Tạp chí Tin học và Điều khiển học, T.20, S.4 (2004), 293–304

GIẢI PHÁP TÌM KIẾM TRANG WEB TƯƠNG TỰ TRONG MÁY TÌM KIẾM VIETSEEK

PHẠM THỊ THANH NAM, BÙI QUANG MINH, HÀ QUANG THỤY

Khoa Công nghệ, Đại học Quốc gia Hà Nội

Abstract. This article describes some of our propositions to upgrade the search function of the Vietseek by adding a vector representation solution for web pages. It alsoproposes the vector representation for web pages, a calculating formula for components of the vector, a "text-based similar" measure of two web pages, and algorithms to find out text-based similar pages of a given web page. Some realizations for above propositions in the Vietseek are described too.

Tóm tắt. Bài báo này trình bày một số đề xuất giải pháp nâng cấp chức năng tìm kiếm của máy tìm kiếm tiếng Việt Vietseek thông qua việc bổ sung biểu diễn vector cho trang web. Phương pháp biểu diễn vector cho trang web, công thức tính toán thành phần vector biểu diễn, độ đo "tương tự theo nội dung" giữa hai trang web và thuật toán tìm kiếm các trang web tương tự với một trang web đã cho được đề xuất. Phương pháp cài đặt các đề xuất trên đây trong máy tìm kiếm Vietseek cũng được trình bày.

1. MỞ ĐẦU

Khai phá text, đặc biệt là khai phá web, hiện được rất nhiều tổ chức, nhà khoa học quan m nghiên cứu, triển khai và kết quả của nhiều công trình nghiên cứu đã được công bố (xem :ang http://www.kdnuggets.com/publications/web-mining.html). Một số bài toán điển hình rong khai phá web là biểu diễn trang web, xử lí (tìm kiếm, phân lớp, khám phá luật), khai phá web-site.... Mô hình vector là mô hình biểu diễn văn bản điển hình và được sử dụng rộng rãi nhất. Có rất nhiều cách xác định giá trị thành phần của vector biểu diễn. Các giải pháp xử lý văn bản thường gắn bó mật thiết với cách biểu diễn được chọn. Mặc dù vậy, với mỗi cách biểu diễn văn bản đã cho, nghimười ta có thể sử dụng nhiều giải pháp xử lý khác nhau; chẳng hạn với cùng một cách biểu diễn vector, có thể sử dụng nhiều thuật toán phân lớp dựa trên các tiếp cận Bayes, k người láng giềng gần nhất (k-NN), cây phân lớp....

Máy tìm kiếm, điển hình như Yahoo, Google, Altavista, là công cụ tìm kiếm rất hữu ích khi làm việc trên Internet. Do định hướng mục tiêu giải quyết bài toán tìm kiếm, biểu diễn trang web trong máy tìm kiếm có một số nét độc đáo. Mặt khác, các máy tìm kiếm hiện tại chưa đề cập nhiều tới những giải pháp khai phá web khác ngoài bài toán tìm kiếm.

Trong bài báo này, chúng tôi định hướng vào việc nâng cấp chức năng tìm kiếm nhờ bổ sung biểu diễn vector trang web đối với máy tìm kiếm tiếng Việt thử nghiệm Vietseek do chúng tôi nghiên cứu, xây dựng.

Mục 2 của bài báo giới thiệu một số công trình nghiên cứu có nội dung liên quan đến bài báo. Mục 3 giới thiệu một số nội dung cơ bản về cấu trúc và hoạt động của máy tìm kiếm Vietseek. Các đề xuất giải pháp trong bài báo này (biểu diễn vector trang web, độ đo "gần

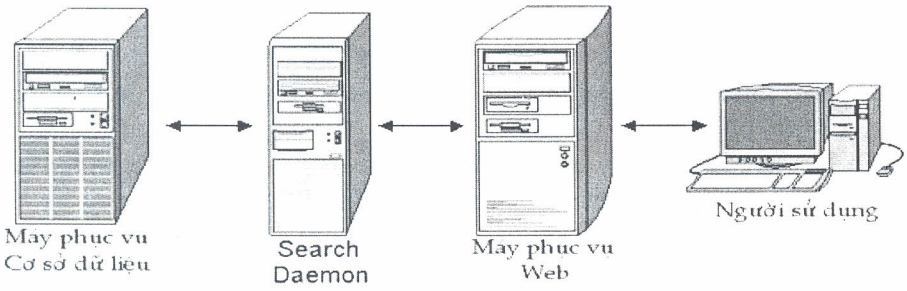
nhau theo nội dung" giữa hai trang web, công thức tính toán thành phần vector biểu diễn, thuật toán tìm kiếm các trang web tương tự) được trình bày trong Mục 4. Mục 5 giới thiệu một số kết quả cài đặt trong máy tìm kiếm Vietseek và bàn luận.

2. MỘT SỐ CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Trong [6], các tác tác giả đã trình bày một số kết quả nghiên cứu về khai phá text sử dụng mô hình vector. Giải pháp từ đồng nghĩa, đa ngôn ngữ và thử nghiệm giải pháp cây phân lớp cũng đã được trình bày ở bài báo này. Trong [7], Sen Slattery trình bày tổng hợp các phương pháp biểu diễn và xử lí siêu văn bản (hypertext), đặc biệt là các thuật toán phân lớp (Bayes, k-NN, FOIL, v.v.). Holger Billhardt, Daniel Borrajo và Victor Maojo [3], Son Doan và Horiguchi [8] đề xuất các giải pháp biểu diễn mới cho phép tăng ngữ nghĩa của vector biểu diễn văn bản khi tính đến tính phụ thuộc ngữ nghĩa của các từ khóa. Thorsten Joachims [9], Hwanjo Yu, Jiawei Han và Kevin Chen-Chuan [4] trình bày những giải pháp tăng cường chất lượng xử lý văn bản theo định hướng tới người sử dụng. Martin Ester, Hans-Peter Kriegei và Matthias Schubert [5] giới thiệu giải pháp phân lớp web site của các công ty loại nhỏ trên cơ sở thiết lập cây biểu diễn có sử dụng mô hình vector. Nội dung các bài báo khác [1, 2, 7] bổ sung nội dung các bài nói trên đây nhằm cho phép nhận được một cái nhìn toàn diện hơn về khai phá web hiện thời.

3. MÁY TÌM KIẾM VIETSEEK

Vietseek là một máy tìm kiếm tiếng Việt, được chúng tôi nghiên cứu phát triển từ phần mềm mã nguồn mở ASPseek trong khuôn khổ Đề tài QG-02-02 và được triển khai trong một dự án thử nghiệm của Mạng TTVN Online hợp tác với VDC1. Trong phương án ban đầu, Vietseek có cấu trúc của một máy tìm kiếm thông thường. Mô hình hoạt động của Vietseek được mô tả trong hình 1.



Hình 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 thỏa mãn yêu cầu của người dùng và 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 (Web Server) làm nhiệm vụ lấy kết quả trả về từ môđun 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, số vòng lặp tính toán là khoảng 20 (cho khoảng vài triệu trang).

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

1. Vị trí xuất hiện của từ khóa trong văn bản.
2. Vị trí tương đối giữa các từ khóa trong trang.
3. Thuộc tính của từ khóa (từ tìm kiếm đặt trong thẻ $H_1, H_2, ..., H_5$).
4. Giá trị hạng của trang.

Cơ sở dữ liệu của Vietseek

Cơ sở dữ liệu của Vietseek được chia thành 2 phần. Phần 1: dữ liệu về nội dung trang web, miền (site), từ khóa... được lưu trữ trong các bảng của cơ sở dữ liệu Mysql. Phần 2: dữ liệu chỉ mục (index) được lưu trữ riêng và có cơ cấu riêng. Để đạt được tốc độ xử lí cao nên không dùng cơ sở dữ liệu 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; ngược lại là 0
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, do máy chủ HTTP trả về
etag	Giá trị "Etag header" do 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 đại diện nào đó
hops	Độ sâu của trang trong cây liên kết
redir	Mã nhận dạng (url_id) nếu url hiện thời được gặp lại hoặc 0 nếu url chưa được gặp lại
origin	Mã nhận dạng của trang gốc mà 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 khóa
word_id	Lưu giữ mã của từ khóa
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

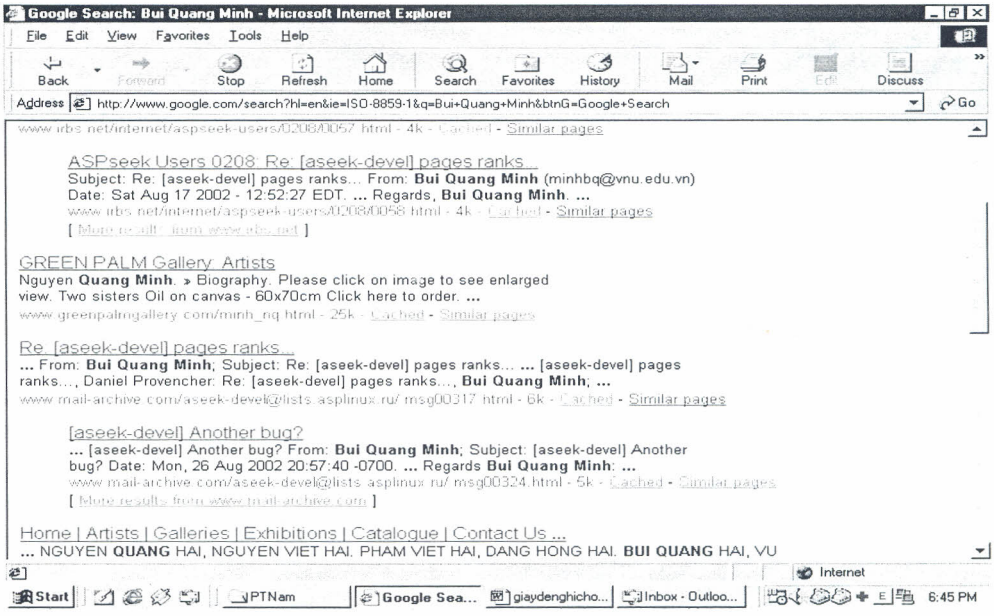
Phần 2: Dữ liệu chỉ mục được lưu trong các file nhị phân

Cấu trúc 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		

4. THUẬT TOÁN TÌM KIẾM THEO NỘI DUNG TRONG MÁY TÌM KIẾM VIETSEEK

Nhằm định hướng vào việc tìm kiếm theo từ khóa nên đối tượng chính của cách biểu diễn trong ASPseek là các từ khóa , thông tin về sự xuất hiện của các từ khóa 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.



Hình 2. Một phần kết quả tìm kiếm của Google đối với cụm từ "Bui Quang Minh"

Các máy tìm kiếm hiện nay cho phép người dùng đưa câu hỏi vào thường 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. Vì lẽ đó, máy tìm kiếm cần có giải pháp để hiển thị các trang web kết quả sao cho những trang có hạng càng cao càng được hiển thị trước. Để tính hạng của một trang, trong các máy tìm kiếm, thường sử dụng công thức bao hàm được mối quan hệ giữa các giá trị hạng của các trang web có liên kết lẫn nhau. Tuy nhiên, bài toán 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 hiển thị trang không chứa cụm từ "Bui Quang Minh" lại xuất hiện trước một trang có chứa cụm từ đó (hình 2). Vì vậy, vấn đề nghiên cứu đề xuất cách thức để máy tìm kiếm tiếp nhận dạng câu hỏi phức tạp hơn, biểu diễn đầy đủ hơn nội dung người dùng cần quan tâm và cho câu trả lời chính xác hơn vẫn đang được tiếp tục nghiên cứu hiện nay [3, 5, 6, 8]. Máy tìm kiếm Google đã cung cấp một kiểu hỏi dạng "Similar pages" song trong nhiều trường hợp, kết quả hiển thị trang "tương tự" có nội dung khác nhiều so với nội dung của trang đang xem xét (hình 3). Dưới đây là những đề xuất mở rộng dạng câu hỏi và giải pháp tìm kiếm được áp dụng cho máy tìm kiếm Vietseek thông qua việc bổ sung chức năng tìm kiếm các trang web "tương tự theo nội dung" với trang web hiện thời được hiển thị cho người dùng.

Khái niệm "tương tự theo nội dung" của các trang web được xác định thông qua một độ

đo "gần nhau" giữa các trang web theo một cách biểu diễn trang web được chọn. Như vậy, cần bổ sung cho máy tìm kiếm một cách biểu diễn mới cho trang web và xác định một độ đo gần nhau giữa các trang web theo cách biểu diễn đã cho.

Hình 3. Trang kết quả tìm kiếm "Similar pages" của Google

4.1. Biểu diễn trang web

Định hướng tới mục tiêu tối thiểu về không gian lưu trữ và tăng tốc độ tìm kiếm, chúng tôi lựa chọn một phương pháp mới biểu diễn vector cho trang web và có tính đến việc liên kết nội dung các trang web láng giềng.

Trong [7], Sen Slattery trình bày bốn phương pháp biểu diễn trang web theo mô hình vector, trong đó ba phương pháp biểu diễn sau sử dụng nội dung của các trang web láng giềng. Qua thực nghiệm, tác giả chỉ ra rằng phương pháp thứ ba cho kết quả tốt hơn phương pháp thứ nhất (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 gấp đôi mà thời gian tính toán cho các bài toán phân lớp và tìm kiếm cũng tăng lên với hệ số như vậy.

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 tính đến việc phân biệt sự xuất hiện của từ khóa trong trang web hiện thời khác với sự xuất hiện của chính từ khóa đó trong các trang web láng giềng. Tuy nhiên, độ dài vector biểu diễn lại tăng nhanh (gấp đôi theo cách thứ ba, và gấp nhiều lần theo cách thứ tư). Cải tiến được đề xuất ở bài báo này là dung hòa cách biểu diễn thứ hai và hai cách biểu diễn cuối.

Nội dung chủ yếu theo cách biểu diễn của chúng tôi là:

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

- Đưa vào trọng số phân biệt về sự xuất hiện các từ khóa trong trang web đang xét và các trang web láng giềng của nó. Chi tiết hơn, trọng số là khác nhau đối với 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, trọng 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.

- Vector biểu diễn được "chuẩn hóa" theo nghĩa các thành phần của vector là các số nguyên và tổng các thành phần là một hằng số. Như vậy, với vector biểu diễn bất kỳ $X = (X_1, X_2, ..., X_N)$ thì $X_1 + X_2 + \cdots + X_N = C$ (C là hằng số, chúng tôi chọn $C = 100$ theo nghĩa "số phần trăm"). Ngoài tác dụng thuận tiện trong tính toán, giải pháp này còn mang một ý nghĩa là hệ thống không phân biệt vai trò các trang web theo độ dài.

4.2. Xác định độ gần nhau về nội dung các trang web

Như trình bày ở trên, cách biểu diễn vector được chọn nhằm thể hiện nhiều ngữ nghĩa về nội dung của trang web. Dưới đây chúng tôi đưa ra độ đo về tính "tương tự theo nội dung" của hai trang web thông qua một độ đo gần nhau của hai vector biểu diễn. 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 [6]. Giả sử có vector biểu diễn $X = (X_1, X_2, ..., X_N)$ và $Y = (Y_1, Y_2, ..., Y_N)$ thì độ gần nhau $Sm(X, Y)$ của hai vector này là $\cos(X, Y)$ của góc tạo bởi X và Y được tính theo công thức (1):

$$Sm(X,Y) = \cos(X,Y) = \frac{\sum\limits_l X_l * Y_l}{\sqrt{\sum\limits_l X_l^2 \sum\limits_l Y_l^2}}. \quad (1)$$

Khi cài đặt trong Vietseek, chúng tôi tính toán giá trị hạng hiển thị các trang web gần nhau là tổ hợp giữa độ gần nhau theo công thức (1) với giá trị hạng của trang web cần hiển thị (công thức (3) sau Thuật toán 2 tại Mục 4.5).

4.3. Xây dựng vector biểu diễn trong máy tìm kiếm

Trong máy tìm kiếm, nội dung các bảng chỉ mục (chỉ mục nội dung, chỉ mục liên kết, chỉ mục ngược...) cho đầy đủ thông tin để chúng ta xây dựng được hệ thống các vector biểu diễn. Dưới đây là mô tả sơ lược về nội dung này (các thuật toán chi tiết cho việc xây dựng các vector biểu diễn được trình bày trong Mục 4.5).

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. Giả sử đang xem xét trang web P và từ khóa W, nhận được đánh giá xuất hiện của từ khóa W trong P là n_1, tổng đánh giá xuất hiện của từ khóa W trong tất cả các láng giềng có liên kết hai chiều với P là n_2, tổng đánh giá xuất hiện của từ khóa W trong tất cả các trang web láng giềng còn lại là n_3. 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...). Khái niệm này tương tự khái niệm "trọng số xuất hiện" (weight values for all of appearances) từ khóa W trong văn bản D [6]. Chúng tôi tính giá trị n_W tương ứng với thành phần W trong vector biểu diễn trang web P như sau:

$$n_W = [(4 * n_1 + 2 * n_2 + n_3)/7] \quad (1)$$

trong đó kí hiệu [.] chỉ hàm lấy phần nguyên. Theo cách tính này, từ khóa xuất hiện trong chính trang Web có trọng số cao hơn từ khóa xuất hiện trong trang Web láng giềng có cả liên kết đi và đến và cuối cùng từ khóa xuất hiện trong trang Web láng giềng chỉ có một liên kết có trọng số thấp nhất.

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 (2):

$$N_W = \frac{n_W * 100}{\sum\limits_W n_W} \qquad (\text{chú ý} \quad \sum_W N_W = 100). \tag{2}$$

Chú ý rằng, khi cài đặt Vietseek đối với một tổ chức cụ thể, chúng tôi định hướng tới việc cho phép người dùng hệ thống định nghĩa tập từ khóa chuyên ngành và vì thế độ dài vector biểu diễn không lớn.

4.4. Cài đặt trong Vietseek

Để tính được tổng đánh giá xuất hiện (trọng số xuất hiện) của từ khóa trong trang web, cách biểu diễn bổ sung cần coi URL là một đối tượng chính. Xuất phát từ bảng *urlword* lưu trữ các thông tin về các URL, chúng tôi xây dựng vector biểu diễn của trang web.

Phương pháp thực hiện như sau: trong bảng *urlword*, thêm một trường mới, có tên **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 trong bảng 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 lấy

được thông tin về tần số xuất hiện của các từ trong mỗi trang và thông tin về mối liên kết giữa trang đang xét với các trang láng giềng, và từ đó tính được trọng số của mỗi từ. Khi cơ sở dữ liệu được tạo chỉ mục lại (sau khoảng thời gian nhất định) thì giá trị của trường này cũng được tính toán luôn trong quá trình tạo chỉ mục.

Việc thêm trường **content_vector** 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 việc thêm trường mới hoàn toàn không ảnh hưởng tới các hoạt động sẵn có 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. Giải pháp khắc phục của chúng tôi là, với mỗi URL, chúng tôi tạo luôn một danh sách các URL tương tự với nó, tức là gần nhất với nó. 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 một ngưỡng về số lượng (khoảng 100 URL có độ tương tự cao nhất), vì thông thường người sử dụng chỉ quan tâm nhiều nhất đến 20 trang đầu tiên.

4.5. Các thuật toán

Thuật toán 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 ra danh 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)}

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

{hết while (2)}

{Hết Thuật toán 1}

Thuật toán 2. (Tạo danh sách các URL "gần nội dung" ứng với URL)

{Các URL được xếp theo tăng theo chỉ số s : $1, 2, ..., N$, trong đó N là số lượng trang Web trong hệ thống}

1. $I \leftarrow 1$
2. $J \leftarrow I + 1$
3. Tính d_{IJ} = độ gần nhau của URL_I với URL_J
4. If d_{IJ} có thể 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} có thể được đưa vào URL_J
   then Đưa d_{IJ} vào URL_J (bao gồm giá trị d_{IJ} và chỉ số I)
6. $J \leftarrow J + 1$
7. If $J \leq N$
   then Chuyển về 3
8. $I \leftarrow I + 1$
9. If $I < N$
   then Chuyển về 2
10. Kết thúc

{Hết Thuật toán 2}

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

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

Để thuận tiện cho việc tính toán, các $d_{I,J}$ trong một URL được xếp theo giá trị giảm dần và dùng thuật toán chèn nhị phân phần tử $d_{I,J}$ vào danh sách đã được sắp. Nếu vị trí của $d_{I,J}$ vượt quá 100 thì không đưa $d_{I,J}$ vào danh sách.

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

Sử dụng kết quả của Thuật toán 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.

5. KẾT QUẢ THỰC NGHIỆM VÀ BÀN LUẬN

Khi triển khai thử nghiệm, Vietseek đã xây dựng được chỉ mục cho khoảng 3000 site tiếng Việt với khoảng 3 triệu trang web. Khoảng 2,5 triệu từ khóa đã được lưu trữ.

Hiện tại, Vietseek đã có chức năng tìm kiếm theo văn bản của một máy tìm kiếm thông thường (hình 4). Các kết quả tìm kiếm được trả về rất nhanh và chính xác do đã thực hiện được việc tính hạng trang web dựa theo 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 trên. 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.

Những chức năng tìm kiếm hình ảnh, tìm kiếm trang web tương tự theo nội dung với trang web hiện thời theo các thuật toán được đề xuất trên đây còn đang được chúng tôi tích hợp vào Vietseek.

Chúng tôi đang tiếp tục tiến hành những nghiên cứu định hướng tới đề xuất biểu diễn mới trang web tinh túy hơn, chẳng hạn cải tiến biểu diễn trang web dựa trên lý thuyết tập mờ [7], bổ sung chức năng tự phát hiện luật [2] hoặc cung cấp các khung nhìn của Vietseek cho từng lĩnh vực hoạt động của người dùng (khoa học tự nhiên, khoa học xã hội, công nghệ thông tin, kinh doanh...).

Hình 4. Giao diện một trang kết quả tìm kiếm Vietseek

Lời cảm ơn. Chúng tôi chân thành cảm ơn Mạng TTVN On line và Cơ quan VDC1 đã hỗ trợ, giúp đỡ chúng tôi trong việc triển khai thử nghiệm máy tìm kiếm Vietseek.

TÀI LIỆU THAM KHẢO

[1] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan, *Searching the Web*, Technical Report, Computer Science Department, Stanford University, 2000.

[2] Bettina Berendt, *Web Usage Mining, Site Semantics, and the Support of Navigation*, Humboldt University Berlin, Institute of Pedagogy and Informatics, Berlin, Germany, 2000.

[3] Holger Billhardt, Daniel Borrajo, and Victor Maojo, Context vector model for information retrieval, *Journal of American Society for Information Science and Technology (JASIS)* **53** (2002) 236–249.

[4] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan, 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*, Aberta, Canada, July 23-26, 2002, 239–248.

[5] Martin Ester, Hans-Peter Kriegei, and Matthias Schubert, Web site mining: A new way to spot competitors, customers and suppliers in the world wide web, *Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aberta, Canada, July 23-26, 2002, 249–258.

[6] Nguyen Ngoc Minh, Nguyen Tri Thanh, Ha Quang Thuy, Luong Song Van, and Nguyen Thi Van, A knowledge discovery model in fulltext databases, *Proceedings of the First Workshop of International Joint Research: "Parallel Computing, Data Mining and Optical Networks"*, Japan Advanced Institute of Science and Technology (JAIST), Tatsunokuchi, Japan, March 7, 2001, 59–68.

[7] Sen Slattery, "Hypertext classification", Doctoral dissertation (CMU-CS-02-142), School of Computer Science, Carnegie Mellon University, 2002.

[8] Son Doan and Susumu Horiguchi, *A new Text Representation Method using Fuzzy Concepts in Text Categorization*, JAIST Science Reports, 2002.

[9] Thorsten Joachims, Optimizing search engines using clickthrough data, *Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aberta, Canada, July 23-26, 2002, 133–142.

Nhận bài ngày 25 - 8 - 2003
Nhận lại sau sửa ngày 21 - 6 - 2004