# MỤC LỤC

# PHẦN MỞ ĐẦU

Cùng với sự phát triển mạnh mẽ của Internet là một khối lượng khổng lồ dữ liệu được phát sinh, tuy nhiên (theo thông tin từ tập đoàn Oracle) khoảng 90% dữ liệu ở dạng phi cấu trúc hoặc nửa cấu trúc. Nhu cầu khai thác, tìm kiếm thông tin một cách chính xác trên internet đã ngày càng trở nên bức thiết hơn, do đó xuất hiện các hệ tìm kiếm theo từ khoá (cụm từ khoá) như Yahoo, Google ... Tuy nhiên việc tìm kiếm theo từ khoá vẫn chưa đủ để giúp người sử dụng nhanh chóng tìm được trang Web cần thiết vì số lượng kết quả trả lại rất lớn và nhiều khi chỉ là các trang Web ít có liên quan. Vì vậy các hệ thống tìm kiếm cần được cải tiến để ngày càng thông minh hơn. Xuất hiện những hệ hướng tới mục tiêu cụ thể như tra cứu thông tin về các chủ đề y tế, giáo dục, luật pháp, âm nhạc ... Tuy vậy, việc nghiên cứu các giải pháp tìm được các trang thông tin theo một nội dung nào đó sát với yêu cầu người sử dụng vẫn còn nhiều hạn chế. Đã có nhiều mô hình tìm kiếm được đề xuất, song những mô hình lý tưởng về mặt lý thuyết thì lại chưa có tính khả thi khi cài đặt. Do đó, trong các hệ tìm kiếm, người ta tìm cách cải tiến các phương pháp có sẵn để áp dụng trong thực tế. Luận văn này hướng tới việc nghiên cứu, phân tích, đánh giá  một số thuật toán tìm kiếm theo nội dung, từ đó đề xuất phương án cải tiến để nâng cao hiệu quả về tính chính xác của nội dung cũng như về tốc độ.

Từ việc tìm hiểu, đánh giá và phân tích ưu, nhược điểm của các phương pháp tiếp cận khác nhau, dựa theo mục tiêu nâng cao hiệu quả tìm kiếm, luận văn đề xuất giải pháp thực hiện "***Phương pháp biểu diễn ngữ nghĩa lân cận siêu liên kết cho máy tìm kiếm VietSeek***".

Nội dung của luận văn được định hướng vào các vấn đề sau:

1. Mô hình toán học biểu diễn trang văn bản Web,

2. Khái quát các phương pháp tiếp cận trong tìm kiếm trang Web có nội dung tương tự. Đánh giá ưu điểm và nhược điểm của mỗi phương pháp được khảo sát.
3. Đề xuất phương pháp kết hợp để nâng cao hiệu quả trong tìm kiếm trang Web có nội dung tương tự

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

Chương 1 với tiêu đề là ***Tổng quan về các phương pháp biểu diễn và tìm kiếm thông tin trên web*** giới thiệu khái quát về các phương pháp biểu diễn và tìm kiếm trên web.

Tiêu đề của chương 2 là ***Phương pháp biểu diễn trang web theo ngữ nghĩa lân cận siêu liên kết***. Chương này trình bày cơ sở, nội dung của phương pháp được đề xuất và đánh giá phương pháp được đề xuất với các phương pháp khác. Luận văn cũng trình bày chi tiết các lựa chọn được đề xuất trong mỗi bước của phương pháp, từ đó chọn ra giải pháp tốt nhất.

Chương 3 ***Máy tìm kiếm VietSeek và thử nghiệm Thuật toán tìm kiếm theo ngữ nghĩa lân cận siêu liên kết*** giới thiệu kiến trúc logic của máy tìm kiếm VietSeek, thiết kế logic về dữ liệu theo biểu diễn vector và thuật toán tìm kiếm theo nội dung trên cơ sở biểu diễn trang web do luận văn đề xuất. Chương này cũng đề xuất những cải tiến khi áp dụng vào thực tế để nâng cao hiệu suất thực hiện của phương pháp biểu diễn.

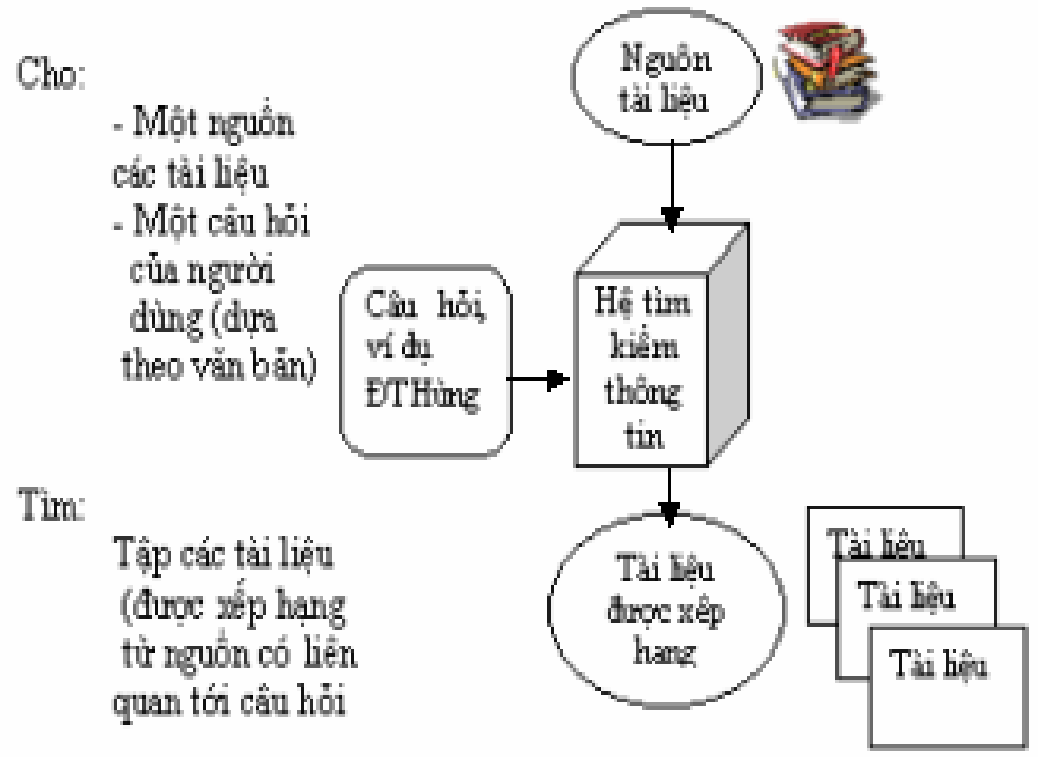
***Phần kết luận*** tổng hợp những kết quả nghiên cứu chính của luận văn và chỉ ra một số hạn chế của luận văn. Đồng thời luận văn đề xuất một số hướng nghiên cứu cụ thể tiếp theo của luận văn.

***Phần phụ lục*** bổ sung một số thông tin chi tiết về việc áp dụng thuật toán cho máy tìm kiếm VietSeek như sơ đồ khối một số module cần bổ sung chức năng, những lệnh bổ sung vào cơ sở dữ liệu của VietSeek.

# CHƯƠNG 1. TỔNG QUAN VỀ TÌM KIẾM THÔNG TIN TRÊN WEB

## 1.1 Giới thiệu về tìm kiếm thông tin

Khai phá dữ liệu trên web (Web Mining) là quá trình khảo sát và phân tích dữ liệu web một cách tự động hoặc bán tự động để phát hiện ra thông tin. Từ thông tin được khai phá, tìm kiếm thông tin (Infomartion Retrieval) trên web là phương pháp để truy cập một cách hiệu quả nhất đến thông tin mà người dùng quan tâm, kỳ vọng cung cấp một tập hợp nhỏ các văn bản gần nhất đến lĩnh vực hoặc chủ đề mà người dùng mong muốn tiếp cận.

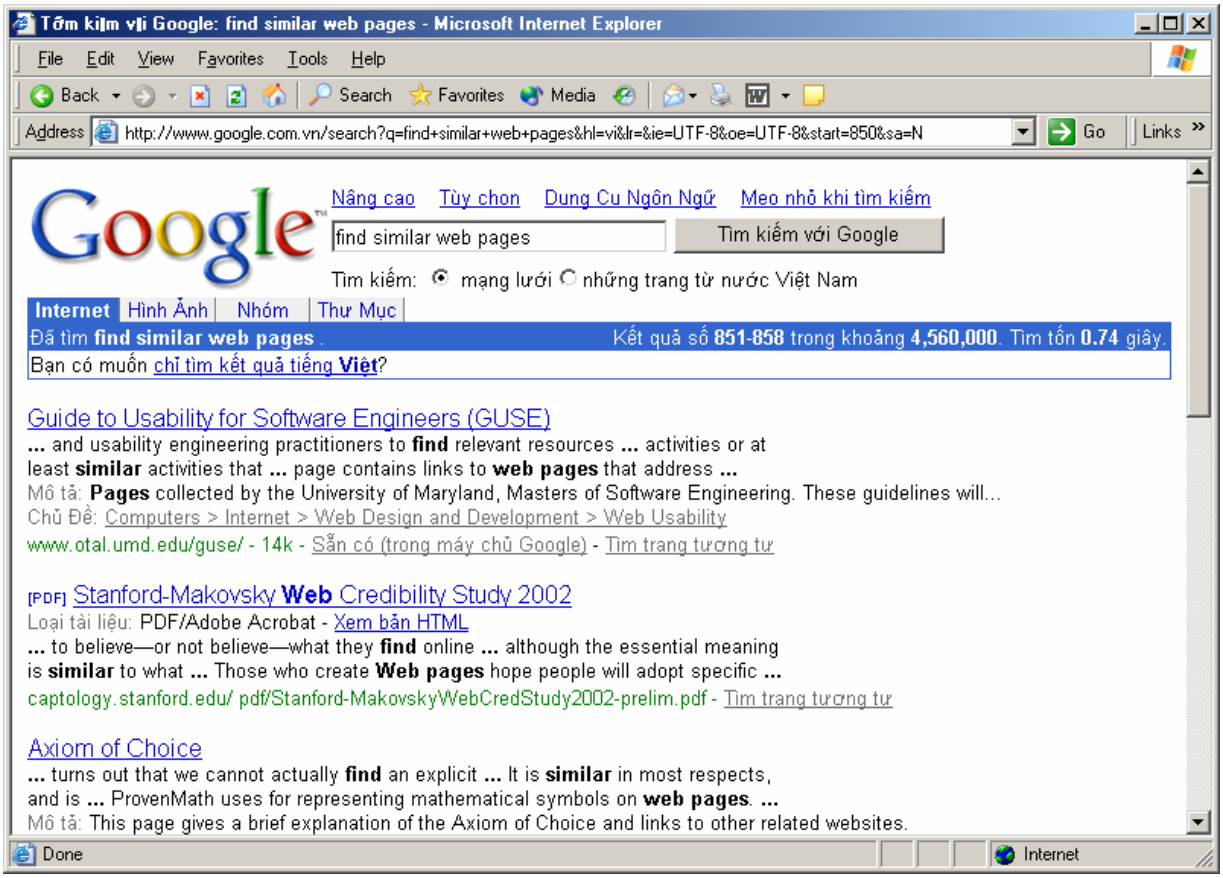



*Hình 1. Tìm kiếm thông tin*

## 1.2 Bài toán tìm kiếm thông tin

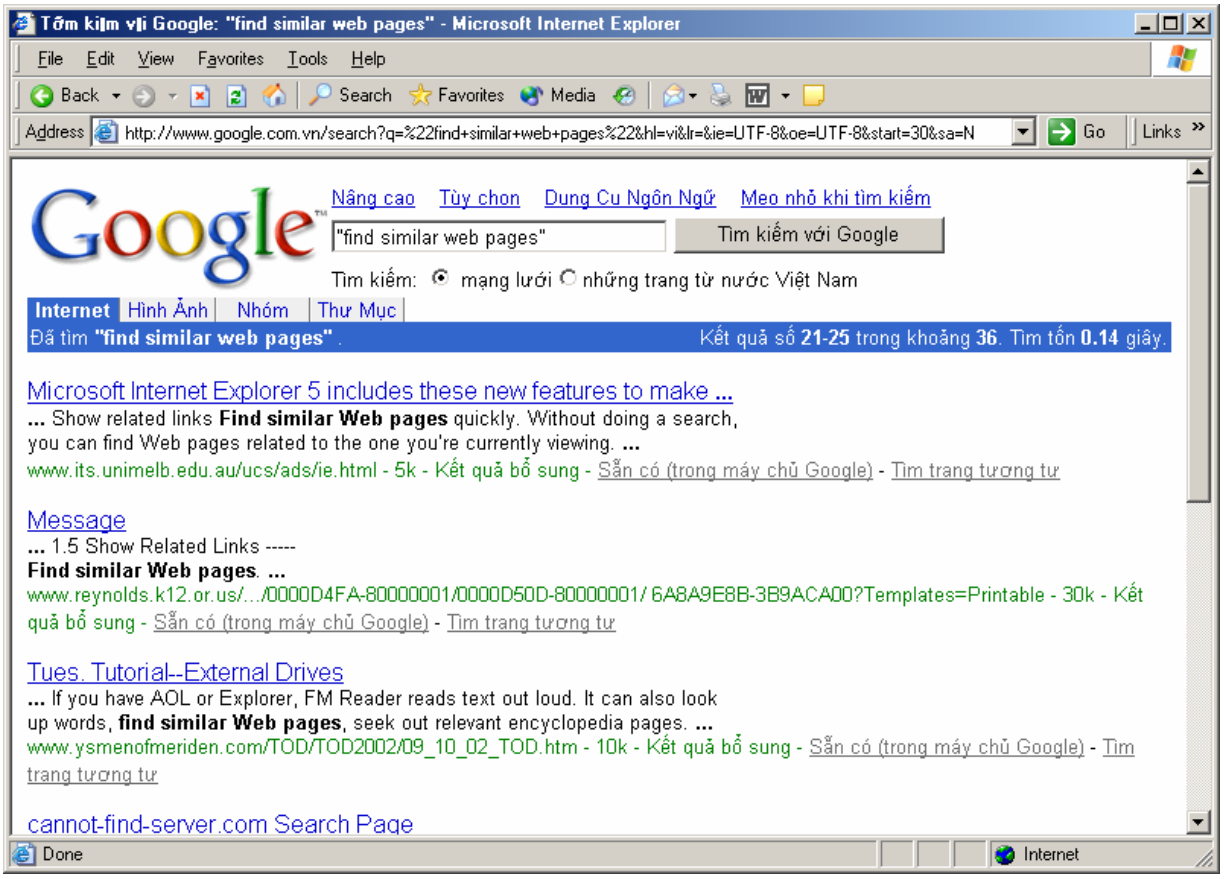
Có 2 bài toán cơ bản trong tìm kiếm thông tin là tìm kiếm theo từ khoá và tìm kiếm theo nội dung. Bài toán tìm kiếm theo từ khoá là bài toán tìm kiếm thông tin theo

các từ khóa do người dùng cung cấp [1][1]. Hệ tìm kiếm sẽ trả về cho người dùng các trang web có chứa những từ khoá trong câu hỏi. Tuy vậy, với số lượng khổng lồ các trang web trên internet như hiện nay thì số lượng kết quả tìm được theo từ khoá là quá lớn. Ví dụ nếu tìm các trang web có từ khoá *find similar web page* thì cho kết quả 858 trang web.



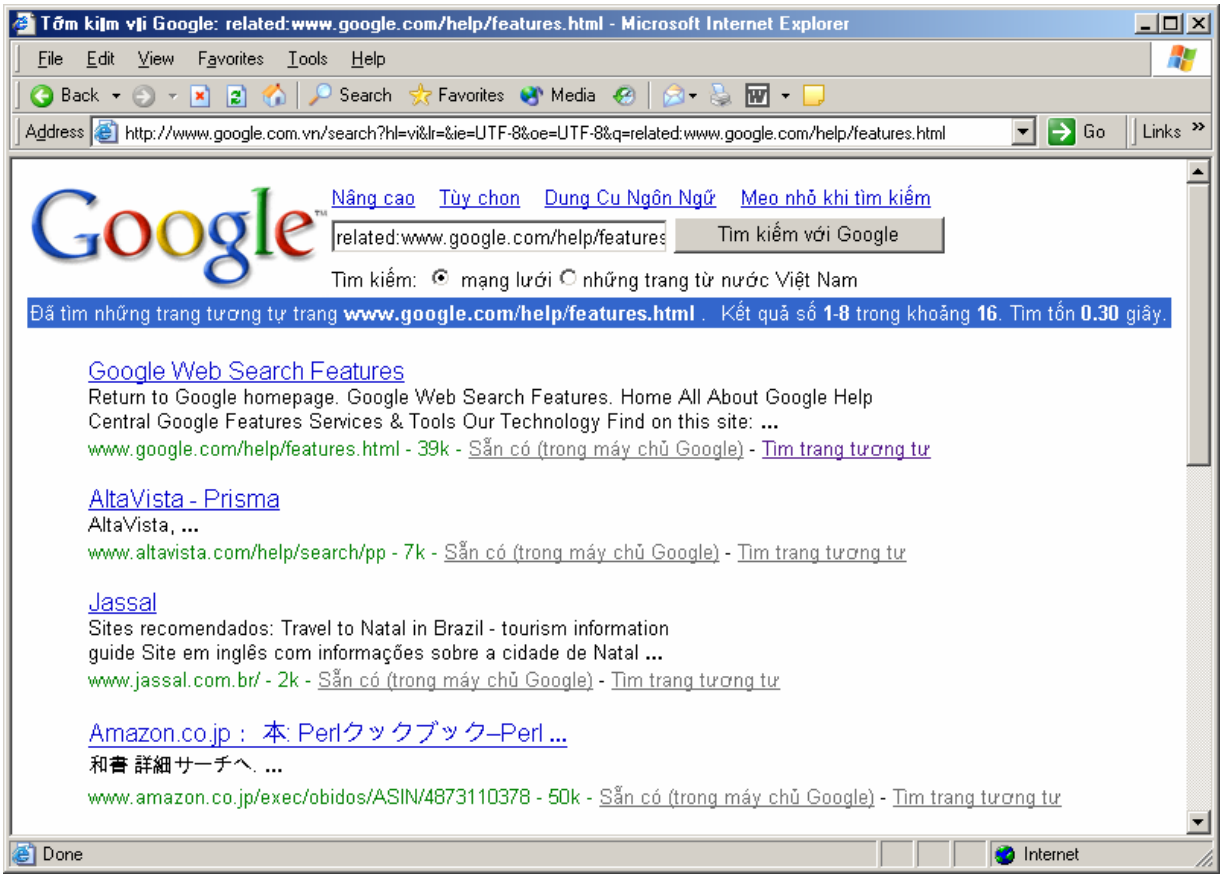

*Hình 2. Tìm kiếm thông tin theo từ khoá*

Bằng cách tìm kiếm theo cụm từ khoá thì số lượng kết quả trả về chính xác hơn, số kết quả trả về là 25 trang web.




*Hình 3. Tìm kiếm thông tin theo cụm từ khoá*

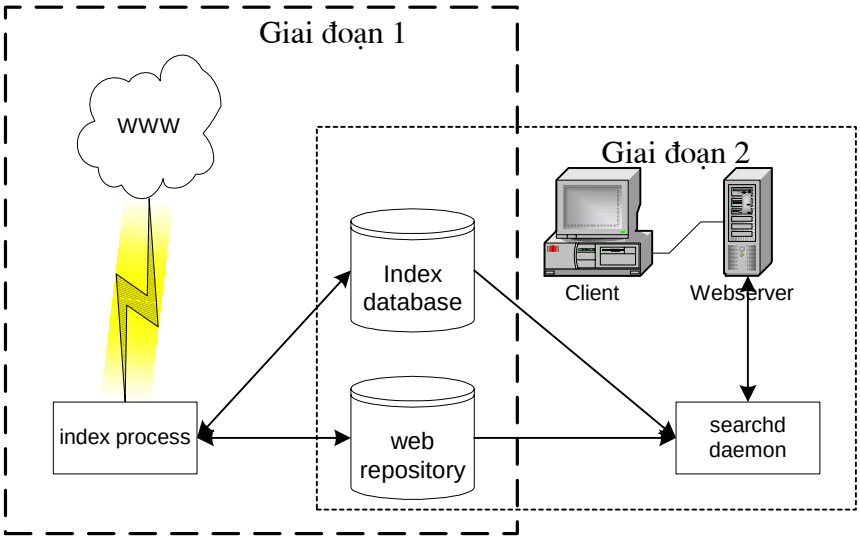
Nếu tìm trang web tương tự với một trang web mẫu thì số lượng kết quả chỉ là 8 trang web.




*Hình 4. Tìm kiếm thông tin theo nội dung một trang web mẫu*

Một cách tiếp cận khác là tìm kiếm theo các site được đề cập trong luận văn của Phạm Thanh Nam [1] vì số lượng các site ít biến động và ít hơn rất nhiều so với các trang web. Tuy vậy, do lượng thông tin ứng với mỗi lĩnh vực đều rất lớn nên vẫn quá khó khăn để tiếp cận các trang văn bản đáp ứng mong muốn với yêu cầu người dùng. Chính vì lý do đó mà các đề tài nghiên cứu những năm gần đây đi sâu về lĩnh vực tìm kiếm theo nội dung tương tự với trang văn bản mẫu như luận văn thạc sĩ của Phạm Thanh Nam năm 2003 [1], luận án tiến sĩ của Seán Slattery năm 2002 [13] hoặc trong một số báo cáo về WWW được tổ chức năm 2002[12], năm 2003. Để đáp ứng các yêu cầu tìm kiếm thông tin của người dùng một cách nhanh nhất, tất cả các giải pháp tìm kiếm thông tin đều chia thành 2 giai đoạn thực hiện tương đối độc lập với nhau

- **Giai đoạn 1: Thu thập và phân tích thông tin về các trang web.**
- **Giai đoạn 2: Xử lý câu hỏi và trả lời**




*Hình 5: Kiến trúc các hệ tìm kiếm thông tin*

Do giai đoạn 1 không tương tác trực tiếp với người dùng nên các thông tin được phân tích một cách đầy đủ nhất để giảm thiểu các phân tích ở giai đoạn sau. Số lượng các trang web được phân tích rất lớn (hàng triệu trang) nên thời gian thực hiện giai đoạn 1 rất lớn (tính bằng giờ) còn thời gian thực hiện giai đoạn 2 là rất nhỏ (tính bằng phần trăm giây).

### 1.2.1 Giai đoạn 1: Thu thập và phân tích thông tin

Các bước xử lý chính:

- ***Tìm duyệt các trang web***. Từ các danh sách địa chỉ ban đầu, bộ phận tìm duyệt sẽ tải trang web và chuyển cho bộ phận phân tích nội dung trang web. Các trang web ban đầu có độ sâu là 0, các liên kết có trong trang web sẽ được bộ phận phân tích ghi nhận lại với độ sâu là 1. Sau khi đã phân tích xong các trang web có độ sâu là 0 thì bộ tìm duyệt tiếp tục tải nội dung các trang web có độ sâu là 1 để phân tích và tìm ra các trang web có độ sâu là

2. Quá trình tải trang web sẽ dừng lại khi đạt đến một độ sâu nhất định nào đó do người dùng đặt tham số như trong VietSeek là 256.

- ***Phân tích và lưu trữ thông tin biểu diễn trang web***. Đây là bước cơ bản quyết định đến chất lượng của các hệ tìm kiếm. Các trang web được phân tích về mặt nội dung để xây dựng thành vector biểu diễn trang web. Các liên kết có trong trang web cũng được ghi nhận lại. Các trang web cũng được đánh giá mối tương quan với các trang khác theo mục tiêu của bài toán, ví dụ như sự tương tự về nội dung so với các trang web khác hoặc phân vào lớp các chủ đề. Toàn bộ thời gian và tài nguyên của các hệ tìm kiếm được sử dụng trong bước này. Do đó bước này cũng được chia thành bài toán nhỏ hơn cần phải giải quyết là *xây dựng cấu trúc biểu diễn thông tin được cung cấp từ các văn bản được phân tích, phân tích cú pháp/ngữ nghĩa, sinh vector biểu diễn, phân lớp văn bản, phân cụm văn bản, phân tích kết quả*. Những nội dung này sẽ được trình bày trong mục 1.3, 1.4 và 1.5 của chương này.
- ***Lưu trữ bản sao trang web***. Để nhanh chóng truy xuất đến nội dung trang web tìm thấy, thông thường các hệ tìm kiếm thường lưu trữ sẵn bản sao các trang web dưới dạng nén cung cấp cho người dùng. Phương pháp nén thường được dùng zip. Việc chọn một kỹ thuật nén thường được cân nhắc giữa tốc độ và tỷ lệ nén. Tỷ lệ nén của zip là 3/1 tuy có nhỏ hơn so với các phương pháp nén khác nhưng tốc độ nén và giải nén của zip lại nhanh đáng kể.

### 1.2.2 Giai đoạn 2: Xử lý câu hỏi và trả lời

Các bước xử lý chính:

- ***Phân tích câu hỏi của người dùng***. Các hệ tìm kiếm thông thường cho phép người dùng tìm kiếm các trang web dưới dạng biểu thức logic, ngoài ra để thuận tiện và nâng cao tính chính xác của câu hỏi, các hệ tìm kiếm

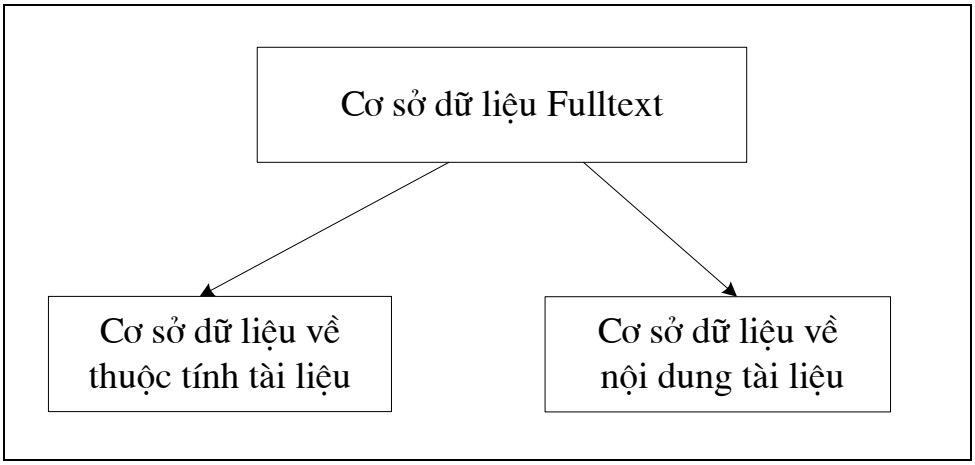
cũng cho phép người dùng đưa vào các điều kiện nâng cao như tìm từ trong chủ đề, tìm các trang theo nội dung của một trang web, tìm theo thời gian xuất hiện, tìm theo ngôn ngữ ..v.v. Câu hỏi của người dùng sẽ được phân tích thành các điều kiện để hệ tìm kiếm có những ứng xử phù hợp.

- ***Định vị các trang web kết quả và xếp hạng***. Dựa trên các điều kiện của người dùng và các trang web đã được phân tích trong giai đoạn "***thu thập và phân tích thông tin"*** hệ tìm kiếm nhanh chóng định vị ra được các trang web kết quả, hơn nữa các trang web cũng được lấy ra theo mức độ tương quan với câu hỏi của người dùng theo một số tiêu chí sắp xếp, ví dụ như thứ tự có xuất hiện các từ khoá trong câu hỏi, mức độ gần với nội dung trang web mẫu. Mức độ chính xác của trang web đối với câu hỏi của người dùng (hạng của trang web) cũng được tính toán và cung cấp cho người dùng. Một số hệ tìm kiếm còn bổ sung thêm tính năng xử lý các phản hồi của người dùng với kết quả để nâng cao độ chính xác cho các lần trả lời sau như ghi nhận số lần truy cập của trang web để tăng độ ưu tiên về hạng của trang web, thay đổi độ tương tự của các trang web đã phân tích, chuyển trang web vào nhóm văn bản có chủ đề chính xác hơn.
- ***Hiển thị nội dung trang web sẵn có***. Người dùng có thể lấy trang web từ địa chỉ được cung cấp bởi hệ tìm kiếm hoặc có thể xem nội dung trang web sẵn có trong kho lưu trữ của hệ tìm kiếm. Thao tác này yêu cầu hệ tìm kiếm giải nén trang web và hiển thị. Thông thường thì hệ tìm kiếm sẽ tô sáng các thành phần có trong câu hỏi của người dùng bằng các màu sắc để người dùng nhanh chóng nhận ra vị trí của chúng trong trang web kết quả.

## 1.3 Mô hình biểu diễn thông tin của văn bản

Cơ sở dữ liệu Fulltext là cơ sở dữ liệu phi cấu trúc biểu diễn thông tin của văn bản mà dữ liệu chứa trong đó bao gồm các nội dung văn bản và các thuộc tính của các 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 văn bản, và phần tập hợp nội dung các văn bản được quản lý [3].




*Hình 6. Mô hình tổ chức của cơ sở dữ liệu Fulltext*

Hiện nay có ba mô hình cơ sở dữ liệu Fulltext điển hình là

1. Mô hình logic
2. Mô hình cú pháp
3. Mô hình vector

Mô hình vector là mô hình được sử dụng phổ biến nhất trong các hệ tìm kiếm hiện nay.

### 1.3.1 Mô hình biểu diễn thông tin theo từ khoá

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* [3]... 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 khoá.

Ví dụ văn bản thứ nhất có nội dung "*VietKey 32-Bit là chương trình hỗ trợ gõ tiếng Việt trong các môi trường Windows 32-Bit của Microsoft*".

Và văn bản thứ 2 "*VietKey có thể nhúng được tiếng Việt trong hầu hết các ứng dụng 16-bit và 32-bit trong môi trường Windows 32-bit"*

Vector biểu diễn văn bản sẽ gồm các thành (từ khoá, tần suất của từ trong văn bản):

| **Từ khoá** | **Vector biểu diễn văn bản 1** | **Vector biểu diễn văn bản 2** |
|---|---|---|
| *16* | 0 | 1 |
| *32* | 2 | 2 |
| *bit* | 1 | 3 |
| *các* | 1 | 1 |
| *có* | 0 | 1 |
| *của* | 1 | 0 |
| *chương* | 1 | 0 |
| *dụng* | 0 | 1 |
| *được* | 0 | 1 |
| *gõ* | 1 | 0 |
| *hầu* | 0 | 1 |
| *hết* | 0 | 1 |
| *hỗ* | 1 | 0 |
| *là* | 1 | 0 |

| | | |
|---|---|---|
| *môi* | 1 | 1 |
| *microsoft* | 1 | 0 |
| *nhúng* | 0 | 1 |
| *thể* | 0 | 1 |
| *tiếng* | 1 | 1 |
| *trình* | 1 | 0 |
| *trường* | 1 | 1 |
| *trợ* | 1 | 0 |
| *trong* | 1 | 2 |
| *ứng* | 0 | 1 |
| *và* | 0 | 1 |
| *vietkey* | 1 | 1 |
| *việt* | 1 | 1 |
| *windows* | 1 | 1 |

*Bảng 1. Vector biểu diễn văn bản*

### 1.3.2 Mô hình biểu diễn thông tin theo nội dung

Đối với bài toán tìm kiếm theo nội dung, phần lớn các giải pháp tìm kiếm thông tin đều lựa chọn mô hình vector. Có ba phương pháp tiếp cận trong việc xác định từ khoá trong vector biểu diễn văn bản.

1. Phương pháp biểu diễn theo nội dung văn bản: Từ khoá trong vector biểu diễn văn bản ***u*** là những từ có mặt trong văn bản ***u***.
2. Phương pháp tiếp cận theo liên kết: Từ khoá trong vector biểu diễn văn bản ***u*** là những từ khóa có trong định danh của những văn bản ***v*** có liên kết đến văn bản u.

3. Phương pháp tiếp cận theo ngữ nghĩa lân cận liên kết: Từ khoá trong vector biểu diễn văn bản ***u*** là những từ xuất hiện trong cửa sổ ngữ nghĩa lân cận liên kết từ những văn bản ***v*** đến văn bản ***u***.

Luận văn đề cập tới giải pháp kết hợp các phương pháp tiếp cận trên đây.

## 1.4 Phân tích cú pháp và ngữ nghĩa

Trong trang web không chỉ có thông tin thể hiện nội dung mà còn các thông tin phụ trợ như các comment, các đoạn mã, các thẻ HTML. Do đó cần phải tách lọc thông tin mà trang web biểu diễn, tách thông tin về các liên kết. Cần phải xác định từ gốc của từ biểu diễn văn bản, xác định vị trí của từ trong văn bản, xác định các biên của đoạn văn theo cú pháp câu (dấu ngắt câu) hoặc biên theo chủ đề đoạn văn (ngắt đoạn, ngắt bảng, ngắt trang).

## 1.5 Phân lớp văn bản

Phân lớp văn bản được xem như là quá trình gán các văn bản vào một hay nhiều lớp văn bản đã được xác định trước. Sau khi được phân lớp, các văn bản sẽ được đánh chỉ số đối với  từng lớp tương ứng. Người dùng có thể yêu cầu hệ tìm kiếm giới hạn số kết quả trong một chủ đề hoặc lớp văn bản mong muốn. Phân lớp văn bản có thể thực hiện tự động bằng các phương pháp cây quyết định [3], mạng Bayer, máy vector trợ giúp. Ngoài ra, các trang web có thể thể được phân lớp bằng thủ công nhờ sự tình nguyện của người dùng trên internet như thư mục chủ đề các trang web ODP (Open Directory Project) [17].

## 1.6 Phân cụm văn bản

Phân cụm văn bản là việc tự động sinh ra các lớp văn bản dựa vào sự tương tự của các văn bản. Các lớp văn bản ở đây là chưa biết trước, người dùng có thể chỉ yêu cầu số lượng các lớp cần phân loại, hệ sẽ đưa ra các văn bản theo từng tập hợp, từng cụm, mỗi tập hợp chứa các văn bản tương tự nhau.

## 1.7 Khai thác thông tin cấu trúc web

Trong tìm kiếm thông tin trên web, các trang web đã chứa đựng thông tin nửa cấu trúc, đó chính là các liên kết giữa các trang web. Thông thường, các web đem lại nhiều thông tin sẽ được trích dẫn nhiều do đó có thể khai thác thông tin liên kết giữa các trang web để đánh giá trọng số của trang web như Slattery đã đề xuất [13].

## 1.8 Khai thác thông tin sử dụng web

Thông tin sử dụng web được chứa trong một tập hợp các file liên quan được định sẵn trên những máy chủ web. Mục đích của việc khai thác thông tin sử dụng web để phát hiện ra những mẫu dữ liệu có ý nghĩa được sinh ra trong những giao dịch khách/chủ. Thông thường các dữ liệu đó ở phía máy chủ là access logs, referrer logs, agent logs và phía máy trạm là cookies. Một dạng thông tin về người dùng web là các profile của họ.

Trong tìm kiếm thông tin, các trang web đem lại nhiều thông tin thường được truy cập nhiều hơn các trang web khác trong cùng chủ để. Do đó tần suất truy cập (thông tin sử dụng web) của các trang web cũng là một thành phần cần xem xét khi đánh giá trọng số của trang web.

Tuy nhiên, với mỗi người dùng thì có thể có tập hợp các trang web được yêu thích của riêng mình. Người sử dụng có thể yêu cầu mà hệ tìm kiếm cho phép giới hạn các trang kết quả trong một tên miền nào đó như .com.vn và những tham số như vậy có thể được định nghĩa trong các profile.

### KẾT LUẬN CHƯƠNG 1

Trong chương này, luận văn đã giới thiệu tổng quát bài toán tìm kiếm thông tin trên web và các phương pháp tìm kiếm thông tin trên web:

1. Các phương pháp tìm kiếm theo từ khoá gồm mô hình cú pháp, mô hình logic và mô hình vector. Các phương pháp này đã được nghiên cứu khá kỹ lưỡng và tiêu biểu nhất là mô hình vector.

2. Các phương pháp tìm kiếm theo nội dung đang được nghiên cứu hiện nay là tìm kiếm theo nội dung toàn văn, theo liên kết và theo ngữ nghĩa lân cận liên kết.

Luận văn đã phân tích nguyên tắc hoạt động cũng như ưu điểm và nhược điểm của mỗi phương pháp. Từ những phân tích trên, luận văn sẽ trình bày phương pháp biểu diễn văn bản mới trong chương 2 và đề xuất thuật toán tìm kiếm theo nội dung trong chương 3.

# CHƯƠNG 2. PHƯƠNG PHÁP BIỂU DIỄN TRANG WEB THEO NGỮ NGHĨA LÂN CẬN SIÊU LIÊN KẾT

## 2.1 Giới thiệu

Mục tiêu của việc tìm kiếm trang Web tương tự là cho phép người sử dụng tìm những trang Web tương tự với trang Web mẫu. Về cơ bản, khi đưa ra một văn bản, một thuật toán tìm kiếm tương tự phải cung cấp danh sách thứ tự của các văn bản tương tự với văn bản mẫu.

Trong chương này, luận văn sẽ trình bày một số phương pháp tiếp cận của giải pháp tìm kiếm theo nội dung và đánh giá chất lượng của mỗi phương pháp. Trên cơ sở phương pháp biểu diễn trang web theo ngữ nghĩa lân cận siêu liên kết [12], luận văn đề xuất một số bổ sung, cải tiến thành giải pháp tìm kiếm theo nội dung. Căn cứ trên những kết quả đánh giá qua thử nghiệm, giải pháp tìm kiếm theo nội dung do luận văn đề xuất được xem là có chất lượng tốt hơn so với các phương pháp đã khảo sát khác và được áp dụng cho máy tìm kiếm VietSeek.

Thuật toán tìm kiếm gồm hai bước:

1. Tiền xử lý các trang web: Tạo vector biểu diễn trang web. So sánh các trang web trong cùng chủ đề của ODP để tính toán sẵn độ tương tự các trang web.
2. Thực hiện tìm kiếm thông tin, chỉ đơn thuần là thao tác định vị và đọc dữ liệu sẵn có trong cơ sở dữ liệu.

Phương pháp này đã được thử nghiệm bằng tập dữ liệu lớn và chứng tỏ tính khả thi của nó. Các vấn đề chính cần phải giải quyết trong phương pháp biểu diễn ngữ nghĩa lân cận siêu liên kết là:

1. Xác định phương pháp đánh giá chất lượng cho độ đo tương tự.
2. Xác định mô hình vector biểu diễn trang web.
3. Xác định nghĩa độ đo tương tự với mô hình biểu diễn đã chọn

4. Khảo sát các thành phần của vector biểu diễn trang web
5. Xây dựng các thuật toán:

- Thuật toán tạo vector biểu diễn trang web
- Thuật toán tính độ tương tự giữa các trang web
- Thuật toán tìm kiếm trang web tương tự

Các vấn đề 1, 2, 3 và 4 sẽ được trình bày trong chương 3 của luận văn. Vấn đề 5 có trong đề xuất phương án thực hiện cho máy tìm kiếm VietSeek trong chương 4.

## 2.2 Phương pháp đánh giá chất lượng độ đo tương tự

### 2.2.1 Chọn phương pháp đánh giá

Khi khảo sát các cách tiếp cận để tìm ra được một giải pháp tìm kiếm thông tin tốt nhất thì cần thiết phải có một phương pháp đánh giá chất lượng cho các mỗi phương án. Chất lượng xếp hạng trang web của máy tìm kiếm thường được đánh giá bởi người dùng dựa trên các độ đo về khoảng cách và đặc trưng của văn bản. Tuy nhiên, sử dụng trực tiếp sự đánh giá của người dùng thường tốn thời gian và công sức, nên điều đó không thích hợp cho những nghiên cứu mà đòi hỏi sự so sánh đánh giá của nhiều tham số.

Trong văn bản về phân cụm, nhiều phương pháp đánh giá chất lượng tự động đã được đề xuất [20]. Steinback [20] chia những phương pháp này thành 2 lớp tổng quát. Phương pháp đánh giá sử dụng các độ đo chất lượng nội tại, như độ tương tự trung bình, chỉ ra chất lượng của một cụm văn bản được đề xuất dựa hoàn toàn trên nội tại hình học và thống kê, không dựa trên một tập chân lý nền có sẵn. Phương pháp đánh giá dựa trên các độ đo chất lượng ngoài, như độ đo entropy, kiểm tra sự tương quan của một cụm với một tập chân lý nền có sẵn. Đây cũng là phương pháp đánh giá được sử dụng để đo chất lượng của một phương án.

Cây phân loại chủ đề các trang web ODP [17] được xây dựng và phổ dụng trên Internet. Trong ODP, các trang web được sắp phân lớp theo các chủ đề và thứ tự của nó

trong chủ đề có thể coi là hạng của trang web trong chủ đề tương ứng. Độ đo tương tự của các văn bản tương ứng với một phương án biểu diễn thông tin về văn bản cung cấp một tập thứ tự. Do đó, có thể dùng ODP làm tập thứ tự nền để kiểm tra chất lượng xếp hạng của một độ đo tương tự. Các độ đo đánh giá độ tương quan giữa hạng của trang web trong ODP và hạng của trang web tương ứng với độ đo tương tự được xây dựng được coi như là sự đánh giá gián tiếp của người dùng về chất lượng xếp hạng. Tất nhiên là không thể sử dụng trực tiếp ODP làm thứ tự cho giải pháp tìm kiếm vì nó chỉ chứa một bộ phận các trang web có mặt trên Internet.

### 2.2.2 Xác định thứ tự nền trong ODP

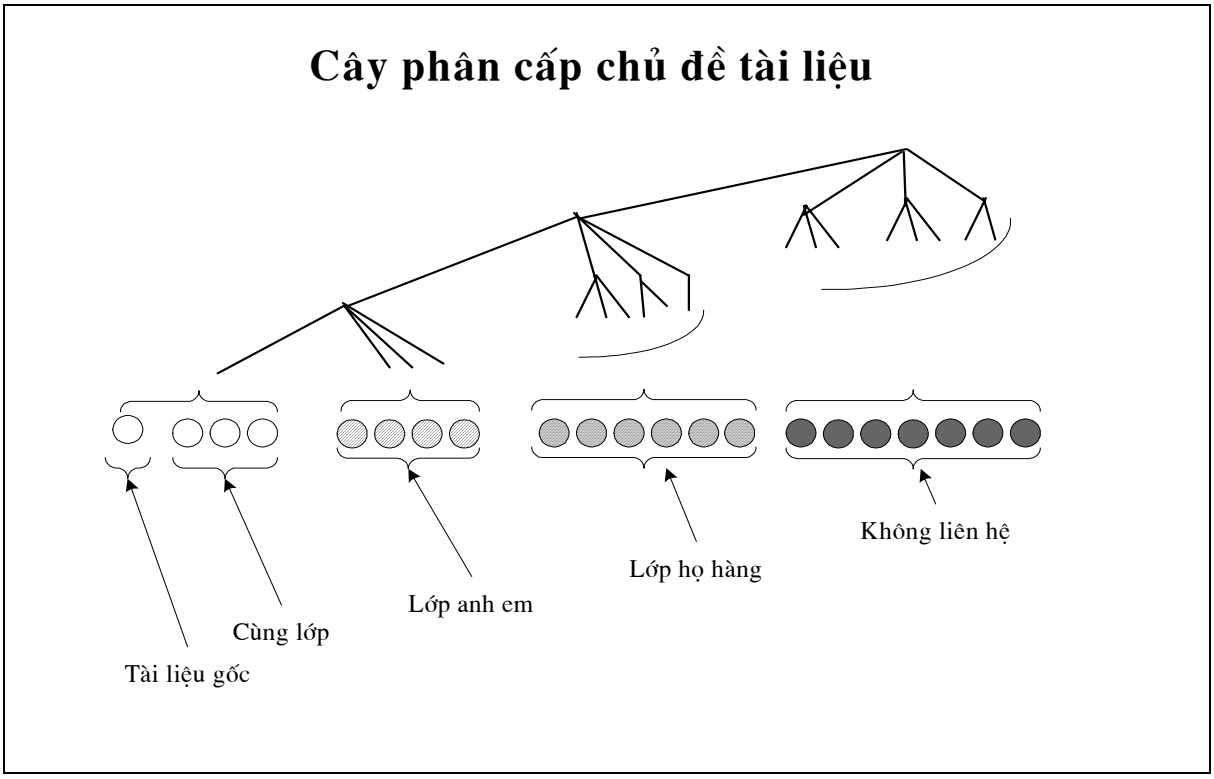
Dựa theo việc phân lớp sẵn có các văn bản của ODP, dễ thấy rằng các văn bản cùng một lớp (cùng chủ đề) sẽ gần nhau về nội dung hơn so với các văn bản ở lớp khác (chủ đề khác). Ví dụ, một văn bản trong lớp recreation/aviation/un-powered thường có nội dung gần với các văn bản khác cùng lớp so với các văn bản không thuộc lớp đó. Hơn nữa, văn bản này lại "gần" với các văn bản khác của lớp recreation/aviation hơn là các văn bản ở khu vực khác của cây.

Tất nhiên là vị trí của văn bản trong cây phân loại chủ đề không thể mang lại sự chính xác về nội dung một cách tuyệt đối. Ví dụ trong chủ đề recreation/autos, hầu hết gần với các văn bản ở shopping/autos hơn là các văn bản ở recreation/smoking. Tuy vậy có thể căn cứ vào đó để xây dựng một tiêu chuẩn cho độ đo tương tự vì các cây phân loại chủ đề đã có sự sắp xếp độ tương tự về mặt nội dung.

Để chuẩn hoá khái niệm khoảng cách từ một văn bản này đến một văn bản khác trong cây, khoảng cách tương quan đã được xác định như dưới đây.

#### ❑ Khoảng cách tương quan

Khoảng cách tương quan $d_f(s,d)$ từ một văn bản mẫu ***s*** đến một văn bản ***d*** khác trong một cây phân lớp là khoảng cách từ lớp chứa ***s*** đến lớp có khoảng cách gần nhất chứa cả ***s*** và ***d***.




*Hình 7. Khoảng cách họ hàng của một văn bản mẫu trong cây phân cấp chủ đề*

Tuy nhiên trong các hệ thống thực tế, độ sâu của các lớp văn bản được giới hạn là 3 và bỏ qua những văn bản có độ sâu lớn hơn (cũng có ít sự liên hệ hơn). Do đó, chỉ có 4 giá trị có thể cho *khoảng cách họ hàng* được định nghĩa như dưới đây (minh họa trong Hình 7):

**Khoảng cách 0**

*Cùng lớp – Những văn bản cùng lớp (cùng một chủ đề lá)*

**Khoảng cách 1**

*Anh em – Những văn bản có chung lớp cha*

**Khoảng cách 2**

*Họ hàng – Những văn bản ở cùng lớp ông bà*

**Khoảng cách 3**

*Không liên hệ – Những văn bản ở lớp khác những lớp nói trên*

Từ cây phân lớp chủ đề ODP, dễ nhận thấy "*Về trung bình, sự tương tự nhau giữa các văn bản với văn bản mẫu là đơn điệu giảm với khoảng cách họ hàng của những văn bản đó*"

Do đó, với bất kì một văn bản mẫu nào trong cây thư mục cũng có thể tìm được thứ tự tương tự bộ phận đối với tập các văn bản khác trong cây thư mục. Chú ý rằng, ở đây không đưa ra bất kì diễn giải về mặt số học nào cho những giá trị khoảng cách này mà chỉ dựa trên nguyên lý đơn điệu đã được phát biểu: về mặt trung bình, đối với một văn bản mẫu cho trước thì văn bản cùng lớp là tương tự hơn so với văn bản cùng lớp cha, và văn bản cùng lớp cha lại tương tự hơn so với văn bản cùng lớp ông bà, ....

- **Tập (quan hệ) t**hứ tự khoảng cách họ hàng $\prec_{d_f(s)}$ **cho mọi văn bản liên quan đến văn bản mẫu** *s* **là :**

$$\prec_{d_f(s)} = \{(a,b) \mid d_f(s, a) < d_f(s,b)\} \qquad (1)$$

Đối với bất kì văn bản mẫu ***s***, tập thứ tự bộ phận này là rất yếu vì hầu hết các cặp văn bản đều không thể so sánh được (do tính thô sơ của khoảng cách họ hàng). Điều quan trọng là tập thứ tự này cho biết những văn bản nào có nội dung gần  nội dung của văn bản mẫu hơn so với các văn bản khác. Đặc biệt, tập thứ tự này tạo ra sự khác biệt giữa các văn bản tương tự nhau và các văn bản khác không liên quan với văn bản mẫu, trong khi đó các văn bản không liên quan thường chiếm phần lớn các văn bản trong kho dữ liệu. Những văn bản có khoảng cách xa thì không có sự khác biệt về thứ tự (tất cả các văn bản có khoảng cách lớn hơn hoặc bằng 3 thì đều có khoảng cách là 3). Tập thứ tự thu được từ ODP với một văn bản mẫu ***q*** được coi là tập thứ tự nền $\prec_t$.

Tất nhiên, như đã trình bày ở đầu mục này, nguyên tắc độ tương tự là đơn điệu giảm theo khoảng cách họ hàng không phải lúc nào cũng được đảm bảo. Tuy nhiên, về mặt trung bình, một hệ thống xếp hạng các trang web phù hợp hơn với thứ tự nền được coi là cho kết quả tốt hơn.

### 2.2.3 So sánh sự tương quan giữa các tập thứ tự

Như vậy, từ một văn bản mẫu ***s*** trong ODP có thể xác định được một tập thứ tự nền cho các văn bản trong ODP so với ***s***. Tập thứ tự nền này được sử dụng để đánh giá chất lượng xếp hạng độ đo tương tự được xây dựng theo một độ tương quan nào đó giữa hai tập thứ tự. Độ đo tương tự nào có độ tương quan với tập thứ tự nền càng cao thì được xem là có chất lượng xếp hạng càng tốt hơn các độ đo tương tự khác. Dù tồn tại một số phương pháp đánh giá độ tương quan giữa các tập thứ tự, tuy nhiên, đa số các phương pháp sử dụng hệ số tương quan Spearman để so sánh hai tập thứ tự. Độ đo theo hệ số tương quan này này là phù hợp nhất để so sánh hai thứ tự hoặc rất ít hoặc không có ràng buộc nào, và giá trị của nó tương ứng với hệ số Pearson $\rho$ [20]. Tuy nhiên, có hai thách thức lớn khi sử dụng hệ số tương quan Spearman để đánh giá chất lượng xếp hạng. Thứ nhất, có rất nhiều ràng buộc lớn đối với tập thứ tự nền. Hai là, vùng chắc chắn của tập thứ tự được quan tâm nhiều hơn những những vùng khác (vùng văn bản tương tự với văn bản mẫu). Do đó, độ đo tương quan Kruskal-Goodman $\Gamma$ [4] (hệ số tương quan $\Gamma$**,** hệ số Gama) là phù hợp hơn, và vì vậy trong luận văn, chúng tôi sử dụng nó để đánh giá chất lượng độ đo tương tự.

❑ Xác định hệ số $\Gamma$ **cho hai tập thứ tự**

Cho hai tập thứ tự $\prec_a$ và $\prec_b$ đối với một tập các tài liệu. Một cặp văn bản (***x***,***y***) mà có thứ tự trong cả $\prec_a$ và $\prec_b$ thì gọi cặp văn bản (***x***,***y***) là phù hợp với $\prec_a, \prec_b$ hoặc $\prec_a, \prec_b$ là phù hợp nhau tại (***x***,***y***). Gọi n là tổng số cặp tài liệu, m là số cặp phù hợp với $\prec_a, \prec_b$. Khi đó hệ số tương quan (hệ số chất lượng xếp hạng) giữa $\prec_a$ và $\prec_b$ được xác định bởi công thức:

$$\Gamma(\prec_a, \prec_b) = 2 \times [m/n] - 1 \qquad (2)$$

Chỉ có một số cặp tài liệu quyết định đến giá trị của $\Gamma$ bởi khi so sánh hai tập thứ tự chỉ xét đến những cặp tài liệu có thứ tự (được xếp hạng) trong cả hai tập thứ tự.

Xét trường hợp khi một trong hai tập thứ tự trên là tập thứ tự nền. Trong trường hợp đó, nếu tất cả các tập văn bản trong thứ tự nền đều đúng thứ tự theo độ đo tương tự thì $\Gamma = 1$ và trường hợp này là hoàn hảo. Nếu $\Gamma = 0$ chứng tỏ tập thứ tự được cung cấp theo độ đo tương tự là ngẫu nhiên. Nếu $\Gamma = -1$ chứng tỏ tập thứ tự được cung cấp bởi độ đo tương tự rất tồi, hoàn toàn không phù hợp với tập thứ tự nền. Với hai tập thứ tự $\prec_a$ và $\prec_b$ mà $\Gamma(\prec_a, \prec_t)$ khác $\Gamma(\prec_b, \prec_t)$ thì tập thứ tự nào có giá trị $\Gamma$ lớn hơn sẽ được coi là có chất lượng tốt hơn (gần với thứ tự nền hơn).

### 2.2.4 Miền của tập thứ tự

Với một cây thư mục chủ đề như ODP, một văn bản mẫu ***s*** và một độ đo tương tự ***sim***, chúng ta có thể xây dựng 2 tập thứ tự cho các văn bản trong thư mục: thứ tự nền $\prec_{d_f(s)}$, và thứ tự của độ đo tương tự $\prec_{sim(s)}$. Độ đo tương quan $\Gamma$ giữa hai tập thứ tự sẽ cho biết chất lượng của độ đo tương tự (thông qua thứ tự nền). Tuy nhiên, cần phải đánh giá khả năng xếp hạng được khảo sát qua các văn bản kết quả. Để tính được tập thứ tự cho tất cả các tài liệu, thông tin trạng thái của $\Gamma$ được mở rộng bằng cách lặp ***s*** cho tất cả các văn bản, tính tổng tất cả các cặp phù hợp và không phù hợp, sau đó chia cho tổng số cặp.

Để cho kết quả chính xác hơn thì cần phải tính toán ba miền của giá trị $\Gamma$ để làm rõ hơn về các miền khác nhau của khoảng cách tương quan. Mỗi miền của $\Gamma$ dựa trên tỉ lệ giữa cặp có thể so sánh được cho *một kiểu nhất định* nào đó. Các kiểu miền của $\Gamma$ là:

**$\Gamma$ -Anh em:**

Chỉ tính toán cho các cặp văn bản ***($d_1$, $d_2$)*** mà ***$d_1$*** cùng lớp với văn bản mẫu và ***$d_2$*** thuộc lớp anh em với văn bản mẫu.

**$\Gamma$-Họ hàng:**

Chỉ tính toán cho các cặp văn bản ***($d_1$, $d_2$)*** mà ***$d_1$*** cùng lớp với văn bản mẫu và ***$d_2$*** ở có cùng họ hàng với văn bản mẫu.

**Γ- Không liên hệ:**

Chỉ tính toán cho các cặp văn bản $(d_1,d_2)$ mà $d_1$ cùng lớp với văn bản mẫu và $d_2$ lớp văn bản khác.

Để thấy rõ sự khác biệt, trong phần dưới đây chỉ ra một trường hợp tốt nhất khi độ đo tương tự cho tập thứ tự gần nhất với thứ tự nền với văn bản mẫu và trường hợp tồi nhất với văn bản mẫu.

## ❑ Văn bản mẫu

http://www.aabga.org

Tiêu đề: American Assoc. of Botanical Gardens and Arboreta

Chủ đề văn bản mẫu: /home /gardens/clubs_and_associations

## ❑ Trường hợp độ đo tương tự phù hợp nhất với tập thứ tự nền Γ= 0.53

Độ đo tương tự trong trường hợp này có sử dụng kích thước cửa sổ liên kết là 32, phương pháp lược bớt từ cùng gốc, lược bớt từ dừng, có sử dụng khoảng cách từ khoá và tần suất từ khoá.

| Thứ tự | Độ tương tự sim | Loại chủ đề |
|---|---|---|
| 1 | 0.16 | /home/gardens/clubs_and_associations |
| 2 | 0.15 | /home/gardens/clubs_and_associations |
| 5 | 0.13 | /home/gardens/clubs_and_aasociations |
| 10 | 0.11 | /home/gardens/plants |
| 20 | 0.10 | /home/gardens/clubs_and_aasociations |
| 60 | 0.07 | /home/gardens/plants |
| 100 | 0.06 | /hone/apartnent_living/gardening |

*Bảng 2. Tập thứ tự với độ đo tương tự tốt nhất*

### ❑ Trường hợp độ đo tương tự ít phù hợp nhất với tập thứ tự nền Γ= 0.30

Độ đo tương tự trong trường hợp này có sử dụng kích thước cửa sổ = 0, không lọc từ cùng gốc, không sử dụng tần suất khoá.

| Thứ tự | Độ tương tự sim | Loại chủ đề |
|---|---|---|
| 1 | 0.17 | /reference/libraries/independent_libraries |
| 2 | 0.15 | /home/gardens/clubs_and_aasociations |
| 5 | 0.14 | business/industries/construction_and_maintenance |
| 10 | 0.14 | /business/industries/agriculture_and_forestry |
| 20 | 0.13 | /recreation/travel/reservations |
| 50 | 0.13 | /recreation/travel/reservations |
| 100 | 0.13 | business/industries/construction_and_maintenance |

*Bảng 3. Tập thứ tự với độ đo tương tự tối nhất*

Ba thành phần giá trị Γ cho phép đánh giá hiệu quả khác nhau về độ đo với những vùng khác nhau của tập thứ tự. Thành phần Γ-anh em giúp nâng cao chất lượng của độ đo tương tự để độ đo này khuyếch đại độ tương tự của những văn bản cùng lớp (là các văn bản gần nhau nhất về nội dung). Thành phần Γ-không liên hệ giúp nâng cao chất lượng của độ đo tương tự để độ đo này làm yếu đi độ tương tự của những văn bản không liên hệ (là các văn bản xa văn bản nhất về nội dung).

## 2.3 Định nghĩa mô hình vector biểu diễn thông tin văn bản

Mô hình biểu diễn thông tin của các trang web được sử dụng là mô hình vector do mô hình này đảm bảo được tìm kiếm theo từ khoá như các hệ tìm kiếm truyền thống và dễ dàng cải tiến các thành phần của vector để biểu diễn thông tin theo nội dung.

### 2.3.1 Vector biểu diễn thông tin văn bản

Mô hình biểu diễn thông tin về văn bản bằng vector (trong các cấu trúc dữ liệu) được áp dụng nhiều trong các hệ tìm kiếm trên thực tế. Văn bản Web ***u*** được trình diễn bằng một vector là tập hợp từ khoá và trọng số tương ứng (còn được gọi là túi từ – bag of words)

$$B_u = \{(w_u^1, f_u^1),...,(w_u^k, f_u^k)\} \quad (3)$$

trong đó $w_u^i$ là từ có nghĩa (từ khoá: keyword / term) được sử dụng để thể hiện ***u*** (ví dụ từ có nghĩa được tìm thấy trong nội dung và cửa sổ lân cận liên kết của ***u***, hoặc liên kết đến ***u***), và $f_u^i$ là trọng số tương ứng.

### 2.3.2 Lựa chọn từ khoá biểu diễn

Từ khoá để biểu diễn thông tin về văn bản được chọn sau khi loại bỏ các chú thích, mã lệnh Javascript, thẻ HTML, và các kí tự không phải là chữ cái. Một danh sách các từ dừng cũng được sử dụng theo định nghĩa trong máy tìm kiếm VietSeek.

Với cách tiếp cận dựa trên liên kết, cần phải xác định có bao nhiêu từ bên trái và bên phải một liên kết. $\boldsymbol{A_{vu}}$ (neo liên kết từ trang ***u*** đến trang ***v***) sẽ bao gồm trong $\boldsymbol{B_u}$. Trong mọi trường hợp, các từ trong liên kết của $\boldsymbol{A_{vu}}$ được bao gồm như là tiêu đề của văn bản ***u***. Các phương pháp để xác định biên cửa sổ liên kết như sau được trình bày như dưới đây.

#### ❑ Phương pháp biên cửa sổ cố định

Kích thước cửa sổ cố định là W, với ý nghĩa nó luôn chứa W từ bên trái và W từ bên phải của neo liên kết $\boldsymbol{A_{vu}}$. Tập các giá trị của $W \in \{0, 4, 8, 16, 32\}$. Lý do để chọn các giá trị trên để thuận lợi trong các đánh giá vì chúng là bội số của 2. Giá trị tối đa của cửa sổ là 32 và một câu văn trong văn bản thông thường có tối đa 32 từ, do đó giá trị này đảm bảo lấy trọn vẹn một câu văn trong phần liên kết.

### ❑ Phương pháp phân tích cú pháp

Chúng ta sử dụng các câu, đoạn văn và kỹ thuật phát hiện vùng HTML để giới hạn động khu vực lân cận $A_{vu}$ mà chứa trong $B_u$. Các đặc điểm chính của văn bản mà có khả năng khoanh vùng cửa sổ là biên của một đoạn văn bản, biên của ô trong bảng, biên của một danh sách và các dấu ngắt cứng theo sau biên của các câu. Kết quả của kỹ thuật này thu được cửa sổ khá hẹp với trung bình khoảng 3 từ lân cận theo mỗi hướng.

### ❑ Phương pháp phân tích chủ đề

Chúng ta sử dụng một kỹ thuật đơn giản trong việc ước chừng biên của chủ đề tại chỗ biên của khu vực. Các đặc điểm chính để xác định biên là bắt đầu của tiêu đề, kết thúc danh sách, kết thúc bảng. Một trường hợp đặc biệt là văn bản được soạn trên nhiều vùng, mỗi vùng được bắt đầu với một tiêu đề mô tả và gồm một danh sách các url trong chủ đề được nêu. Khu vực được tìm theo chủ đề có kích thước trung bình khoảng 21 từ mỗi bên của neo liên kết.

## 2.3.3 Lược bớt từ khoá

### ❑ NoStem- bớt từ dừng

Các từ khoá biểu diễn thông tin của văn bản chính là các từ xuất hiện trong văn bản. Trong văn bản có các từ chỉ dùng để biểu diễn cấu trúc câu chứ bản thân nó không có nghĩa, chẳng hạn như liên từ, giới từ (ví dụ “thì”, “là”...) và được gọi là từ dừng. Do đó, nếu từ mới được phát hiện qua phân tích cú pháp nằm trong danh sách từ dừng thì loại bỏ từ đó.

### ❑ Stem - Lược từ cùng gốc

Đối với một số tiếng nước ngaòi (tiếng Anh và một số tiếng khác) các từ khoá biểu diễn nội dung văn bản được chuyển thành từ nguyên gốc theo thuật toán Porter [21] nhất thể các hình thái của một từ. Nếu nguyên gốc của từ nằm trong danh sách các nguyên gốc của từ dừng thì cũng loại bỏ từ đó.

### ❑ StopStem - Lược bớt gốc từ dừng

Như trên đã nói, với nhiều ngôn ngữ nước ngoài, nhiều từ trong ngôn ngữ được xây dựng từ một nguyên gốc từ. Các từ khoá biểu điễn thông tin của văn bản chính là các từ xuất hiện trong văn bản. Nếu nguyên gốc của từ khoá mà nằm trong danh sách các nguyên gốc của từ dừng thì từ khoá bị loại bỏ. Phương pháp này có ích đối với các trường hợp từ không có ý nghĩa được phát hiện chính xác hơn với từ nguyên gốc.

### 2.3.4 Xác định trọng số của từ khoá

Một trong các thành phần quan trọng đối với trọng số từ khoá là phương pháp chuẩn hoá số lần xuất hiện của từ khoá trong văn bản. Một số phương pháp thường dùng được giới thiệu dưới đây.

### ❑ Phương pháp dựa trên tần số từ mục (TF-Term Frequency)

Các giá trị của các từ khoá được tính dựa trên số lần xuất hiện của các từ khoá trong văn bản. Gọi $tf_{ij}$ là số lần xuất hiện của từ khoá $t_i$ trong văn bản $d_j$, khi đó $w_{ij}$ được tính bởi công thức:

$$w_{ij} = \sqrt{tf_{ij}} \ or \ w_{ij} = 1 + \log(tf_{ij}) \text{ or } w_{ij} = tf_{ij} \qquad (4)$$

### ❑ Phương pháp dựa trên tần số văn bản nghịch (IDF - Inverse Document Frequency)

Gọi m là số lượng các văn bản, *df* là số lượng văn bản có chứa từ khoá. Khi đó trọng số được tính bởi công thức sau:

$$w_{ij} = \log\frac{m}{df_i} = \log(m) - \log(df_i) \qquad (5)$$

### ❑ Phương pháp TF*IDF

Phương pháp này là tổng hợp của hai phương pháp TF và IDF, giá trị của ma trận trọng số được tính như sau:

$$w_{ij} = \begin{cases} \left[1 + \log(tf_{ij})\right]\log(\frac{m}{df_i}) \text{ nếu } tf_{ij} \geq 1. \\ 0 \qquad\qquad\qquad\qquad \text{nếu } tf_{ij} = 0. \end{cases} \tag{6}$$

Phương pháp *TF.IDF* nhằm mục đích khuyếch đại trọng số của các từ khoá có số lần xuất hiện cao trong văn bản. Khi tìm thông tin theo các từ khoá thì các văn bản có số lần xuất hiện từ khoá nhiều hơn thì sẽ có thứ tự cao hơn. Ngược lại, các phương pháp không đơn điệu lại nhằm mục đích khuyếch đại trọng số của các từ khoá có ít văn bản chứa nó. Các từ khoá mà có ít văn bản đề cập đến chứng tỏ đó là các vấn đề chuyên biệt như là các tên hiếm gặp, lĩnh vực chuyên sâu, vấn đề mới xuất hiện ...v.v. Sự khuyếch đại này trong thực tế sẽ tốt cho các yêu cầu tìm thông tin theo từ khoá chuyên biệt và văn bản có chứa từ khoá chuyên biệt sẽ có thứ tự cao hơn các văn bản khác.

Vấn đề đáng quan tâm là sự tương tự giữa các văn bản, nghĩa là xét chung cả nội dung của văn bản chứ không phải xét riêng một vài từ khoá (hoặc cụm từ khoá). Vì vậy các từ khoá có tần suất cao và thấp đều có ảnh hưởng không tốt đến độ đo tương tự. Từ nhận xét trên, phương pháp chuẩn hoá từ khoá trong độ đo tương tự sẽ làm giảm bớt trọng số của các từ khoá có tần suất cao và tần suất thấp [21].

Một thành phần của trọng số từ khoá cần phải xem xét là khoảng cách giữa vị trí của từ khoá đối với vị trí của liên kết. Trong phương pháp tiếp cận theo liên kết cho một kích thước cửa sổ, thay vì xem xét mọi từ khoá gần với neo liên kết $A_{vu}$ như nhau, trọng số của từ khoá phụ thuộc vào khoảng cách từ khoá đến neo liên kết (những từ trong liên kết có khoảng cách là 0). Thực nghiệm cho thấy rằng, khi áp dụng phương pháp chuẩn hoá trọng số từ khoá có sử khoảng cách vị trí từ khoá đối với neo liên kết làm cho chất lượng của độ đo tương tự tăng lên (giá trị của $\Gamma$)

## 2.4 Định nghĩa độ đo tương tự

Độ đo được chúng ta sử dụng để đo độ tương tự của tập hợp từ của hai văn bản là hệ số Jaccard. Hệ số Jaccard của 2 tập A và B được định nghĩa

$$sim_J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (7)$$

Hệ số Jaccard được mở rộng để áp dụng cho vector biểu diễn văn bản như sau.

Gọi $f_a(w_i)$ là trọng số từ khoá $w_i$ trong vector biểu diễn văn bản A, $f_b(w_i)$ là trọng số từ khoá $w_i$ trong vector biểu diễn văn bản B. Nếu một từ khoá mà chỉ xuất hiện trong một văn bản thì trọng số của nó trong văn bản còn lại là 0. Khi đó:

$$|A \cap B| = \sum \min(f_a(w_i), f_b(w_i)) \text{ trong đó } w_i \in A \text{ và } w_i \in B \quad (8)$$

$$|A \cup B| = \sum \max(f_a(w_i), f_b(w_i)) \text{ trong đó } w_i \in A \text{ hoặc } w_i \in B \quad (9)$$

## 2.5 Đánh giá chất lượng xếp hạng đối với mỗi phương pháp xây dựng vector

Chất lượng xếp hạng của độ đo tương tự qua các phương pháp lựa chọn từ khoá và trọng số từ khoá được đánh giá bằng hệ số tương quan $\Gamma$. Trong kết quả thực nghiệm [12], 300 cặp văn bản mẫu nằm ở ba lớp của ODP [17] được sử dụng làm tập thứ tự nền. Có 51,469 trang văn bản có liên quan đến 300 cặp văn bản mẫu trong số 42 triệu trang web của Stanford WebBase được sử dụng làm tập dữ liệu [21]. Tập các trang web được thử nghiệm đã được liên kết bởi gần một triệu trang trong kho dữ liệu và chúng cũng được sử dụng để hỗ trợ quá trình khảo sát chất lượng các độ đo tương tự.

Do đồ thị các thành phần của $\Gamma$ có dạng tương tự nhau nên trong phần minh hoạ trình bày đồ thị của $\Gamma$-anh em thể hiện chất lượng xếp hạng các văn bản gần nhau về nội dung trong tập thứ tự nền.

Tuy trong một vài đồ thị chỉ cho thấy chất lượng của độ đo tương tự được cải thiện có vẻ là rất ít (khoảng hai con số sau dấu phẩy thập phân). Tuy nhiên, mỗi đồ thị chỉ ra hiệu quả đối với một thành phần của độ đo tương tự, do đó khi kết hợp tất cả các thành phần với nhau thì sự khác biệt cũng trở nên đáng kể.

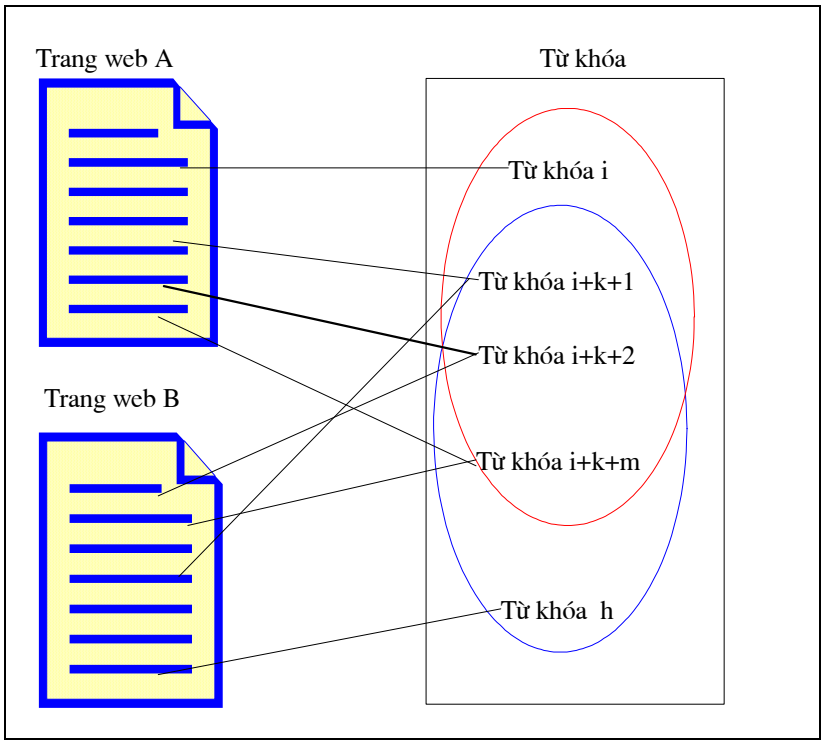
| Chú thích trong biểu đồ | Mô tả |
|---|---|
| Hệ số Γ | Hệ số tương quan giữa tập thứ tự nền và tập thứ tự cung cấp bởi độ đo tương tự |
| Thuần nội dung | Từ khoá biểu diễn văn bản là các từ trong nội dung toàn văn của văn bản được xét |
| Thuần liên kết | Từ khoá biểu diễn văn bản là các liên kết đến văn bản được xét |
| Cửa sổ liên kết | Từ khoá biểu diễn văn bản là các từ khoá xuất hiện trong cửa sổ lân cận liên kết của văn bản được xét |
| w32 | Kích thước cửa sổ liên kết cố định là 32 |
| w16 | Kích thước cửa sổ liên kết cố định là 16 |
| w8 | Kích thước cửa sổ liên kết cố định là 8 |
| w4 | Kích thước cửa sổ liên kết cố định là 4 |
| w0 | Kích thước cửa sổ liên kết cố định là 0 |
| Ngữ nghĩa | Phân tích biên cửa sổ liên kết động theo ngữ nghĩa |
| Cú pháp | Phân tích biên cửa sổ liên kết động theo cú pháp |

*Bảng 4: Bảng chú giải đầy đủ cho chú thích trong các biểu đồ*

### 2.5.1 Đánh giá chất lượng đối với cách chọn từ khoá

***Cách tiếp cận theo nội dung*** chỉ xét đến nội dung toàn văn của trang web. Ưu điểm của các tiếp cận này là mọi trang web đều được đối xử bình đẳng với nhau mà không phụ thuộc và số lượng trích dẫn hay thời gian xuất hiện. Tuy nhiên, nhược điểm của phương pháp này là ý nghĩa nội dung của trang web chỉ dựa vào nội dung văn bản do tác giả trang web cung cấp, mà bỏ qua những quan điểm của tác giả đối với những

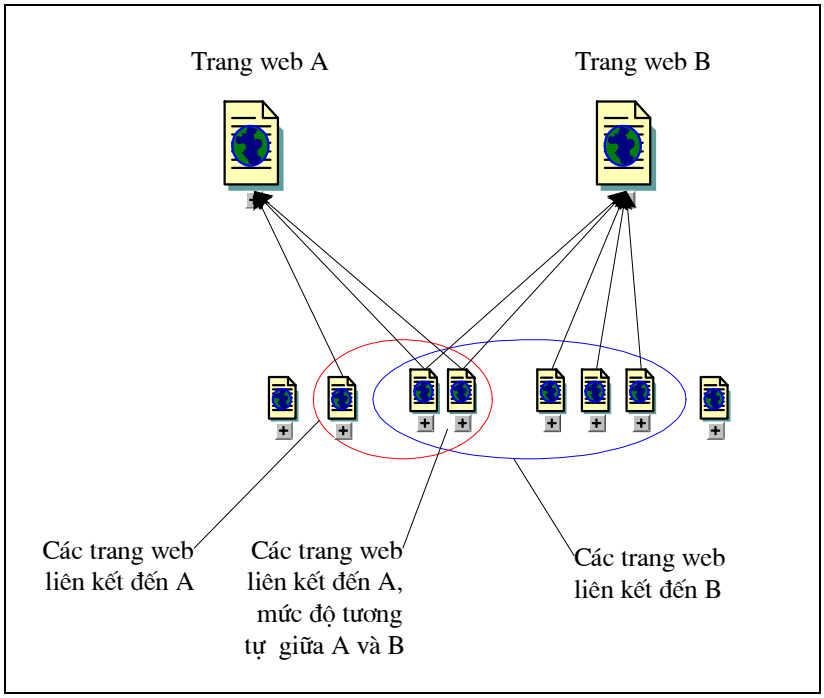
trang web khác (các trích dẫn từ các trang web khác) nên trọng tâm của trang web bị dàn trải trên toàn bộ trang web. Cách tiếp cận này cũng đòi hỏi phải xử lý vấn đề ngôn ngữ như từ đồng nghĩa, từ dừng, cú pháp, ngôn ngữ khác nhau ...




*Hình 8. Cách tiếp cận theo nội dung toàn văn*

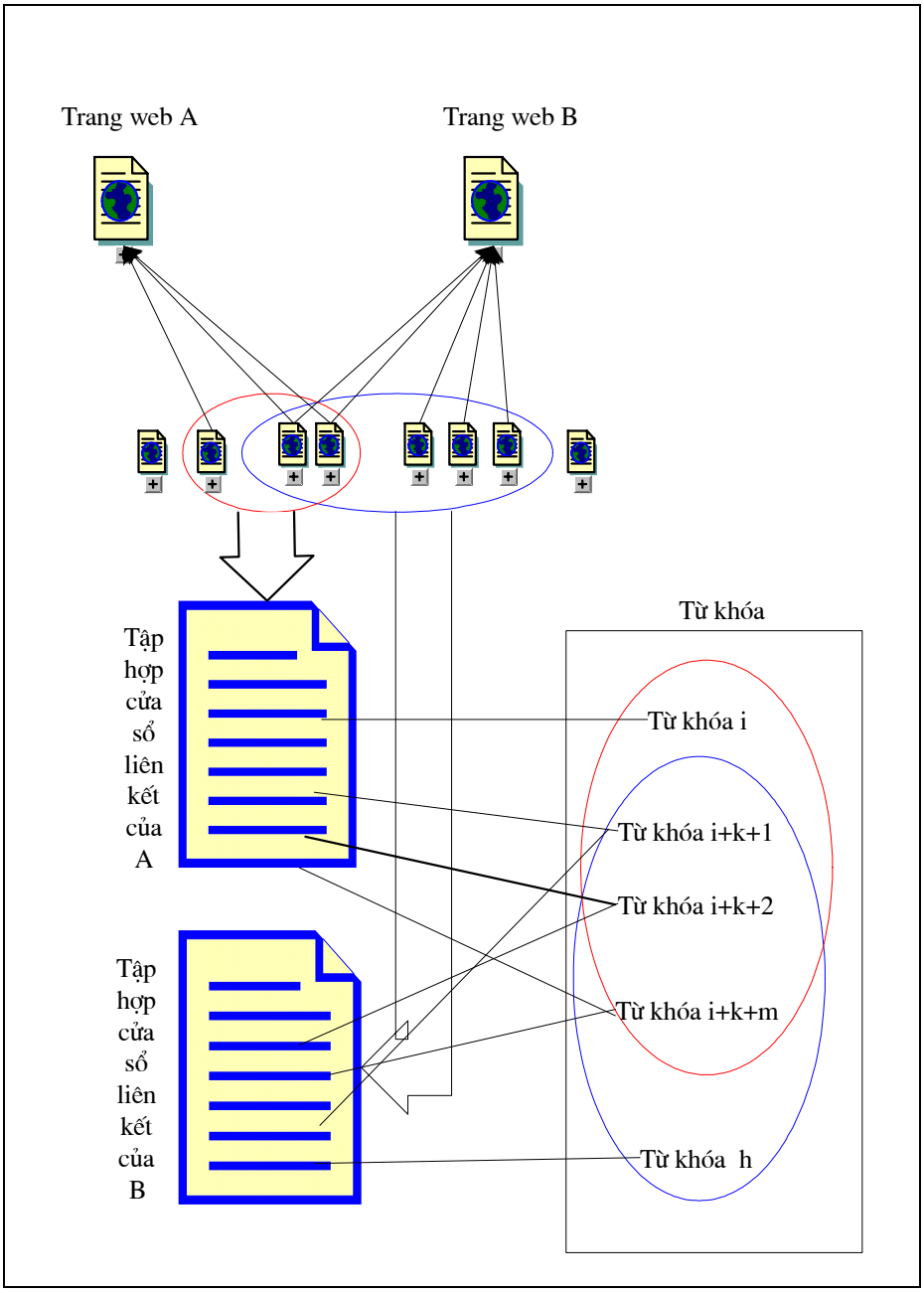
***Cách tiếp cận theo liên kết 0*** chỉ xét số lượng các trang web liên kết đến, nghĩa là mức độ được trích dẫn của trang web. Ưu điểm của cách tiếp cận này là các đánh giá các trang web theo lợi ích thông tin do trang web mang lại vì trang trang web càng có ích thì càng có nhiều trang web trích dẫn đến nó. Một ưu điểm khác nữa của cách tiếp cận này là không phải xử lý vấn đề về ngôn ngữ vì các trang web cùng chủ đề tuy ngôn ngữ khác nhau (ví dụ tin tiếng Anh, tin tiếng Việt) vẫn có thể được trích dẫn như nhau (vì cùng một nguồn thông tin đối với cả hai ngôn ngữ). Tuy nhiên, nhược điểm của cách tiếp cận này là các trang web mới xuất hiện thì không được đối xử bình đẳng với các trang web khác. Do đó, cách tiếp cận này thường xuất hiện các tình huống trực giao

các văn bản, nghĩa là các văn bản có nội dung toàn văn là giống nhau nhưng tập các văn bản đồng thời trích dẫn đến cả hai văn bản lại rất ít (hoặc không trùng nhau).




*Hình 9. Cách tiếp cận theo liên kết*

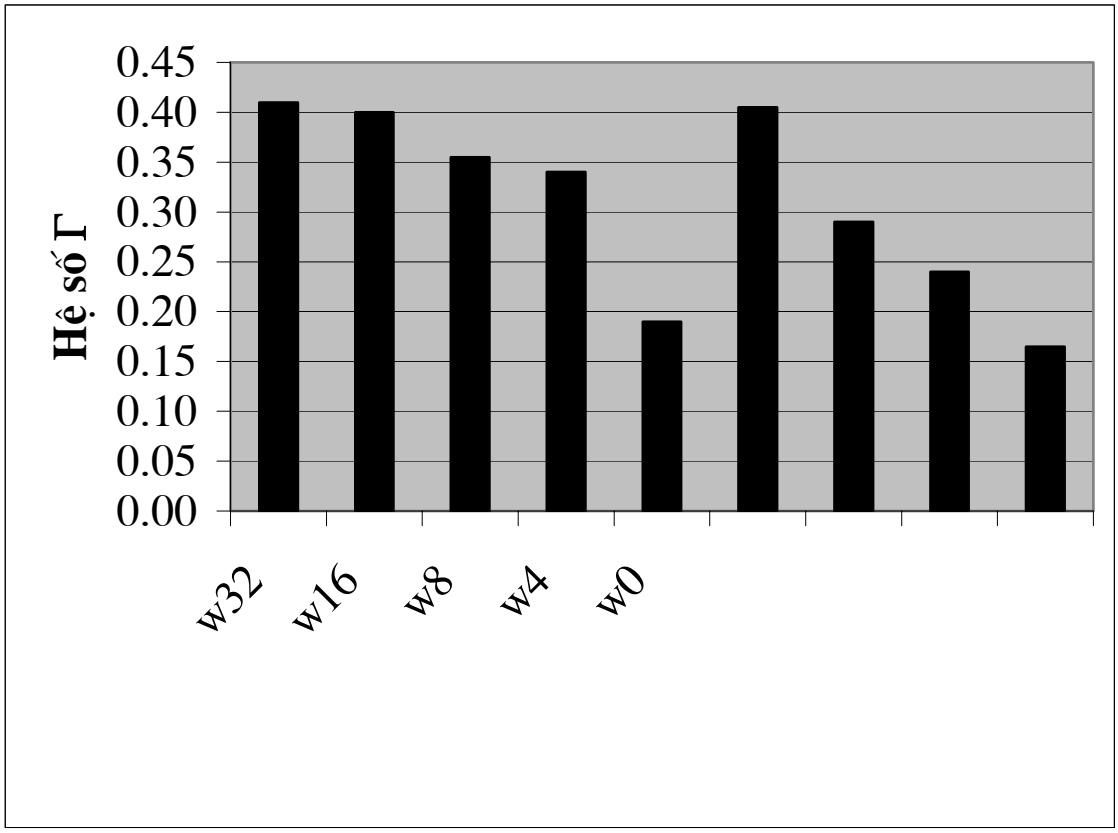
***Cách tiếp cận theo ngữ nghĩa lân cận liên kết***, theo đó từ khoá trong vector biểu diễn văn bản là những từ khóa xuất hiện trong lân cận vị trí liên kết, được hiểu như là cửa sổ liên kết. Các tiếp cận này có ưu điểm là thông tin trong cửa sổ liên kết thường được tạo bởi con người tóm tắt thông tin về văn bản được liên kết đến. Cách tiếp cận này không chỉ quan tâm đến số lượng của các liên kết mà còn quan tâm đến chất lượng của liên kết, nghĩa là thông tin gì được thể hiện trong mỗi liên kết. Nếu hai văn bản có cùng tập liên kết đến nhưng các văn bản trong tập liên kết đến có nhiều chủ đề. Giả sử tập liên kết đến trang web A vì chủ đề của nó là thể thao, tập liên kết đến B vì chủ đề của nó là chính trị. thì trang web A vẫn khác trang web B. Ngược lại, nếu hai trang có tập liên kết của chúng lại trực giao với nhau nhưng cửa sổ ngữ nghĩa lân cận liên kết vẫn thể hiện về cùng một chủ đề thì hai trang web này vẫn tương tự nhau. Nhược điểm của phương pháp này vẫn là vấn đề phải xử lý ngôn ngữ.




*Hình 10: Cách tiếp cận theo cửa sổ liên kết*

Biểu đồ dưới đây thể hiện kết quả đánh giá chất lượng xếp hạng của độ đo tương tự với các cách tiếp cận chọn từ khoá cho vector biểu diễn văn bản. Kết quả cho thấy cửa sổ ngữ nghĩa lân cận liên kết cố định lớn luôn cho kết quả tốt hơn, nhưng cửa sổ

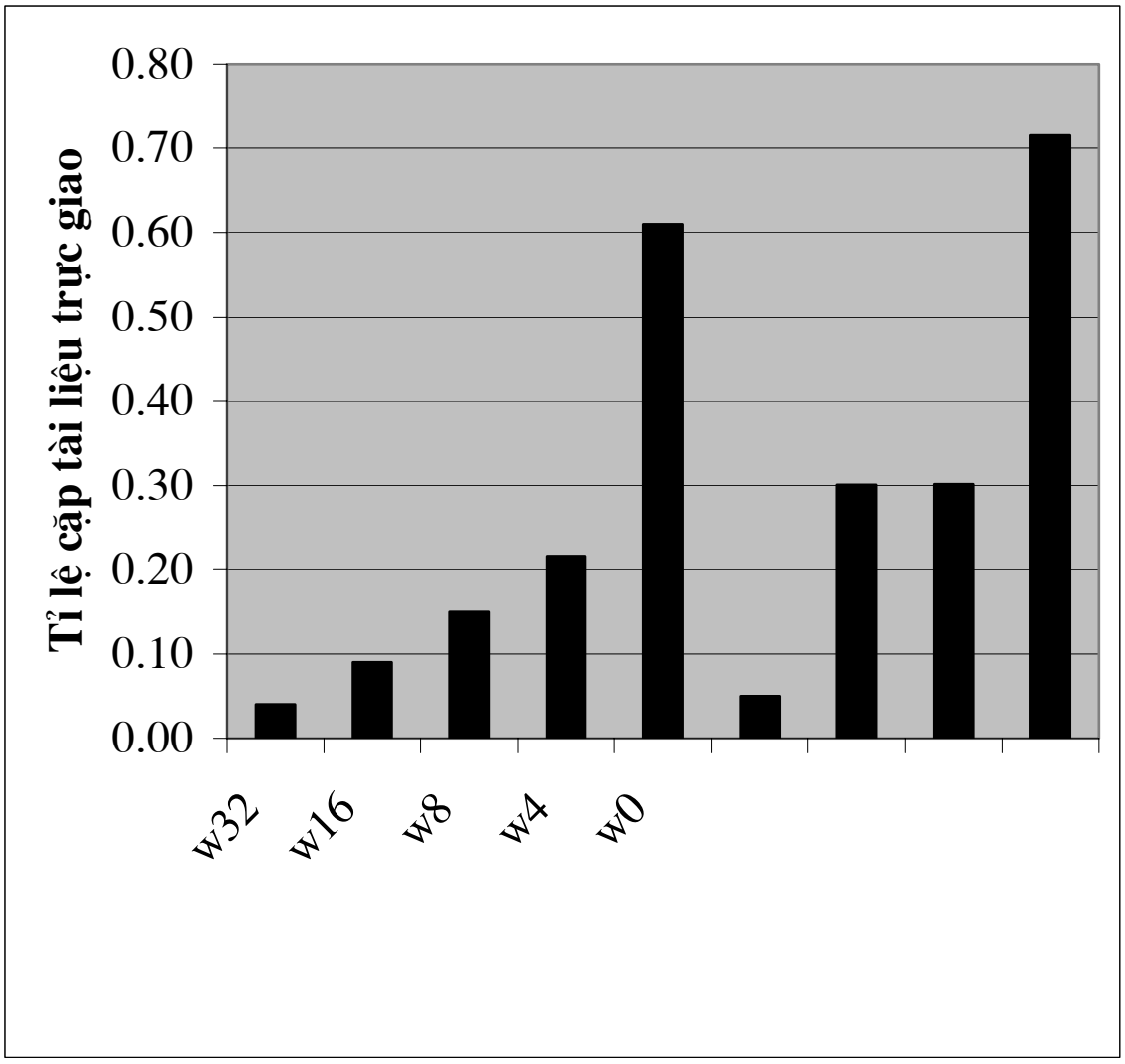
động của chủ đề cũng đạt được kết quả tương tự mà kích thước trung bình của cửa sổ lại nhỏ hơn. Do đó, các từ khoá được lựa chọn là từ khóa trong cửa sổ liên kết động có biên cửa sổ được phát hiện theo phương pháp phân tích ngữ nghĩa.




*Hình 11. Biểu đồ hệ số Gamma với các phương pháp chọn từ khoá.*

Cửa sổ nhỏ với thuần liên kết cho kết quả trong tập hợp từ với có khả năng trực giao lớn, làm cho độ tương tự khó được xác định

Kết quả cho thấy cách tiếp cận dựa trên ngữ nghĩa lân cận liên kết sử dụng cửa sổ lớn cho kết quả tốt nhất. Điều này có vẻ như có mâu thuẫn với mong muốn cửa sổ liên kết nhỏ để giảm bớt sự có mặt của các từ khoá ít có nghĩa xuất hiện trong tập hợp từ của một văn bản, chủ đề được thể hiện súc tích hơn. Tuy nhiên thực tế lại cho thấy cửa sổ liên kết lớn đem lại nhiều lợi ích hơn.

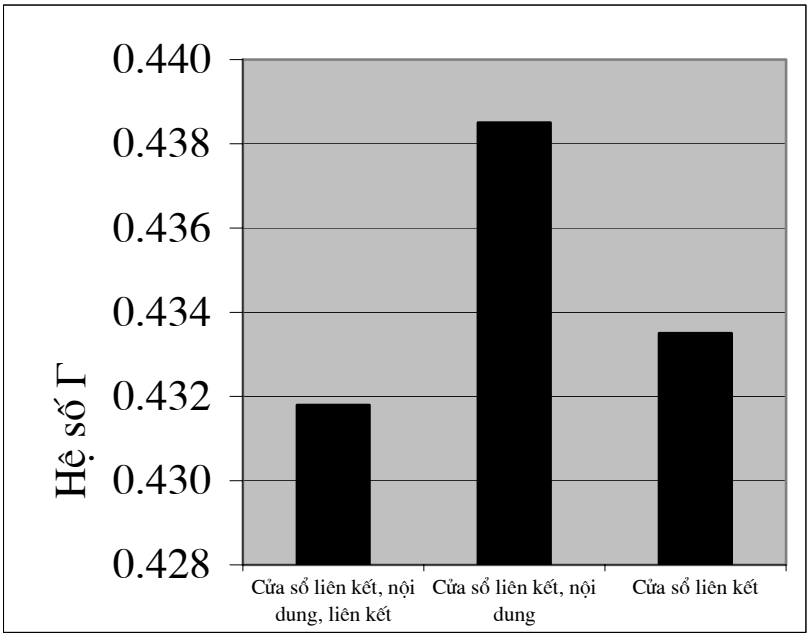



*Hình 12. Biểu đồ tỉ lệ trực giao với các phương pháp chọn từ khoá*

Biểu đồ tỷ lệ trực giao trên cho biết tỉ lệ các cặp văn bản trong cùng nhóm ODP mà trực giao. Chúng thấy rằng với kích thước cửa sổ nhỏ, nhiều văn bản có thể được cho là tương tự trong thực tế là trực giao. Trong trường hợp này, không thể cải thiện kết quả bằng phương pháp chuẩn hoá trọng số vì phương pháp biểu diễn này không cung cấp đủ thông tin có thể sử dụng được về độ tương tự giữa các văn bản đối với những cặp văn bản trực giao. Một nhận xét nữa, theo cách tiếp cận về nội dung và liên kết, số

lượng những văn bản ở cùng một nhóm mà trực giao rất lớn. Theo cách tiếp cận dựa trên liên kết, hầu hết các văn bản trong cùng một nhóm là các cặp văn bản trực giao, đây là một khám phá quan trọng về giới hạn của cách tiếp cận liên kết. Những liên kết đến có thể được mô tả không rõ ràng. Nếu hai trang có nhiều liên kết đến, nhưng giao của những liên kết này là rỗng thì thông tin về chúng là rất ít. Có thể chúng đề cập đến cùng một chủ đề, nhưng bởi vì chúng còn mới, chúng có thể không bao giờ xác định được tập liên  kết chung. Trong trường của các tiếp cận cửa sổ liên kết, khả năng hai tập hợp từ khoá trực giao là thấp hơn rất nhiều. Với mỗi liên kết, thay vì được thể hiện bởi liên kết có nghĩa không rõ ràng, nó được thể hiện bởi ngữ nghĩa của các từ khoá cấu thành các liên kết.

Các các tiếp cận đơn thuần như trên thì kích thước cửa sổ động cũng không cho kết quả mong muốn đối với tỉ lệ trực giao. Bất kì khu vực nào có chất lượng cao đều hướng đến các cửa sổ lớn để cho kết quả tốt hơn [0]. Với tổ hợp các cách tiếp cận khác nhau được khảo sát, kết quả cho thấy nếu kết hợp cả ba cách tiếp cận lại cho kết quả tồi hơn cách tiếp cận cửa sổ liên kết. Cách kết hợp nội dung toàn văn của văn bản và cửa sổ liên kết cho kết quả khả quan nhất. Dễ nhận thấy rằng nếu các văn bản có rất ít liên kết đến thì nội dung toàn văn của văn bản sẽ chiếm ưu thế. Ngược lại, nếu văn bản có nhiều liên kết đến thì từ khoá dựa trên cửa số liên kết sẽ chiếm ưu thế. Bằng cách này, tập hợp từ biểu diễn văn bản sẽ tự động dựa trên thông tin về chủ đề của văn bản nhiều nhất có thể sử dụng được. Đây chính là cách tiếp cận được luận văn dùng để áp dụng cho máy tìm kiếm VietSeek sau này.




*Hình 13. Biểu đồ hệ số Γ với các phương pháp tiếp cận*

### 2.5.2 Đánh giá chất lượng đối với cách chuẩn hoá trọng số từ khoá

Kết quả khảo sát về biên cửa sổ liên kết cho thấy cách tiếp cận dựa trên ngữ nghĩa lân cận liên kết với cửa sổ lớn cho kết quả tốt hơn. Tuy nhiên, dễ thấy rằng cửa sổ nhỏ cung cấp sự thể hiện nội dung của trang web súc tích hơn. Thực tế, để nâng cao chất lượng hơn nữa, trọng số của từ khoá có thể được chuẩn hoá dựa trên khoảng cách từ liên kết đến vị trí của từ khoá. Những từ khoá càng gần liên kết thì có ý nghĩa càng quan trọng đối với liên kết đó. Phương pháp này sẽ giảm được số cặp văn bản trực giao thay vì chọn kích thước cửa sổ nhỏ, tuy nhiên kích thước cửa sổ lại không đến mức quá lớn (khi đó trọng số theo khoảng cách của các từ khoá này là 0). Biểu thức chuẩn hóa trọng số của từ khóa theo khoảng cách được chọn

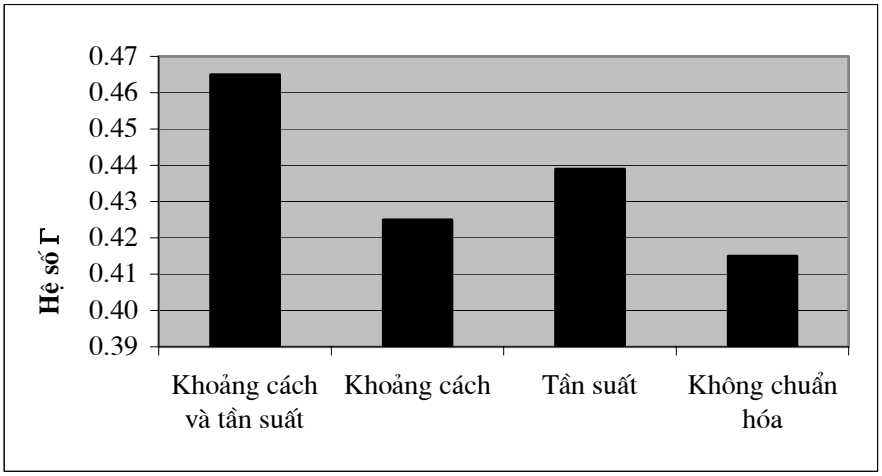
$$\log = \log_2\left(\frac{32}{1+\text{distance}(t, A_{vu})}\right) \qquad (10)$$

Trong đó, với

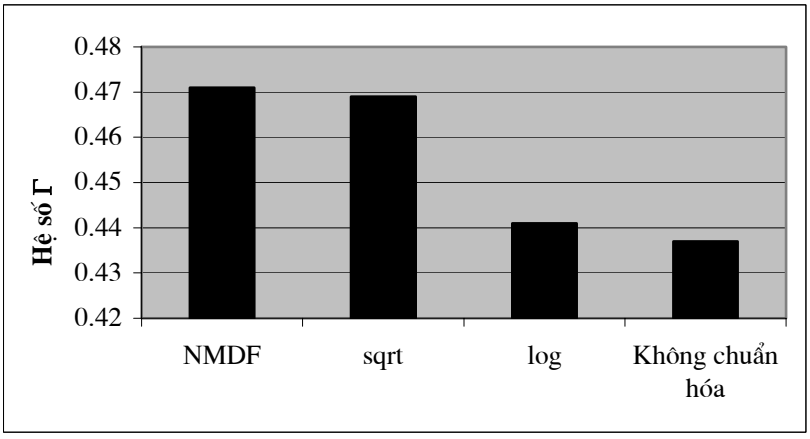
***u*** là văn bản cần tìm vector biểu diễn

***v*** là văn bản có liên kết $\boldsymbol{A}_{vu}$ đến văn bản ***u***

distance(***t***, $\boldsymbol{A}_{vu}$) là khoảng cách vị trí từ khoá ***t*** đến liên kết $\boldsymbol{A}_{vu}$. Những từ nằm trong chính liên kết thì có khoảng cách là 0.




*Hình 14. Biểu đồ hệ số Γ đối với khoảng và tần suất*



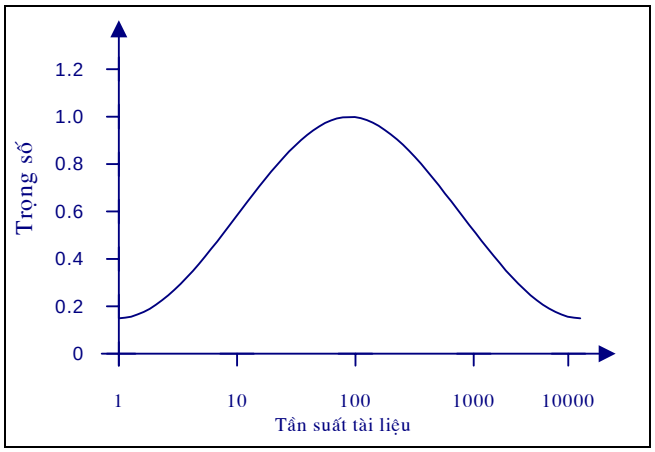

*Hình 15. Biểu đồ hệ số Γ với các công thức chuẩn hoá trọng số*

Bằng phương pháp giảm bớt trọng số của các từ khoá có tần suất cao và thấp của từ khoá trong văn bản, kết quả của trọng số dựa trên tần suất đã được nâng cao hiệu quả. Với $t_f$ là một tần suất của từ khoá trong tập hợp từ biểu diễn văn bản, và $d_f$ là tần suất của từ khoá trong mọi văn bản. Công thức chuẩn hóa tần suất

$$\log = \frac{tf}{1+\log_2(df)} \tag{11}$$

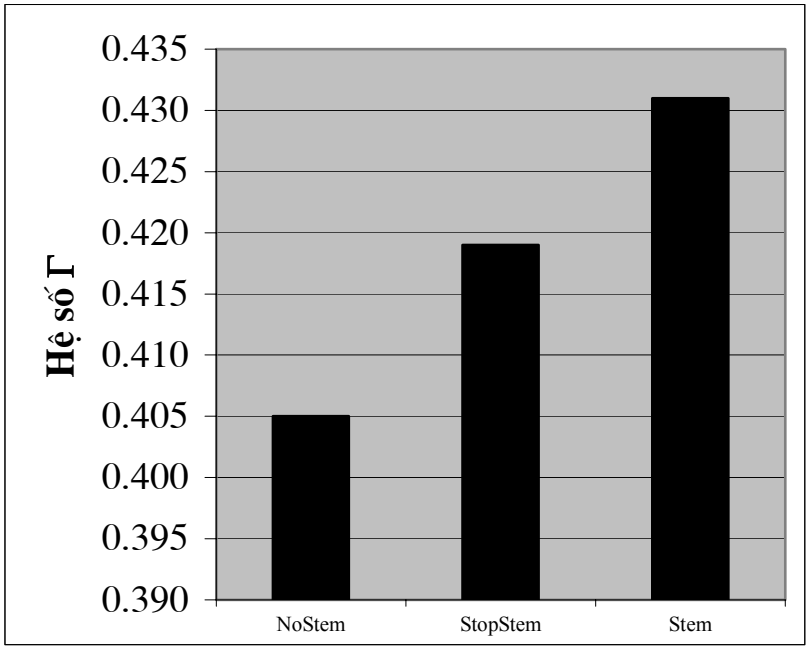
$$\text{sqrt} = \frac{tf}{\sqrt{df}} \tag{12}$$

$$\text{NMDF} = tf \times e^{-\frac{1}{2}\left(\frac{\log(df)-\mu}{\sigma}\right)^2} \tag{13}$$




*Hình 16. Đồ thị chuẩn hoá trọng số của từ khoá*

### 2.5.3 Đánh giá chất lượng đối với phương pháp lược bớt từ khoá




*Hình 17. Biểu đồ hệ số Γ đối với các phương lược bớt từ khoá*

Trong ba phương pháp lược bớt từ khoá NOSTEM, STOPSTEM và STEM, biểu đồ cho thấy phương pháp Stem đạt hiệu quả cao nhất. Lý do là nó lược bỏ số từ dừng một cách tối đa, hơn nữa nó giảm bớt số lượng các từ khoá do loại bỏ thêm được các biến thể của từ khoá.

## 2.6.Thiết kế các thuật toán tìm kiếm theo mô hình vector

Các thuật toán dưới đây sẽ được trình bày chi tiết trong chương 3 khi áp dụng vào máy tìm kiếm Vietseek.

### ❑ Thuật toán 2.6.1: Tạo vector biểu diễn trang web

**Input**:

- Các trang web cần được tạo chỉ mục $w_1$, $w_1$, ..., $w_n$

**Output**:

- Vector biểu diễn các trang web theo ngữ nghĩa lân cận liên kết $B_1$, $B_1$, ..., $B_n$

**Các bước thuật toán:**

1. Vector biểu của các trang web được khởi tạo là rỗng.
2. Đặt $i$=1
3. Xác định từ khóa trong nội dung toàn văn trang web và trọng số từ khóa tương ứng. Cập nhật từ khóa nội dung toàn văn trang web vào vector biểu diễn $B_i$
   - Nếu từ khóa chưa có trong $B_i$, đưa từ khóa và trọng số tương ứng vào $B_i$
   - Nếu từ khóa đã có trong $B_i$, cộng trọng số của nó vào trọng số từ khóa tương ứng trong vector $B_i$
4. Xác định cửa sổ liên kết từ $w_i$ liên kết đến $w_j$ có trong $w_i$ chưa được xử lý
   - Xác định các từ khóa trong cửa sổ liên kết và trọng số tương ứng.
   - Cập nhật từ khóa trong cửa sổ liên kết vào vector biểu diễn $B_{.j}$
5. Lặp lại bước 4
6. Đặt $i = i + 1$. Nếu $i <= n$ và lặp lại bước 3

### ❑ Thuật toán 2.6.2: Tính độ tương tự giữa các trang web

**Input**:

- Vector trang web mẫu B
- Vector biểu diễn của trang web cần đánh giá độ tương tự với trang web mẫu $B_1$, $B_2$, ..., $B_n$

**Output**:

- Độ tương tự của các trang web cần đánh giá $S_1$, $S_2$, ..., $S_n$

**Các bước thuật toán:**

1. Đặt $i = 1$

2. Tính tổng trọng số $B \cap B_i$ là $Min_i$
3. Tính tổng trọng số $B \cap B_i$ là $Max_i$
4. Độ tương tự trang web mẫu với trang web đang xét là $S_i = Min_i / Max_i$
7. Đặt $i = i + 1$. Nếu $i <= n$ và lặp lại bước 2

### ❑ Thuật toán 2.6.3: Tìm kiếm trang web tương tự

**Input**:

- Văn bản mẫu cần tìm q

**Output**:

- Danh sách các văn bản tương tự với văn bản mẫu. Với mỗi văn bản trong danh sách cho biết mức độ tương tự với văn bản mẫu

**Các bước thuật toán:**

1. Xác định mã số của trang mẫu
2. Xác định danh sách các trang web tương tự với trang web mẫu lớn hơn ngưỡng $\alpha$,
3. Sắp xếp các trang web tìm được theo thứ tự giảm dần và trả lại kết quả.

## KẾT LUẬN CHƯƠNG 2

Trong chương 2, luận văn đã hệ thống các cơ sở lý thuyết của phương pháp biểu diễn trang web theo lận cận ngữ nghĩa. Một nội dung quan trọng được trình bày trong chương này là sử dụng thứ tự nền để đánh giá chất lượng độ đo tương tự định nghĩa trên các tập văn bản. Luận văn cũng đã có những đề xuất chi tiết cho các công thức được nêu trong phần lý thuyết.

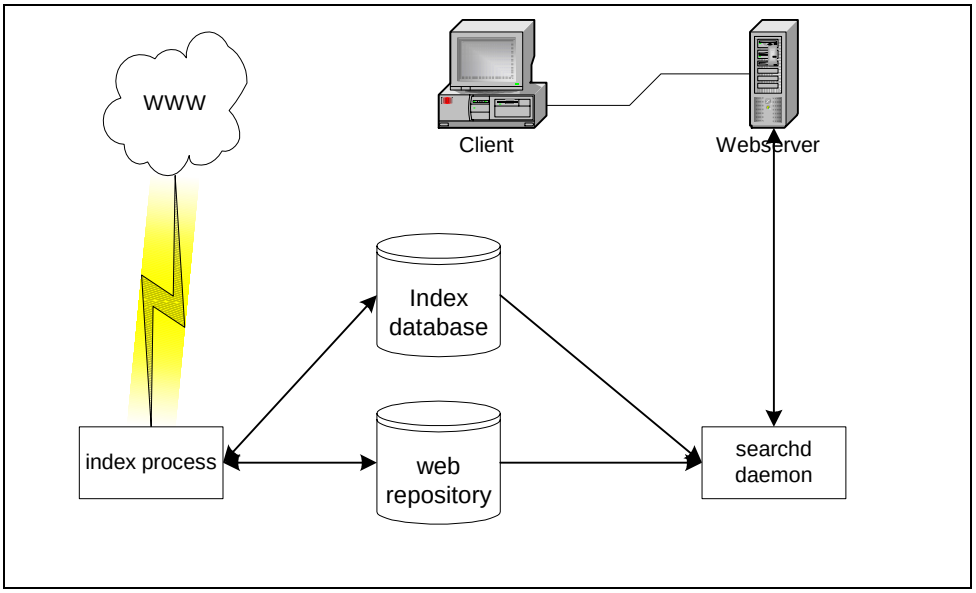
Trong chương 3, luận văn tập trung trình bày đề xuất áp dụng cụ thể của các của phương pháp đã mô tả trong chương 2 áp dụng vào máy tìm kiếm VietSeek.

# CHƯƠNG 3. MÁY TÌM KIẾM VIETSEEK VÀ THỬ NGHIỆM THUẬT TOÁN TÌM KIẾM THEO NGỮ NGHĨA LÂN CẬN SIÊU LIÊN KẾT

## 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]. Hiện tại, nhóm tác giả VietSeek sử dụng tên gọi Vinahoo thay thế cho tên gọi VietSeek bởi hai lý do. Lý do thứ nhất, một trang web tiếng Việt với tên VietSeek cũng đã được giới thiệu gần đây, và lý do thứ hai, tương tự như Yahoo, về mặt ngôn ngữ thì "Vinahoo" được coi là viết tắt của "VIet NAm Halirious Online Organization"




*Hình 18. Sơ đồ hoạt động của máy tìm kiếm VietSeek*

Cấu trúc dữ liệu của VietSeek đã được Phạm Thanh Nam phân tích tương đối chi tiết [1]. Trong luận văn này chỉ mô tả thêm về cấu trúc locgic các chức năng của VietSeek, đặc biệt là các chức năng cần bổ sung các modul tìm kiếm tương tự.

Máy tìm kiếm VietSeek gồm 2 modul chính:

- Modul index thực hiện công việc tìm duyệt các trang web. Phân tích các trang web và tạo các chỉ mục tương ứng. Lưu trữ các trang web. Các công việc này nhằm xử lý, tính toán trước các dữ liệu cần thiết để phục vụ cho phần giao diện với người dùng.

- Module 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. Module giao diện (máy phục vụ web) làm nhiệm vụ lấy kết quả trả về từ module tìm kiếm, trộn lại rồi hiển thị dưới dạng web cho người dùng.

### 3.1.2 Cơ sở dữ liệu của Vietseek

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

1. Phần 1: các dữ liệu về từ điển các trang web, từ khoá, site, domain, các trang web có kích thước nhỏ, các chỉ mục có kích thước nhỏ được lưu trữ dạng bảng trong cơ sở dữ liệu
2. Phần 2: Các trang web có kích thước lớn, các chỉ mục có kích thước lớn được lưu trữ thành file. VietSeek có modul chuyên xử lý vấn đề lưu trữ cung cấp dịch vụ cho các modul khác mà không cần biết dữ liệu tổ chức được nằm trên file hay trong cơ sở dữ liệu.

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 đã có sự cải tiến để hoạt động được trong thực tế chứ không lưu trữ theo cơ sở dữ liệu quan hệ đơn thuần. Các hệ quản trị cơ sở dữ liệu nói chung đều bị hạn chế về số lượng và kích thước của các bảng. Do đó VietSeek đã lưu trữ thông tin chi tiết kèm luôn vào từ điển. Với mỗi từ khoá thì thông tin các url mà từ khoá xuất hiện được lưu kèm theo dưới dạng nhị phân. Như vậy sẽ tiết

kiệm dung lượng, giảm bớt số lượng bản ghi, cho phép truy xuất trực tiếp đến dữ liệu về danh sách các url. Nếu thông tin về từ khoá và url mà lưu trữ riêng thành bảng thì mỗi từ khoá và mỗi url sẽ phải nằm trên một bản ghi. Do đó số lượng bảng ghi là tích Đề-các giữa bảng từ điển từ khoá và bảng từ điển url. Hơn nữa thông tin về mỗi từ khoá (word_id) sẽ bị lặp lại cho toàn bộ các url mà từ khoá đó xuất hiện. Còn khi lưu dạng nhị phân các url kèm vào bảng danh mục từ khoá thì không

Trong các modul được xây dựng bổ sung chức năng tìm kiếm tương tự cho VietSeek ta chỉ cần quan tâm đến bảng từ điển các từ khoá và bảng từ điển các trang web và sử dụng các bảng dữ liệu bổ sung thêm.

♦ Từ điển từ khoá **wordurl** của VietSeek. Bảng này luôn luôn được lưu trong cơ sở dữ liệu

| ***Tên trường*** | ***Mô 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 khoá |
| **totalcount** | Tổng số lần xuất hiện của từ khoá trong tất cả các trang web (URL) |

*Bảng 5. Mô tả cấu trúc bảng dữ liệu từ điển của VietSeek*

♦ 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*** | ***Mô 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 hoá 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 |
| **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 6. Mô tả cấu trúc bảng dữ liệu URL của VietSeek*

♦ Thông tin về chỉ mục đảo của các siêu liên kết ***citation***

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

*Bảng 7. Mô tả cấu trúc bảng dữ liệu chỉ mục đảo của VietSeek*

## 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

Trong cơ sở dữ liệu của VietSeek chỉ lưu trữ nội dung các trang web và chỉ mục nhị phân các url theo khía cạnh từ khoá, vì vậy chỉ mục từ khoá theo khía cạnh url sẽ được lưu trữ trong bảng sim_urlcontent. Để tăng tốc độ cũng như vượt qua giới hạn về kích thước của bảng dữ liệu, sim_urlcontent có thể được phân mảnh thành các bảng dữ liệu thành phần theo miền giá trị của url_id.

Mặt khác do các cửa sổ liên kết nằm ở các trang web khác, việc thay đổi của các cửa sổ liên kết có ảnh hưởng đến vector biểu diễn nhưng lại độc lập với sự thay đổi nội dung của trang web cho nên ta sẽ lưu trữ riêng các cửa sổ liên kết trong bảng sim_urlwnd để đảm bảo sự thay đổi đối với vector biểu là nhỏ nhất.

Như vậy vector biểu diễn gồm hai thành phần: nội dung của trang web chính và các cửa sổ liên kết nằm trong các trang web khác.

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 một danh sách các URL tương tự với nó được lưu trữ trong sim_urlsim, tức là có độ gần nhau lớn hơn ngữ $\alpha$ nào đó. Qua kinh nghiệm bản thân cũng như tham khác các văn bản khác thì nói chung $\alpha$ nên giới hạn ở giá trị 100 do sự quan tâm của người dùng thường chỉ dừng lại

khoảng 20 kết quả ban đầu. Bảng dữ liệu sim_urlsim có thể được phân mảnh bằng cách phân chia theo chủ đề của trang web.

♦ Bảng chỉ mục nội dung của trang web sim_urlcontent

| Tên trường | Mô tả |
|---|---|
| **url_id** | Mã số của trang web được tham chiếu đến bảng urlword |
| **word_count** | Số lượng tập hợp từ khoá (không lặp lại giá trị từ khoá) có mặt trong văn bản |
| **words** | Danh sách các từ khoá có mặt trang web theo thứ tự của word_id, mỗi từ khoá gồm các thành phần<br>- word_id: mã số của từ khoá<br>- df : tần suất từ khoá trong nội dung văn bản<br>- tf: tần suất từ khoá trong vector biểu diễn<br>- weight: trọng số của từ khoá |

*Bảng 8. Mô tả cấu trúc bảng dữ liệu chỉ mục nội dung của VietSeek*

♦ Chỉ mục cửa sổ liên kết của trang web sim_urlwnd

| Tên trường | Mô tả |
|---|---|
| **Id** | Số hiệu của cửa sổ |
| **url_id** | Mã số của trang web có vector biểu diễn |
| **refer_by** | Mã số của trang web chứa cửa sổ liên kết đến trang web có mã số là url_id |
| **word_count** | Số lượng tập hợp từ khoá (không lặp lại giá trị từ khoá) có mặt trong văn bản |

| | |
|---|---|
| **Words** | Danh sách các từ khoá có mặt trang web theo thứ tự của word_id, mỗi từ khoá gồm các thành phần<br>- word_id: mã số của từ khoá<br>- df : tần suất từ khoá trong nội dung văn bản<br>- tf: tần suất từ khoá trong vector biểu diễn<br>- weight: trọng số của từ khoá |

*Bảng 9. Mô tả cấu trúc bảng dữ liệu chỉ mục cửa sổ liên kết của VietSeek*

♦ Chỉ mục độ tương sự giữa các trang web sim_urlsim

| Tên trường | Mô tả |
|---|---|
| **id** | Số hiệu của cặp của hai trang web có mã số (url1, url2). Chỉ duy nhất có một cặp giá trị tương ứng với hai trang web có mã số url1, url2 mà không kể thứ tự mã số của chúng trong cặp. |
| **url_id1** | Mã số của trang web thứ nhất. |
| **url_id2** | Mã số của trang web thứ hai |
| **sim** | Độ tương tự giữa hai trang web có mã số là url1 và url2 |

*Bảng 10: Mô tả cấu trúc bảng dữ liệu chỉ mục các trang web tương tự của VietSeek*

♦ Danh sách các trang web cần tính toán độ tương tự sim_urlsim

| Tên trường | Mô tả |
|---|---|
| **url_id** | Mã số của trang web cần tính lại. |

*Bảng 11. Mô tả cấu trúc bảng dữ liệu chứa danh sách cần chỉ mục lại của VietSeek*

♦ Danh mục các chủ đề Category. Dữ liệu của bảng này được lấy từ Open Directory ở địa chỉ http://rdf.dmoz.org/rdf/structure.rdf.u8.gz và được đưa vào cơ sở dữ

liệu mysql bằng perl script cung cấp tại địa chỉ http://odp.locallink.net/setup/odpstructure.txt

| Tên trường | Mô tả |
|---|---|
| **Topic** | Đường dẫn đầy đủ của chủ đề.<br>Ví dụ: Top/Computers/Databases |
| **TopicShort** | Chủ đề hiện tại. Ví dụ: Databases |
| **ParentTopic** | Đường dẫn đầy đủ của chủ đề cha. Ví dụ: Top/Computers |
| **Description** | Mô tả về chủ đề |
| **LastUpdate** | Thời điểm cập nhật cuối cùng. |

*Bảng 12: Mô tả cấu trúc bảng dữ liệu các chủ đề của VietSeek*

♦ Các trang web trong cây chủ đề  Link. Dữ liệu của bảng này được lấy từ Open Directory ở địa chỉ http://rdf.dmoz.org/rdf/content.rdf.u8.gz và được đưa vào cơ sở dữ liệu mysql bằng perl script cung cấp tại địa chỉ http://odp.locallink.net/setup/odpcontent.txt. Chú ý rằng một trang web có thể thể hiện nhiều chủ đề và ở các cấp khác nhau như trang web http://www.arttoday.com/netscape/ vừa ở chủ đề Top/Netscape/Community đồng thời cũng xuất hiện trong chủ đề Top/Netscape/Computing_and_Internet/Image_Library.

| Tên trường | Mô tả |
|---|---|
| **LinkID** | Mã số chủ đề của trang web trong cây chủ đề. Mã số này khác với mã số của bộ tìm duyệt điền cho các trang web được index. |
| **Page** | Url của trang web |
| **ParentTopic** | Đường dẫn đầy đủ trong cây chủ đề. Ví dụ<br>Top/Computers/Databases |

| | |
|---|---|
| **Title** | Tiêu đề của trang web |
| **Description** | Phần mô tả của trang web |

*Bảng 13. Mô tả cấu trúc bảng dữ liệu các trang web theo chủ đề của VietSeek*

### 3.2.2 Các thuật toán áp dụng cho máy tìm kiếm VietSeek

Các thuật toán này được áp dụng cho modul index của VietSeek. Để giảm bớt khả năng các trang web có vector biểu diễn thay đổi nhiều lần trong một phiên tìm duyệt web của modul index, các trang web không được tính toán ngay độ tương tự với các trang web khác mà được đưa vào hàng đợi xử lý sim_urltmp. Lý do là vector biểu diễn trang web có sự thay đổi khi có một trang web liên kết đến nó hoặc một trang web đã từng liên kết đến nó có sự thay đổi trong cửa sổ liên kết. Trước khi kết thúc quá trình tìm duyệt, modul index sẽ gọi đến modul xử lý các trang web cần tính toán độ tương tự và xoá chúng ra khỏi hàng đợi.

Với 2 trang web A và B được định nghĩa:

$$SIM(A,B) = |A \wedge B|/|A \vee B| \quad (14)$$

Vì $|A \wedge B|$ lấy tổng các trọng số (lấy trọng số nào nhỏ hơn trong hai trọng số của một từ khoá trong A và B) của từ khoá mà vừa có mặt trong A, vừa có mặt trong B nên $|A \wedge B| < W(A)$ (tổng các trọng số của A) và $|A \wedge B| < W(B)$ (tổng các trọng số của B). Tương tự, vì $|A v B|$ lấy tổng các trọng số (lấy trọng số nào lớn hơn trong hai trọng số của một từ khoá trong A và B) của từ khoá mà vừa có mặt trong A, vừa có mặt trong B nên $|A v B| > W(A)$ (tổng các trọng số của A) và $|A \vee B| > B$ (tổng các trọng số của B). Do đó ta có

$$SIM(A,B) = |A \wedge B|/|A \vee B| < \min(A,B)/\max(A,B) \quad (15)$$

Như vậy min(A,B)/max(A,B) chính là cận trên của độ tương tự giữa A và B, ta gọi biểu thức này là SUPPER(A,B). Nếu cận trên này nhỏ hơn ngưỡng được cho là tương tự

giữa A và B thì ta có thể không cần phải so sánh hết toàn bộ các thành phần của hai vector biểu diễn A và B và cải thiện đáng kể thời gian xử lý.

Khi phân tích các cửa sổ liên kết, các cửa sổ liên kết phải đảm bảo các yêu cầu sau đây:

- Không được vượt quá 16 từ khoá mỗi bên trái và phải của cửa sổ
- Không được vượt quá 500 kí tự (mỗi từ là 30 kí tự thì 16 từ cũng chỉ 480 kí tự)
- Bên trái cửa sổ phải đảo lại để thuận lợi cho việc tính khoảng cách từ khoá với liên kết.
- Toàn bộ các từ khoá nằm trong liên kết đều nằm trong cửa sổ nhưng cũng không vượt quá 500 kí tự
- Kết thúc khi sang một câu khác (dấu chấm câu và dấu ngăn cách)
- Kết thúc khi sang một đoạn văn khác trong bảng (dấu chấm câu và dấu ngăn cách)
- Kết thúc khi sang một trang khác khác trong bảng (dấu chấm câu và dấu ngăn cách)
- Kết thúc khi sang một ô khác trong bảng
- Kết thúc khi sang một mục khác trong danh sách
- Kết thúc khi sang một liên kết khác

### ❑ Thuật toán 3.3.1: Phân tích cửa sổ liên kết

**Input**:

- Nội dung trang web có chứa cửa sổ liên kết
- Vị trí bắt đầu liên kết đến trang web cần tạo vector biểu diễn
- Mã số url_id của trang web chứa cửa sổ liên kết
- Mã số url_id của trang web được liên kết đến

Các tham số đầu vào đã được modul phân tích của VietSeek tính toán trước

**Output**:

- Cửa sổ liên kết được lưu trữ trong bảng sim_urlwnd
- Danh sách các trang web cần tính lại độ tương tự vector biểu diễn có thay đổi

**Các bước thuật toán:**

8. Tìm phần trái của cửa sổ liên kết

1.1 Khởi tạo:

- Vị trí được xét lùi 1 vị trí so với vị trí bắt đầu liên kết
- Số lượng từ khoá = 0
- Số lượng kí tự bên trái = 0
- Vị trí cửa sổ trái bắt đầu từ 0
- Vị trí trong bộ đệm cho từ hiện tại bắt đầu từ cuối bộ đệm

1.2 Khi nào chưa thoả mãn các giới hạn về biên cửa sổ và còn lớn hơn điểm bắt đầu của văn bản thì tiếp tục xét

1.3 Nếu là một thẻ HTML hoặc khoảng trống thì:

- Chuyển từ hiện tại trong bộ đệm vào cửa sổ liên kết trái
- Số lượng từ trong cửa sổ liên kết trái cộng thêm 1
- Số lượng kí tự trong cửa sổ liên kết trái cộng thêm số lượng kí tự chuyển vào
- Vị trí bắt đầu cửa sổ liên kết trái dịch đi 1 từ
- Vị trí trong bộ đệm cho từ hiện tại bắt đầu từ cuối bộ đệm
- Phân tích nếu là thẻ HTML, vị trí đang xét lùi qua các kí tự của thẻ HTML
- Nếu là khoảng trống thì vị trí đang xét lùi 1 kí tự
- Trở lại bước 1.2 để kiểm tra điều kiện

1.4 Là kí tự bình thường

- Chuyển kí tự hiện tại vào bộ đệm từ
- Vị trí đang xét lùi đi một kí tự
- Trở lại bước 1.2 để kiểm tra điều kiện

1.5 Chuyển từ khoá cuối cùng trong bộ đệm từ vào bên trái cửa sổ liên kết

9. Tìm phần trung tâm của cửa sổ liên kết

9.1. Khởi tạo:

- Số lượng kí tự của trung tâm cửa sổ là 0
- Vị trí đang xét là vị trí bắt đầu liên kết

9.2. Khi nào chưa thoả mãn các giới hạn về biên cửa sổ và còn nhỏ hơn điểm kết thúc của văn bản thì tiếp tục xét

- Nếu là thẻ HTML thì phân tích thẻ HTML và vị trí đang xét dịch chuyển qua thẻ HTML
- Nếu là khoảng trống và vị trí đang xét dịch chuyển qua hết khoảng trống và chuyển một kí tự trắng vào trung tâm cửa sổ
- Nếu là kí tự bình thường thì chuyển vào trung tâm cửa sổ và vị trí đang xét chỉ đến kí tự tiếp theo
- Kiểm tra lại điều kiện của bước 2.2

10. Tìm phần phải của cửa sổ liên kết

10.1. Khởi tạo:

- Số lượng kí tự của bên phải cửa sổ là 0
- Vị trí đang xét là vị trí tiếp theo của bước trước

10.2. Khi nào chưa thoả mãn các giới hạn về biên cửa sổ và còn nhỏ hơn điểm kết thúc của văn bản thì tiếp tục xét

- Nếu là thẻ HTML thì phân tích thẻ HTML và vị trí đang xét dịch chuyển qua thẻ HTML

- Nếu là khoảng trống và vị trí đang xét dịch chuyển qua hết khoảng trống và chuyển một kí tự trắng vào bên phải cửa sổ
- Nếu là kí tự bình thường thì chuyển vào bên phải cửa sổ và vị trí đang xét chỉ đến kí tự tiếp theo
- Kiểm tra lại điều kiện của bước 3.2

11. Lưu trữ cửa sổ liên kết

11.1. Lưu trữ chung

- Lưu trữ bên trái cửa sổ với khoảng cách từ khoá bắt đầu từ 1, khoảng cách từ khoá tiếp theo sẽ tăng lên 1
- Lưu trữ trung tâm cửa sổ với khoảng cách từ khoá bắt đầu từ 0, khoảng cách từ khoá tiếp theo sẽ tăng lên 0
- Lưu trữ bên trái cửa sổ với khoảng cách từ khoá bắt đầu từ 1, khoảng cách từ khoá tiếp theo sẽ tăng lên 1

11.2. Lưu trữ một thành phần

- Khi nào bộ đệm còn thì còn thực hiện
- Lấy từ khoá hiện tại
- Tính trọng số từ khoá theo khoảng cách
- Tính trọng số từ khoá theo tần suất
- Lưu trữ từ khoá hiện thời vào bộ đệm window_vector
- Tăng khoảng cách từ khoá đến giá trị tiếp theo

11.3. Lưu trữ cửa sổ vào cơ sở dữ liệu

- Xoá bảng sim_urlwnd có giá trị (url_id, refer_by) = (Mã số url_id của trang web chứa cửa sổ liên kết, Mã số url_id của trang web được liên kết đến)
- Thêm vào sim_urlwnd bộ giá trị giá trị (url_id, refer_by, window_vector)

- Bổ sung mã số của trang web được liên kết đến vào danh sách các web cần tính lại độ tương tự

### ❑ Thuật toán 3.3.2: Lưu trữ nội dung của một trang web

**Input**:

- Các từ khoá của trang web
- Mã số url_id của trang web

**Output**:

- Nội dung trang web được lưu trữ trong bảng sim_urlcontent
- Danh sách các trang web cần tính lại độ tương tự vector biểu diễn có thay đổi

**Các bước thuật toán:**

1. Tạo vector
   - Lần lượt lấy từng từ khoá
   - Tính toán trọng số và tổng số từ khoá (word_count) có trong nội dung trang web
   - Đưa từ khoá vào bộ đệm content_vector
2. Lưu trữ
   - Xoá nội dung cũ nếu có trong bảng sim_urlcontent với url_id = url_id của trang web hiện tại
   - Thêm vào bảng sim_urlcontent bộ giá trị (url_id, word_count, content_vector)
   - Thêm url_id của trang web hiện tại vào danh sách các trang web cần tính độ tương tự

### ❑ Thuật toán 3.3.3: Tính toán độ tương tự của các văn bản

**Input**:

- Danh sách các url_id của trang web cần tính toán độ tương tự sim_urltmp
- Cây chủ đề trang web sim_urltopic
- Vector nội dung của các trang web trong sim_urlcontent
- Vector cửa sổ liên kết của các trang web trong sim_urlwnd

**Output**:

- Cửa sổ liên kết được lưu trữ trong bảng sim_urlwnd
- Danh sách các trang web cần tính lại độ tương tự vector biểu diễn có thay đổi

**Các bước thuật toán:**

1. Lấy url_id nhỏ nhất ra khỏi hàng đợi sim_urltmp (và xoá url_id này khỏi bảng sim_urltmp). Gọi trang web này là A.
2. Lấy danh sách các trang web cùng chủ đề
   2.1. Tìm chủ đề của trang web hiện tại cần xử lý trong cây thư mục. Chỉ lấy chủ đề có độ sâu là 3.
   2.2. Nếu không tìm thấy thì bổ sung trang web hiện tại vào chủ đề khác (chủ đề Top/World)
   2.3. Lấy danh sách các trang khác web cùng chủ đề
3. Lần lượt lấy mỗi trang web trong danh sách cùng chủ đề ra xử lý. Gọi trang web này là B
   3.1. Lấy vector biểu diễn trang web A từ sim_urlcontent và sim_urlwnd. Tính toán lại trọng số của mỗi từ khoá. Tính tổng trọng số của A gọi là W(A)
   3.2. Lấy vector biểu diễn trang web B từ sim_urlcontent và sim_urlwnd. Tính toán lại trọng số của mỗi từ khoá. Tính tổng trọng số của A gọi là W(B)

3.3. Khởi tạo $|A \wedge B| = 0$, $|A \vee B| = W(B)$. Giả sử $W(A) < W(B)$. Gọi MIN(A,B) là W(A). Gọi MAX(A,B) là W(B). SUPPER(A,B) = MIN(A, B)/MAX(A,B). Nếu cận trên nhỏ hơn ngưỡng tương tự thì xử lý trang web khác.

3.4. Lần lượt xét mỗi từ khoá xuất hiện trong vector biểu diễn A, giả sử là word(i) và trọng số tương ứng là wa(i).

- Nếu từ khoá không có mặt trong B thì trọng số của word(i) trong B là wb(j) = 0.
- Nếu từ khoá có mặt trong B và có trọng số wb(j).
- Nếu wa(i) <= wb(j):

  $|A \wedge B| = |A \wedge B| + wa(i)$

  $|A \vee B| = |A \vee B|$ vì wb(j) đã nằm trong W(B)

  MIN(A,B) = MIN(A,B) vì wa(i) đã nằm trong W(A)

  MAX(A,B) = MAX(A,B) vì wb(i) đã nằm trong W(B)
- Nếu wa(i) > wb(j):

  $|A \wedge B| = |A \wedge B| + wb(j)$

  $|A \vee B| = |A \vee B| + wa(i) - wb(j)$

  MIN(A,B) = MIN(A,B) – wa(i) + wb(j)

  MAX(A,B) = MAX(A,B) + wa(i) – wb(j)

  SUPPER(A,B) = MIN(A, B)/MAX(A,B). Nếu SUPPER(A,B) < ngưỡng tương tự thì coi A không tương tự với B. và xử lý trang web khác.
- Tiếp tục xử lý từ khoá khác trong vector biểu diễn A

3.5. Tính $SIM(A,B) = |A \wedge B|/|A \vee B|$. Nếu SIM(A,B) > ngưỡng tương tự thì

- Xoá bộ giá trị (A, B, sim) hoặc (B, A, sim) trong sim_urlsim.
- Nếu số lượng các trang web tương tự với A lớn hơn 100 và có độ tương tự nhỏ hơn SIM(A,B) thì xoá trang web có độ tương tự nhỏ nhất trong sim_urlsim.

- Nếu số lượng các trang web tương tự với B lớn hơn 100 và có độ tương tự nhỏ hơn SIM(A,B) thì xoá trang web có độ tương tự nhỏ nhất trong sim_urlsim.
- Thêm bộ giá trị (A, B, sim) vào sim_urlsim

3.6. Tiếp tục xử lý trang web khác

### ❑ Thuật toán 3.3.4: Tìm kiếm các trang web "gần" với trang web hiện thời

**Input**: url của trang web mẫu

**Output**: Danh sách các url và độ tương tự của các trang web khác theo thứ tự giảm dần của độ tương tự

**Các bước thuật toán:**

1. Tìm mã số url_id mẫu tương ứng với url trang mẫu trong bảng từ điển urlword
2. Lấy ra url_id1, url_id2, sim từ bảng sim_urlsim với điều kiện url_id1 = url_id mẫu hoặc url_id2 bằng url_id mẫu và sắp theo thứ tự giảm dần của sim
3. Lấy địa của url của các trang web tương tự từ url_word với mã số url_id bằng url_id1 + url_id2 – url_id mẫu (vì url_id mẫu là một giá trị trong cặp url_id1, url_id2 nhưng ta không biết là cái nào ).
4. Hiển thị kết quả cho người dùng

***Nhận xét***

Thuật toán này thể hiện khả năng tìm kiếm "gần về nội dung" dựa trên biểu diễn vector thông qua việc lưu trữ sẵn 100 chỉ số trang web gần nhất nhưng làm giảm khối lượng dữ liệu sử dụng còn 1/2 như cách thông thường.

Cụ thể nếu A và B tương tự nhau thì chỉ lưu trữ một cặp giá trị (A,B) thay cho (A,B) nghĩa là B tương tự A và (B,A) nghĩa là là A tương tự B. Thuật toán này có thể áp dụng cho máy tìm kiếm VietSeek để thực hiện các công việc:

- Loại bỏ các trang trùng thừa khi hiển thị kết quả tìm kiếm,
- Liệt kê các trang web có liên quan với trang web tìm được theo từ khoá,

- Tìm kiếm các trang web tương tự theo chủ đề.

### 3.2.3 Kết quả thực hiện

Giả sử chúng ta cần tìm ra các trang web tương tự với trang web http://190.2.180.188/manual/install.html

Ta phải tìm mã số của nó trong bảng từ điển các url bằng lệnh sau

```
select u.url_id, u.url
from urlword u
where url = 'http://190.2.180.188/manual/mod/';
```

```
+--------+---------------------------------+
| url_id | url                             |
+--------+---------------------------------+
|      7 | http://190.2.180.188/manual/mod/|
+--------+---------------------------------+
1 row in set (0.01 sec)
```

*Bảng 14. Lệnh và kết quả thực hiện khi lấy mã số một trang web*

Khi biết mã số url tương ứng là 7, ta thực hiện lấy ra danh sách các trang web tương tự với nó sắp xếp theo độ tương tự giảm dần

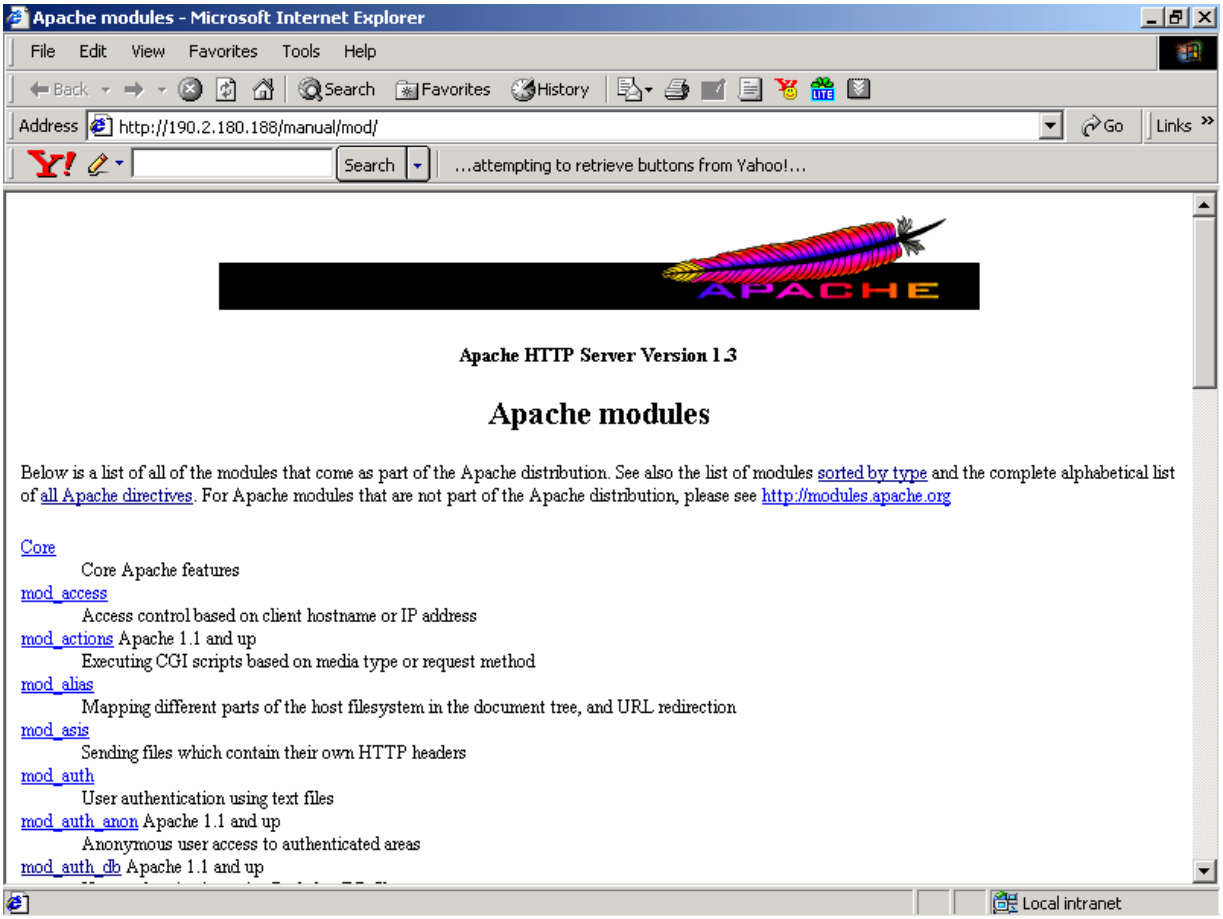
```
select u.url_id, u.url, s.sim
from urlword u, sim_urlsim s
where (s.url_id1 = 7 or s.url_id2 = 7)
and u.url_id = s.url_id1+s.url_id2-7
order by s.sim desc;
```

```
+--------+-------------------------------------------------------+-------+
| url_id | url                                                   | sim   |
+--------+-------------------------------------------------------+-------+
|     14 | http://190.2.180.188/manual/mod/index-bytype.html     | 0.797 |
|      5 | http://190.2.180.188/manual/sitemap.html              |  0.27 |
|      8 | http://190.2.180.188/manual/new_features_1_3.html     | 0.196 |
|     65 | http://190.2.180.188/manual/mod/mod_speling.html      | 0.188 |
```

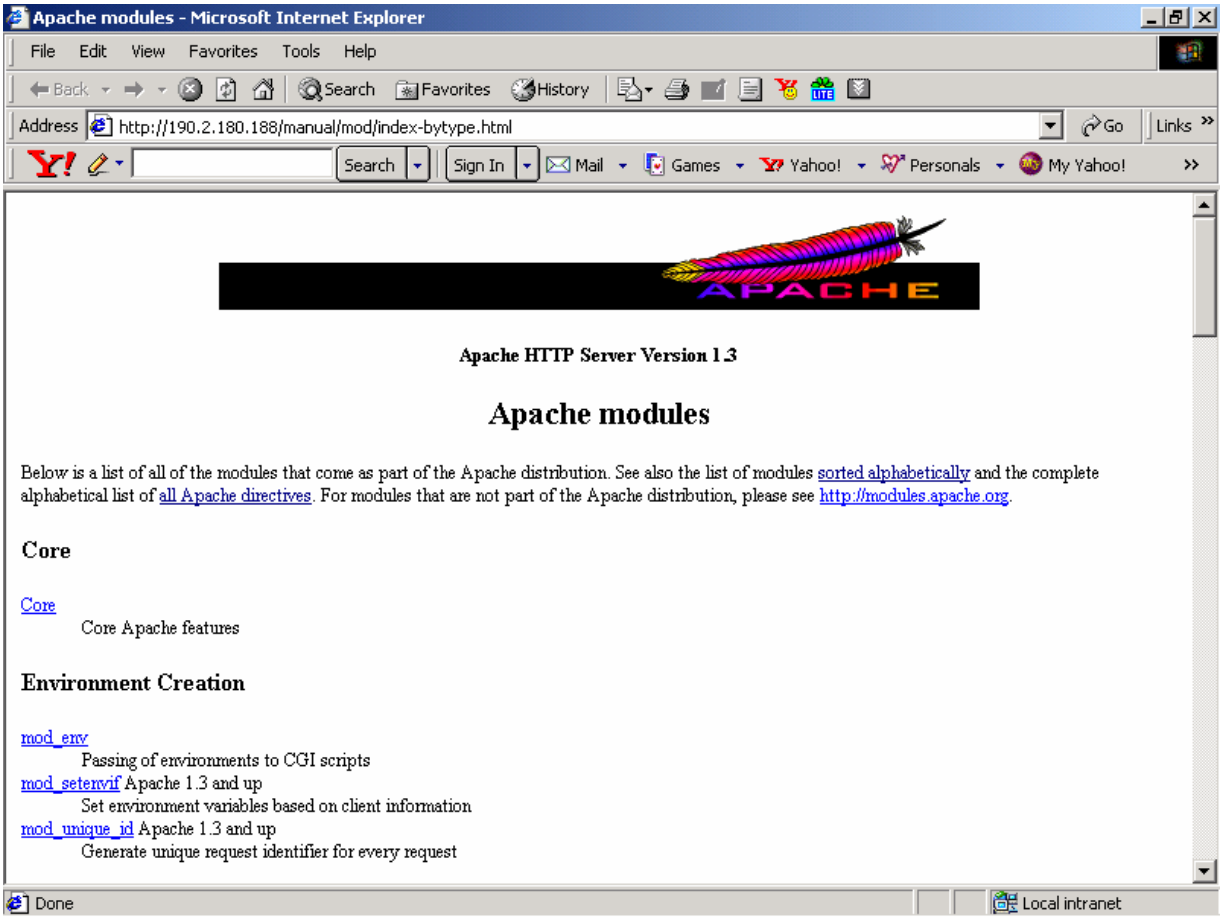
| | | | |
|---|---|---|---|
| | 101 | http://190.2.180.188/manual/mod/mod_cern_meta.html | 0.185 |
| | 48 | http://190.2.180.188/manual/mod/mod_info.html | 0.182 |
| | 113 | http://190.2.180.188/manual/mod/mod_so.html | 0.172 |
| | 103 | http://190.2.180.188/manual/mod/mod_digest.html | 0.171 |
| | 104 | http://190.2.180.188/manual/mod/mod_example.html | 0.171 |
| | 68 | http://190.2.180.188/manual/mod/mod_setenvif.html | 0.169 |
| | 97 | http://190.2.180.188/manual/mod/module-dict.html | 0.167 |
| | 9 | http://190.2.180.188/manual/upgrading_to_1_3.html | 0.166 |
| | 96 | http://190.2.180.188/manual/mod/directive-dict.html | 0.166 |
| | 130 | http://190.2.180.188/manual/mod/mod_cookies.html | 0.165 |
| | 66 | http://190.2.180.188/manual/mod/mod_actions.html | 0.164 |
| | 46 | http://190.2.180.188/manual/mod/mod_usertrack.html | 0.162 |
| | 131 | http://190.2.180.188/manual/mod/mod_browser.html | 0.16 |
| | 111 | http://190.2.180.188/manual/mod/mod_mmap_static.html | 0.159 |
| | 57 | http://190.2.180.188/manual/mod/mod_env.html | 0.156 |
| | 107 | http://190.2.180.188/manual/mod/mod_isapi.html | 0.156 |
| | 117 | http://190.2.180.188/manual/mod/mod_vhost_alias.html | 0.155 |
| | 62 | http://190.2.180.188/manual/mod/mod_auth_db.html | 0.153 |
| | 100 | http://190.2.180.188/manual/mod/mod_auth_digest.html | 0.153 |
| | 133 | http://190.2.180.188/manual/mod/mod_log_common.html | 0.153 |
| | 114 | http://190.2.180.188/manual/mod/mod_status.html | 0.151 |
| | 59 | http://190.2.180.188/manual/mod/mod_expires.html | 0.15 |

26 rows in set (0.02 sec)

*Bảng 15. Danh sách các trang web tương tự với trang web mẫu*




*Hình 19. Trang web mẫu* `http://190.2.180.188/manual/mod/`




*Hình 20. Trang web tương tự* `http://190.2.180.188/manual/mod/index-bytype.html`

Cả hai trang web trên đều thể hiện chung một vấn đề là mô tả các module của apache và được trình bày theo hai cách khác nhau. Chúng có độ tương tự là 0.797.

## KẾT LUẬN CHƯƠNG 3

Chương 3 trình bày cấu trúc thành phần của máy tìm kiếm tiếng Việt VietSeek và sơ đồ locgic 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 (các bảng), bổ sung các modul phân tích trang web để tìm ra vector biểu diễn trang web theo ngữ nghĩa lân siêu liên kết (thuật toán 3.3.1, 3.3.2).

Luận văn đề xuất thuật toán so sánh độ tương tự giữa các vector biểu diễn trang web. Hơn nữa, qua quá trình nghiên cứu, phân tích và áp dụng trong thực tế, luận văn đề xuất phương pháp tính xấp xỉ cận trên (thuật toán 3.3.3) của độ đo tương tự để cắt bớt nhánh xử lý trong khi so sánh giữa hai vector. Điều này tăng đánh kể tốc độ phân tích và làm cho các thuật toán do luận văn đề xuất có ý nghĩa trong thực tế.

Để tăng tốc độ phân tích trang web, luận văn đã đề xuất phương án lưu các trang web có vector biểu diễn thay đổi vào một hàng đợi để xử lý sau (thuật toán 3.3.1, 3.3.2). Điều đó đảm bảo cho vector biểu diễn có thay đổi bao nhiêu lần trong một phiên tìm duyệt thì cũng chỉ cần xử lý cho lần thay đổi cuối cùng (thuật toán 3.3.3).

Luận văn đã đề xuất thuật toán thể hiện khả năng tìm kiếm "gần về nội dung" dựa trên biểu diễn vector (thuật toán 3.3.4) bằng việc lưu trữ sẵn 100 chỉ số trang web gần nhất nhưng giảm kích thước còn 1/2 như cách thông thường.

# PHẦN KẾT LUẬN

### *1. Kết quả đạt được của luận văn*

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

• Đã trình bày tổng quan về bài toán tìm kiếm thông tin trên web (chương 1). Đã đã trình bày, khảo sát, phân tích, so sánh và đánh giá chất lượng một số phương pháp tiếp cận điển hình để giải quyết bài toán này (chương 2),

• Thông qua việc khảo sát, phân tích, đánh giá từng phương pháp nói trên, luận văn đã:

- Đề xuất một cách thức biểu diễn trang web theo ngữ nghĩa lân cận siêu liên kết làm cơ sở so sánh nội dung toàn văn văn bản và khai thác được ngữ nghĩa lân cận các siêu liên kết (mục 2.6).

- Đề xuất một phương pháp giảm bớt số lần so sánh độ tương tự các trang web (mục 3.2).

- Đề xuất một phương pháp tính cận trên của độ tương tự và cách thức xấp xỉ (cắt bớt nhánh xem xét), do đó giảm được đáng kể số phép tính phải thực hiện, làm tăng tốc độ thực hiện (mục 3.2).

- 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).

Tuy nhiên, do hạn chế về thời gian hoàn thành luận văn nên việc triển khai phát triển máy tìm kiếm VietSeek vẫn chưa bổ sung được giao diện đối với người sử dụng để khai thác phản hồi của người dùng với kết quả tìm kiếm.

Luận văn tuy đã đề xuất một số cải tiến có ý nghĩa về giải pháp biểu diễn và tìm kiếm, đồng thời xây dựng được một số môđun chương trình thuật toán cho phương pháp cải tiến song chỉ mới thử nghiệm bước đầu mà chưa cài đặt tích hợp vào trong VietSeek. Đây cũng là một hạn chế của luận văn.

***2. Phương hướng nghiên cứu tiếp theo***

Web Mining luôn là lĩnh vực nghiên cứu và triển khai thời sự và những hạn chế kết quả của luận văn chính là phương hướng phát triển nội dung luận văn. Những bài toán dưới đây là nội dung nghiên cứu tiếp theo của luận văn này:

- Nghiên cứu cải tiến hệ thống thông qua giải pháp thu nhận đánh giá phản hồi của người dùng đối với chất lượng tìm kiếm để chất lượng tìm kiếm định hướng hơn tới người dùng.

- Tự động phân lớp các trang web tiếng Việt bổ sung thêm vào cây chủ đề ODP.

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

[1]. Phạm Thanh Nam (2003). *Một số giải pháp cho bài toán tìm kiếm trong cơ sở dữ liệu Hypertext*. Luận văn thạc sĩ Công nghệ thông tin - Đại học Quốc gia Hà Nội.

[2]. Phạm Thanh Nam, Bùi Quang Minh, Hà Quang Thụy (2004). *Giải pháp tìm kiếm trang Web tương tự trong máy tìm kiếm VietSeek*. Tạp chí Tin học và Điều khiển học (nhận đăng 1-2004).

[3]. Đoàn Sơn (2002). *Các phương pháp biểu diễn và ứng dụng trong khai phá dữ liệu văn bản*. Luận văn thạc sĩ Công nghệ thông tin - Đại học Quốc gia Hà Nội.

**Tiếng Anh**

[4]. J. Dean and M. Henzinger (1999). *Finding Related Pages in the World Wide Web*. Proceedings of WWW8, 1999.

[5]. L. A. Goodman and W. H. Kruskal. *Measures of association for cross classifications*. J. of Amer. Stat. Assoc., 49:732-764, 1954. ???

[6]. T.H. Haveliwala, A. Gionis, and P. Indyk (2000). *Scalable Techniques for Clustering the Web*.Informal Proceedings of the International Workshop on the Web and Databases, WebDB*, 2000*.

[7]. J. Hirai, S. Raghavan, H. Garcia-Molina, and A. Paepcke (2000). *WebBase: A Repository of Web Pages*.Proceedings of WWW9, 2000.

[8]. A.K. Jain, M. Narasimha Murty, and P.J. Flynn (1999). *Data clustering: A review ACM Computing Surveys*, 31(3), 1999.

[9]. H. P. Luhn. *The Automatic Creation of Literature Abstracts*. IBM Journal of Research and Development, 2:159-165, 1958.

[10]. Nguyen Ngoc Minh, Nguyen Tri Thanh, Ha Quang Thuy, Luong Song Van, Nguyen Thi Van (2001). *A Knowledge Discovery Model in Full-text 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.

[11]. M. Porter (1980). *An Algorithm for Suffix Stripping*. Program: Automated Library and Information Systems, 14(3):130-137, 1980.

[12]. G. Salton and M.J. McGill (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

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

[14]. S. Siegel and N. J. Castellan (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 1988.

[15]. M. Steinbach, G. Karypis, and V. Kumar (2000). *A comparison of document clustering techniques*. TextMining Workshop, KDD, 2000.

[16]. Taher H. Haveliwala, Aristides Gionis, Dan Klein, Piotr Indyk (2002). *Evaluating Strategies for Similarity Search on the Web*. WWW2002 - USA.

[17]. BBC. http://www.bbc.com.

[18]. CNN http://www.cnn.com.

[19]. Open Directory Project (ODP). http://www.dmoz.com.

[20]. Web page www.*InfoWorld.com* (Theo công bố ngày 17/02/2004 thì trong kho dữ liệu của Google đã có 4,28 tỷ trang web, 880 triệu hình ảnh và 845 triệu thông điệp Internet. Mảng thông tin đang tăng nhanh gần đây là các trang web liên quan đến sách, bao gồm các chương đầu, phần phê bình, tham khảo. Hệ thống thông tin này được Google truy xuất qua dịch vụ Google Print đang được vận hành thử nghiệm. Số liệu thống kê

gần đây của Google là 3,3 tỷ trang web được kết nối vào tháng 8-2003, là 400 triệu hình ảnh vào tháng 11/2002).

[21]. Yahoo! http://www.yahoo.com/.

# PHỤ LỤC

1. Script để tạo các bảng lưu trữ chỉ mục tương tự

```
DROP table IF EXISTS sim_urlcontent;
DROP table IF EXISTS sim_urlwnd;
DROP table IF EXISTS sim_urlsim;

DROP table IF EXISTS Alias;
DROP table IF EXISTS Category;
DROP table IF EXISTS Editor;
DROP table IF EXISTS Link;
DROP table IF EXISTS Newsgroup;

#table sim_urlword
#url_id: id of url
#bag: bag of word = (word_id1,df1;word_idi,dfi; ...;word_idn,dfn)
CREATE TABLE sim_urlcontent
    (url_id integer primary key
    ,word_count  integer not null
    ,words longblob
    );

# table url window
# url_id: id of url
# refer_id: url_id references to this url
# left url window in content of refer_id references to this url
# center url window in content of refer_id references to this url
# right url window in content of refer_id references to this url

CREATE TABLE sim_urlwnd
```

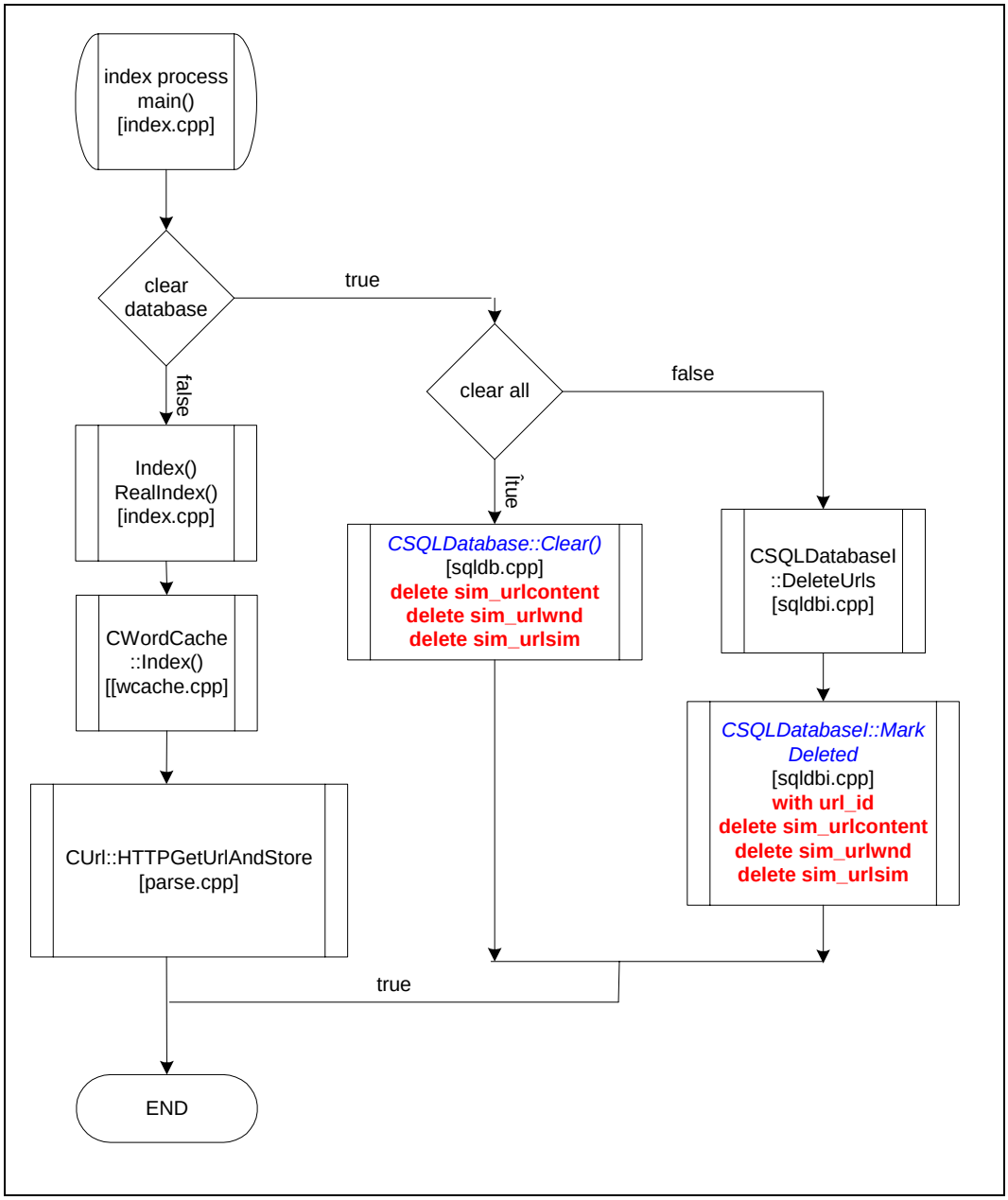
```
        (id integer auto_increment primary key
        ,url_id integer not null
        ,refer_by integer not null
        ,word_count  integer not null
        ,words longblob
        ,unique index (url_id, refer_by)
        ,index (url_id, refer_by)
        );

    #table url sim
    #url_id: id of url
    #url_sim:     similation     url     =     (url_id1,sim1;url_idi,simi;
...;url_idn,simn)
    CREATE TABLE sim_urlsim
        (id integer auto_increment primary key
        ,url_id1 integer not null
        ,url_id2 integer not null
        ,sim float not null
        ,unique index(url_id1, url_id2)
        ,index(url_id1)
        ,index(url_id2)
        );

    CREATE TABLE sim_urltmp
        (url_id integer primary key
        );
    # using tool from http://odp.locallink.net/setup/
    # Table structure for table 'Alias'
    #
    CREATE TABLE Alias (
      aliasID int(10) NOT NULL auto_increment,
      title varchar(255) DEFAULT '' NOT NULL,
```

```
  targetCategory varchar(255) DEFAULT '' NOT NULL,
  parentTopic varchar(255) DEFAULT '' NOT NULL,
  PRIMARY KEY (aliasID),
  KEY alias_targetCategory_index (targetCategory),
  KEY alias_parentTopic_index (parentTopic)
);

#
# Table structure for table 'Category'
#
CREATE TABLE Category (
  topic varchar(255) DEFAULT '' NOT NULL,
  topicShort varchar(50) DEFAULT '' NOT NULL,
  parentTopic varchar(255),
  description varchar(255) DEFAULT '' NOT NULL,
  lastUpdate varchar(255) DEFAULT '' NOT NULL,
  PRIMARY KEY (topic),
  KEY category_parentTopic_index (parentTopic),
  KEY category_topicShort_index (topicShort)
);

#
# Table structure for table 'Editor'
#
CREATE TABLE Editor (
  editorID int(10) NOT NULL auto_increment,
  parentTopic varchar(255) DEFAULT '' NOT NULL,
  editorName varchar(50) DEFAULT '' NOT NULL,
  PRIMARY KEY (editorID),
  KEY category_parentTopic_index (parentTopic)
);
```

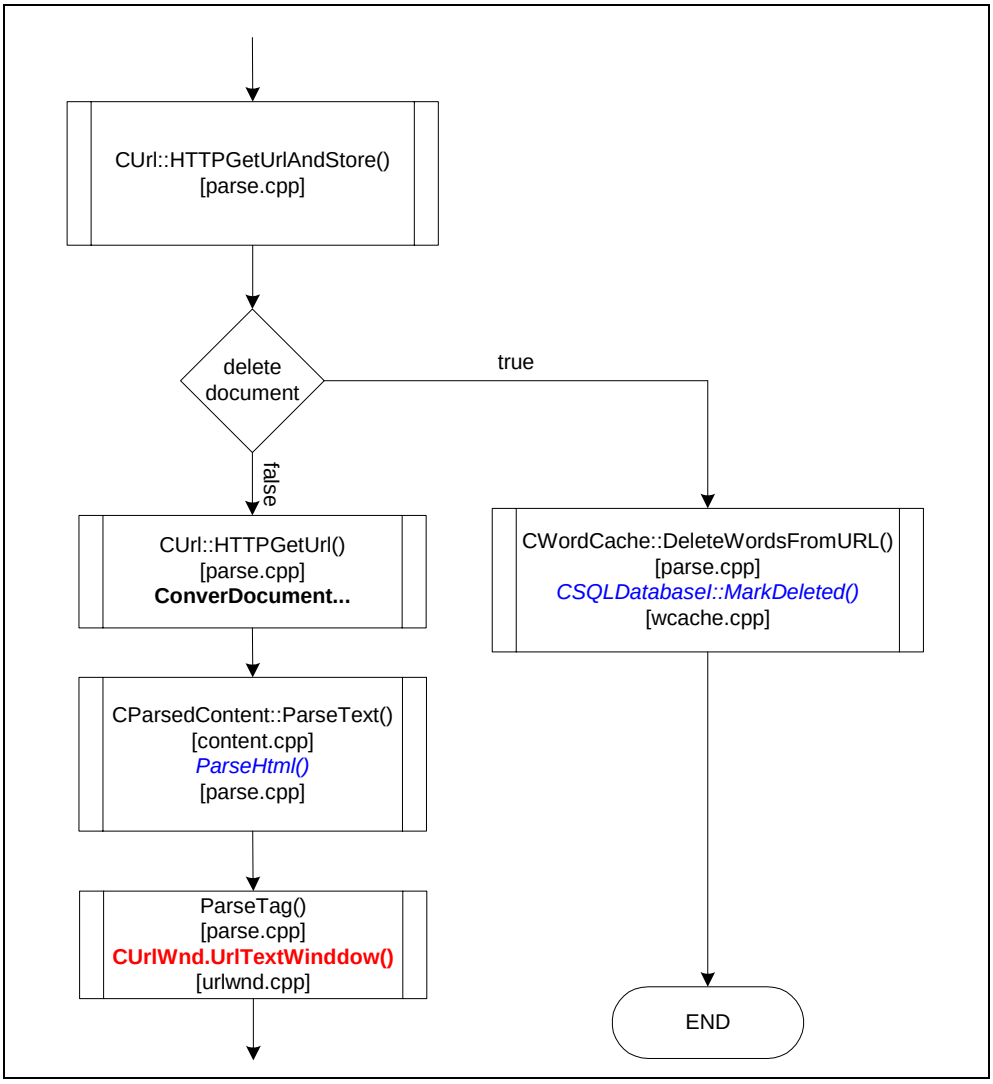
```
#
# Table structure for table 'Link'
#
CREATE TABLE Link (
  linkID int(10) NOT NULL auto_increment,
  page varchar(255) DEFAULT '' NOT NULL,
  parentTopic varchar(255) DEFAULT '' NOT NULL,
  title varchar(255) DEFAULT '' NOT NULL,
  description varchar(255) DEFAULT '' NOT NULL,
  PRIMARY KEY (linkID),
  KEY link_parentTopic_index (parentTopic),
  KEY link_page_index (page),
  KEY link_title_index (title),
  KEY link_description_index (description)
);

#
# Table structure for table 'Newsgroup'
#
CREATE TABLE Newsgroup (
  newsID int(10) NOT NULL auto_increment,
  newsgroupName varchar(255) DEFAULT '' NOT NULL,
  parentTopic varchar(255) DEFAULT '' NOT NULL,
  PRIMARY KEY (newsID),
  KEY newsgroup_parentTopic_index (parentTopic)
);
```

Bảng 16. Nội dung các lệnh tạo cấu trúc dữ liệu bổ sung cho VietSeek

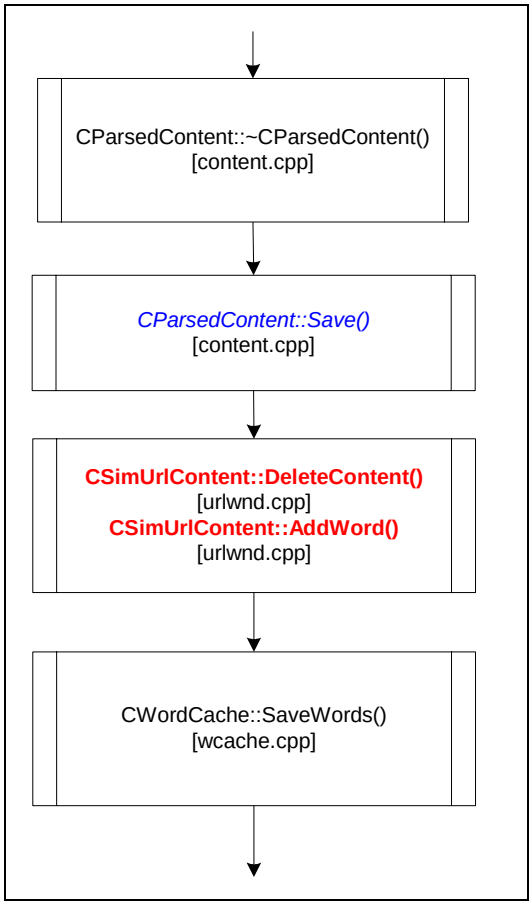
2. Phân tích các modul của VietSeek cần hiệu chỉnh để bổ sung chức năng tìm kiếm tương tự




*Hình 21. Sơ đồ khối của modul index*




*Hình 22. Sơ đồ khối của modul HTTPGetAndStore*




*Hình 23. Sơ đồ khối của modul CParsedContent*