là trang bắt đầu. Trong khi đọc trang đó, sử dụng phân tích cú pháp HTML để xác định các liên kết đến các trang khác trong cùng website. Chú ý rằng các thẻ HTML có tên là FRAME và EMBED là các liên kết cần thiết để có thể hoàn thành được toàn bộ đồ thị của cả website. Sau khi các liên kết này được phân tích thì tất cả các liên kết bắt đầu từ cùng một tên miền sẽ được xem xét. Một việc cần thực hiện là phải đánh dấu lại các trang web đã được đến thăm để tránh quẩn (chẳng hạn, sử dụng giải pháp của quá trình indexing trong các máy tìm kiếm). Vì vậy, tất cả các trang có thể đi tới được thì đều được thăm và tất cả các liên kết tìm được sẽ được thăm cho đến khi hoàn thành được đồ thị biểu diễn website này.

Cách thông thường nhất để phân loại các trang web là sử dụng máy phân lớp Bayes tự nhiên hoặc sử dụng máy vector trợ giúp (SVM - Support Vector Machine) trong không gian các từ khóa. Độ chính xác của kết quả phân lớp phụ thuộc rất nhiều vào việc lựa chọn các từ khóa.

Bài toán phân lớp các website được xác định như sau: Ký hiệu C là tập các lớp website đã được biết, và S là một website mới (website S có thể bao gồm một tập các trang P, hoặc bất cứ một cấu trúc dữ liệu nào như đồ thị). Bài toán đặt ra (bài toán phân lớp website) là xác định xem website S phù hợp nhất với lớp (thành phần) nào của C.

Cách đơn giản nhất để phân lớp website là mở rộng phương pháp phân lớp trang web sao cho phù hợp với định nghĩa về website. Cách đơn giản là chỉ cần xây dựng các vector đặc trưng đơn để đếm tần số các từ trong tất cả các trang web nằm trong toàn bộ website, nghĩa là có thể coi website là một siêu trang (superpage) bao gồm các trang đơn. Và cách tiếp cận này có thể gọi là cách phân lớp các siêu trang. Có thể coi cách tiếp cận này sử dụng phương pháp biểu diễn trang web thứ hai [11] với thay đổi là không chỉ kể đến các trang web láng giềng mà kể tới tất cả các trang web trong website. Điểm thuận lợi của cách tiếp cận này là không quá phức tạp so với việc phân lớp các trang đơn. Chỉ cần duyệt qua các nút trong biểu đồ của các trang web trong một website rồi đếm các từ khóa và xây dựng vector biểu diễn. Sau đó vector biểu diễn có thể được phân lớp bởi một máy phân lớp chuẩn bất kỳ được chọn.

Tuy nhiên cách tiếp cận phân lớp siêu trang lại tồn tại một số vấn đề hạn chế về mặt nhận thức. Ví dụ như chúng ta đã biết một website có thể bao gồm rất nhiều trang viết bằng các ngôn ngữ khác nhau, hay các thuộc tính cấu trúc (ví dụ các frame trong một tab) có thể làm mất hầu hết các ý nghĩa của chúng. Và một vấn đề quan trọng nữa là cách phân lớp này làm mất ngữ cảnh cục bộ của các trang trong website, do tất cả các từ xuất hiện trong site đều được sử dụng để xây dựng nên vector biểu diễn. Mà ngữ cảnh xuất hiện các từ khóa trong các trang web lại đóng một vai trò quan trọng. Một ví dụ minh hoạ đơn giản về tính quan trọng của ngữ cảnh như sau: nghĩa của cụm từ "quản trị mạng" và "dịch vụ" nằm trong cùng một trang của một công ty ngụ ý rằng công ty đó cung cấp các dịch vụ và trong đó có dịch vụ quản trị mạng. Nhưng nếu các từ khóa này không cùng xuất hiện trong một trang mà nằm riêng rẽ ở các trang khác nhau thì ý nghĩa lại khác đi rất nhiều. Chẳng hạn một công ty, cung cấp dịch vụ bất kỳ (không phải dịch vụ quản trị mạng) và đang tìm kiếm một người "quản trị mạng" cũng đều đưa các cụm từ đó lên các trang web trong website của mình. Qua việc đánh giá kết quả thực nghiệm đã được tiến hành, Martin Ester, Hans-Peter Kriegel và Matthias Schubert [6] đã chỉ ra rằng cách phân lớp siêu trang web cho kết quả không tốt.

Để khắc phục các tồn tại của phương pháp phân lớp siêu trang web, cần đưa ra việc cải tiến, trước hết là cách biểu diễn các website sao cho tự nhiên hơn và mang ý nghĩa nhiều hơn. Thay vì cách tập trung vào các từ đơn để phân loại website, chúng ta tập trung vào việc biểu diễn website thông qua việc tóm tắt nội dung các trang web trong website đó. Việc tóm tắt nội dung trang web được thực hiện thông qua việc ấn định trang web đó một chủ đề trong một tập các chủ đề đã được xác định trước đó. Khi đó nội dung của các từ khóa chỉ ảnh hưởng đến nội dung các trang web chứa nó, và như vậy là ngữ cảnh cục bộ được bảo toàn.

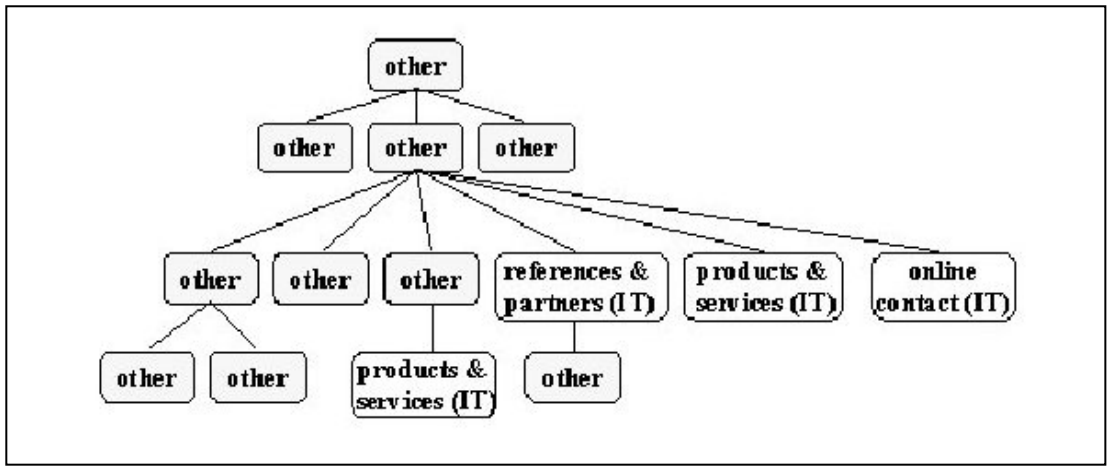
Có hai bài toán cần được giải quyết ở đây. Thứ nhất, bài toán tiền xử lý phân trang web theo chủ đề được giải quyết nhờ việc sử dụng tất cả các kỹ thuật đã được áp dụng cho việc phân lớp các trang web qua việc thu thập từ khóa. Thứ hai, bài toán lựa chọn tập các chủ đề dùng cho việc gán một chủ đề tương ứng tới một trang web được giải quyết dựa vào quá trình nghiên cứu, đánh giá các trang web của rất nhiều website kinh doanh khác nhau. Kết luận qua việc nghiên cứu, đánh giá đó cho thấy mặc dù các công ty thuộc vào rất nhiều lĩnh vực kinh doanh khác nhau, nhưng hầu hết các trang web trong các website của chúng thuộc vào mười chủ đề sau đây: company, company philosophy, online contact, places and opening hours, product and services, references and patners, employees, directory, vacancies và “other”. Chủ đề “other” là chủ đề dùng cho một trang bất kỳ mà không được xác định chính xác thuộc vào một trong các chủ đề trước đó. Chú ý rằng tập các lớp chủ đề trong danh sách trên đây đề cập tới một ứng dụng phân lớp riêng biệt, vì vậy vẫn mang tính chất minh hoạ, tuy nhiên, phương pháp đã được trình bày có thể áp dụng tốt cho bất cứ lớp website nào.

Tiếp theo đó, dựa vào chủ đề (nhãn) của các trang web thuộc website, Martin Ester, Hans-Peter Kriegel and Matthias Schubert đưa ra hai phương pháp biểu diễn website như sau:

Phương pháp thứ nhất là phương pháp xây dựng vector tần số chủ đề cho website. Theo phương pháp này, mỗi một website tương ứng với một vector có số thành phần (số chiều) bằng số lượng chủ đề trong tập chủ đề đã được khám phá (trong ví dụ

nói trên, vector biểu diễn website có 10 thành phần). Mỗi thành phần của vector biểu diễn website có giá trị số là số lượng các trang web thuộc website có chủ đề tương ứng. Cách biểu diễn này tuy không khai thác được cấu trúc liên kết của website nhưng nó cho phép nhìn nhận một website như một tập các trang web đã được gán chủ đề. Do tập chủ đề được chọn là không nhiều cho nên kích thước vector biểu diễn website là nhỏ.

Sau khi đã biểu diễn các website thành các vector tần số chủ đề, chúng ta có thể phân lớp các website bằng các máy phân lớp thông thường như Bayes tự nhiên hay cây quyết định. Việc phân lớp này có điểm thuận lợi là số chiều của vector tần số chủ đề ít hơn rất nhiều so với số chiều của vector tần số từ (theo cách tiếp cận siêu trang) cho nên thời gian phân lớp nhanh .



Hình 1.4. Một cây biểu diễn website

Phương pháp thứ hai là phương pháp xây dựng cây biểu diễn website. Theo phương pháp này, để lưu giữ được bản chất của cấu trúc liên kết trong website, chúng ta dùng một cây gắn nhãn để trình bày website. Cây gắn nhãn này được bắt đầu từ một nút gốc (duy nhất và tương ứng với trang khởi đầu hay trang gốc), và tiếp theo là các nút trong (hoặc là các trang thư mục - để cung cấp một cái nhìn khái quát về các chủ đề trong website và các liên kết đến các trang web khác, hoặc là các trang vừa bao gồm các mục thư mục vừa bao gồm nội dung) và cuối cùng là các lá (tương ứng với các trang web nội dung). Hơn nữa, hầu hết các website thường bắt đầu từ các thông tin

chung nhất, gắn bó nhất với lĩnh vực hoạt động của công ty tại các trang gần với trang gốc, và càng chuyên biệt hoá đối với các trang càng ở xa trang gốc. Để xây dựng cây website, Martin Ester, Hans-Peter Kriegei, Matthias Schubert sử dụng số lượng nhỏ nhất các liên kết như là một giá trị đo khoảng cách giữa hai trang web trong website và xây dựng một cây như một tập các đường nhỏ nhất từ trang gốc đến tất cả các trang trong đồ thị. Tiếp theo, thực hiện việc tìm kiếm theo chiều ngang trên toàn bộ đồ thị, bỏ qua tất cả các liên kết đến các trang web đã được thăm. Lưu ý rằng, trong trường hợp có hai đường cùng độ dài dẫn đến cùng một trang web thì đường nào xuất hiện trước sẽ được chọn. Hình 1.4. trình bày một cây website được sinh ra bởi phương pháp này khi thực hiện cho một website ví dụ với lớp chủ đề đã có.

Tiếp theo đó, thực hiện việc xây dựng một máy phân lớp các cây website, được gọi là cây Markov thứ tự O (0-order Markov tree), bằng cách sử dụng ý tưởng về chuỗi Markov kết hợp với máy phân lớp Bayes tự nhiên [14]. Kết quả thực nghiệm cho thấy máy phân lớp cây Markov thứ tự O cho kết quả chính xác hơn so với máy phân lớp truyền thống Bayes tự nhiên và máy phân lớp siêu trang.

Qua tìm hiểu về phương pháp tiếp cận theo website, có thể thấy đây là một phương pháp tiếp cận mới với ưu điểm là thu hẹp được không gian tìm kiếm, và trong một vài ứng dụng đặc biệt thì cho kết quả tốt. Tuy nhiên cách tiếp cận này cũng có một số điểm yếu, đó là đã bỏ mất giá trị thẻ của ngôn ngữ HTML, không tận dụng được một số thông tin tiềm tàng của văn bản web và đặc biệt là không còn giữ được ý nghĩa cục bộ của các trang web đơn.

- ***Đề xuất một phương pháp xây dựng cây website***

Martin Ester, Hans-Peter Kriegei, Matthias Schubert [6] giới thiệu sơ lược về quá trình xây dựng cây website cho một website. Chúng tôi đề xuất thuật toán cụ thể sau đây (Thuật toán 1.1) nhằm giải quyết bài toán xây dựng cây website. Tư tưởng của thuật toán dựa trên quá trình "loang" dần các trang web trong website đó. Mặt khác, các URL chỉ dẫn tới trang web không thuộc website nói trên được bỏ qua.

Thuật toán sử dụng tập các trang web đã được khám phá Tap_hien_thoi được làm được mở rộng dần cho đến khi không mở rộng được nữa thì thuật toán kết thúc. Trong mỗi bước, thuật toán thiết lập tập các nút ký hiệu là Tap_muc_I gồm các nút trong cây website có mức bằng I.

Thuật toán 1.1. (Xây dựng cây website)

Input: U_o là URL trang chủ của một website

Output: Cây website có U_o là trang chủ

Nội dung:

Bước 1. (Khởi tạo)

(1.1) $Tap\_muc_o \leftarrow \{ U_o\}$

(1.2) $I \leftarrow 0$

(1.3) $Tap\_hien\_thoi \leftarrow Tap\_muc_o$

Bước 2. (loang dần theo mức các nút)

Repeat

(2.1) $I \leftarrow I + 1$

(2.2) $Tap\_muc_I \leftarrow \varnothing$

(2.3) $\forall U \in Tap\_muc_{I-1}$ thực hiện

(2.3.1) Đọc trang web có địa chỉ là U,

(2.3.2) $\forall$ V là URL xuất hiện trong trang web vừa đọc (V chỉ dẫn tới trang web được tổ chức trong host chứa U_o)

(2.3.2.1) Nếu $V \in Tap\_hien\_thoi \cup Tap\_muc_I$ thì bỏ qua

(2.3.2.2) Ngược lại ($V \notin Tap\_hien\_thoi \cup Tap\_muc_I$)

(i) $Tap\_muc_I \leftarrow Tap\_muc_I \cup \{V\}$

(ii) $Tap\_hien\_thoi \leftarrow Tap\_hien\_thoi \cup \{V\}$

(iii) Tạo một cung đi từ U tới V (V là con của U)

Until $Tap\_muc_I = \varnothing$

Việc "loang dần" theo mức các nút (Bước 2) với việc kiểm tra các nút con thuộc cây (bước 2.3.2) phù hợp với thuật toán "loang" các nút trong một cây chứng tỏ thuật toán 1.1 chính xác xây dựng cây website cần có.

KẾT LUẬN CHƯƠNG MỘT

Cơ sở dữ liệu các trang Web đang được nhiều nhà khoa học trên thế giới quan tâm, trong đó các bài toán biểu diễn trang Web, tìm kiếm và phân lớp là những bài toán trọng tâm nhất. Tồn tại một số phương pháp biểu diễn và xử lý văn bản Web; đáng chú ý là biểu diễn trang web (dựa theo/không dự theo mô hình vector) và biểu diễn website trong đó hầu hết các kết quả nghiên cứu được công bố liên quan đến lĩnh vực biểu diễn và xử lý trang web. Mỗi phương pháp nói trên đều có những ưu điểm riêng trong mỗi phạm vi ứng dụng, tuy nhiên cũng còn một số tồn tại như tốn nhiều không gian bộ nhớ hoặc khối lượng tính toán lớn. Chương một cũng trình bày chi tiết về một cách tiếp cận theo phương pháp biểu diễn website và luận văn đề xuất thuật toán 1.1 để giải quyết bài toán xây dựng cây website.

Trong chương hai, cùng với việc trình bày và phân tích kỹ lưỡng hơn về ưu, nhược điểm của các phương pháp tiếp cận trên đây, luận văn đề xuất một phương pháp tiếp cận kết hợp để giải quyết bài toán tìm kiếm trong cơ sở dữ liệu trang web nhằm nâng cao hiệu quả tìm kiếm. Ý tưởng của phương pháp mới được đề xuất dựa trên việc kết hợp giữa phương pháp tiếp cận của các máy tìm kiếm thông thường (để tận dụng được tính nửa cấu trúc và các mối liên kết của dữ liệu trang web) với phương pháp biểu diễn vector để bổ sung thêm trọng số cho các từ khóa (tần số xuất hiện của các từ khóa trong các trang). Đồng thời, luận văn đề xuất việc bổ sung thêm chức năng tìm kiếm các trang web có nội dung gần với nội dung của trang web hiện thời vào máy tìm kiếm.

CHƯƠNG II. MỘT SỐ PHƯƠNG PHÁP BIỂU DIỄN TRANG WEB VÀ GIẢI PHÁP KẾT HỢP

Biểu diễn dữ liệu là một công việc rất quan trọng đối với các bài toán tìm kiếm, lưu trữ, phân lớp hay phân cụm dữ liệu. Bất cứ là công việc gì thực hiện với dữ liệu thì vấn đề biểu diễn dữ liệu cũng là tiền đề quan trọng và có ảnh hưởng rất lớn đối với các quá trình sau đó. Nếu dữ liệu trong hệ thống được biểu diễn và xử lý bằng các phương pháp tốt và phù hợp thì sẽ giúp cho các công việc tiếp sau đó được thực hiện dễ dàng và hiệu quả hơn rất nhiều.

Biểu diễn văn bản là cách trình bày văn bản của các tài liệu trong cơ sở dữ liệu trang Web để có thể dễ dàng quản lý, tìm kiếm và làm việc với chúng.

Như đã được giới thiệu trong chương 1, trong lĩnh vực xử lý dữ liệu cho các bài toán tìm kiếm trang web tồn tại một số phương pháp biểu diễn như: sử dụng mô hình vector, logic mờ, mạng ngữ nghĩa hay sử dụng file cơ sở dữ liệu các bản ghi... mỗi phương pháp biểu diễn đều có các ưu nhược điểm riêng và phù hợp với từng hướng khác nhau giải quyết các bài toán đó. Vì vậy, trước khi giải quyết một bài toán, chúng ta cần tìm hiểu kỹ các yêu cầu của bài toán cũng như các ưu, nhược điểm riêng của từng phương pháp biểu diễn trang web để có thể chọn được phương pháp biểu diễn phù hợp nhất áp dụng cho bài toán của mình.

Cơ sở dữ liệu trang web thường chứa đựng số lượng cực lớn các tài liệu được lưu trữ ở nhiều máy tính khác nhau trên toàn thế giới, mà nội dung một số trang web lại có thể thay đổi thường xuyên nên các phương pháp biểu diễn truyền thống (biểu diễn dữ liệu fulltext thông thường) không còn phù hợp nữa, hoặc là hoạt động không hiệu quả. Một nhu cầu được đặt ra là phải xây dựng các phương pháp biểu diễn mới, hoặc cải tiến các phương pháp biểu diễn dã có cho phù hợp với các điều kiện mới.

Sau đây, chúng tôi trình bày chi tiết hai lớp phương pháp biểu diễn trang web phổ biến hiện nay để chỉ ra được sự thay đổi và cải tiến phù hợp với điều kiện của từng bài toán tìm kiếm khác nhau. Lớp phương pháp thứ nhất được dùng trong các hệ thống máy

tìm kiếm, trong đó nhấn mạnh ngữ nghĩa của việc liên kết các trang web trong việc tính hạng của trang web. Trong quá trình tiền xử lý văn bản trang web, hạng của nó được hoàn thiện dần theo công thức tính dần từng bước cho đến khi hoàn thiện hệ thống. Sau đó, hạng của trang web được dùng cho việc hiển thị các trang web kết quả tìm kiếm cho người dùng. Lớp thứ hai dựa trên việc phát triển mô hình vector trong biểu diễn dữ liệu fulltext. Đại diện cho lớp phương pháp theo hướng này được Sean Slattery trình bày [11]. Mỗi trang web được tương ứng với một vector biểu diễn. Câu hỏi tìm kiếm đa dạng và phong phú hơn lớp thứ nhất và kết quả tìm kiếm được hiển thị dựa theo "độ gần nhau" của câu hỏi với các trang web.

2.1 Phương pháp biểu diễn trong các máy tìm kiếm

Sự phát triển nhanh chóng của mạng Internet và Intranet đã sinh ra một khối lượng khổng lồ các trang web. Cùng với sự phát triển và thay đổi hàng ngày hàng giờ về nội dung cũng như số lượng của các trang web trên Internet thì vấn đề tìm kiếm thông tin đối với người sử dụng lại ngày càng khó khăn. Một vấn đề cần được giải quyết là: Làm thế nào để tìm ra được các trang web có mang thông tin cần thiết trong số hàng tỷ các trang web? Việc này chỉ có thể thực hiện được nhờ vào các máy tìm kiếm (search engine) hiện đang được cung cấp rộng rãi cho mọi người sử dụng trên Internet, chẳng hạn như Yahoo, Google, Altavista...

Máy tìm kiếm là các hệ thống được xây dựng có khả năng tiếp nhận các yêu cầu tìm kiếm của người dùng (thường là một tập các từ khoá), sau đó phân tích và tìm kiếm trong cơ sở dữ liệu đã có sẵn và đưa ra các kết quả các trang web cho người sử dụng.

Như đã biết, bài toán biểu diễn và tìm kiếm thông tin trên Internet đặt ra nhiều thách thức. Thứ nhất, tập hợp trang web trên Internet là một tập dữ liệu khổng lồ, phân tán trên rất nhiều máy tính khắp nơi trên thế giới. Thứ hai, nội dung các trang web không hoàn toàn đồng nhất, chẳng hạn vấn đề ngôn ngữ trình bày trang web bao gồm rất nhiều loại ngôn ngữ khác nhau (cả ngôn ngữ diễn tả nội dung lẫn ngôn ngữ lập trình), nhiều loại định dạng khác nhau (text, HTML, PDF, hình ảnh, âm thanh,...),

nhiều loại từ vựng khác nhau (địa chỉ email (email addresses), các liên kết (links), các mã nén (zip code), số điện thoại (phone number),...). Và thứ ba là nội dung trang web thay đổi liên tục và không ai có thể kiểm soát nổi. Các nghiên cứu về kích thước của hệ thống web đã đưa ra các số liệu sau đây để minh chứng cho các khó khăn đó [6]. Hiện nay có khoảng hơn một tỷ các trang web được cung cấp cho người sử dụng, giả sử kích thước trung bình của mỗi trang web là 5-10 KB, thì kích thước tổng cộng của hệ thống ít nhất khoảng 10 terabyte. Mặt khác, tốc độ tăng số lượng các trang web cũng rất nhanh, chẳng hạn, trong hai năm gần đây số lượng các trang web đã tăng lên gấp đôi. Ngoài số lượng lớn các trang web được tạo mới thì các trang web đang tồn tại trên Internet cũng không ngừng cập nhật thông tin. Theo kết quả nghiên cứu hơn 500.000 trang web trong hơn 4 tháng thì 23% các trang thay đổi hàng ngày. Trong các site mà tên miền có đuôi .com thì 40% các trang thay đổi hàng ngày, và khoảng 10 ngày thì 50% các trang trong các tên miền đó biến mất, nghĩa là địa chỉ URL của chúng không còn tồn tại nữa.

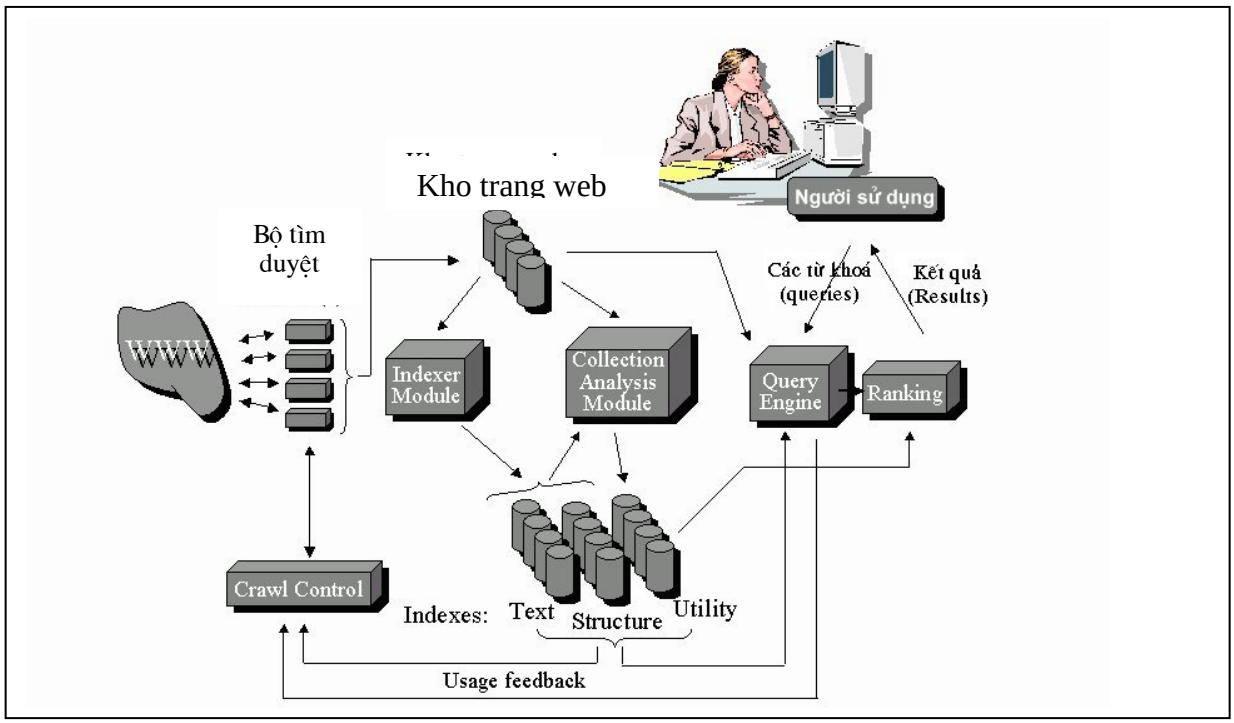
Các thách thức trên đây cho thấy việc biểu diễn dữ liệu trong các máy tìm kiếm là rất quan trọng. Biểu diễn các trang web như thế nào để vừa có khả năng lưu trữ được một số lượng khổng lồ các trang web đó, vừa cho phép máy tìm kiếm thực hiện việc tìm kiếm nhanh chóng và chính xác. Trước hết chúng ta khảo sát cấu trúc cơ bản của máy tìm kiếm và hoạt động của nó.

2.1.1 Cấu trúc cơ bản và hoạt động của một máy tìm kiếm

Cấu trúc điển hình của một máy tìm kiếm được mô tả như trong hình 2.1. Trong thực tế thì mỗi máy tìm kiếm lại có các sửa đổi riêng theo cách riêng, tuy nhiên về cơ bản vẫn dựa trên các bộ phận được mô tả trong hình 2.1.

Bộ tìm duyệt (Crawler): Hầu hết các máy tìm kiếm hoạt động dựa vào các bộ tìm duyệt là các chương trình có kích thước nhỏ đảm nhận chức năng cung cấp dữ liệu (các trang web) cho máy tìm kiếm hoạt động. Bộ tìm duyệt thực hiện công việc duyệt web. Hoạt động của nó tương tự như hoạt động của con người khi truy cập web là dựa

vào các mối liên kết để đi từ trang web này tới trang web khác. Các bộ tìm duyệt được cung cấp các địa chỉ URL xuất phát, đọc trang web tương ứng, phân tích và tìm ra các URL có trong các trang web đó. Sau đó bộ tìm duyệt cung cấp các URL kết quả cho **bộ điều khiển tìm duyệt (Crawl Control)**. Bộ điều khiển tìm duyệt sẽ quyết định xem URL nào sẽ được tìm duyệt tiếp theo và gửi lại kết quả quyết định cho bộ tìm duyệt (trong một số máy tìm kiếm, bộ tìm duyệt thực hiện luôn chức năng của bộ phận điều khiển tìm duyệt). Các bộ tìm duyệt cũng chuyển luôn các trang web đã duyệt vào **kho trang web (Page Repository)**. Sau đó, các bộ tìm duyệt tiếp tục đi thăm các trang web



Hình 2.1. Mô hình cấu trúc của máy tìm kiếm

khác trên Internet cho đến khi các nguồn chứa cạn kiệt.

Bộ tạo chỉ mục (Indexer Module) thực hiện việc khảo sát tất cả các từ khóa trong từng trang web có trong kho trang web, và ghi lại các địa chỉ URL của các trang web có chứa mỗi từ. Kết quả sinh ra một bảng chỉ mục rất lớn (thực sự, bảng chỉ mục giới hạn trong các trang web đã qua bộ tìm duyệt). Nhờ có bảng chỉ mục này, máy tìm kiếm cung cấp tất cả các địa chỉ URL của các trang web khi có yêu cầu: Khi cho một từ

khóa bất kỳ thì qua bảng chỉ mục, máy tìm kiếm sẽ nhận được tất cả các địa chỉ URL của các trang web có chứa từ khóa đó. Chỉ mục này được gọi là chỉ mục nội dung (Text Index).

Việc tạo chỉ mục cho hệ thống web thực sự là một việc làm rất khó khăn do kích thước đồ sộ của hệ thống web cũng như sự thay đổi nhanh chóng của nó và tính phức tạp trong dữ liệu web. Vì vậy tồn tại rất ít cách thức tạo chỉ mục chung. Thông thường, bộ tạo chỉ mục tạo ra **chỉ mục nội dung** và **chỉ mục cấu trúc (Structure Index)** hoặc một số loại **chỉ mục tiện ích (Utility Index)**. Để tạo chỉ mục nội dung thì như đã nói ở trên, bộ tạo chỉ mục phân tích nội dung trang web và chiết xuất ra tất cả các từ xuất hiện trong đó. Để xây dựng chỉ mục cấu trúc (ứng với các siêu liên kết) thì bộ tạo chỉ mục sẽ tạo ra một mô dạng một đồ thị gồm các nút và các cung. Mỗi nút trong đồ thị tương ứng với một trang web, còn mỗi cung nối từ nút A đến nút B tương ứng là siêu liên kết từ trang web A đến trang web B. Cho phép dễ dàng thay đổi các chỉ mục cấu trúc để có thể cập nhật được thông tin về sự thay đổi không ngừng của siêu liên kết trong các trang web. Như vậy chỉ mục cấu trúc là chỉ mục phản ánh mối liên kết giữa các trang web, và việc tạo chỉ mục này cho phép sử dụng đặc tính quan trọng của dữ liệu web là có chứa các siêu liên kết. Chỉ mục cấu trúc thường là không có trong các cơ sở dữ liệu fulltext do các văn bản fulltext không chứa các liên kết.

Bộ phân tích tập (Collection Analysis Module) hoạt động dựa vào thuộc tính của bộ truy vấn (**Query Engine**). Ví dụ nếu bộ truy vấn chỉ đòi hỏi việc tìm kiếm hạn chế trong một số website đặc biệt, hoặc giới hạn trong một tên miền thì công việc sẽ nhanh và hiệu quả hơn khi phải xây dựng một bảng chỉ mục các website mà trong đó có kết nối mỗi tên miền tới một danh sách các trang web thuộc miền đó. Công việc như thế được thực hiện bởi bộ phân tích tập; nó sử dụng thông tin từ hai loại chỉ mục cơ bản (chỉ mục nội dung và chỉ mục cấu trúc) do bộ tạo chỉ mục cung cấp cùng với thông tin từ khoa trang web, và các thông tin được sử dụng bởi phương pháp tính hạng (ranking) để tạo ra các chỉ mục tiện ích.

Bộ truy vấn chịu trách nhiệm nhận các yêu cầu của người sử dụng. Bộ phận này hoạt động thường xuyên dựa vào bảng chỉ mục và thỉnh thoảng dựa vào kho trang web. Do số lượng các trang web là rất lớn, và trong thực tế thì người sử dụng chỉ đưa vào khoảng một hoặc vài từ khoá, cho nên tập kết quả thường rất lớn. Vì vậy bộ xếp hạng (Rangking) có chức năng sắp xếp kết quả thành một danh sách các trang web theo thứ tự giảm dần về độ liên quan (theo máy tìm kiếm) tới vấn đề mà người sử dụng đang quan tâm, và sau đó hiển thị danh sách kết quả tìm được cho người sử dụng.

Khi muốn tìm kiếm các trang web về một vấn đề nào đó, người sử dụng đưa vào một số các từ khoá mà họ coi là liên quan đến vấn đề cần quan tâm (gọi là từ khóa tìm kiếm). Bộ truy vấn dựa theo các từ khoá tìm kiếm và tìm trong bảng chỉ mục địa chỉ các trang web có chứa các từ khoá tìm kiếm. Sau đó, bộ truy vấn chuyển các trang web kết quả cho **bộ xếp hạng** để sắp xếp các kết quả theo thứ tự rồi hiển thị kết quả cho người sử dụng.

Vấn đề quan tâm ở đây là cách biểu diễn trang web trong máy tìm kiếm (phần Index trong hình 2.1), trong đó chú trọng tới cách thức bộ tạo chỉ mục xây dựng chỉ mục cho trang web và phương pháp lưu trữ các chỉ mục đó trong bảng chỉ mục để đáp ứng được yêu cầu hoạt động của máy tìm kiếm. Cần phân biệt cách biểu diễn dữ liệu theo cách đánh chỉ mục nội dung và cách đánh chỉ mục cấu trúc cũng như cách đánh chỉ mục tiện ích.

2.1.2 Phương pháp biểu diễn dữ liệu trong các máy tìm kiếm

- ***Biểu diễn chỉ mục nội dung***

Chỉ mục nội dung trợ giúp cho việc tìm kiếm theo nội dung (text-based retrieval), giúp cho máy tìm kiếm có thể sử dụng bất cứ một phương pháp truy nhập truyền thống nào để tìm kiếm trong các bộ dữ liệu. Máy tìm kiếm sử dụng chỉ mục liên kết ngược (**inverted index**) cho việc biểu diễn tài liệu.

Một chỉ mục liên kết ngược bao gồm một tập các danh sách ngược (inverted list), mỗi danh sách ngược tương ứng với một từ khóa. Một danh sách ngược đối với một từ

khóa là một danh sách ngắn các định vị nơi xuất hiện từ khóa đó trong bộ dữ liệu. Trường hợp đơn giản nhất, định vị bao gồm mã trang web (trong kho trang web) chứa từ khóa và vị trí của từ khóa đó trong trang web. Tuy nhiên các thuật toán tìm kiếm thường sử dụng thêm các thông tin phụ liên quan đến vị trí xuất hiện của từ khóa trong trang web. Ví dụ, từ khóa xuất hiện nằm trong cặp thẻ <B>, nằm trong phần tiêu đề (heading), hay từ khóa nằm trong siêu liên kết ... thì có thể sẽ cho độ quan trọng khác nhau trong thuật toán xếp hạng. Để điều tiết việc này thì một thông số trọng tải phụ được thêm vào định vị. Thông số trọng tải này mã hoá bất cứ một thông tin phụ nào cần thiết để bảo toàn tính chất của mỗi lần xuất hiện từ khóa. Cho một từ khóa w và định vị là l, hệ thống trình bày một cặp (w,l) tương ứng như là một mã cho w.

Để minh hoạ cho điều trình bày trên đây, ví dụ có 4 tài liệu với nội dung như sau (dãy kí tự nằm trong cặp dấu ngoặc “” , để đơn giản các ký tự là chữ thường):

Tài liệu 1: “i love you”

Tài liệu 2: “god is love”

Tài liệu 3: “love is blind”

Tài liệu 4: “blind justice”

Việc tạo các chỉ mục cho các tài liệu này được thực hiện như sau:

1. Chiết xuất tất cả các từ khóa có mặt trong cả 4 tài liệu
2. Lưu trữ chúng theo thứ tự từ điển a, b, c, ...
3. Lưu trữ các thông tin về tài liệu (bao gồm mã tài liệu, địa chỉ URL, tiêu đề, mô tả ngắn gọn...)

Kết quả thu được một chỉ mục ngược là một danh sách các thông tin như sau:

Từ	Mã tài liệu	Vị trí xuất hiện	địa chỉ URL	Tiêu đề	Miêu tả ngắn gọn
blind	3	8	...	...	...
blind	4	0	...	...	...
god	2	0	...	...	...
i	1	0	...	...	...
is	2	4	...	...	...

is	3	5	...	...	...
justice	4	6	...	...	...
love	1	2	...	...	...
love	2	7	...	...	...
love	3	0	...	...	...
you	1	7	...	...	...

Từ "blind" trong tài liệu 3 bắt đầu tại ký tự thứ 8, vì vậy có giá trị mã tài liệu là 3 và vị trí xuất hiện là 8, tương tự như vậy đối với các từ khác.

Khi có yêu cầu tìm kiếm tài liệu với từ khóa là "is" và "love" thì đầu tiên máy tìm kiếm tìm ra danh sách tất cả các trang web có chứa từ "is" và tất cả các trang web có chứa từ "love", sau đó lấy phần giao của hai danh sách này. Trong trường hợp này thì tài liệu số 2 và 3 đều có chứa cả 2 từ khóa. Như vậy máy tìm kiếm nhanh chóng tìm ra các trang web có chứa các từ khoá tìm kiếm.

Chỉ mục ngược được lưu trữ qua file cơ sở dữ liệu các bản ghi. Mỗi một danh sách ngược lưu trữ thông tin về một từ và tương ứng là một bản ghi trong cơ sở dữ liệu .Việc xây dựng một cơ sở dữ liệu để lưu trữ danh sách ngược cho một bộ dữ liệu lớn như tập các trang web trên Internet đòi hỏi một kiến trúc phân tán với độ mềm dẻo cao. Trong môi trường web có hai chiến lược cơ bản cho việc chia các danh sách ngược thành một tập các nút khác nhau để có thể lưu trữ phân tán tại nhiều nơi khác nhau.

Kiểu thứ nhất là file liên kết ngược cục bộ (local inverted file - IFL). Trong tổ chức kiểu IFL, tập các trang web trong kho trang web được chia thành một số các tập con và mỗi nút sẽ lưu trữ các danh sách ngược của một trong tập con nói trên. Khi có một yêu cầu tìm kiếm, bộ truy vấn sẽ truyền yêu cầu đó đi tất cả các nút, và sau đó mỗi nút sẽ trả lại một danh sách các trang có chứa các từ khóa đang tìm kiếm.

Kiểu thứ hai là file liên kết ngược toàn cục (Global inverted file - GFL). Trong tổ chức kiểu GFL, chỉ mục ngược được chia theo các từ khóa, vì vậy mỗi một dịch vụ truy vấn (query server) lưu trữ danh sách ngược của một tập con các từ khóa trong bộ dữ liệu. Ví dụ, trong hệ thống với hai dịch vụ truy vấn là A và B, thì A sẽ lưu trữ danh sách

ngược của tất cả các từ khóa có ký tự đầu tiên từ a đến q, trong khi đó B lưu trữ danh sách ngược của tất cả các từ khóa còn lại (có ký tự đầu tiên từ r đến z). Vì vậy khi bộ truy vấn muốn tìm các trang có chứa từ "process" thì sẽ chỉ yêu cầu tới A.

- ***Biểu diễn chỉ mục cấu trúc***

Trong quá trình tạo chỉ mục, bộ tạo chỉ mục sẽ phân tích tất cả các siêu liên kết có trong tất cả các trang web và lưu trữ mọi thông tin quan trọng về các siêu liên kết đó trong các file neo (anchor file). Các file này chứa đầy đủ các thông tin để xác định mỗi siêu liên kết xuất phát từ đâu và đi đến đâu cũng như cụm từ được dùng để đặt cho siêu liên kết. Một chương trình con của bộ tạo chỉ mục có chức năng chuyển địa chỉ quan hệ (relative URL) giữa các siêu liên kết thành địa chỉ tuyệt đối (absolute URL), và đưa địa chỉ đó vào phần định danh trang web (docID), đồng thời sinh ra cơ sở dữ liệu các siêu liên kết, trong đó có chứa từng đôi định danh trang web tương ứng với mỗi siêu liên kết. Cơ sở dữ liệu siêu liên kết được sử dụng để tính hạng cho các tài liệu.

- ***Xếp hạng và phân tích các liên kết***

Vấn đề tiếp theo là sắp xếp các kết quả tìm kiếm. Tập hợp dữ liệu trang web trên Internet là khổng lồ và luôn biến đổi, số lượng từ khoá người dùng đưa vào trong một câu hỏi lại rất ít (khoảng một đến vài từ khóa), do đó kết quả tìm kiếm được là rất lớn và hầu hết các trang web kết quả tuy chứa các từ khóa tìm kiếm nhưng chất lượng thông tin trong các trang đó lại quá nghèo nàn hoặc không có liên quan gì tới vấn đề người dùng quan tâm. Hơn nữa, rất nhiều trang web không có đủ các thông tin tự miêu tả, vì vậy với các kỹ thuật tìm kiếm truyền thống chỉ dựa vào việc xem xét nội dung của các trang web sẽ dẫn tới kết quả công việc là không chính xác.

Đặc điểm của dữ liệu web là nửa cấu trúc vì ngoài nội dung các trang web còn chứa các siêu liên kết để liên kết giữa các trang với nhau (thông thường người ta tạo siêu liên kết khi có sự liên quan về nội dung). Cấu trúc liên kết các trang web chứa các thông tin quan trọng có thể giúp cho việc lọc hoặc tính hạng của trang web. Nhìn chung một liên kết từ A sang B có thể coi như là một sự tiến cử đến trang B của tác giả trang A. Hơn nữa các trang web thường được viết bằng ngôn ngữ HTML, là ngôn ngữ nửa

cấu trúc, nó có chứa các thẻ có chức năng giúp cho người viết trang web muốn nhấn mạnh các vấn đề cho người đọc (ví dụ các thẻ tiêu đề, thân nội dung, các thẻ FONT hay các thẻ heading...). Chính vì lý do đó, một số thuật toán mới đã được đề xuất nhằm khai thác cấu trúc liên kết này.

Trong giai đoạn đánh chỉ mục, bộ tạo chỉ mục cũng tạo ra các chỉ mục cấu trúc, và trong giai đoạn tính hạng của trang web bộ xếp hạng có thể sử dụng các thông tin này để sắp xếp thứ tự các trang kết quả thứ tự ưu tiên các trang có nội dung gần với các từ khoá tìm kiếm nhất để giúp cho người sử dụng khai thác thông tin hiệu quả hơn.

Việc tính toán thứ hạng các trang web dựa vào một số quy tắc sau đây:

1) Dựa vào vị trí xuất hiện của từ khoá trong trang web: Từ khoá tìm kiếm xuất hiện tại tiêu đề trang hay tại các phần miêu tả (discription) ... thì chắc chắn sẽ quan trọng hơn khi nó xuất hiện trong thân của trang web;

2) Dựa vào vị trí tương đối giữa các từ khoá tìm kiếm trong trang web, các trang có chứa các từ khoá trong cụm từ tìm kiếm đứng liền nhau thì sẽ được tính hạng cao hơn các trang mà các từ trong cụm từ tìm kiếm đứng tách nhau. Ví dụ, khi người sử dụng đưa vào cụm từ khoá tìm kiếm là "*công nghệ thông tin*" thì trang web có chứa nội dung "...khoa *công nghệ thông tin*, Đại học Quốc gia Hà Nội..." sẽ được tính hạng cao hơn trang web có chứa nội dung "...*thông tin* về khoa *công nghệ*...";

3) Dựa vào thuộc tính của từ khoá trong trang web, chẳng hạn chúng được đặt trong các thẻ H1, H2,...., H5;

4) Dựa vào giá trị hạng trang.

Lý do đặt ra các quy tắc từ 1 đến 3 cho việc sắp xếp các trang là rõ ràng, còn quy tắc dựa vào giá trị hạng trang được trình bày như dưới đây.

- ***Tính hạng trang web***

Tính hạng trang web là một kỹ thuật tính toán độ quan trọng của các trang web dựa trên cấu trúc của các mối liên kết. Kỹ thuật này dựa vào quan điểm là các trang web quan trọng thì sẽ được nhiều trang web khác liên kết đến. Có nghĩa là trang web A

có hạng lớn hơn (quan trọng hơn) trang web B nếu số các trang web liên kết đến trang A nhiều hơn số các trang web liên kết đến trang B.

Hạng trang web được tính toán như sau:

Cho u là một trang web, gọi R_u là hạng của u: $R_u = PageRank(u)$

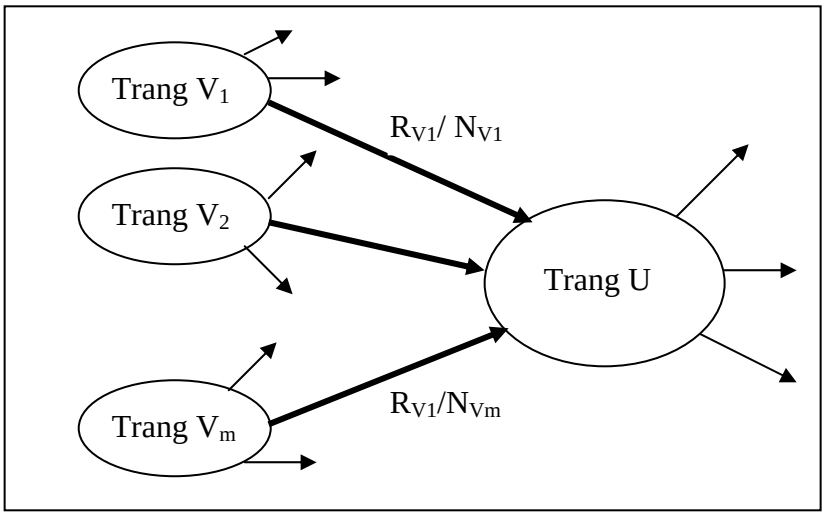
Gọi N_u là số các siêu liên kết ra từ trang u (số lượng siêu liên kết từ u đến các trang web khác)

Gọi v1, v2, ..., vm là các trang web có siêu liên kết đến trang u

Ta có $R_u = d\{ R_{v1} / N_{v1} + ... + R_{vm} / N_{vm} \} + (1-d)$, trong đó d là hệ số hãm.

Quá trình tính toán sẽ được lặp đi lặp lại cho đến khi hội tụ. Việc tính hạng trang web không tốn nhiều thời gian. Máy tìm kiếm Google chỉ cần sử dụng một máy trạm cỡ trung bình để tính toán trong vài giờ khi thực hiện tính hạng cho khoảng 26 triệu trang web.

Chú ý rằng hạng trang web là đại lượng đại diện cho sự phân bố xác suất của các trang web trong một tập các trang web xác định, do đó tổng các hạng của tất cả các trang web trong kho trang web có giá trị bằng 1.



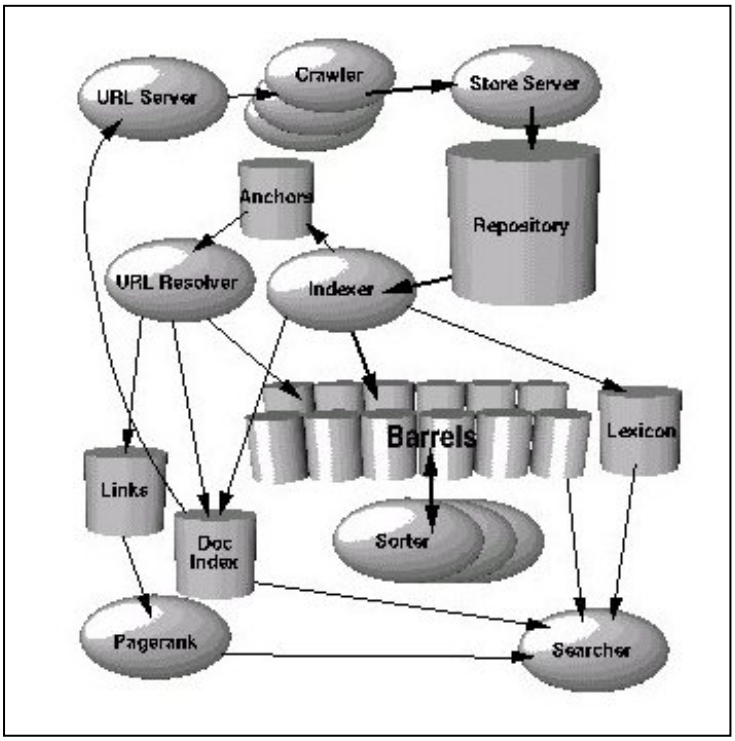
Hình 2.2. Tính hạng trang web

❖ Biểu diễn dữ liệu trong máy tìm kiếm Google

Phần này trình bày chi tiết cách biểu diễn dữ liệu trong máy tìm kiếm Google, một máy tìm kiếm đang được đánh giá cao hiện nay và được sử dụng rất phổ biến trên thế giới. Tất cả các chương trình trong máy tìm kiếm Google đều được viết bằng ngôn ngữ C và C^{++} để có thể chạy được trên cả hai hệ điều hành Linux và Solaris.

- ***Hoạt động của máy tìm kiếm Google***

Trong Google, chức năng tìm duyệt các trang web được thực hiện bởi một vài bộ tìm duyệt phân tán. Bộ dịch vụ URL (**URLserver**) gửi danh sách các địa chỉ URL đã định sẵn lộ trình cho các bộ tìm duyệt. Bộ tìm duyệt đi theo các siêu liên kết và các địa chỉ đó để tải các trang web về rồi gửi tới dịch vụ lưu trữ (storeserver). Sau đó, dịch vụ lưu trữ nén và lưu trữ các trang web đó trong kho trang web. Tất cả các trang web đều được gán cho một mã định danh duy nhất (**docID**). Mã định danh dạng này sẽ được ấn định cho mỗi trang web khi một địa chỉ URL mới (chỉ đến trang đó) được phân tích ra từ các trang web đã có. Chức năng tạo chỉ mục được thực hiện bởi **bộ tạo chỉ mục (indexer)** và **bộ sắp xếp (sorter)**. Bộ tạo chỉ mục thực hiện một số chức năng như đọc các trang trong kho trang web, giải nén trang web và phân tích chúng. Mỗi một trang web được chuyển thành một tập các từ khóa xuất hiện trong trang web đó, tập này được gọi là ***hit***. Các hit này ghi lại các từ khóa, vị trí của các từ khóa trong tài liệu, kích thước font chữ và kiểu chữ (chữ hoa hay chữ thường). Bộ tạo chỉ mục sẽ phân tán các hit này vào một tập các **thùng chứa (barrel)**, và tạo nên bảng **chỉ mục chuyển tiếp (forward index)** đã được sắp xếp cục bộ. Sau đó, bộ chỉ mục phân tích ra tất cả các siêu liên kết trong tất cả các trang web rồi lưu trữ các thông tin quan trọng về chúng trong một **file neo**. đặc điểm về File neo đã được nói ở trên.

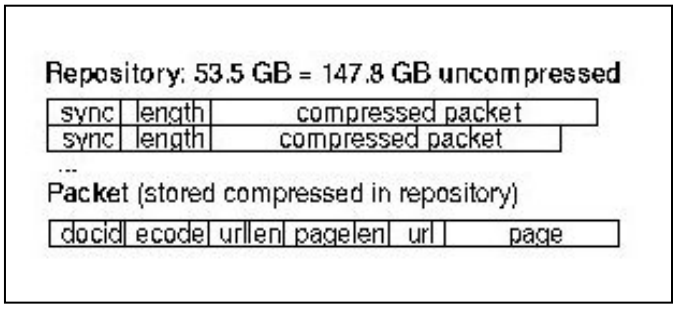


Hình 2.3.Mô hình kiến trúc của máy tìm kiếm Google

Bộ phân tích URL (**URLresolver**) thực hiện chức năng chuyển địa chỉ URL quan hệ thành các địa chỉ URL tuyệt đối rồi lần lượt đưa vào các docID. Nó cũng đưa các đoạn text gắn với siêu liên kết vào trong chỉ mục chuyển tiếp, kết hợp với docID mà siêu liên kết đó chỉ tới. Bộ phân tích URL cũng sinh ra cơ sở dữ liệu các liên kết ghép đôi với docID được sử dụng để tính hạng trang web như đã biết.

Bộ sắp xếp sử dụng các thùng chứa chứa các hit đã được sắp xếp theo các docID, sắp xếp chúng theo wordID để sinh ra bảng **chỉ mục liên kết ngược**. Việc này được thực hiện ngay tại chỗ, do đó đòi hỏi một không gian bộ nhớ nhất định. Bộ sắp xếp cũng tạo ra một danh sách các wordID và các offset vào trong bảng chỉ mục liên kết ngược. Sử dụng các danh sách này cùng với các **từ vựng (lexicon)** do bộ tạo chỉ mục tạo ra, một chương trình có tên là **DumLexicon** sinh ra một bộ phân tích từ vựng mới phục vụ cho **bộ tìm kiếm**. Bộ tìm kiếm được thực hiện bởi **webserver** và sử dụng **bộ từ vựng** (được xây dựng bởi chương trình DumLexicon) cùng với bảng chỉ mục liên kết ngược và giá trị hạng trang web để trả lời các yêu cầu tìm kiếm.

- **Cấu trúc dữ liệu của Google**



Hình 2.4. Cấu trúc dữ liệu của kho trang web

Kho trang web lưu trữ toàn bộ nội dung của tất cả các trang web, mỗi trang được nén bằng phương pháp zlip. Việc chọn một kỹ thuật nén thường được cân nhắc giữa tốc độ và tỷ lệ nén. Tỷ lệ nén của zlip là 3/1 (nhỏ hơn so với tỷ lệ 4/1 của phương pháp nén bzip) nhưng tốc độ của zlip nén lại nhanh đáng kể. Lần lượt các trang web được lưu trữ vào kho và bổ sung vào phần đầu các thông tin về docID, độ dài, và địa chỉ URL. Kho trang web không đòi hỏi một cấu trúc dữ liệu nào khác để truy nhập nó, hơn nữa từ repository cho phép xây dựng lại tất cả các cấu trúc dữ liệu khác.

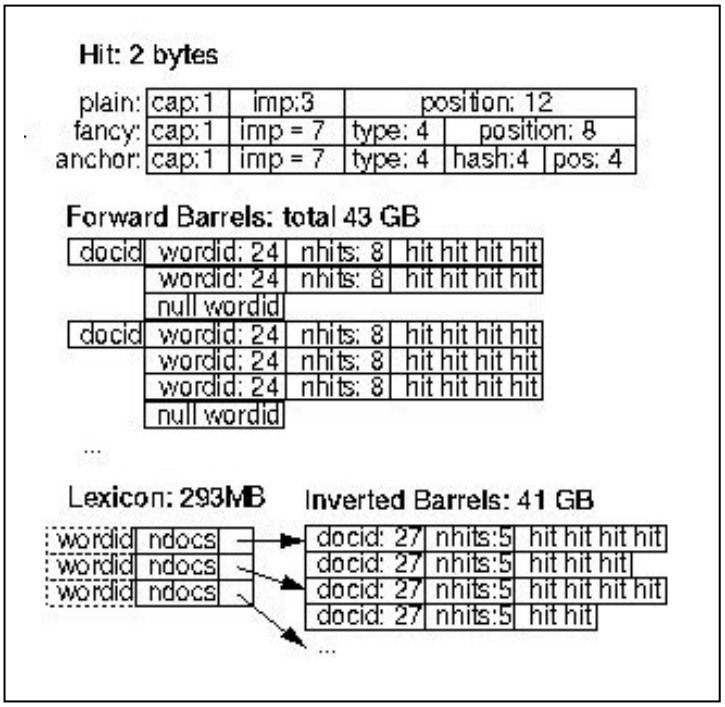
Chỉ mục tài liệu lưu giữ các thông tin về mỗi tài liệu. Nó được cố định với kiểu chỉ mục ISAM (mô hình truy nhập chỉ số kế tiếp: Index Sequel Access Model), và được sắp

xếp theo giá trị của docID. Các thông tin được lưu trữ trong chỉ mục tài liệu bao gồm tình trạng hiện tại của tài liệu, con trỏ chỉ tới vị trí trong kho trang web, giá trị tổng kiểm tra và một số giá trị thống kê khác. Nếu tài liệu đã được bộ tìm duyệt xử lý thì nó còn chứa con trỏ để trỏ tới một file kích thước động (được gọi là docinfo) chứa các địa chỉ URL và các tiêu đề. Ngoài ra, còn có các con trỏ trỏ tới danh sách các URLchỉ chứa các địa chỉ URL. Nhu cầu cần có một cấu trúc dữ liệu hợp lý và khả năng tìm được các bản ghi trong một bước tìm kiếm đĩa trong quá trình tìm kiếm đã đưa đến việc thiết kế bổ sung này.

Hơn nữa, sử dụng một file để chuyển các URL thành các docID, được gọi là file tổng kiểm tra. Đó là một danh sách các tổng kiểm tra URL (URL checksum) tương ứng với các docID, và được sắp xếp theo giá trị tổng kiểm tra. Nhằm mục đích tìm ra một docID của một URL nào đó, thì tổng kiểm tra của URL đó được tính toán và việc tìm kiếm nhị phân trên file tổng kiểm tra để tìm ra docID tương ứng với URL đó. URL cũng có thể được chuyển vào docID theo từng mẻ bằng cách trộn với file này. Đây chính là kỹ thuật mà bộ phân tích URL sử dụng để chuyển URL vào docID.

Bộ từ vựng của Google có một vài định dạng khác nhau, bao gồm 14 triệu từ khóa (một vài từ khóa rất hiếm thì không được đưa vào) và được lưu trữ trong 2 phần, phần một là một danh sách các từ (được móc nối vào nhau nhưng tách riêng nhau bởi giá trị null) và phần hai là một bảng băm các con trỏ. Do cần đáp ứng một chức năng khác nên danh sách các từ được bổ sung một số các thông tin bổ trợ khác.

Danh sách hit: một danh sách hit tương ứng với một danh sách các xuất hiện của một từ khoá trong một tài liệu, bao gồm vị trí, font chữ, và thông tin về kiểu chữ hoa hay chữ thường. Danh sách hit này được lập ra để sử dụng trong cả chỉ mục liên kết ngược và chỉ mục chuyển tiếp, vì vậy cách biểu diễn nó một cách hiệu quả là rất quan trọng. Phương pháp mã hoá cho danh sách hit là mã hóa compact (compact encoding) vì đòi hỏi ít không gian nhớ hơn phương pháp mã hóa giản đơn (simple encoding) và sử dụng ít bit thao tác hơn mã hóa huffman. Chi tiết của hit được chỉ ra trong hình 2.5. Mã hóa compact sử dụng 2 byte cho tất cả các hit. Có hai kiểu hit là fancy hit và plain hit.



Hình 2.5.Cấu trúc của hit, chỉ mục chuyển tiếp, từ vựng và chỉ mục ngược

Fancy hit bao gồm các hit xuất hiện trong một URL, tiêu đề, thẻ neo hoặc các thẻ meta. Còn plain hit thì bao gồm tất cả các thứ còn lại. Một plain hit bao gồm 1 bit lưu giữ thông tin về chữ hoa hay chữ thường, kích thước font, và 12 bit lưu giữ vị trí của các từ trong tài liệu (tất cả các vị trí cao hơn 4095 thì đều được gán nhãn là 4096). Kích thước font chữ được biểu diễn liên quan đến phần còn lại của tài liệu sử dụng 3 bit (chỉ có 7 giá trị thực sự được sử dụng, bởi vì giá trị 111 đã được sử dụng cho cờ báo hiệu đó là một fancy hit). Một fancy hit bao gồm một bit quy định chữ thường, chữ hoa, kích

thước font không cần biểu diễn, nên giá trị các bit trong danh sách được đặt đặt giá trị bằng 7 để chỉ ra nó là một fancy hit, 4 bit để mã hoá kiểu của fancy hit, và 8 bit cho vị trí. Với một hit neo sử dụng 8 bit cho vị trí, 8 bit này được chia thành 2 phần, 4 bit cho vị trí trong thẻ neo và 4 bit cho hàm băm của docID mà thẻ neo xuất hiện. Việc này gây ra một vài hạn chế khi tìm kiếm theo cụm từ khi mà không có nhiều thẻ neo cho một từ (nghĩa là rất ít khi các liên kết được đặt vào một từ). Độ dài của danh sách hit được lưu trữ trước khi lưu trữ chính nó. Để tiết kiệm không gian thì độ dài của danh sách hit được liên kết với wordID trong chỉ mục chuyển tiếp và liên kết với docID trong bảng chỉ mục liên kết ngược, và được giới hạn là 8 và 5 bit tương ứng với mỗi loại. Nếu như chiều dài dài hơn số bit thì một mã thoát sẽ được sử dụng trong các bit đó và 2 byte tiếp theo chứa chiều dài thực sự của tài liệu.

Bộ chỉ mục chuyển tiếp: chỉ mục chuyển tiếp thực sự đã được sắp xếp cục bộ. Nó được sắp xếp trong số các thùng chứa. Mỗi thùng chứa một tập các wordID. Nếu tài liệu bao gồm các từ rơi vào một thùng chứa nào đó thì docID của nó cũng sẽ được ghi lại trong thùng chứa đó, và theo đó là một danh sách các wordID cùng với danh sách các hit tương ứng với các từ đó. Lược đồ này tuy đòi hỏi bổ sung một chút không gian lưu trữ vì đã nhân đôi các docID (tuy nhiên chỉ là rất nhỏ nếu số lượng các thùng là hợp lý) tuy nhiiên lại cho phép tiết kiệm đáng kể được thời gian cũng như độ phức tạp mã hoá trong giai đoạn tạo chỉ mục cuối cùng do bộ sắp xếp thực hiện.

Bộ chỉ mục liên kết ngược: chỉ mục liên kết ngược bao gồm các thùng chứa giống như chỉ mục chuyển tiếp, ngoại trừ việc chúng được xử lý bởi bộ sắp xếp. Với tất cả các wordID hợp lệ thì bộ từ vựng chứa các con trỏ chỉ đến các thùng chứa mà wordID đang nằm trong đó. Chúng chỉ đến một doclist (danh sách tài liệu) của docID cùng với các danh sách hit tương ứng của chúng. Doclist này biểu diễn cho tất cả các xuất hiện của từ khóa đó trong tất cả các tài liệu.

Một điều quan trọng là cách mà docID xuất hiện trong các doclist. Giải pháp đơn giản là lưu trữ chúng theo thứ tự sắp xếp của docID. Điều này cho phép trộn nhanh các doclist khác nhau cho các yêu cầu tìm kiếm gồm nhiều từ khóa. Một cách khác là lưu

trữ chúng theo sắp xếp hạng của sự xuất hiện các từ khóa trong mỗi tài liệu. Mỗi cách nói trên đều có các ưu nhược điểm riêng. Google đã chọn cách thoả hiệp giữa hai lựa chọn này bằng cách giữ cả hai tập thùng ngược (inverted barrel), một tập cho danh sách các hit (bao gồm các tiêu đề hay các thẻ neo) và tập kia cho tất cả các danh sách hit. Với cách này, cho phép kiểm tra trong tập các thùng nhỏ trước và nếu không thấy phù hợp thì lại tiếp tục tìm ở thùng lớn hơn.

2.2 Phương pháp biểu diễn trang web theo mô hình vector

Biểu diễn trang web theo mô hình vector (Seán Slattery [11]) phát triển từ phương pháp biểu diễn tài liệu fulltext theo mô hình vector. Một số đề xuất cải tiến của chúng tôi về cơ bản cũng dựa trên việc biểu diễn trang web theo mô hình vector. Vì vậy, trước tiên chúng ta xem xét những nội dung cơ bản nhất của phương pháp biểu diễn theo mô hình vector.

2.2.1 Phương pháp biểu diễn vector

Phương pháp biểu diễn dữ liệu bằng mô hình vector (Space Vector Model) là một phương pháp phổ biến nhất hiện nay [3,8-13]. Theo cách này, mỗi văn bản được biểu diễn như một vector có các thành phần là thể hiện từ khoá tương ứng có mặt hoặc không có mặt trong văn bản đó. Mỗi từ khoá lại có một trọng số biểu diễn về mức độ quan trọng của nó trong văn bản. Quá trình gán các giá trị đó được gọi là quá trình đánh chỉ số (indexing). Hiện nay có nhiều phương pháp đánh chỉ số như TF, IDF, TF*IDF, LSI... trong đó chủ yếu dựa vào tần số xuất hiện của các từ hoặc mối quan hệ giữa sự xuất hiện của các từ trong văn bản. Như vậy thì số chiều của không gian vector là lực lượng của tập các từ khóa.

Như đã biết, định nghĩa chung nhất (đối với tiếng Anh cũng như các ngôn ngữ sử dụng bảng chữ cái latin) thì từ là một chuỗi các ký tự và số viết liền nhau, ngoại trừ các khoảng trống (các dấu tab hoặc các ký tự xuống dòng) hay các dấu câu như dấu chấm, dấu phẩy... Thông thường khi tạo vector cho các văn bản thì tất cả các chữ hoa trong văn bản đều được chuyển hết thành chữ thường nên quy ước chỉ xem xét chữ thường.

Sau đây chúng ta cùng xét cách biểu diễn tài liệu bằng vector dưới dạng các từ cùng với hàm f biểu diễn tần số xuất hiện của các từ trong tài liệu đó. Cách biểu diễn này còn gọi là cách biểu diễn theo túi các từ (bag of words). Cách biểu diễn này được sử dụng rộng rãi trong các máy phân lớp Text bao gồm Bayes tự nhiên (Naive Bayes), Máy vector trợ giúp (Support Vector Machine - SVM), k- người láng giềng gần nhất (k Nearest Neighbour - kNN), Mạng nơron (Neural Net) ... Phương pháp này biểu diễn mỗi tài liệu bằng một tập duy nhất các từ khóa xuất hiện trong chính nó cùng với tần số xuất hiện của mỗi từ.

Ví dụ, giả sử có một tài liệu 1 với nội dung như sau:

> *The plentiful content of the World-Wide Web is useful to millions. Some simply browse the web through entry points such as Yahoo!. But many information seekers use a search engine to begin their web activity.*

và tài liệu 2 có nội dung như sau:

> *Many of search engines use well-know information retrieval algorithms and techniques.*

Lúc đó các vector biểu diễn hai tài liệu này như sau:

Từ	***Vector cho văn bản 1***	***Vector cho văn bản 2***
a	*1*	*0*
activity	*1*	*0*
algorithms	*0*	*0*
and	*0*	*1*
as	*1*	*0*
begin	*1*	*0*
browse	*1*	*0*
but	*1*	*0*
content	*1*	*0*
engine	*1*	*0*
engines	*0*	*1*
entry	*1*	*0*
information	*1*	*1*
is	*1*	*0*
many	*1*	*1*

millions	*1*	*0*
of	*1*	*1*
plentiful	*1*	*0*
points	*1*	*0*
retrieval	*0*	*1*
search	*1*	*1*
seekers	*1*	*0*
simply	*1*	*0*
some	*1*	*0*
such	*1*	*0*
techniques	*0*	*1*
the	*3*	*0*
their	*1*	*0*
through	*1*	*0*
to	*2*	*0*
use	*1*	*1*
useful	*1*	*0*
web	*3*	*0*
well-know	*0*	*1*
wide-World	*1*	*0*
yahoo	*1*	*0*

Nhìn vào bảng các vector biểu diễn, có thể biết từ "activity" xuất hiện một lần trong văn bản 1 và không xuất hiện lần nào trong văn bản 2. Mặt khác, dễ dàng thấy rằng cách biểu diễn tài liệu này đã bỏ qua các thông tin về vị trí của mỗi từ và các thông tin về trật tự từ trong tài liệu. Vì vậy mà cách biểu diễn này không thể cho biết là trong tài liệu 1 có cụm từ "search engine" đi liền nhau hay không mà chỉ có thể cho biết là trong tài liệu có chứa từ "search" và từ "engine"

Hơn nữa, dễ dàng nhận thấy là chiều của vector theo cách biểu diễn này là rất lớn, bởi vì chiều của nó được xác định bằng số lượng các từ khác nhau trong tập hợp văn bản. Ví dụ số lượng các từ có thể từ 10^3 đến 10^5 trong một tập văn bản nhỏ, còn trong tập văn bản lớn thì có thể số lượng sẽ nhiều hơn, đặc biệt là trong môi trường web. Vì vậy đã có một số phương pháp giảm bớt số chiều của vector được áp dụng. Chẳng hạn, một phương pháp rất đơn giản và hiệu quả là loại bỏ các từ dừng. Từ dừng (stop word) là từ được dùng để biểu diễn cấu trúc câu chứ không biểu đạt nội dung của văn bản, ví

dụ như các từ nối, các giới từ... Những từ như vậy xuất hiện rất nhiều trong văn bản nhưng lại không liên quan đến chủ đề và nội dung của văn bản. Do đó việc loại bỏ các từ này đi cho phép giảm được số chiều của vector biểu diễn mà lại không làm ảnh hưởng đến hiệu quả tìm kiếm. Ví dụ về các từ dừng trong tiếng Anh và tiếng Việt trong bảng sau:

Tiếng Việt	Tiếng Anh
Và	a
Hoặc	the
Cũng	do
	about

2.2.2 Phương pháp biểu diễn trang web theo mô hình vector

Phần này trình bày chi tiết cách thức biểu diễn trang web được Seán Slattery trình bày trong [11].

Xuất phát từ việc sử dụng phương pháp biểu diễn trang web bằng vector, cùng với quan điểm là sử dụng các thông tin về liên kết nhằm tăng độ chính xác tìm kiếm cũng như phân lớp các trang web nên cần thiết phải đưa thêm các thông tin về các trang web láng giềng vào vector biểu diễn của trang web đang xét (trang láng giềng của trang web đang xét là các trang web có liên kết đến hoặc đi của trang web) .

Để hiểu rõ về cách biểu diễn này xem xét một ví dụ đơn giản: cho 4 trang web chứa các từ tương ứng và các liên kết giữa các trang như hình 2.6. Mỗi hình chữ nhật biểu diễn cho một trang web, với nội dung là các ký tự nằm trong đó. Các liên kết được biểu diễn bởi các mũi tên, với chiều mũi tên là chiều chỉ tới các trang được liên kết đến. Và giả sử trang A là đang được quan tâm. Tồn tại bốn cách biểu diễn trang web như sau:

- ***Cách biểu diễn thứ nhất***

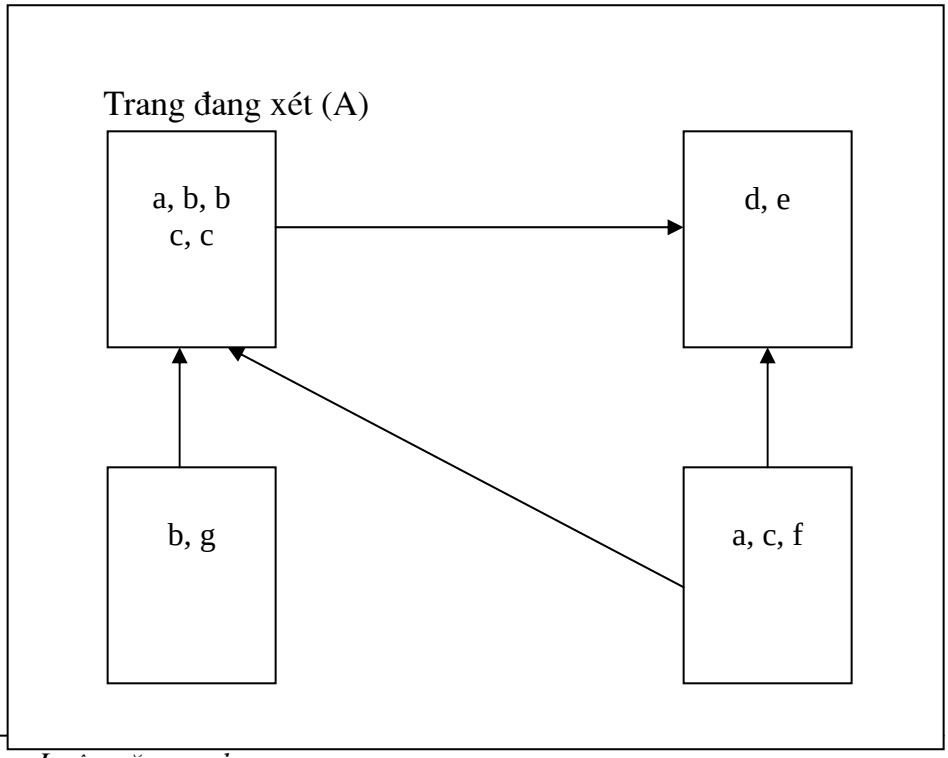
Cách này không quan tâm đến bất cứ một liên kết nào cũng như bất cứ một trang láng giềng nào mà chỉ biểu diễn trang A bằng vector các từ khóa trong nó. Cách biểu diễn này giống như cách biểu diễn túi các từ khóa. Theo cách này, mỗi trang web được biểu diễn bằng một danh sách các từ khóa trong nó. Trong danh sách này, mỗi từ khóa trong một trang web được lưu trữ cùng tần số xuất hiện nó ở trong trang web. Như vậy là cách này bỏ qua tất cả các thông tin về vị trí của từ khóa trong trang, thứ tự của các từ trong trang cũng như các thông tin về các siêu liên kết. Kết quả, trang A được biểu diễn bởi vector sau:

a	b	c	d	e	f	g
1	2	2	0	0	0	0

Trong nhiều trường hợp khi mà các tài liệu đã liên kết độc lập với các nhãn của các lớp thì cách biểu diễn này là lựa chọn tốt nhất. Tuy nhiên trong một số trường hợp khác thì cách biểu diễn này không cung cấp cho máy học cơ hội khai thác được tính cân đối trong các tài liệu liên kết.

- ***Cách biểu diễn thứ hai***

Cách đơn giản nhất để sử dụng các thông tin về liên kết của trang web là móc nối nó với tất cả các trang láng giềng để tạo ra một siêu trang (super-document). Theo cách này,



Hình 2.6. Tập gồm 4 trang web liên kết

vector biểu diễn bao gồm các từ xuất hiện trong A cùng với tất cả các từ xuất hiện trong các trang láng giềng của A cùng với tần số xuất hiện của các từ. Cách này cũng bỏ qua các thông tin về vị trí của các từ trong trang và thứ tự của chúng. Với ví dụ trên, nhận được vector biểu diễn sau cho A:

a	b	c	d	e	f	g
2	3	3	1	1	1	1

Mối nguy hiểm của cách biểu diễn này là làm loãng đi nội dung của trang A, và do đó có thể dẫn đến việc tạo ra thêm nhiễu cho việc phân lớp. Cách biểu diễn này là sự lựa chọn rất tốt trong trường hợp cần biểu diễn một tập các trang web có nội dung về cùng một chủ đề.

- ***Cách biểu diễn thứ ba***

Để biểu diễn được kỹ lưỡng hơn, có thể suy nghĩ về một cách tiếp cận là dùng một vector có cấu trúc để biểu diễn các trang web. Một vector có cấu trúc được chia một cách logic thành hai phần hoặc nhiều hơn. Mỗi phần được sử dụng để biểu diễn một tập các trang (láng giềng). Độ dài của một vector thì cố định nhưng mỗi phần của vector thì chỉ dùng để biểu diễn các từ xuất hiện trong một tập nào đó. Ví dụ, vector biểu diễn được chi thành hai phần, phần một được dùng để biểu diễn các từ xuất hiện trong trang A, còn phần thứ hai sẽ được dùng để biểu diễn các từ xuất hiện trong các trang láng giềng của A. Theo cách này, nhận được vector biểu diễn cho A như sau

	phần 1								phần 2						
a	b	c	d	e	f	g		a	b	c	d	e	f	g
1	2	2	0	0	0	0		1	1	1	1	1	1	1

Cách biểu diễn này tránh được khả năng các trang láng giềng có thể làm loãng nội dung của trang A. Nếu như thông tin về các trang láng giềng hữu ích cho việc phân lớp trang A thì máy học vẫn có thể truy nhập đến toàn bộ nội dung của chúng để học.

- ***Cách biểu diễn thứ tư***

Chúng ta xây dựng một vector cấu trúc như sau:

1. Xác định một số d được coi là bậc cao nhất của các trang trong tập
2. Xây dựng một vector cấu trúc với d+1 phần như sau
   - ❖ Phần đầu tiên biểu diễn chính tài liệu A
   - ❖ Các phần tiếp theo từ phần thứ 2 đến phần d+1 biểu diễn các tài liệu láng giềng của A, mỗi tài liệu được biểu diễn trong một phần.

Như vậy, có thể thấy rằng đây là một vector chứa rất nhiều thông tin tiềm năng, tuy nhiên còn một vấn đề cần giải quyết trong cách biểu diễn này, đó là chuẩn hóa cách biểu diễn cho tài liệu theo lược đồ này, nếu không việc biểu diễn là không xác định. Chẳng hạn, với 4 trang web trong ví dụ đã cho thì có ít nhất hai khả năng biểu diễn bằng cách thay đổi thứ tự trang láng giềng trong các phần biểu diễn.

Phần 1							Phần 2							Phần 3							Phần 4						
a	b	c	d	e	f	g	a	b	c	d	e	f	g	a	b	c	d	e	f	g	a	b	c	d	e	f	g
1	2	2	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1
1	2	3	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0

Trong trường hợp biểu diễn chưa được chuẩn hóa sẽ nảy sinh khó khăn là máy học trong quá trình xây dựng giả thuyết.

Seán Slattery đã làm thực nghiệm để đối sánh cách biểu diễn mới với cách biểu diễn truyền thống. Tập dữ liệu huấn luyện và kiểm tra là tập các website của các bộ môn Khoa học máy tính của một số các trường đại học: trường đại học Cornell (Cornell University), trường đại học Texas (Texas University), trường đại học Washington (University of Washington) và trường đại học Wisconsin (University of Wisconsin). Tổng số các trang web được thu thập là 4,168 trang và được phân loại bằng tay theo các nhóm sau:

Student: các trang chủ về sinh viên

Course: các trang chủ về các khoá học

Faculty: các trang chủ cho thành viên của các khoa

Project: các trang chủ cho các dự án nghiên cứu

Staff: các trang chủ cho các nhân viên

Department: các trang chủ của các bộ môn

Other: các trang không thuộc 6 nhóm trên

Số lượng các trang web thuộc mỗi loại được liệt kê trong bảng sau

	Cornell	**Texas**	**Washington**	**Wisconsin**	**Tổng**
Student	128	148	126	156	558
Course	44	38	76	85	243
Faculty	34	46	31	42	153
Project	20	20	21	25	86
Staff	21	3	10	12	46
Department	1	1	1	1	4
Other	620	570	942	946	3078
Tổng	868	826	1207	1267	4168

Số lượng siêu liên kết giữa các trang web trong tập dữ liệu này là 10353 liên kết, tất cả đều là các liên kết nằm trong phạm vi của tập dữ liệu và không có liên kết ra các trang bên ngoài.

Hoạt động của hệ thống được đánh giá qua hai thông số là độ chính xác phân lớp và độ hồi tưởng tìm kiếm được tính theo các công thức dưới đây.

Độ chính xác (Precision) là tiêu chuẩn để đánh giá độ chính xác dự đoán của máy phân lớp và độ hồi tưởng (Recall) tiêu chuẩn để đánh giá độ chính xác của máy tìm kiếm trong việc tìm được một ví dụ dương được tính toán theo các công thức sau đây:

$$\mathrm{Pr}e = \frac{n_{ppc}}{n_{pp}} \qquad \mathrm{Re}c = \frac{n_{cpp}}{n_{pe}}$$

Trong đó,

Pre: độ chính xác phân lớp (Precision),

Rec: Độ hồi tưởng (Recall),

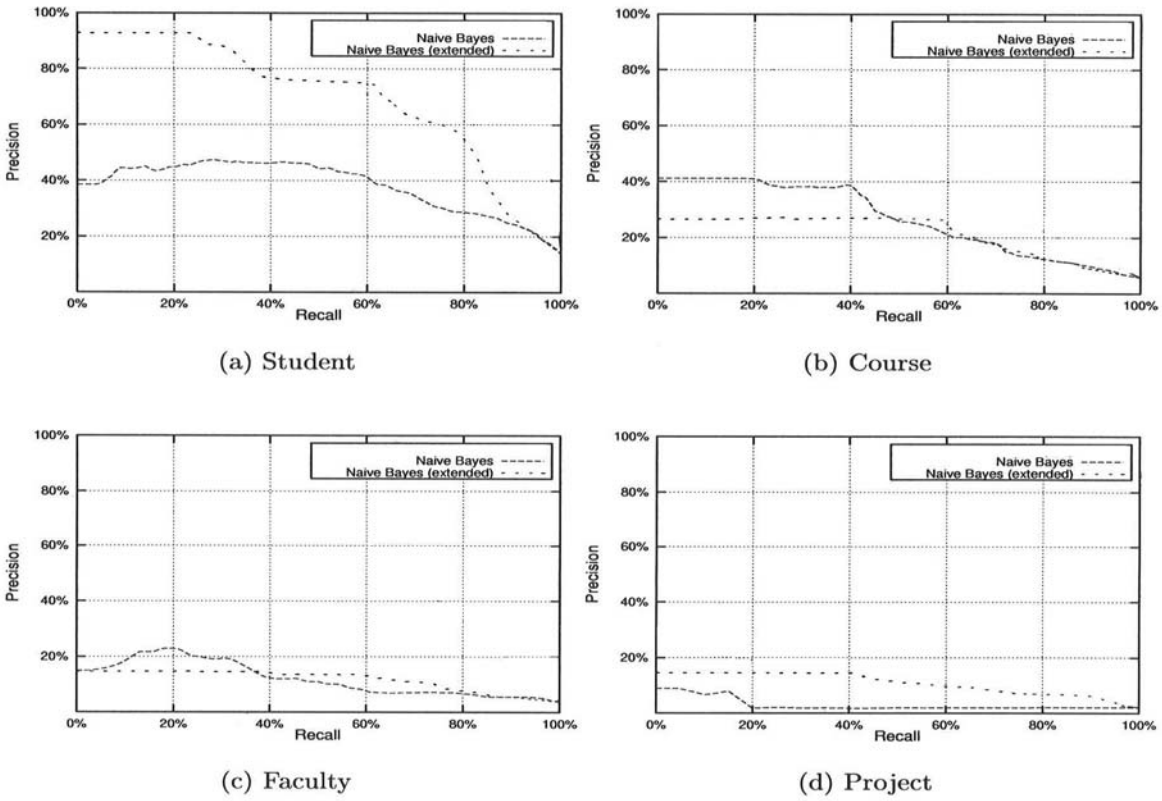
n_{ppc}: số lượng kết quả dương thực sự (correct positive predictions)

n_{pp}: số lượng kết quả dương (positive predictions)

n_{pe}: số lượng ví dụ dương (positive examples)

Seán Slattery sử dụng máy phân lớp Bayes tự nhiên để đối sánh cách biểu diễn thứ ba với cách biểu diễn thứ nhất. Kết quả thử nghiệm được biểu diễn trong hình 2.7, trong đó đường đậm nét tương ứng với cách biểu diễn thông thường (cách 1) còn đường rời nét tương ứng với cách biểu diễn vector kết hợp (cách 3).

Quan sát kết quả thử nghiệm trong hình 2.7, chúng ta thấy rằng trong hầu hết các trường hợp thì phương pháp biểu diễn mới (phương pháp biểu diễn vector có kết hợp các thông tin về các trang web láng giềng) cho chúng ta kết quả phân lớp tốt hơn so với phương pháp truyền thống (phương pháp vector với thông tin về tần số xuất hiện của các từ).



Hình 2.7. Kết quả thử nghiệm phân lớp

• Đề xuất cải tiến phương pháp biểu diễn có tính đến các trang web liên kết

Như nhận xét đánh giá theo kết quả thử nghiệm trên đây, phương pháp biểu diễn thứ ba cho kết quả tốt hơn phương pháp biểu diễn thứ nhất (là phương pháp biểu diễn không sử dụng thông tin liên kết với các trang web khác). Tuy nhiên, theo cách biểu diễn như vậy thì độ dài vector biểu diễn trang web lại tăng lên gấp đôi (do vector biểu diễn được tổ chức thành hai phần). Điều đó không chỉ đòi hỏi không gian lưu trữ dữ liệu phải tăng lên gấp đôi mà thời gian tính toán cho các bài toán phân lớp và tìm kiếm cũng tăng lên với hệ số như vậy.

Đề xuất cải tiến của chúng tôi hướng tới một phương án dung hòa cách biểu diễn thứ hai và hai cách biểu diễn cuối. Cách biểu diễn thứ hai coi sự xuất hiện các từ khóa trong các trang láng giềng có trọng số bằng sự xuất hiện các từ khóa của trang web

đang xem xét. Hai cách biểu diễn cuối cho sự phân biệt trọng số sự xuất hiện của từ khóa trong trang xem xét khác sự xuất hiện trong các trang láng giềng song độ dài vector biểu diễn lại tăng nhanh (gấp đôi trong cách thứ ba, và gấp nhiều lần theo cách thứ tư). Nội dung chủ yếu của biểu diễn mới là:

- Kích thước của vector biểu diễn không tăng: bằng số lượng các từ khóa trong hệ thống,

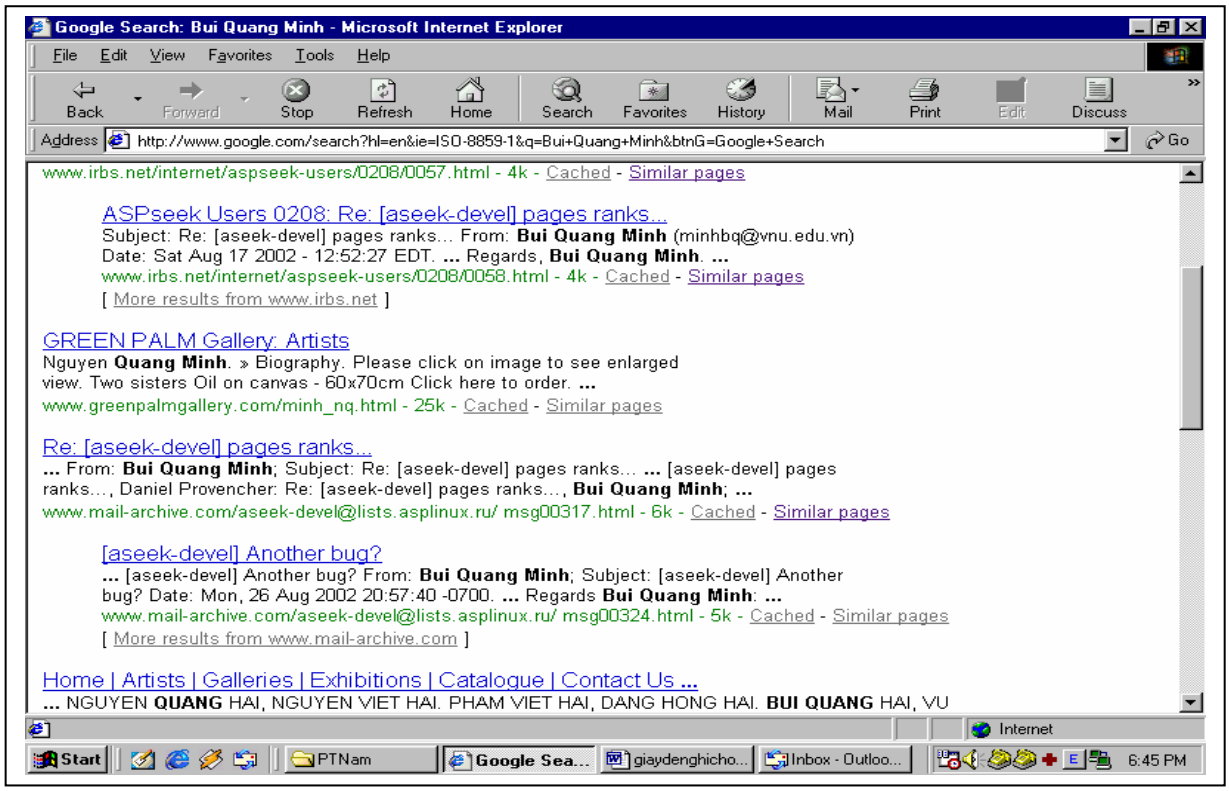
- Có sự phân biệt trọng số của sự xuất hiện các từ khóa trong trang web đang xét và các trang web láng giềng. Không những thế, có hệ số phân biệt giữa ba loại trang web láng giềng: có cả liên kết đi và tới, chỉ có liên kết đi, chỉ có liên kết tới. Chẳng hạn, hệ số cho trang web đang xét có hệ số 4, trang web có cả liên kết đi và tới có hệ số 2 và trang web láng giềng thuộc một trong hai dạng cuối có hệ số 1.

2.3 Đề xuất giải pháp biểu diễn vector trong máy tìm kiếm

Qua phân tích hoạt động của các máy tìm kiếm (mục 2.1) cho thấy câu hỏi người dùng đưa vào ở dạng rất đơn giản gồm một hoặc một số (không nhiều) các từ khóa. Vì vậy, máy tìm kiếm thường cho tập hợp gồm rất nhiều trang web kết quả chứa các từ khóa trong câu hỏi. Chính vì lẽ đó, máy tìm kiếm phải tìm cách hiển thị các trang web kết quả sao cho những trang có giá trị (hạng) càng cao càng được hiển thị trước. Để tính hạng của một trang, máy tìm kiếm đã sử dụng một công thức cho phép thể hiện mối quan hệ giữa các giá trị hạng của các trang web có liên kết lẫn nhau. Tuy nhiên, cách tính hạng hiển thị vẫn còn một số vấn đề cần giải quyết. Chẳng hạn, khi người dùng yêu cầu máy tìm kiếm Google tìm các trang web có chứa cụm từ "Bui Quang Minh" thì hệ thống cung cấp kết quả trong đó trang không chứa cụm từ "Bui Quang Minh" lại hiển thị trước một trang có chứa cụm từ đó (hình 2.8). Tuy vậy, do dạng câu hỏi người dùng là quá đơn giản cho nên vấn đề nghiên cứu đề xuất cách thức cho phép máy tìm kiếm tiếp nhận câu hỏi phức tạp hơn, biểu diễn đầy đủ hơn vấn đề người dùng cần hỏi và cho câu trả lời chính xác hơn hiện nay vẫn đang được tiếp tục nghiên cứu. Trong máy tìm kiếm Google cho cung cấp một kiểu hỏi dưới dạng "Similar pages" song kết quả hiển

thị trang kết quả lại có nội dung khác nhiều so với nội dung của trang đang xem xét (hình 2.9).

Chúng tôi đề xuất cách thức cho phép mở rộng dạng câu hỏi mà người dùng đưa cho máy tìm kiếm tuy đơn giản song lại rất tự nhiên. Đối với máy tìm kiếm (chúng tôi



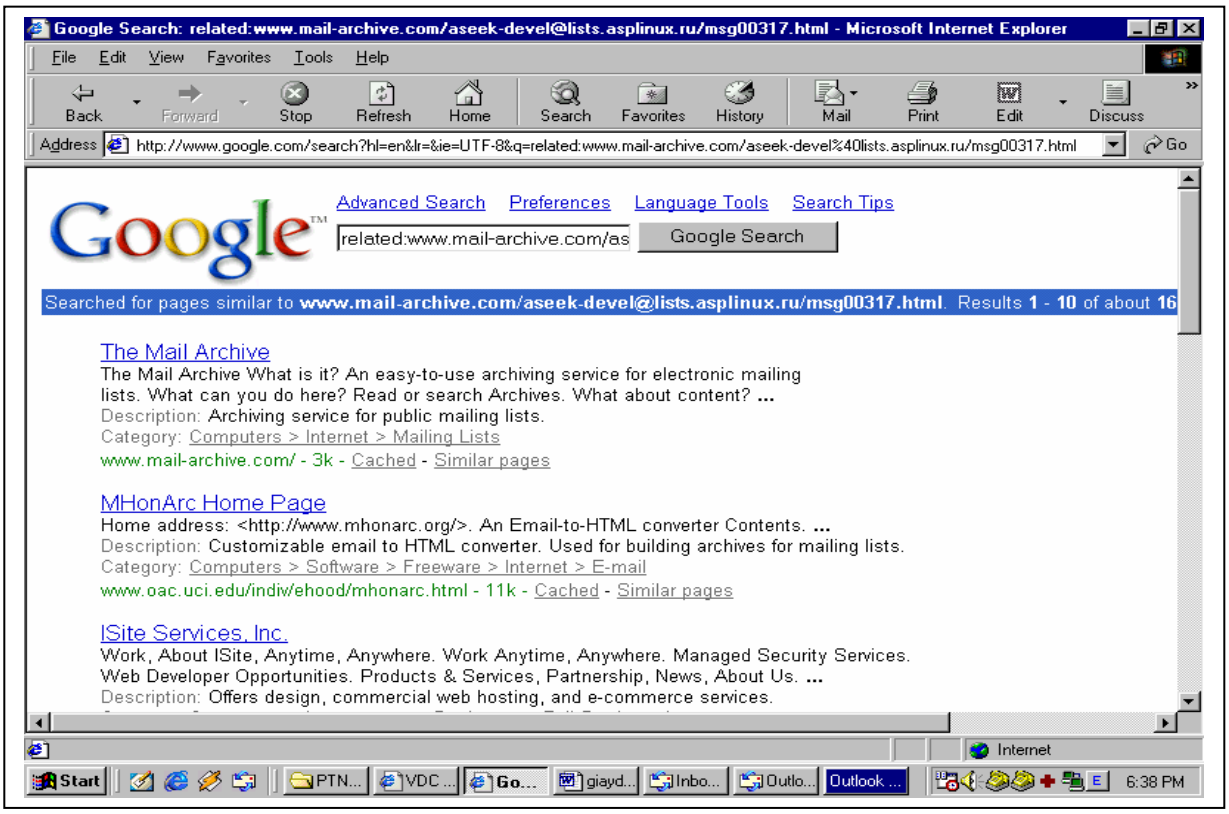
Hình2.8. Một phần kết quả tìm kiếm của Google đối với cụm từ "Bui QuangMinh"

đang triển khai cho máy tìm kiếm VietSeek), đề xuất của chúng tôi là cho thêm chức năng tìm kiếm các trang web "gần về nội dung" với trang web hiện thời mà người dùng đang xem (Việc hiển thị trang web vẫn thuộc phạm vi của máy tìm kiếm).

Khái niệm "gần về nội dung" được hiểu như sau: Theo một cách biểu diễn nào đó cho các trang web, máy tìm kiếm xác định một độ đo "gần nhau" giữa các trang web theo cách biểu diễn đã cho. Như vậy, cần bổ sung cho máy tìm kiếm một cách biểu diến trang web mới và xác định cho nó một độ đo gần nhau giữa các trang web.

- ***Vấn đề biểu diễn trang web***

Như đã được phân tích trong mục 2.2, phương pháp biểu diễn vector với việc sử dụng thông tin từ các trang web láng giềng cho nhiều "ngữ nghĩa về nội dung" của trang web. Định hướng vào mục tiêu đòi hỏi tối thiểu về không gian lưu trữ và tốc độ



Hình2.9. Trang kết quả tìm kiếm "Similar pages" của Google

tìm kiếm nhanh, chúng tôi lựa chọn phương pháp do chúng tôi đề xuất tại cuối mục trước; đồng thời hệ số phân biệt trang web đang xét với các loại trang web láng giềng tương ứng là 4, 2, 1 như đã được trình bày ở trên.

Chi tiết về quá trình nhận được tập hợp vector biểu diễn được trình bày trong phần dưới đây và thiết kế lôgic chi tiết về dữ liệu được trình bày trong chương 3.

• Vấn đề xác định độ đo gần nhau về nội dung

Như đã nói ở trên, cách biểu diễn vector được chọn cho nhiều ngữ nghĩa về nội dung của trang web và độ đo gần nhau về nội dung được tính theo độ gần nhau của hai vector biểu diễn. Giả thiết các vector biểu diễn đã được chuẩn hóa theo một nghĩa nào

đó (tổng giá trị các thành phần trong một vector cho một giá trị xác định, chẳng hạn 100). Với hai vector cho trước, chúng tôi đề nghị sử dụng cosin của góc giữa hai vector đó làm độ gần nhau Sm của chúng [9].

Giả sử vector $X = (X_1, X_2, ..., X_N)$ và vector $Y = (Y_1, Y_2, ..., Y_N)$ thì độ gần nhau Sm (X, Y) là Cos (X, Y) của góc tạo bởi X và Y được tính theo công thức sau:

$$Sm(X,Y) = Cos(X,Y) = \frac{\sum_l X_l * Y_l}{\sqrt{\sum_l X_l^2 \sum_l Y_l^2}}$$

- ***Quá trình xây dựng các vector biểu diễn***

Như đã biết, nội dung các bảng chỉ mục (chỉ mục nội dung, chỉ mục liên kết, chỉ mục ngược ...) trong máy tìm kiếm có đầy đủ thông tin để chúng ta xây dựng được hệ thống các vector biểu diễn. Dưới đây là sự mô tả sơ lược về quá trình này (Thuật toán chi tiết cho việc xây dựng các vector biểu diễn được mô tả tại chương 3):

- Xây dựng vector chưa chuẩn hóa: số lượng thành phần bằng số lượng từ khóa trong hệ thống, mỗi thành phần trong vector tương ứng với từ khóa theo chỉ số WordID (xem 2.2). Giả sử đang xem xét trang web W và từ khóa T, chúng ta nhận được tổng đánh giá xuất hiện của từ khóa T trong W là n_1, tổng đánh giá xuất hiện của từ khóa T trong tất cả các láng giềng có hai liên kết với W là n_2, tổng đánh giá xuất hiện của từ khóa T trong tất cả các trang web láng giềng còn lại là n_3, thế thì giá trị n_W là thành phần tương ứng với từ khóa W trong vector biểu diễn được tính:

$n_W = [(4*n_1 + 2* n_2 + n_3)/7]$ trong đó ký hiệu [.] chỉ hàm lấy phần nguyên.

Khái niệm "đánh giá xuất hiện" từ khóa W trong một trang web được hiểu là tổng của các lần xuất hiện của từ khóa W trong trang web đó với hệ số vị trí của từng lần xuất hiện (ở tiêu đề, ở thẻ thuộc tính, ở siêu liên kết, ở thân trang web ...).

- Chuẩn hóa vector biểu diễn theo tính toán sau: từ các giá trị thành phần n_W nhận được, tính giá trị thành phần sau chuẩn hóa N_W theo công thức sau đây:

$$N_W = \frac{n_W * 100}{\sum_W n_W}$$

Chú ý rằng, trong một số lĩnh vực ứng dụng cụ thể, cho phép sử dụng không nhiều từ khóa chuyên ngành trong máy tìm kiếm và vì thế độ dài vector biểu diễn không lớn.

• ***Thực hiện chức năng tìm kiếm trang gần theo nội dung***

Cho trang web hiện thời là W, chức năng tìm kiếm các trang gần nội dung với W được thực hiện theo các bước sau:

(1) Tính độ gần nhau giữa vector biểu diễn W với vector biểu diễn trang web X bất kỳ trong hệ thống: Tính Sm(W,X)

(2) Xếp lại các trang web X theo thứ tự giảm dần của Sm(W,X)

(3) Hiển thị danh sách tóm tắt các trang web đã được sắp xếp.

Để bước (1) và bước (2) được thực hiện nhanh và cung cấp cho người dùng những trang web "gần về nội dung" với trang web W, có thể đưa thêm một số nội dung sau:

- Sắp xếp hệ thống các vector tăng dần theo hệ số góc của nó so với vector chỉ chứa chiều thứ nhất (100, 0, ... , 0),

- Cho một ngưỡng δ để lọc bỏ mọi vector X mà độ gần Sm(W,X) nhỏ hơn δ.

Nội dung chi tiết cho đề xuất ở đây sẽ được trình bày trong chương tiếp theo.

KẾT LUẬN CHƯƠNG 2

Việc xây dựng các hệ thống xử lý dữ liệu trang web được tiến hành theo hai hướng chính là hướng sử dụng mô hình vector biểu diễn trang web và hướng hoạt động trong các máy tìm kiếm. Một số máy tìm kiếm điển hình (Yahoo, Google ...) đã hoạt động khá hiệu quả, tuy nhiên câu hỏi tìm kiếm ở dạng rất đơn giản. Trong mô hình biểu diễn vector, các nghiên cứu chú trọng việc khai thác ngữ nghĩa suy rộng của các từ khóa trong các trang web láng giềng. Luận văn đã đề xuất một cách thức biểu diễn vector cho các trang web (mục 2.2).

Trên cơ sở tìm hiểu và phân tích các phương pháp biểu diễn trang web theo hai hướng nói trên, luận văn đã đề xuất việc bổ sung một cách biểu diễn vector cho trang web trong các máy tìm kiếm và chức năng tìm kiếm trang web "gần theo nội dung" (mục 2.3). Trong chương này, luận văn cũng trình bày những bước sơ bộ để triển khai những đề xuất trên đây.

Trong chương 3, luận văn tập trung trình bày thể hiện cụ thể của các đề xuất trên đây áp dụng vào máy tìm kiếm VietSeek..

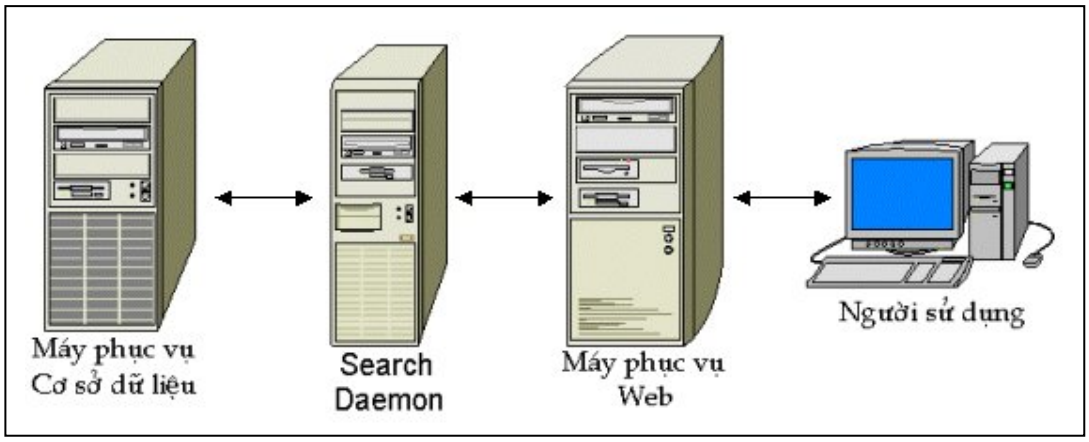
CHƯƠNG III. MÁY TÌM KIẾM VIETSEEK VÀ THỬ NGHIỆM THUẬT TOÁN TÌM KIẾM THEO NỘI DUNG

3.1 Máy tìm kiếm VietSeek

3.1.1 Các đặc điểm cơ bản của Vietseek

Vietseek là một trong số ít các máy tìm kiếm tiếng Việt đã được xây dựng và sử dụng hiện nay (như Panvietnam của công ty Netnam, VinaSEEK của công ty Tinh Vân, Hoa Tiêu của Vương Quang Khải). Vietseek được phát triển dựa trên ASPseek (là một phần mềm mã nguồn mở) bởi Bùi Quang Minh trong khuôn khổ của Đề tài QG-02-02 và công ty TTVNOnline [1].

Về cơ bản, cấu trúc của Vietseek giống với cấu trúc của một máy tìm kiếm thông thường (hình 2.1). Tuy nhiên Vietseek chưa có chức năng phản hồi lại thông tin từ bộ truy vấn đến bộ điều khiển tìm duyệt. Vietseek đã xây dựng được chỉ mục cho khoảng 3000 site tiếng Việt với khoảng 3 triệu trang web, và khoảng 2,5 triệu từ khoá đã được lưu trữ. Hiện nay Vietseek đang tiếp tục tiến hành tạo chỉ mục cho khoảng 7 triệu trang web khác. Mô hình hoạt động của Vietseek được mô tả trong hình 3.1



Hình 3.1.Mô hình hoạt động của Vietseek

Cơ sở dữ liệu về các trang web và chỉ mục được lưu trữ trong máy phục vụ cơ sở dữ liệu. Môđun tìm kiếm (Search Deamon) là một tiến trình chạy ngầm hoạt động theo cơ chế client/server, có nhiệm vụ lập danh sách các URL thoả mãn yêu cầu của người

dùng. Sau đó tính hạng hiển thị cho tất cả các trang theo bốn yếu tố rồi nhóm theo site và sắp xếp từ trên xuống. Môđun giao diện (máy phục vụ web) làm nhiệm vụ lấy kết quả trả về từ môdun tìm kiếm, trộn lại rồi hiển thị dưới dạng web cho người dùng.

Khi tính hạng trang web, hệ số hãm d được chọn là 0.85 , và số vòng lặp khi tính toán là khoảng 20 (cho khoảng vài triệu trang).

Vietseek tính hạng hiển thị cho một trang web dựa vào bốn yếu tố sau:

1. Vị trí xuất hiện của từ khoá trong văn bản,
2. Vị trí tương đối giữa các từ khoá trong trang,
3. Thuộc tính của từ khoá (từ tìm kiếm đặt trong thẻ H1, H2,...., H5),
4. Giá trị hạng của trang.

3.1.2 Cơ sở dữ liệu của Vietseek

Cơ sở dữ liệu của Vietseek được chia thành 2 phần:

1. Phần 1: dữ liệu về văn bản web, domain, word... được lưu trữ trong các bảng của cơ sở dữ liệu Mysql
2. Phần 2: dữ liệu chỉ mục (index) được lưu trữ riêng và có cơ cấu riêng. Để đạt được tốc độ xử lý cao nên không dùng Mysql mà được lưu trữ trong các file nhị phân khác nhau.

Quá trình tìm kiếm chỉ truy nhập đến phần 2, còn khi hiển thị kết quả mới truy nhập đến phần 1. Sau đây là chi tiết cách biểu diễn các dữ liệu trong hai phần.

❖ ***Phần 1: dữ liệu được lưu trữ trong các bảng của cơ sở dữ liệu MySQL***

♦ Thông tin về các site được lưu trữ trong bảng ***sites***

Tên trường	Miêu tả
Site_id	Mã nhận dạng của site
Site	Nội dung cụ thể của tên site (ví dụ www. Yahoo.com)

♦ Thông tin về các URL (là thông tin về các trang web) được lưu trong bảng ***urlword*** (bảng này lưu giữ thông tin về tất cả các URL đã được tạo chỉ mục và các URL chưa tạo chỉ mục).

Tên trường	Miêu tả
url_id	Mã nhận dạng của URL (của trang web)
site_id	Mã nhận dạng của site chứa trang đó
deleted	Được gán giá trị 1 nếu máy chủ trả về lỗi 404, hoặc các quy định (được thiết đặt cho chương trình) không cho phép tạo chỉ mục cho trang này
url	Nội dung của URL của trang
next_index_time	Thời gian của lần tạo chỉ mục tiếp theo, giá trị là "giây"
status	Là giá trị kiểm tra tình trạng HTTP do máy chủ trả về, hoặc có giá trị là 0 nếu trang này chưa được tạo chỉ mục.
crc	Mã kiểm tra của trang (MD5 checksum: thuật toán mã hóa MD5)
last_modified	Giá trị kiểm tra "HTTP header" của trang, được máy chủ HTTP trả về
etag	Giá trị "Etag header" được máy chủ HTTP trả về
last_index_time	Thời gian của lần tạo chỉ mục trước, giá trị là "giây"
referrer	Mã nhận dạng (url_id) của trang đầu tiên tham khảo đến trang này
tag	Một thẻ tuỳ ý nào đó
hops	Độ sâu của trang trong cây liên kết
redir	
origin	Mã nhận dạng của trang gốc mà nó (trang hiện tại) là bản sao. Nếu nó không phải là bản sao thì trường này nhận giá trị là 0

♦ Bảng ***wordurl*** (lưu giữ các thông tin về mỗi từ trong cơ sở dữ liệu, mỗi bản ghi tương ứng với một từ)

Tên trường	Miêu tả
word	Lưu giữ từ khoá
word_id	Lưu giữ mã của từ khoá
urls	Lưu giữ thông tin về các site và các URL mà từ xuất hiện. Nếu kích thước thông tin lớn hơn 1000 byte thì giá trị của trường này sẽ rỗng và thông tin sẽ được lưu giữ ở trong các file riêng biệt khác có tên là wordurl.urls
urlcount	Tổng số lượng các trang web (URL) chứa từ khóa
totalcount	Tổng số lần xuất hiện của từ khóa trong tất cả các trang web (URL)

♦ Bảng ***citation*** (lưu giữ các thông tin về chỉ mục đảo của các siêu liên kết)

Tên trường	Miêu tả
url_id	Mã nhận dạng của URL
referrers	Một mảng gồm các url_id của các trang có liên kết đến trang này

❖ ***Phần 2: dữ liệu chỉ mục được lưu trong các file nhị phân***

♦ File ***wordurl.urls*** (file này lưu trữ các thông tin về các site và các URL mà từ khóa xuất hiện, nếu kích thước phần này trong giới hạn 1000 byte thì được lưu trữ trong trường **urls** thuộc bảng ***wordurl***)

Các thông tin về các site, được sắp xếp theo site_id		
Offset	**Độ dài**	**Miêu tả chi tiết**
0	4	Giá trị offset bắt đầu thông tin về site thứ nhất mà từ xuất hiện

4	4	Mã nhận dạng của site thứ nhất nơi từ xuất hiện
8	4	Giá trị offset bắt đầu thông tin về site thứ hai mà từ xuất hiện
12	4	Mã nhận dạng của site thứ hai nơi từ xuất hiện
.....		
(N-1)*8 + 4	4	Giá trị offset bắt đầu về site thứ N, với N có giá trị bằng tổng số các site mà từ xuất hiện.
(N-1)*8 + 8	4	Mã nhận dạng của site thứ N nơi từ xuất hiện
Thông tin về các URL, được lưu trữ tiếp ngay sau thông tin về site. Giá trị offset được tính từ 0		
0	4	url_id của trang thứ nhất trong site thứ nhất trong phần thông tin về các site
4	2	Tổng số từ trong URL này
6	2	Vị trí thứ nhất
8	2	Vị trí thứ hai
.....		
6 + (N-1)*2	2	Vị trí thứ N, với N là tổng số từ xuất hiện trong URL
Lặp lại với các thông tin cho các URL của cùng site, nhưng có url_id lớn hơn url_id của phần trên		
.....		
Lặp lại với các thông tin về URL của site tiếp theo trong phần thông tin về site		

❖ ***Ví dụ về cách lưu trữ dữ liệu trong CSDL của Vietseek***

Ví dụ đơn giản sau đây cho phép hình dung ra cách lưu trữ dữ liệu trong Vietseek.

Giả sử có hai site là http://www.vanban.vn và http://www.luat.vn, cùng một số trang nằm trong hai site đó và chúng được gán cho các mã nhận dạng. Chúng ta nhận được các bảng thông tin như sau:

♦ Bảng sites

site_id	Nội dung
1	htttp://www.vanban.vn
2	htttp://www.luat.vn

♦ Bảng urlword (đã lược bớt một số trường không quan trọng)

url_id	Site_id	Nội dung
1	1	htttp://www.vanban.vn/index1.htm
2	1	htttp://www.vanban.vn/index2.htm
3	1	htttp://www.vanban.vn/index3.htm
4	1	htttp://www.vanban.vn/index4.htm
5	1	htttp://www.vanban.vn/index5.htm
6	1	htttp://www.vanban.vn/index6.htm
7	2	htttp://www. luat.vn/index1.htm
8	2	htttp://www. luat.vn/index2.htm
9	2	htttp://www. luat.vn/index3.htm
10	2	htttp://www. luat.vn/index4.htm
11	2	htttp://www. luat.vn/index5.htm
12	2	htttp://www. luat.vn/index6.htm

Ví dụ nội dung của trang htttp://www.vanban.vn/index3.htm là *"giới thiệu luật giao thông. Luật có hiệu lực từ ngày 1/1/1999 "*

Nội dung của trang htttp://www.vanban.vn/index5.htm là *"giới thiệu luật hình sự. Bộ luật có 300 điều. Luật có hiệu lực từ ngày 1/1/1999 "*

Nội dung của trang htttp://www.luat.vn/index2.htm là *"bộ luật hình sự"*

♦ Bảng ***wordurl*** lưu giữ tất cả các sự xuất hiện của mỗi từ trong mỗi trang, do kích thước nên trường ***urls*** của bảng này được lưu ở trong các file nhị phân. Đối với từ "luật" thì sẽ được lưu trong bảng ***wordurl*** và trong file nhị phân tương ứng như sau:

word	luật
word_id	1
urls	(Thông tin về từ có trong các URL, kết nối đến file nhị phân *wordurl.urls*)
urlcount	3
totalcount	6

♦ Nội dung của file nhị phân ***wordurl.urls*** như sau:

url	Vị trí byte	Giá trị
	0	16 (offset bắt đầu thông tin về site thứ nhất mà từ xuất hiện)
	4	1 (site-id của site thứ nhất)
	8	38 (offset bắt đầu thông tin về site thứ hai mà từ xuất hiện)
	12	2 (site-id của site thứ 2)
	16	3 (URL thứ 3 trong site 1)
	20	2 (xuất hiện 2 lần)
	22	3 (từ thứ 3 trong URL 3)
	24	6 (từ thứ 6 trong URL 3)
	26	5 (URL thứ 5 của site 1)
	30	3 (xuất hiện 3 lần)
	32	3 (từ thứ 3 trong URL 5)
	34	7 (từ thứ 7 trong URL 5)
	36	11 (từ thứ 11 trong URL 5)
	38	8 (URL thứ 8 của site 2)
	42	1 (xuất hiện 1 lần)
	44	2 (từ thứ 2 trong URL 8)

Vietseek đã xây dựng xong chức năng tìm kiếm theo văn bản, và chức năng tìm kiếm hình ảnh hiện đang được xây xây dựng. Các kết quả tìm kiếm được trả về rất nhanh và chính xác do đã thực hiện được việc tính hạng trang web dựa vào các liên kết ngay từ khi tạo chỉ mục cho các trang và việc xếp hạng hiển thị trang kết quả đã được tính toán dựa theo bốn tiêu chí được nêu ở phần 3.1.1. Vietseek đã chuyển đổi được tất cả các loại mã tiếng Việt khác nhau (TCVN, VNI, VIQR) sang mã Unicode, và kết quả được trả lại dưới dạng mã Unicode. Tuy nhiên, còn một số vấn đề mà Vietseek chưa giải quyết được. Thứ nhất, chưa phân tán cơ sở dữ liệu vào các nút lưu trữ khác nhau,

Hình 3.1.Giao diện một trang kết quả tìm kiếm của máy tìm kiếm Vietseek

nên trong tương lai khi số lượng các trang web tiếng Việt phát triển nhiều hơn nữa sẽ rất khó khăn trong việc lưu trữ. Do chưa phân tán được cơ sở dữ liệu vào nhiều nút nên Vietseek chưa sử dụng kỹ thuật phân hoạch chỉ mục (index partitional). Thứ hai, chưa xây dựng được chức năng tự học của máy tìm kiếm từ danh sách các URL được người dùng sử dụng trong kết quả trả về. Và cuối cùng, giống như hầu hết các máy tìm kiếm khác, Vietseek chưa quan tâm đến việc xếp hạng các trang web dựa vào tần số xuất hiện các từ khoá tìm kiếm trong trang web đó.

3.2 Đề xuất thuật toán tìm kiếm mới cho máy tìm kiếm VietSeek

3.2.1 Những cơ sở để đề xuất thuật toán

Qua phân tích chi tiết cách biểu diễn dữ liệu của máy tìm kiếm Vietseek, chúng ta thấy việc tổ chức lưu trữ trong cơ sở dữ liệu khá hợp lý. Do việc tìm kiếm được thực hiện theo từ khoá nên đối tượng chính của cách biểu diễn trong Vietseek là các từ khoá, thông tin về sự xuất hiện của các từ khoá trong các trang được sắp xếp theo word_id và được lưu trữ trong các file nhị phân. Tổ chức lưu trữ như vậy giúp cho việc tìm kiếm nhanh và hiệu quả. Trong mục 2.3, chúng tôi đã đề xuất việc bổ sung vào máy tìm kiếm cách biểu diễn trang web theo mô hình vector. Trong phần này, chúng tôi trình bày chi tiết các thiết kế cho việc biểu diễn đó. Để tính được trọng số xuất hiện (đánh giá xuất hiện) của các từ trong các trang, chắc chắn là cách biểu diễn này phải coi đối tượng chính là các URL. Vì trong cơ sở dữ liệu của Vietseek có bảng ***urlword*** lưu trữ các thông tin về các URL, cho nên chúng tôi sử dụng luôn bảng này làm cơ sở cải tiến để biểu diễn thông tin theo cách mới.

Cách biểu diễn như sau: chúng ta thêm vào bảng ***urlword*** một trường mới, tên là **content_vector,** trường này có kiểu giống như kiểu của trường **urls** trong bảng ***wordurl***. Trường này lưu trữ các thông tin về vector biểu diễn cho trang web tương ứng có mã nhận dạng lưu trong trường **url_id** của cùng bảng. Các trường trong bảng ***urlword*** được mô tả như sau (đã lược bớt các trường không liên quan):

Tên trường	Miêu tả
url_id	Mã nhận dạng của URL (của trang web)
site_id	Mã nhận dạng của site chứa trang đó
url	Nội dung của URL của trang
content_vector	Thông tin về vector biểu diễn URL (nhận giá trị rỗng nếu kích thước thông tin > 1000 byte, và thông tin sẽ được lưu trữ trong file nhị phân có tên là urlword.content_vector)

...	....

Cấu trúc của file ***urlword.content_vector*** được miêu tả như sau

Thông tin về các từ xuất hiện trong URL, được sắp xếp theo word_id		
Vị trí	**Độ dài**	**Miêu tả**
0	4	Word_id (mã nhận dạng của từ thứ nhất xuất hiện trong URL)
4	2	Trọng số của từ thứ nhất xuất hiện trong URL
6	4	Word_id (mã nhận dạng của từ thứ hai xuất hiện trong URL)
10	2	Trọng số của từ thứ hai xuất hiện trong URL
.....		
Lặp cho các từ tiếp theo xuất hiện trong URL		

Việc tạo nội dung trường **urlword.content_vector** cho dữ liệu đã có trong cơ sở dữ liệu Vietseek được thực hiện bằng cách duyệt file ***wordurl.urls*** và file ***citation***. Từ hai file này chúng ta lấy được các thông tin về tần số xuất hiện của các từ trong mỗi trang và thông tin về mối liên kết giữa một trang đang xét với các trang láng giềng, và từ đó tính toán được trọng số của mỗi từ. Khi cơ sở dữ liệu được tạo chỉ mục lại (sau một khoảng thời gian nhất định) thì giá trị của trường này được tính toán luôn trong quá trình tạo chỉ mục.

Việc thêm trường **content_vector** mới vào cơ sở dữ liệu không làm ảnh hưởng đến sự hoạt động của toàn bộ hệ thống Vietseek cũng như các modun tìm kiếm, tạo chỉ mục... vì các lệnh thao tác với CSDL dữ liệu đều chỉ rõ các trường cần thao tác. Do đó nếu thêm trường mới mà không có ràng buộc gì không làm ảnh hưởng tới các hoạt động của hệ thống.

Do số lượng các trang web là rất lớn nên việc tính toán và so sánh độ gần nhau giữa vector biểu diễn của một trang đang xét với các trang còn lại trong cơ sở dữ liệu chắc chắn sẽ tốn thời gian. Do đó với mỗi URL chúng tôi tạo luôn 1 danh sách các

URL tương tự với nó, tức là có độ gần nhau lớn. Việc lưu trữ các URL này được tổ chức tương tự như việc tổ chức lưu trữ các siêu liên kết giữa các trang. Cụ thể là tương tự như bảng ***citation***. Số lượng các URL này được giới hạn bởi ngưỡng δ được giới hạn về số lượng (khoảng 100 URL có độ tương tự cao nhất), vì thông thường người sử dụng chỉ quan tâm đến nhiều nhất là 20 giá trị đầu tiên.

3.2.2 Thuật toán

❖ ***Thuật toán 3.1 (tạo content_vector)***

(1) **word** ← từ khóa đầu tiên trong bảng **wordurl** (**word** chưa được xét)

(2) while (trong bảng **wordurl** còn từ khóa chưa được xét) thực hiện

{ Xét **word**}

(2.1) Lấy radanh sách URL tương ứng với **word**,

(2.2) **url** ← URL đầu tiên trong danh sách (**url** chưa được xét)

(2.3) while (trong danh sách còn URL chưa được xét) thực hiện

{ Xét **url** - Tính trọng số của **word** trong **url** }

(2.3.1) Lấy n_1 = tổng số từ xuất hiện trong **url** (có sẵn trong bảng wordurl.urls)

(2.3.2) Tham chiếu theo url_id đến bảng citation để có được thông tin về các URL có liên kết đến **url**,.

(2.3.3) Tính n_2 và n_3

(2.3.4) Tính n_W theo công thức $n_W = [(4*n_1 + 2* n_2 + n_3)/7]$

(2.3.5) Bổ sung thông tin về **word** hiện tại (gồm word_id, trọng số n_W) vào cuối file ***urlword.content_vector***

(2.3.6) **url** ← URL tiếp theo trong danh sách

{hết while (2.3)}

(2.2) **word** ← từ khóa tiếp theo trong bảng **wordurl**

{hết while (2)}

{hết thuật toán 3.1}

❖ ***Thuật toán 3.2 (tạo danh sách các URL "gần nội dung" ứng với URL)***

{Các URL được xếp theo tăng của chỉ s: 1, 2, ..., N}

1. I ← 1
2. J ← I + 1
3. Tính d_{IJ} = độ gần nhau của URL_I với URL_J
4. If d_{IJ} được đưa vào URL_I

   then

   Đưa d_{IJ} vào URL_I (bao gồm giá trị d_{IJ} và chỉ số J). Để thuật toán hoạt động nhanh chúng ta sử dụng danh sách các d_{IJ} trong URL_I được sắp xếp giảm dần về giá trị.
5. If d_{IJ} được đưa vào URL_J

   then Đưa d_{IJ} vào URL_J (bao gồm giá trị d_{IJ} và chỉ số I).
6. J ← J + 1
7. If J≤ N

   then chuyển về 3
8. I ← I + 1
9. If I < N

   then chuyển về 2
10. Kết thúc

Trong thuật toán này có hai bài toán con cần giải quyết:

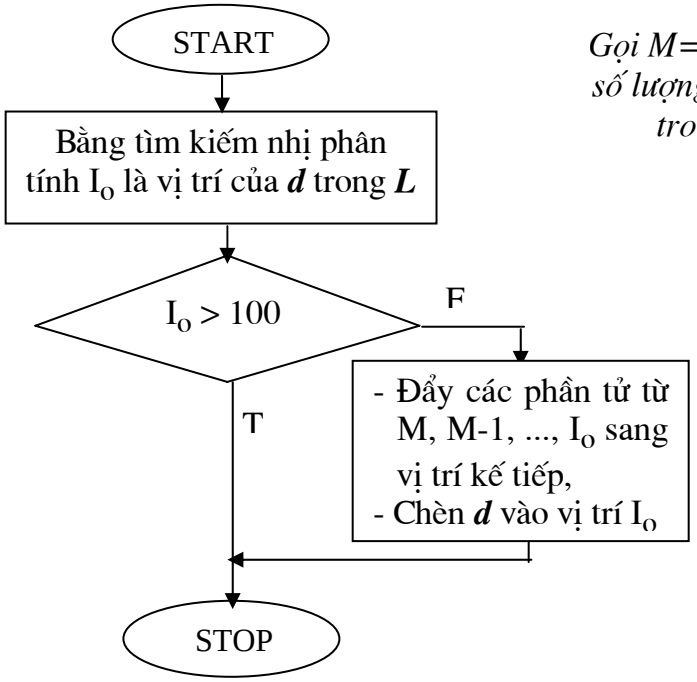
- Kiểm tra có đưa $d_{I,J}$ vào URL_I (hoặc URL_J) hay không. Vì mỗi URL chỉ cần lưu 100 lân cận gần nhất với nó cho nên khi thuật toán hoạt động, mỗi URL chỉ cần chứa không quá 100 lân cận "hiện thời" gần nhất. Khi có thêm một lân cận mới, nếu số

lượng lân cận có trong URL nhỏ thua 100 thì bổ sung lân cận mới vào; trong trường hợp đã có 100 lân cận rồi, nếu độ gần nhau mới lớn hơn ít nhất một lân cận đã có thì loại lân cận nhỏ nhất trong những lân cận đang tạm thời lưu giữ ra và đưa lân cận mới vào.

- Cho $d_{I,J}$ vào URL_I (hoặc URL_J): Đưa vào hai giá trị đó là giá trị lân cận $d_{I,J}$ và chỉ số J nếu xem xét URL_I (hoặc chỉ số I nếu xem xét URL_J).

Để thuận tiện cho các tính toán các giá trị được lưu trữ trong một URL theo giá trị giảm dần theo độ gần nhau: Sử dụng thuật toán chèn (hoặc chèn nhị phân) một phần tử vào một danh sách đã xếp đối với hai bài toán xem xét và bổ sung.

Thuật toán 3.2.a.theo sơ đồ khối sau đây mô tả sơ lược thuật toán giải quyết hai bài toán con này.



Thuật toán 3.2.a. Xem xét và chèn độ lân cận d vào danh sách L các độ lân cận

Sử dụng kết quả của thuật toán 3.2, chúng ta hoàn toàn có thể xây dựng thuật toán tìm kiếm các trang web gần nội dung với trang web hiện thời bằng cách hiển thị danh sách 100 trang web tương ứng với trang web hiện thời.

Tuy nhiên, chúng tôi xin nêu ra ý tưởng kết hợp giá trị gần nội dung với giá trị hạng của trang web để đưa ra một giá trị kết hợp trong việc sắp xếp các trang web hiển thị. Nội dung đó được trình bày trong thuật toán 3.3 dưới đây.

❖ ***Thuật toán 3.3. (Tìm kiếm các trang web "gần" với trang web hiện thời)***

1. Tính "độ gần" của trang web hiện thời với 100 trang web trong danh sách tương ứng với nó theo công thức tổ hợp giữa độ gần về nội dung với hạng của từng trang web trong danh sách. Chẳng hạn, công thức tổ hợp có thể là:

$$\gamma_i = d*\alpha_i + (1-d)*\beta_i, \qquad (i=1,..., 100)$$

Trong đó, α_i là độ gần về nội dung và β_i là hạng liên kết đã có, γ_i là độ gần cần tính còn d là trọng số ($d \geq 0.8$ để nhấn mạnh độ gần về nội dung).

2. Sắp xếp lại danh sách 100 trang web nói trên theo giá trị giảm dần của γ_i.
3. Hiển thị 100 trang web nói trên theo thứ tự đã được sắp xếp.

{hết thuật toán 3..3}

Chú ý rằng để công việc tìm kiếm được nhanh chóng, hai bước 1 và 2 của thuật toán 3.3 có thể được tính một lần cho toàn bộ hệ thống và thuật toán tìm kiếm lúc đó được tiến hành như trình bày trong bước 3 và đạt được tốc độ cao.

KẾT LUẬN CHƯƠNG 3

Chương 3 trình bày cấu trúc thành phần của máy tìm kiếm tiếng Việt VietSeek và sơ đồ hoạt động của nó. Phát triển những đề xuất của chương 2, luận văn trình bày thiết kế chi tiết việc bổ sung thành phần dữ liệu (biểu diễn trang web theo mô hình vector, thuật toán 3.1) và chức năng tìm kiếm "gần về nội dung" dựa trên biểu diễn vector (thuật toán 3.3). Để tăng tốc độ tìm kiếm, luận văn đề xuất việc lưu trữ sẵn 100 chỉ số trang web gần với mỗi trang web (thuật toán 3.2).

Các thiết kế dữ liệu và chức năng được đề xuất có tính khả thi. Trong thời gian tới, chúng tôi sẽ tiếp tục cài đặt thực sự trên VietSeek.

PHẦN KẾT LUẬN

1. Kết quả đạt được của luận văn

Thông qua việc khảo sát, phân tích, phát triển nội dung một số công trình nghiên cứu gần đây về các bài toán biểu diễn và xử lý dữ liệu trang web, luận văn đã hoàn thành một số kết quả chính sau đây:

• Hệ thống hóa hai phương pháp tiếp cận điển hình để biểu diễn trang web đang được nghiên cứu và triển khai hiện nay trong lĩnh vực xử lý dữ liệu web là phương pháp biểu diễn trong các máy tìm kiếm (mục 2.1) và phương pháp biểu diễn theo mô hình vector (mục 2.2),

• Thông qua việc phân tích, đánh giá đặc điểm của từng phương pháp nói trên, luận văn đã:

- Đề xuất một cách thức trình bày vector biểu diễn trang web vừa đảm bảo việc khai thác các mối liên kết các trang web thông qua siêu liên kết, vừa đảm bảo được độ dài vector biểu diễn không lớn (mục 2.2.2),

- Đề xuất một phương pháp biểu diễn trang web kết hợp trong máy tìm kiếm và thiết kế giải pháp cho các bài toán tìm kiếm, phân lớp trong các máy tìm kiếm theo phương pháp biểu diễn được đề xuất (mục 2.3),

- Thông qua việc khảo sát dữ liệu của máy tìm kiếm tiếng Việt VietSeek, luận văn thiết kế các dữ liệu bổ sung phù hợp với phương pháp biểu diễn mới và từ đó đề xuất bổ sung thêm chức năng tìm kiếm trang web có nội dung "gần" với nội dung trang web hiện thời (mục 3.3),

• Khảo sát các phương pháp biểu diễn website trong đó chú trọng tới cách biểu diễn cây website. Đề xuất thuật toán xây dựng cây website (mục 1.2.2).

Tuy nhiên do hạn chế về thời gian hoàn thành luận văn nên việc triển khai phát triển đối với máy tìm kiếm VietSeek mới dừng ở mức lôgic trong việc thiết kế dữ liệu và chức năng. Dù rằng các thiết kế mà luận văn trình bày là hoàn toàn khả thi song việc chưa cài đặt được các đề xuất phát triển là mặt hạn chế của luận văn.

2. Phương hướng nghiên cứu tiếp theo

Lĩnh vực biểu diễn và xử lý dữ liệu trang web là một lĩnh vực thời sự, các phương pháp biểu diễn đang ngày được nghiên cứu, phát triển nhằm xây dựng các hệ thống cơ sở dữ liệu trang web, các máy tìm kiếm ngày càng tốt hơn nhằm phục vụ người sử dụng ngày càng hiệu quả hơn. Trước tiên, bài toán biểu diễn trang web vẫn chứa đựng nhiều vấn đề cần được nghiên cứu và phát triển. Chẳng hạn, vấn đề chuyển giao "ngữ nghĩa" của các từ khóa từ trang web này sang trang web khác đang được nhiều nhóm nghiên cứu giải quyết theo các cách thức khác nhau, trong đó có giải pháp tính đến khu vực lân cận của các siêu liên kết. Mặt khác, hiện thực hóa các nghiên cứu, đề xuất của luận văn đối với máy tìm kiếm VietSeek cũng cần được cài đặt để các đề xuất đó được đánh giá thông qua hoạt động thực sự của VietSeek.

Những bài toán nói trên là nội dung nghiên cứu tiếp theo của luận văn này.

TÀI LIỆU THAM KHẢO

[1]. Bùi Quang Minh (2002). *Máy tìm kiếm VietSeek*. Báo cáo kết quả nghiên cứu thuộc Đề tài khoa học đặc biệt cấp ĐHQGHN mã số QG-02-02.

[2]. Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, Sriram Raghavan (2000). *Searching the web*. Technical Report, Computer Science Department, Stanford University.

[3]. Holger Billhardt, Daniel Borrajo, Victor Maojo (2002). *Context Vector Model for Information Retrieval*. Journal of American Society for Information Science and Technology (JASIS), **53** (3), 236-249.

[4]. Junghoo Cho and Hector Garcia-Molina (2000). *Estimating frequency of change*. In Submitted for publication, Technical Report, Computer Science Department, Stanford University.

[5]. Bui Cong Cuong (1999). *A Multiple Criteria Group Decision Making Model under Linguistic Assessments*. Institute of Mathematics, Hanoi, Vietnam.

[6]. Martin Ester, Hans-Peter Kriegei, Matthias Schubert (2002). *Web Site Mining: A new way to spot Competitors, Customerrs and Suppliers in the World Wide Web*. Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26,2002, Aberta, Canada, 249-258.

[7] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth (1996). *From Dataming to Knowledge Discovery: An Overview*. Advances Knowledge Discovery and Data Mining. AAAI Press/ MIT Press, 1-36.

[8]. Thorsten Joachims (2002). *Optimizing Search Engines using Clickthrough Data*. Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26,2002, Aberta, Canada, 133-142.

[9]. Nguyen Ngoc Minh, Nguyen Tri Thanh, Ha Quang Thuy, Luong Song Van, Nguyen Thi Van (2001). *A Knowledge Discovery Model in Fulltext Databases*. Proceedings of the First Workshop of International Joint Research: "Parallel Computing, Data Mining and Optical Networks". March 7, 2001, Japan Advanced Institute of Science and Technology (JAIST), Tatsunokuchi, Japan, 59-68.

[10]. Dunja Mladenic' (1998). *Machine Learning on Non-homogeneous, Distbuted Text Data* (Chapter 3. Document representation and learning algorithms). Doctoral dissertation. University of Ljubljana, Slovenia.

[11]. Sen Slattery (2002). *Hypertext Classification*. Doctoral dissertation (CMU-CS-02-142). School of Computer Science. Carnegie Mellon University.

[12]. Son Doan, Susumu Horiguchi (2002). *A new text representation method using fuzzy concepts in text catergozation*. JAIST Science Reports 2002.

[13]. E. Herrera-Viedma (2001). *Modeling the Retrieval Proces of an Information Retrieval System Using an Ordinal Fuzzy Lingguistic Approach*. Journal of American Society for Information Science and Technology (JASIS), **52** (6), 460-475.

[14]. Hwanjo Yu, Jiawei Han, Kevin Chen-Chuan (2002). *PEBL: Positive Example Based Learning for Web Page Classification Using SVM*. Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26,2002, Aberta, Canada, 239-248.