|  |
| --- |
| **ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  **KHOA CÔNG NGHỆ PHẦN MỀM** |

|  |
| --- |
| **KHÓA LUẬN TỐT NGHIỆP** |
| **SỬ DỤNG SOCIAL ANNOTATION, FUZZY NEURAL ARTMAP, WEBLOG ĐỂ LÀM NỔI BẬT CÔNG CỤ TÌM KIẾM CỤC BỘ** |

Giảng viên hướng dẫn: **PGS.TS. VŨ THANH NGUYÊN**

**GV.TRẦN ANH DŨNG**

Sinh viên thực hiện: TRẦN ANH TUẤN (07520392)

LÊ DUY TÂN (07520445)

Lớp: CNPM02

Khóa: 02

***TP. Hồ Chí Minh,tháng năm***

|  |
| --- |
| **ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  **KHOA CÔNG NGHỆ PHẦN MỀM** |

|  |
| --- |
| **KHÓA LUẬN TỐT NGHIỆP** |
| SỬ DỤNG SOCIAL ANNOTATION, FUZZY NEURAL ARTMAP, WEBLOG ĐỂ LÀM NỔI BẬT CÔNG CỤ TÌM KIẾM CỤC BỘ |

Giảng viên hướng dẫn: **PGS.TS. VŨ THANH NGUYÊN**

**GV.TRẦN ANH DŨNG**

Sinh viên thực hiện: TRẦN ANH TUẤN (07520392)

LÊ DUY TÂN (07520445)

Lớp: CNPM02

Khóa: 02

***TP. Hồ Chí Minh,tháng năm***

LỜI MỞ ĐẦU

Từ 20 năm nay, Lý thuyết tập mờ và Mạng nơ-ron nhân tạo đã phát triển rất nhanh và đa dạng. Công nghệ mờ và công nghệ mạng nơ-ron đã cung cấp những công nghệ mới cho các ngành công nghiệp và làm ra nhiều sản phẩm thông minh, đáp ứng nhu cầu thị trường cần có những bộ điều khiển linh hoạt hơn, những thiết bị “biết” làm việc với bài toán khó, phải xử lý nhiều loại thông tin mập mờ, chưa đầy đủ và thiếu chính xác. Hai công nghệ hiện đại này là hai trụ cột chính để tạo nên công nghiệp tích hợp mới, công nghệ tính toán mềm (Soft Computing).

Đồng thời hiện nay, với số lượng các trang Web đã lên đến hàng tỉ, nên việc tìm ra số trang có chứa nội dụng đòi hỏi của từ khoá có thể lên đến hàng triệu trang. Do đó, việc hiển thị các trang tìm thấy theo đúng thứ tự quan trọng của các trang và theo mong muốn của người dùng cũng là một trở ngại lớn đòi hỏi sự chắt lọc từ máy truy tìm và sự khéo léo về cách thức đặt ra từ khoá từ người dùng máy.Để giải quyết vấn đề đó các máy tìm kiếm đã ra đời, họ dùng các thuật toán để đánh giá nội dung trang web và quyết định thứ tự chúng. Hiện trên internet có 3 công cụ hỗ trợ tìm kiếm hiệu quả nhất là:

* <http://www.google.com>
* <http://www.bing.com>
* <http://www.yahoo.com>

Các công cụ này sử dụng rất nhiều cách để đánh giá nội dung trang web (như google dùng tới trên 200 cách) và các thuật toán chính xác này đều được bí mật.

Do đó nhóm chúng tôi dưới sự hướng dẫn của Thầy PGS.TS Vũ Thanh Nguyên và Thầy Trần Anh Dũng về các giải thuật Fuzzy Artmap Neural Network, Social Annotation, Weblog chúng tôi quyết định sử dụng những giải thuật này để tối ưu hóa nội dung các trang web và hỗ trợ các máy tìm kiếm có thể tìm kiếm hiệu quả hơn. Mục tiêu của đề tài là phân tích weblog từ server của các trang web và sắp xếp thứ tự của các trang web theo số lượng từ khóa có chứa trong trang web đó bằng các giải thuật đã được đề ra.

Kết quả là nhờ những thuật toán chính xác và các cách xử lý hợp lý nhóm đã thành công trong việc phân tích đánh giá các trang web thông qua weblog và xây dựng được một công cụ tìm kiếm cục bộ để mình họa các kết quả tìm kiếm sau khi đã được phân tích và đánh giá.

Do nhiều hạn chế, đặc biệt là hạn chế về thời gian, bài báo cáo này không tránh khỏi sự thiếu sót, khuyết điểm. Mong được nhận sự góp ý chân thành từ quý Thầy (Cô).

*Nhóm sinh viên thực hiện*

**Trần Anh Tuấn**

**Lê Duy Tân**

LỜI CẢM ƠN

Chúng em chân thành cám ơn của giáo viên hướng dẫn Thầy Vũ Thanh Nguyên và Thầy Trần Anh Dũng trong thời gian qua đã nhiệt tình giúp đỡ chúng em có thể hoàn thành khóa luận tốt nghiệp đúng thời hạn.

Mặc dù chúng em đã cố gắng và nỗ lực rất nhiềuđể hoàn thànhđề tài này nhưng trong quá trình thực hiện không tránh khỏi một số sai sót.Nhóm chúng em rất mong nhậnđược sự gópý của thầy (cô) và các bạnđểđề tài của chúng em được hoàn thiện hơn.

Một lần nữa chúng em xin chân thành cảmơn thầy cô và các bạn!

TP. Hồ Chí Minh, ngày … tháng … năm …

*Nhóm sinh viên thực hiện*

**Trần Anh Tuấn**

**Lê Duy Tân**

NHẬN XÉT

**(Của giảng viên hướng dẫn)**

NHẬN XÉT

**(Của giảng viên phản biện)**

MỤC LỤC

[Chương 1 - GIỚI THIỆU 1](#_Toc333470512)

[1.1. Giới thiệu tổng quan vấn đề 1](#_Toc333470513)

[1.2. Đặt vấn đề: 3](#_Toc333470514)

[1.3. Định hướng phương pháp giải quyết đề tài: 5](#_Toc333470515)

[1.4. Mô tả nội dung báo cáo 6](#_Toc333470516)

[Chương 2 - FUZZY ARTMAP NEURAL NETWORK 8](#_Toc333470517)

[2.1. Đánh giá trang web: 8](#_Toc333470518)

[2.2. Hệ thống Fuzzy Artmap Neural Network: 9](#_Toc333470519)

[2.3. Giải thuật Fuzzy Artmap Neural Network: 10](#_Toc333470520)

[Chương 3 - WEBLOG VÀ GIẢI THUẬT LPAGERANK 13](#_Toc333470521)

[3.1. Xử lý Weblog: 13](#_Toc333470522)

[3.1.1. Thông tin trên weblog: 13](#_Toc333470523)

[3.1.2. Lọc Weblog: 14](#_Toc333470524)

[3.2. Giải thuật PageRank: 15](#_Toc333470525)

[3.3. Giải thuật LpageRank: 17](#_Toc333470526)

[3.4. Giải thuật xây dựng đồ thị xác suất: 18](#_Toc333470527)

[3.5. Giải thuật LpageRank được cải tiến từ giải thuật PageRank với Weblog: 19](#_Toc333470528)

[3.6. Ứng dụng giải thuật TF- IDF kết hợp cùng với giải thuật LpageRank trong công cụ tìm kiếm: 20](#_Toc333470529)

[3.6.1. Sử dụng giải thuật TF – IDF cho việc đánh trọng số câu truy vấn: 20](#_Toc333470530)

[3.6.2. Giải thuật tính TF – IDF cho một tài liệu: 22](#_Toc333470531)

[3.6.3. Ứng dụng giải thuật TF- IDF trong công cụ tìm kiếm: 24](#_Toc333470532)

[Chương 4 - SOCIAL ANNOTATION VÀ CÁC GIẢI THUẬT LIÊN QUAN 25](#_Toc333470533)

[4.1. Ứng dụng chú thích cộng đồng (Social Annotation) cải thiện việc tìm kiếm thông tin trên mạng: 25](#_Toc333470534)

[Chương 5 - Xây dỰng CÔNG CỤ TÌM KIẾM Ứng dỤng các thuẬt GiẢi Social Annotation, Fuzzy ArTmap, NEURAL netWork 28](#_Toc333470535)

[5.1. Giới thiệu công cụ tìm kiếm: 28](#_Toc333470536)

[5.1.1. Tên công cụ: 28](#_Toc333470537)

[5.1.2. Môi trường phát triển: 28](#_Toc333470538)

[5.2. Phân tích: 28](#_Toc333470539)

[5.2.1. Khảo sát hiện trạng: 28](#_Toc333470540)

[5.2.2. Yêu cầu phi chức năng: 29](#_Toc333470541)

[5.2.3. Yêu cầu chức năng 29](#_Toc333470542)

[5.2.4. Mô hình usecase: 30](#_Toc333470543)

[5.3. Thiết kế: 30](#_Toc333470544)

[5.3.1. Mô hình chương trình: 30](#_Toc333470545)

[5.3.2. Sơ đồ luồng dữ liệu: 35](#_Toc333470546)

[5.4. Phát triển ứng dụng: 35](#_Toc333470547)

[5.4.1. Weblog Spider: 35](#_Toc333470548)

[5.4.2. Log Miner: 43](#_Toc333470549)

[5.4.3. Webpage Classifier: 49](#_Toc333470550)

[5.4.4. Social Annotation 53](#_Toc333470551)

[5.4.5. Search Engine: 58](#_Toc333470552)

[5.4.6. Lưu trữ dữ liệu: 60](#_Toc333470553)

[5.5. Demo: 65](#_Toc333470554)

[5.5.1. Demo: 65](#_Toc333470555)

[5.5.2. Kết quả thực hiện: 69](#_Toc333470556)

[Chương 6 - KẾt LuẬn 71](#_Toc333470557)

[6.1. Kết luận: 71](#_Toc333470558)

[6.1.1. Ưu điểm 71](#_Toc333470559)

[6.1.2. Hạn chế: 72](#_Toc333470560)

[6.1.3. Những nét mới của khóa luận: 72](#_Toc333470561)

[6.2. Hưởng mở rộng đề tài: 72](#_Toc333470562)

[Danh mỤc Tài LiỆu Tham KhẢo 74](#_Toc333470563)

Danh mỤc các bẢng, sơ đỒ, hình

**Danh mục hình**

[Hình 2‑1 Sơ đồ khối của hệ thống Fuzzy ARTMAP 9](#_Toc333470434)

[Hình 3‑1 Quá trình xử lý thông tin web log 14](#_Toc333470435)

[Hình 3‑2 Biếu diễn độ tương tự của vector tài liệu d và vector truy vấn q 21](#_Toc333470436)

[Hình 4‑1 Minh họa ba thành phần trong Social Search 26](#_Toc333470437)

[Hình 5‑1 Kiến trúc công cụ tìm kiếm 35](file:///C:\Users\MartinLe\Dropbox\Do%20an\Bao%20cao%20Search%20engine%20(ver%201.3).docx#_Toc333470438)

[Hình 5‑2 Mô hình cấu trúc dữ liệu của WebSpider 36](#_Toc333470439)

[Hình 5‑3 Thông tin từ khóa và trang web 37](#_Toc333470440)

[Hình 5‑4 Cấu trúc WebSpider 40](#_Toc333470441)

[Hình 5‑5 Lưu đồ giải thuật TF-IDF 42](#_Toc333470442)

[Hình 5‑6 Quá trình phân tích website 45](#_Toc333470443)

[Hình 5‑7 Sơ đồ hoạt động của Log Miner 48](#_Toc333470444)

[Hình 5‑8 Quan hệ giữa các lớp 51](#_Toc333470445)

[Hình 5‑9 Hoạt động của bộ phận lớp 52](#_Toc333470446)

[Hình 5‑10 Chỉ số xếp hạng của một trang web 54](file:///C:\Users\MartinLe\Dropbox\Do%20an\Bao%20cao%20Search%20engine%20(ver%201.3).docx#_Toc333470447)

[Hình 5‑11 Sự kết hợp giữa SSR và SPR (tham khảo Optimizing Web Search Using Social Annotations – Dynamic Ranking). 54](#_Toc333470448)

[Hình 5‑12 Phân bố số lượng trung bình trên PageRank (tham khảo Optimizing Web Search Using Social Annotations). 56](#_Toc333470449)

[Hình 5‑13 Những phân tích phân bố của Social Annotation (tham khảo Optimizing Web Search Using Social Annotations). 57](#_Toc333470450)

[Hình 5‑14Quá trình tìm kiếm của Search Engine 58](#_Toc333470451)

[Hình 5‑15 Chỉ số xếp hạng của một trang web 59](#_Toc333470452)

[Hình 5‑16 Mô tả cơ sở dữ liệu 64](#_Toc333470453)

[Hình 5‑17 Phân tích trang web 65](#_Toc333470454)

[Hình 5‑18 Kêt quả phân tích 66](#_Toc333470455)

[Hình 5‑19 Đánh giá trang web 67](#_Toc333470456)

[Hình 5‑20 Kết quả tìm kiếm với thuật giả PageRank 68](#_Toc333470457)

[Hình 5‑21 Kết quả tìm kiếm với thuật giả LPageRank 69](#_Toc333470458)

[Hình 5‑22 Kết quả tìm kiếm với thuật giả Social Annotation 69](#_Toc333470459)

**Danh mục bảng**

[Bảng 5‑1 Đánh giá kết quả của giải thuật Social Annotation (tham khảo Optimizing Web Search Using Social Annotations). 55](#_Toc333470357)

[Bảng 5‑2 Case Study của SocialPageRank với PageRank (tham khảo Optimizing Web Search Using Social Annotations). 57](#_Toc333470358)

# GIỚI THIỆU

## Giới thiệu tổng quan vấn đề

Khái niệm mạng nơ-ron được bắt đầu vào cuối thập kỷ 1800 khi người ta cố gắng mô tả hoạt động của trí tuệ con người. Ý tưởng này bắt đầu được áp dụng cho các mô hình tính toán từ mạng Perceptron. Đầu thập kỷ 1950 Friedrich Hayek là người đầu tiên khẳng định ý tưởng về trật tự tự phát trong não xuất phát từ các mạng phân tán gồm các đơn vị đơn giản (nơ-ron). Cuối thập kỷ 1940, Donnald Hebb đưa ra giả thuyết đầu tiên về một cơ chế thần kinh mềm dẻo (*neural plasticity*), Hebbian learning (???). Hebbian learning được coi là một quy tắc 'điển hình' của học không có giám sát. Nó (và các biến thể) là mô hình thời kỳ đầu của long term potentiation (*tạo tiềm lực dài hạn*).

Perceptron là một bộ phân loại tuyến tính dành cho việc phân loại dữ liệu  x \in R^n xác định bằng các tham số w \in R^n, b \in R và một hàm đầu ra *f* = *w*'*x* + *b*. Các tham số của nó được thích nghi với một quy tắc tùy biến (ad-hoc) tương tự với xuống dốc ngẫu nhiên (*stochastic steepest gradient descent*). Perceptron chỉ có thể phân loại hoàn hảo một tập dữ liệu mà các lớp khác nhau là phân tách tuyến tính (*linearly separable*) trong không gian đầu vào.Nó thường thất bại hoàn toàn đối với dữ liệu không chia tách được. Sự phát triển của thuật toán này ban đầu đã tạo ra một số hứng khởi, phần vì mối quan hệ của nó đối với các cơ chế sinh học. Sau này, phát hiện về điểm yếu này đã làm cho các mô hình Perceptron bị bỏ mặc cho đến khi các mô hình phi tuyến được đưa ra.

Cognitron (1975) là một mạng nơ-ron đa tầng thời kỳ đầu với một thuật toán huấn luyện.Các chiến lược thần kinh khác nhau sẽ khác nhau về cấu trúc thực sự của mạng và các phương pháp thiết lập trọng số cho các kết nối.Mỗi dạng có các ưu điểm và nhược điểm riêng. Mạng có thể lan truyền thông tin chỉ theo một hướng, hoặc thông tin có thể được đẩy đi đẩy lại cho đến khi tại một nút xuất hiện sự tự kích hoạt và mạng sẽ dừng tại một trạng thái kết thúc. Khả năng truyền dữ liệu hai chiều giữa các nơ-ron/nút còn được sử dụng trong mạng Hopfield (1982), và sự chuyên hóa các tầng nút này cho các mục đích cụ thể đã được đưa ra trong mạng nơ-ron lai (*hybrid neural network*) đầu tiên.

Social Annotation là một chú thích trực tuyến kết hợp với một nguồn tài nguyên Web.CritLink có lẽ là hệ thống chú thích đầu tiên, đến 1997 – 1998 thì được phát triển bởi Ka-Ping Yee của Đại học California. Với hệ thống chú thích, người dùng có thể thêm, sửa đổi hoặc loại bỏ thông tin từ một tài nguyên web mà không sửa đổi tài nguyên riêng của mình. Các chú thích có thể xem như là một lớp trên đầu trang của tài nguyên hiện có, và lớp chú thích này thường được hiển thị cho người dùng khác, những người chia sẽ cùng một hệ thống Social Annotation.

Công cụ tìm kiếm cũng được biết đến như là một trong số những ngôi sao sáng nhất trên internet trong những năm đầu thập niên 90. Năm 1993, Matthew Gray, sau này làm việc tại MIT, tạo ra máy thăm dò trang web đầu tiên, và đến cuối năm 1993, công cụ tìm kiếm web Aliweb đầu tiên xuất hiện. Đến 1994, hai học viên cao học David Filo và Jerry Yang bắt đầu tuyển chọn các web sites thông dụng theo các chủ đề phân cấp và hình thành nên Yahoo. Và cũng bắt đầu từ đây, các công cụ tìm kiếm như AOL, Lycos, DEC hình thành. Năm 1998, Larry Page và Sergey Brin nghiên cứu triển khai Google, công cụ tìm kiếm Google này đã gây một sự chú ý nổi bật.Ngày nay, Google đang thống trị trong lĩnh vực tìm kiếm với 72.25% thị phần trong tháng 12/2009 và vẫn đang tiếp tục duy trì và tăng trưởng.

Tìm kiếm cục bộ trong trang web được định nghĩa là công cụ tìm kiếm trên các trang web của cùng một web site.Đây là một trong những vấn đề khá nổi trội những năm gần đây trong hướng nghiên cứu về tìm kiếm thông tin nhất là tìm kiếm thông tin trên web.Trong một số công cụ tìm kiếm thương mại nổi tiếng, như Google, Yahoo cũng đã phát triển những chức năng cho phép người dùng hạn chế tìm kiếm trong một tên miền, hay tìm với những nội dung riêng biệt trên một trang web cụ thể như tìm dựa trên các định dạng tập tin, dựa trên cấp độ của liên kết, cấp độ thư mục,… Các công cụ tìm kiếm cục bộ thường sử dụng để tìm kiếm sản phẩm, tài liệu hay những thông tin liên quan thuộc về trang web đó mà thường những công cụ tìm kiếm toàn cục khó tìm được một cách chính xác hoặc nếu có tìm được thì cũng rất khó giới hạn phạm vi chính xác của dữ liệu trả về. Chính vì vậy mà các công cụ tìm kiếm cục bộ trên trang web vẫn rất cần thiết và tồn tại song hành cùng với sự phát triển một cách nhanh chóng của các công cụ tìm kiếm toàn cục trên web.Những công cụ này hỗ trợ người dùng rất nhiều và hữu ích khi truy cập vào một trang web cụ thể.Nó góp phần rất lớn cho việc hướng người dùng đến được chính xác nội dung cần tìm trong trang web.

Trong khóa luận này sẽ xây dựng một công cụ tìm kiếm theo từ khóa sử dụng weblog, giải thuật LpageRank, mạng neural fuzzy ARTMAP và Social Annotation để củng cố xếp hạng trang. Như vậy công cụ phân tích và đánh giá không chỉ khắc phục được sự thô cứng trong việc tìm kiếm chính xác các từ khóa trong câu truy vấn mà còn phản ánh độ quan tâm của người dùng, độ tốt của trang web lên kết quả tìm kiếm.

## Đặt vấn đề:

Trong thời đại internet ngày nay, tìm kiếm chính xác thông tin cần thiết ngày càng quan trọng hơn,đặc biệt là thông tin và số lượng tài liệu mỗi ngày tăng theo cấp số nhân, nếu việc tìm kiếm thông tin có liên quan không đáp ứng được nhu cầu của người dùng có thể dẫn đến lãng phí thời gian, giảm năng suất, và giảm thu nhập.Một vấn đề trong việc kinh doanh của các doanh nghiệp là họ thường đưa các thông tin sản phẩm và dịch vụ lên trang web của họ,số lượt truy cập và khách hàng tiềm năng có thể dễ dàng chuyển sang một trang web khác nếu họ không thể tìm thấy những gì họ đang tìm kiếm. Điều này dẫn đến mất khách hàng, giảm doanh thu, và làm giảm lợi nhuận.

Công cụ tìm kiếm là một công cụ phổ biến và cần thiết cho người dùng web, tuy nhiên người dùng không thực sự hài lòng khi sử dụng chúng. Khi một câu truy vấn được tìm kiếm, nhiều kết quả không phù hợp được trả về, và đa số những kết quả mà người dùng mong đợi đều không nằm trong danh sách những kết quả đầu tiên, và người dùng cảm thấy thật khó để có thể chọn một kết quả mà mình mong muốn cũng như đưa ra câu truy vấn chính xác để cho công cụ tìm kiếm hiểu mình mong muốn điều gì. Và đã có những nghiên cứu để cải tiến chất lượng của công cụ tìm kiếm. Đa số những nghiên cứu đó theo một số khía cạnh sau đây:

- Sắp xếp lại các trang web theo độ tương đồng với truy vấn tìm kiếm. Trong lĩnh vực này, một số phương pháp được sử dụng như: trích xuất siêu dữ liệu, phân tích cấu trúc link, khai thác nhật ký tìm kiếm, khai thác dạng câu truy vấn, khai thác mục đích sử dụng web của người dùng, … Khía cạnh này mang tính khả thi cao vì quan tâm đến người dùng, phân tích xem người dùng mong muốn gì và cố gắng trả về kết quả tìm kiếm như mong muốn.

- Một số các tiếp cận khác như sắp xếp theo độ ưu tiên của trang web mà không quan tâm đến truy vấn của người dùng như PageRank, HITS, … Những phương pháp này tính toán hạng của mỗi trang web dựa trên siêu liên kết giữa các trang web với nhau. Nếu một trang web có nhiều link trỏ đến, hạng của trang cũng cao hơn, và trang này cũng quan trọng hơn. Trong một website, các siêu liên kết thường được tạo ra để sắp xếp toàn bộ website theo cấu trúc phân cấp tuyến tính. Khi áp dụng những thuật toán trên cho một trang web cục bộ, những trang như trang chủ sẽ có hạng cao và người dùng web không thường tìm kiếm những trang này.

Và như vậy việc sử dụng những phương pháp nào xây dựng công cụ tìm kiếm cục bộ để công cụ tìm kiếm trả về những kết quả phù hợp nhất cho người sử dụng là mục tiêu của khóa luận.

## Định hướng phương pháp giải quyết đề tài:

Trong đề tài này sẽ kết hợp giữa phân tích và đánh giá cục bộ trang web:

* Sử dụng nhật ký truy cập web của người dùng web server logs. Web serverlogs ghi lại những thông tin chi tiết về hành vi truy cập web của người dùng, khi người dùng duyệt qua một website. Những trang mà họ chọn phản ánh được sự đánh giá của họ về mối liên quan của nội dung trang web với thông tin mà họ cần. Vì thế mà việc khai thác những thông tin trong web server logs như là một tiêu chí xếp hạng của các trang web sẽ phản ánh được độ quan tâm của người dùng lên những thông tin mà người dùng đang tìm kiếm.
* Sử dụng mạng neural fuzzy ARTMAP để phân loại, đánh giá những trang web trong website dựa trên các tiêu chuẩn đánh giá (sẽ được khóa luận nói rõ hơn trong chương 2), từ tập kết quả đầu ra sẽ được dùng để tối ưu hóa cho công cụ tìm kiếm. Việc sử dụng mạng neural ARTMAP là vì tốc độ phân lớp nhanh và có độ chính xác cao trên tập dữ liệu lớn, rất phù hợp cho công cụ tìm kiếm vì phải xử lý khối lượng dữ liệu trên hàng ngàn trang web và tài liệu.
* Sử dụng Social Annotation thường hiệu quả, tóm tắt được nhiều mặt của một trang Web và cung cấp siêu dữ liệu mới cho xếp hạng tương đồng. Chú thích của một số trang Web khá thưa thớt và phương pháp tiếp cận dựa trên term-matching bị hảnh hưởng ít nhiều về vấn đề đồng nghĩa. Vì thế việc đặt ra vấn đề giải quyết đó là khai thác cấu trúc tính đồng nghĩa của Social Annotation thông qua hai giải thuật: SocialSimRank và SocialPageRank.

Trong khóa luận này sẽ xây dựng một công cụ tìm kiếm cho website cục bộ với các chức năng:

+ Xây dựng Spider sẽ dò tìm tất cả các trang web của website và lưu trang web với các thuộc tính: id, nội dung, tiêu đề, diễn tả, định dạng, url. Đồng thời đánh chỉ mục từ khóa, lưu vào cơ sở dữ liệu vị trí, trọng số TF-IDF của từng từ.Spider còn lưu cấu trúc của website đế tính giá trị PageRank, độ sâu của trang web, lưu những trang bị lỗi.

+ Từ thông tin weblog, Spider tính giá trị LpageRank cũng như Social Sim Rank hay Social Page Rank, tần số cập nhật của trang web, số lượng truy cập, thời gian truy cập, tần số lỗi, những giá trị này được sử dụng để tối ưu hóa cho công cụ tìm kiếm trong thuật toán LpageRank, Social Annotation và mạng neural fuzzy ARTMAP.

+ Xử lý truy vấn: bộ xử lý câu truy vấn của người dùng xem người dùng đang tìm kiếm theo tiêu đề, nội dung, url, hoặc cả ba loại tìm kiếm trên, tìm kiếm theo toán tử AND, OR, tìm kiếm theo định dạng của file (ví dụ tìm kiếm theo định dạng file html, pdf, doc, docx, xls, xlsx, ppt, pptx…), tìm kiếm theo phương pháp, TFIDF & LpageRank& mạng neural fuzzy ARTMAP, Social Annanotation &mạng neural fuzzy ARTMAP.

+ Tìm kiếm: bộ tìm kiếm với đầu vào là truy vấn của người dùng, bộ tìm kiếm sẽ tìm kiếm trong cơ sở dữ liệu và trả về danh sách những trang kết quả tìm được, mỗi kết quả được trả về bao gồm tiêu đề, định dạng, một đoạn nội dung của trang web có chứa từ khóa tìm kiếm được in đậm, url, ngày spider trích xuất thông tin trang web.

## Mô tả nội dung báo cáo

Báo cáo khoá luận này gồm 6 chương giới thiệu các kiến thức về mạng nơ-ron, weblog, fuzzy artmap, social annotation và demo chương trình minh họa:

* **Chương 2 – Fuzzy Artmap Neurak Network:** giới thiệu về mạng nơ-ron dùng để phân lớp trang web.
* **Chương 3 – Weblog và giải thuật LpageRank:** giới thiệu về weblog, cách trích xuất thông tin từ weblog và giải thuật LpageRank.
* **Chương 4 – Social Annotation và giải thuật liên quan:**giới thiệu về giải thuật Social Annotation dùng để đánh giá xếp hạng trang.
* **Chương 5 – Xây dựng công cụ tìm kiếm ứng dụng các giải thuật Social Annotation, Fuzzy Neural Artmap:**giới thiệu về demo minh họa chương trình bao gồm các lớp và kết quả đạt được.
* **Chương 6 – Kết luận**: kết luận ưu điểm và hạn chế của khóa luận kèm theo hướng mở rộng đề tài.

# FUZZY ARTMAP NEURAL NETWORK

## Đánh giá trang web:

Mô hình sử dụng hệ thống fuzzy ARTMAP neural network sau khi được huấn luyện sẽ có thể thi hành như một công cụ để phân lớp tập dữ liệu truy cập của người dùng.Các tiêu chuẩn đánh giá các trang web mà người dùng truy cập sẽ được chuyển vào mạng neural để phân lớp các trang có cùng độ tương quan với nhau.Từ tập kết quả đầu ra của mạng neural này, sẽ được sử dụng để tối ưu kết quả cho công cụ tìm kiếm.Sử dụng hệ thống fuzzy ARTMAP neural network làm giảm đi việc tính toán phân lớp các trang web và làm tăng tốc độ cho ra kết quả so với những kỹ thuật phân lớp khác khi áp dụng với công cụ tìm kiếm.

Xét một tập các trang web A{a1, a2,…, an} của người dùng truy cập được trích xuất từ web server logs, mỗi trang web sẽ có một vector ai{c1, c2, c3, c4, c5, c6} biểu diễn độ quan tâm của người dùng và độ quan trọng của trang web với 6 tiêu chuẩn đánh giá. 6 tiêu chuẩn trên bao gồm:

+ c1: điểm của mỗi trang trong giải thuật LpageRank

+ c2: tần số cập nhật của trang web bằng số lần cập nhật của trang / số lần cập nhật của trang được cập nhật nhiều nhất trong tuần.

+ c3: độ sâu của link bằng độ sâu của link / độ sâu lớn nhất trong website.

+ c4: tần suất lỗi của trang bằng số lần lỗi của trang / số lần lỗi của trang có số lần lỗi nhiều nhất.

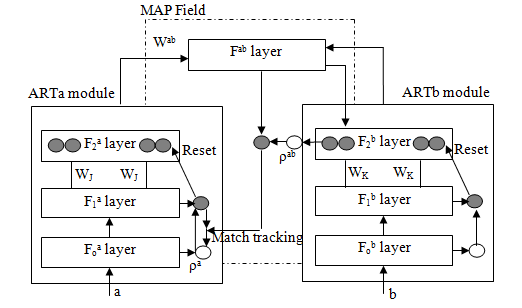
+ c5: tần suất truy cập bằng số lần truy cập của trang / số lần truy cập của trang có số lần truy cập nhiều nhất.

+ c6: thời gian truy cập bằng thời gian truy cập của trang / thời gian của trang truy cập nhiều nhất.

Tập con các vector ak trên được chọn ra cùng với tập các vector bj{b1, b2, …, bk} với bk là kết quả dự đoán lớp chính xác của ai sẽ được sử dụng làm tập dữ liệu học cho mạng. Sau khi tiến hành phân lớp, kết quả các trang web tương tự với nhau sẽ thuộc chung một lớp pj{ak, ak+1,…} trong p lớp.

## Hệ thống Fuzzy Artmap Neural Network:

Hệ thống fuzzy ARTMAP neural network là mạng neural học giám sát có khả năng nhận dạng nhanh và ổn định, có khả năng dự báo chính xác phù hợp cho phân lớp, cấu trúc của mạng là cấu trúc tiến hóa và là sự kết hợp của hai mạng fuzzy ART, ARTa và ARTb. Hai mạng này được liên kết với nhau bởi mạng neural MAP Field.

****

Hình ‑ Sơ đồ khối của hệ thống Fuzzy ARTMAP

Quá trình học của mạng ARTMAP là một quá trình hoạt động đồng thời của 2 mạng ARTa và ARTb. Cho mẫu đầu vào a vào lớp Foa của mạng ARTa và mẫu đầu vào b vào lớp Fob của mạng ARTb với b là mẫu dự đoán chính xác của a. Lớp mã hóa Fo tạo ra vector **A = (a, ac)** trong ARTa và **B = (b, bc)** trong ARTb với ac, bc tương ứng là phần bù của a, b. Lớp F1 với vector đầu vào x (xa = Acho ARTa và xb = B cho ARTb) tạo ra những ma trận trọng số thích nghi Wjliên kết giữa lớp F1 và lớp F2 **(**WjachoARTa và Wjb cho ARTb). Vector y (ya = Acho ARTa và yb = B cho ARTb) biểu diễn sự kích hoạt của lớp F2. Mạng fuzzy ARTMAP có các tham số giống với mạng fuzzy ART là tham số α > 0 được sử dụng trong quá trình so sánh giữa các neural trong F2, tham số huấn luyện β ∈ [0 1] biểu diễn tốc độ học, và tham số vigilance ρ ∈ [0 1] được sử dụng trong quá trình tìm neural chiến thắng trong F2, đồng thời mạng fuzzy ARTMAP có thêm giá trị nhỏ nhất của tham số vigilance của ARTa *ρ­¬a*, tham số vigilance *ρ­ab* và tham số huấn luyện *βab* của lớp MAP.

Quá trình tính toán sẽ tìm ra một neural chiến thắng lớn nhất tại F2a tương ứng với đầu vào A và xem là neural kích hoạt J. Trong khi đó F2b sẽ kích hoạt một neural K từ mẫu vào B. Quá trình so sánh sẽ xảy ra tại MAP Field.Nếu neural kích hoạt K của B trùng với neural kích hoạt J của A thì việc dự đoán của ARTa là chính xác, đồng thời cập nhật Wjab. Ngược lại, một tín hiệu kích hoạt từ Map Field sẽ tác động lên ρa một lượng làm thay đổi ρa, sau đó quá trình sẽ lặp lại đối với ARTa để tìm neural kích hoạt khớp với ARTb.

## Giải thuật Fuzzy Artmap Neural Network:

Giải thuật huấn luyện mạng ARTMAP có thể được trình bày như sau:

Với mỗi dữ liệu (a, b) (a là vector của tập dữ liệu đầu vào, và b là đầu ra mong muốn tương ứng) trong tập dữ liệu đầu vào ban đầu thực hiện các bước sau đây:

**Bước 1**: Biểu diễn dữ liệu đầu vào

ρa  = ρ­¬a, xa = A, xb = B

**Bước 2**: Quá trình kích hoạt của F2

**Bước 2.1**: Tìm kiếm neural phù hợp

Với mỗi module ART, chúng ta tính giá trị kích hoạt Tj với mỗi neural j của F2.Kế tiếp chúng ta chọn neural có giá trị cao nhất, chính là neural chiến thắng. Tj được tính theo công thức:



**Bước 2.2**: Với mỗi neural được chọn trong bước 2.1 (J trong F2a và K trong F2b), chúng ta tính hàm mj:



Nếu giá trị của hàm này của neural J và neural K lớn hơn hoặc bằng vigilance ρ (ρa hoặc ρb), điều này được cho là có sự cộng hưởng và neural tương ứng ở lớp F2 được kích hoạt: ARTa yJ = 1 và yj = 0 cho tất cả j ≠ J (tương tự cho ARTb). Ngược lại thì quay lại bước 2.1 để chọn neural mới trong module tương ứng.

Nếu không có neural nào được chọn, một (hoặc vài) nút mới được tạo một cách tự động, lưu ý rằng J và K là những nút sau đây:

WJa = 1, WJab = 1, yJa = 1 và yja = 0 với j ≠ J

WKb = 1, WKab = 1, yKb = 1 và ykb = 0 với k ≠ K

**Bước 3:** Kiểm tra tham số vigilance trong MAP Field

Trong lớp này, chúng ta tính:



Nếu |xab| / |yb| ≥ ρab, thực hiện bước 4, ngược lại thực hiện bước 5.

**Bước 4**: Huấn luyện và cập nhật các trọng số

, ,  được cập nhật theo công thức:







Lưu ý ở chế độ huấn luyện nhanh giá trị của β được gán bằng 1 (β = 1) trong mỗi lớp.

**Bước 5**: Thay đổi giá trị tham số vigilance của ARTa.



gán TJ = 0, và quay trở lại bước 2.

Giải thuật fuzzy ARTMAP neural network đã trình bày ở trên sẽ được áp dụng vào tập dữ liệu truy cập của người dùng, kết quả đầu ra sẽ là tập các trang web đã được phân lớp, đánh giá và xếp hạng và được sử dụng như một phần của việc tối ưu công cụ tìm kiếm.

# WEBLOG VÀ GIẢI THUẬT LPAGERANK

## Xử lý Weblog:

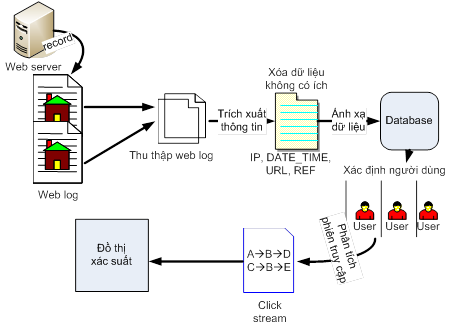
### Thông tin trên weblog:

Web server logs cung cấp thông tin về cách người dùng truy cập trang web. Nếu người dùng sở hữu trang web của riêng họ, họ có thể dễ dàng truy cập chúng. Nếu người dùng chạy trang web của họ trên các máy chủ bên ngoài có thể cần phải thương lượng với administrator để có thể truy cập được log files. Tất cả các máy chủ web đều ghi lại thông tin chi tiếp cho từng trang được truy cập. Yếu tố chính xác những gì các bản ghi máy chủ khác nhau tùy theo loại phần mềm được sử dụng và các tùy chọn thiết lập bởi quản trị viên của trang web.

Hầu hết các máy chủ có một tùy chọn để ghi lại địa chỉ IP trong log file hoặc để thực hiện một yêu cầu cho DNS (domain naming system) để có được nhiều thông tin hơn. Phải mất nhiều sự tính toán mới có thể thực hiện reverse-DSN look up, nhưng hầu hết các máy chủ web có thể xử lý một cách dễ dàng. Các URL tiếp theo mô tả các trang cuối cùng mà người dùng truy cập trước khi yêu cầu trang hiện tại. Yếu tố dữ liệu này giúp chúng ta biết cách mà người dùng truy cập trên các trang web.

Tập tin log là dạng nhật ký của máy chủ lưu trữ các mẫu truy cập của người dùng thành từng dòng (record), mỗi dòng có nhiều hay ít tham số phụ thuộc vào máy chủ ghi log cấu hình.

Định dạng của một dòng trong tập tin nhật ký có nhiều hay ít tham số tùy thuộc vào hệ điều hành và cấu hình ghi log của máy chủ nhưng thông thường bao gồm các thông tin sau: hostname hay IP, RFC931 (thường ít được sử dụng), author user, ngày và giờ truy cập (date), URL truy cập, phương thức truy cập (method), REF, protocol, port, user agent, trạng thái (status).



Hình ‑ Quá trình xử lý thông tin web log

Việc khai thác nhật ký sử dụng web thường trải qua ba giai đoạn chính: tiền xử lý - rút trích thông tin có ích từ tập tin log, xử lý mẫu để lấy thông tin phiên truy cập của người dùng, và cuối cùng là phân tích mẫu truy cập để thu được thông tin có ích cho mục đích nào đó.Hình 3-1 minh họa cho quá trình xử lý thông tin web log. Các mục tiếp theo sẽ trình bày quá trình xử lý chính để khai thác nhật ký sử dụng web.

### Lọc Weblog:

Tập tin log là dạng nhật ký của máy chủ lưu trữ các mẫu truy cập của người dùng thành từng dòng (record), mỗi dòng có nhiều hay ít tham số phụ thuộc vào máy chủ ghi log cấu hình.

Tập tin web log ghi lại tất cả những yêu cầu từ người dùng web. Mỗi dòng trong tập tin web log ghi lại những thông tin sau: *client IP, access time, HTTP request medthod, Referer, URL, protocol, status code, number of bytes transmitted*. Ví dụ về cách định dạng tập tin web log trong file httpd.conf của Apache HTTP Server như sau:

*LogFormat "%h %t \"%r\" \"%{Referer}i\" %>s %b" lpagerank*

với:

*%h*: địa chỉ IP của người dùng truy cập yêu cầu đến server

*%t*: thời gian lúc nhận được yêu cầu của người dùng, định dạng là:

[day/month/year:hour:minute:second zone]

*\"%r\"*: phương thức truy cập, tài nguyên truy cập, giao thức truy cập của người dùng.

*\"%{Referer}i\"*: tài nguyên truy cập trong *\"%r\"* do trang *\"%{Referer}i\"* chuyển hướng đến.

*%>s*: mã trạng thái của server gửi cho client, hiển thị kết quả yêu cầu của client, nếu thành công (mã bắt đầu bằng 2), redirection (mã bắt đầu bằng 3), lỗi nguyên nhân bởi client (mã bắt đầu bằng 4), hoặc lỗi nguyên nhân bởi server (mã bắt đầu bằng 5). Danh sách đầy đủ về các mã trạng thái có thể được tìm thấy trong [6].

*%b*: kích thước của tài nguyên mà server chuyển đến cho client.

Ví dụ về một tập tin web log được ghi lại bằng cách định dạng trên của Apache HTTP Server như sau:

*127.0.0.1 [13/Jan/2010:01:38:10 +0700] "GET /uit.edu.vn/daotao/cncq.html HTTP/1.1" "http://localhost/uit.edu.vn/daotao/index.html" 200 818*

Trong ví dụ trên, người dùng truy cập với IP *127.0.0.1* đã yêu cầu trang */uit.edu.vn/daotao/cncq.html* qua giao thức truy cập *HTTP/1.1*, phương thức truy cập *GET* vào *13/Jan/2010:01:38:10 +0700*. Kích cỡ của trang được gửi lại cho người dùng là *818* bytes.*200* là mã trạng thái của server gửi lại cho client, được ghi nhận là gửi thành công.

Không phải dòng nào trong web log cũng thực sự hữu ích, web logs phải được lọc ra các yêu cầu image, video, audio và các trang yêu cầu thất bại.

## Giải thuật PageRank:

PageRank được phát triển tại trường Đại học Stanford bởi Larry Pagevà sau đó là Sergey Brin như là một phần cùa một dự án nghiên cứu về một loại công cụ tìm kiếm mới. Dự án được bắt đầu từ năm 1995 và đã dẫn đến một nguyên mẫu đầu tiên, được đặt tên là Google, ra đời vào năm 1998. Một thời gian ngắn sau đó, Page và Brin đã thành lập Google Inc..Trong khi chỉ một trong số nhiều yếu tố xác định thứ hạng trên kết quả tìm kiếm của Google thì PageRank tiếp tục cung cấp những cơ sở khác cho tất cả các công cụ tìm kiếm trên web của Google.

Độ “quan trọng” của trang i (PageRank của trang i) sẽ được tính bằng tổng các giá trị PageRank của tất cả các trang trỏ đến trang i. Do một trang có thể trỏ đến nhiều trang khác, nên độ uy tín của nó cũng phải được chia sẻ với các trang đó.

Xem web như là một đồ thị có hướng G = (V, E), gọi n là tổng số các trang thu thập được. Điểm PageRank của một trang i, kí hiệu là P(i) sẽ được định nghĩa bởi công thức sau:

(2.9)

với Oj là số lượng out-link của trang j.

Với một tập gồm n trang web, ta sẽ có một hệ thống gồm n phương trình tuyến tính với n là chưa biết. Ta có thể sử dụng ma trận kề để diễn đạt chúng. Như vậy mỗi cột trong ma trận sẽ biểu diễn cho một trang (tương ứng một đỉnh trong đồ thị) và cho ta biết các trang trỏ đến nó. Hay nói cách khác ta sẽ có P là một vector cột để biểu diễn cho các giá trị PageRank của từng trang:

(2.10)

Với A là ma trận kề biểu diễn cho đồ thị, sẽ có các giá trị tại mỗi ô sau:

(2.11)

Phương trình tính PageRank lúc này sẽ có dạng:

(2.12)

Đây là một phương trình đặc biệt thuộc một hệ thống đặc trưng (eigensystem), tại đó P là một vector đặc trưng (eigenvector) tương ứng với giá trị đặc trưng (eigenvalue) là 1. Điều đó chứng tỏ rằng khi một số điều kiện được thỏa mãn, 1 là giá trị đặc trưng lớn nhất và PageRank vector P là một vector đặc trưng chính (principal eigenvector).

Để tìm P, ta có thể sử dụng một phương pháp toán học được gọi là Power Iteration (còn gọi là eigenvalue algorithm).

Power Iteration được định nghĩa đơn giản như sau: cho trước một ma trận A, thuật toán sẽ tạo ra một giá trị đặc trưng và một vector đặc trưng v khác 0, dạng ; có thể được sử dụng khi A là một ma trận lớn tuy nhiên nó chỉ tìm một giá trị đặc trưng duy nhất (có trị tuyệt đối lớn nhất) và có thể chậm trong quá trình thực hiện.

## Giải thuật LpageRank:

Giải thuật LPageRank được tác giả Qing Cui và Alex Dekhtyar giới thiệu vào năm 2005 với hướng nghiên cứu sử dụng weblog để giảm bớt chi phí thu thập liên kết và cải tiến việc đánh giá mức độ truy cập lên trang trong việc tính toán trọng số của trang web. Nhìn một cách tổng quan, thì LPageRank là PageRank được tính toán dựa trên đồ thị xác suất của trang web được xây dựng từ các mẫu truy cập trong nhật ký sử dụng web của người dùng.

Tập tin log ghi lại sự truy cập của người dùng trên một web site. Do đó, nếu như ta lấy thông tin toàn bộ tập tin weblog được ghi nhận một cách đầy đủ và giả sử rằng tất cả các liên kết (link) trên trang web đều được truy cập ta có thể mô hình hóa lại được chuỗi truy cập của người dùng và từ đó hình thành nên liên kết cấu trúc của toàn bộ trang web. Không chỉ có thế, tập tin weblog còn cung cấp thêm thông tin về việc người dùng quan tâm như thế nào đến một liên kết.Đó là một trong những thông tin rất quan trọng cho việc đánh giá khả năng truy cập trang web.

Đồ thị xác suất là đồ thị được xây dựng từ tập tất cả các liên kết cấu trúc của tập tất cả các trang web trong web site.Mỗi liên kết từ một trang này đến một trang khác tượng trưng cho một cung trong đồ thị.Tần số liên kết giữa các trang với nhau tương ứng với tần số truy cập giữa chúng của người dùng và cũng được xem là trọng số của cung trong đồ thị. Do đó, đồ thị thể hiện khả năng truy cập từ một trang đến các trang khác thông qua trọng số của cung kết nối. Đồ thị xác suất được xây dựng dựa trên kết quả phân tích thông tin weblog.

## Giải thuật xây dựng đồ thị xác suất:

Mỗi mẫu truy cập của weblog mô tả một cung liên kết từ trang một trang này đến một trang khác trong web site. Xét một phiên truy cập của người dùng, ta sẽ có được một chuỗi truy cập từ một trang lần lượt đến một hay nhiều trang khác. Ta sẽ tiến hành mô hình hóa toàn bộ lịch sử truy cập người dùng thành đồ thị xác suất như sau. Với mỗi cung của đồ thị (B,A) tương ứng với một mẫu truy cập từ B liên kết đến A trong một phiên truy cập (session), ta tăng trọng số của cung (B,A) lên 1 nếu trong một phiên truy cập có liên kết từ B đến A. Sau khi duyệt tất cả các session ta sẽ có một đồ thị có trọng số thể hiện mối liên kết giữa các liên kết (link). Sau cùng, ta tiến hành chuẩn hóa trọng số các cung trong đồ thị với bằng cách chia trọng số của cung cho tổng trọng số các cung ra cho mỗi nút trong đồ thị. Trong quá trình này, ta xây dựng một ma trận vuông A có kích thước mxm với m là số URL trong toàn bộ các session thu thập được. Mỗi giá trị trong ma trận vuông A[i,j] ứng với cung (i,j), giá trị của nó được tăng lên sau mỗi liên kết từ i đến j được duyệt qua. Cuối cùng, ta chuẩn hóa ma trận A để thu được ma trận đồ thị xác suất.

🟑**Giải thuật xây dựng đồ thị xác suất:**

*Cho tập trang web N{A1,A2,..,Am} được biểu diễn với ma trận vuông A mxm*

*Tập phiên truy cập S*

***B1****: duyệt tất cả các mẫu truy cập trong S*

*Gán A[i,j]=A[i,j]+1 nếu mẫu chứa liên kết từ trang Ai đến Aj.*

***B2:*** *duyệt tất cả các trang trong tập N*

*Gán A[i,j]=A[i,j]/N(A) với N(A) là số cung ra đi từ Ai đến trang khác.*

## Giải thuật LpageRank được cải tiến từ giải thuật PageRank với Weblog:

Giả sử G=(W,E,P) là đồ thị xác suất cho tập các trang web của site là W. Sau đó tính LPageRank(LPR) của trang web như sau:



Lưu ý rằng LPR(A)=PR(A) đối với đồ thị G nếu P(B,A)= đối với tất cả các cung.

🟑**Giải thuật LpageRank*.***

*B1: Duyệt đồ thị xác suất ta lấy giá trị xác suất cung P(B,A).*

*B2:Khởi tạo giá trị PageRank cho N trang web PR(A1)=1/N, PR(A2)=1/N,…PR(AN)=1/N*

*B3: Tính giá trịPageRank của tất cả các trang bằng công thức sai:*



*B4: Tính sai*

*số của tất cả các trang =LPR(A)i –LPR(A)i-1 (i: là lần lặp thứ i).*

*Nếu của tất cả các trang đều nhỏ hơn ngưỡng thì dừng.*

*Nếu tồn tại >ngưỡng quay lại bước 3.*

Với cách cải tiến này, giá trị LPageRank cho mỗi trang sẽ được tính toán phụ thuộc vào việc người dùng truy cập lên trang web đó nhiều hay ít. Nói cách khác, giá trị này phụ thuộc vào mức độ quan tâm của người dùng đối với trang web. Giải thuật LPageRank đã tối ưu hơn cách thực hiện máy móc của PageRank khi không quan tâm đến hành vi sử dụng web của người dùng của giải thuật PageRank. Đây là sự khác nhau cơ bản mà LPageRank đã sử dụng để tối ưu cho công cụ tìm kiếm.

## Ứng dụng giải thuật TF- IDF kết hợp cùng với giải thuật LpageRank trong công cụ tìm kiếm:

### Sử dụng giải thuật TF – IDF cho việc đánh trọng số câu truy vấn:

Định nghĩa trọng số TF – IDF:

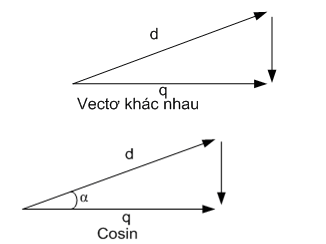
Trọng số TF-IDF (Term Frequency-Inverse Document Frequency) là trọng số thường được sử dụng trong tìm kiếm thông tin và khai thác dữ liệu văn bản. Trọng số này là một độ đo tĩnh được sử dụng để đánh giá mức độ quan trọng của một từ (token) đối với một tài liệu và một từ trong tài liệu với tập các tài liệu. Độ quan trọng này được tăng lên tương ứng với số lần xuất hiện của từ trong tài liệu.TF(Term Frequency) trong một tài liệu là tần số xuất hiện của một từ khóa (term) đã cho trong một tài liệu. Và IDF(Inverse Document Frequency) là độ đo nghịch đảo của một từ khóa trong tất cả các tài liệu có trong kho tìm kiếm có chứa từ khóa.

Trong mô hình tìm kiếm này, một tài liệu là một vectơ thực d trong Rk (trọng số thực), di được xác định dựa trên một hàm tính toán TF-IDF. Cũng tương tự như một tài liệu, ta xem một câu truy vấn cũng là một vectơ thực q trong Rk trong đó qi là trọng số của ti trong q. Hàm tính độ liên quan là f(d,q)= sim(d,q) trong đó sim(d,q) là mức độ giống nhau giữa d và q.

Từ đó, định nghĩa giống nhau giữa một vectơ tài liệu và một vectơ truy vấn như sau:



Phương thức này sẽ gán một lượng đáng kể cho các từ khóa xuất hiện trong tài liệu nhưng không xuất hiện trong câu truy vấn.Vectơ truy vấn thường thưa thớt hơn nhiều so với vectơ tài liệu, vì vậy một phương thức tốt hơn nên loại bỏ hiệu ứng của các từ khóa không xuất hiện trong câu truy vấn.



Hình ‑ Biếu diễn độ tương tự của vector tài liệu d và vector truy vấn q

Phương thức đo sự giống nhau cosin dựa trên quan sát ở trên, là phương thức thông dụng để đo sự giống nhau giữa một vectơ tài liệu và một vectơ truy vấn



Chú ý là, nếu góc giữa hai vectơ nhỏ thì cosin gần tới 1, là giá trị lớn nhất của sự giống nhau.Nếu hai vectơ gần như vuông góc thì cosin gần tới 0 nghĩa là sự giống nhau nhỏ nhất.

TF-IDF là phương pháp thông dụng để “cân nặng” (đánh giá) các từ khóa (term) trong một tài liệu. Ý tưởng cơ bản của phương pháp này là xem xét tính phổ biến của một từ khóa trong một tài liệu khi so sánh với tính phổ biến của thuật ngữ đó trong các tài liệu khác.

### Giải thuật tính TF – IDF cho một tài liệu:

Định nghĩa chính qui của TF-IDF được định nghĩa như sau: Gọi n(d,ti) là tần số xuất hiện của ti trong d và N= là tổng số thuật ngữ trong d. Di chỉ số tài liệu chứa ti và D là tổng số tài liệu có trong tập hợp. Tần số thuật ngữ (term frequency) TF (d, ti) là tần số xuất hiện của ti trong d.

Có vài cách tính tần số thuật ngữ: hai cách thông dụng nhất là chia số lần xuất hiện của thuật ngữ trong tài liệu cho hoặc là tổng số thuật ngữ có trong tài liệu hoặc là số lần xuất hiện của thuật ngữ xuất hiện nhiều nhất trong tài liệu:

TF(d, ti­) = n(d, ti)/N

Hoặc

TF(d, ti­) = n(d, ti)/max(n(d, ti))

Trong bất kì trường hợp nào thuật ngữ xuất hiện nhiều hơn sẽ có điểm TF cao hơn (cao nhất là 1) và thuật ngữ ít xuất hiện sẽ có điểm TF gần bằng 0.

Ngược lại, IDF(ti­) (Inverse Document Frequency) là tần số nghịch đảo của ti­ trong tất cảc các tài liệu có trong tập hợp và số tài liệu trong tập hợp có chứa ti.

IDF(ti) = log(D/Di)

Thuật ngữ thường xuyên xuất hiện trong tài liệu như là “the” vì vậy sẽ có IDF gần bằng 0 và thuật ngữ hiếm gặp sẽ có IDF gần bằng 1. Trọng số TF-IDF được tính bằng cách nhân tổ hợp điểm TF và IDF

TF-IDF(d, ti) =TF(d,ti) x IDF(ti)

Qua công thức trên, ta có thể thấy rằng TF-IDF sẽ cho điểm một thuật ngữ cao hơn nếu nó xuất hiện thường xuyên trong một tài liệu và không xuất hiện thường xuyên trong các tài liệu khác.

🟑**Giải thuật TF-IDF.**

*Cho tập từ khóa t{t0,…tn} trong tập tài liệu D{D0,…DN}.*

*Tính TF-IDF(Di,ti).*

*B1: Duyệt từng tài liệu Dj trong tập tài liệu D.*

*B2: Duyệt từng từ khóa ti trong tập từ khóa t trong từng tài liệu Dj.*

*Nếu ti xuất hiện*

*count\_ti =count\_ti + 1.*

*B3: Tính TF(Dj,ti)=count\_ti/N //N tổng số từ trong tài liệu Dj.*

*Hoặc: TF(Dj,ti)=count\_ti/Max\_count\_term //Max\_count\_term số lần xuất //hiện của từ khóa xuất hiện nhiều nhất trong Dj*

*B4: Duyệt hết một tài liệu Dj, nếu count\_ti>0 thì count\_di=count\_di+1.*

*B5: Duyệt từng từ khóa ti.*

*Tính IDF(ti)=log(N/count\_di) //số tài liệu trong D chia cho tổng số tài liệu chứa ti.*

*B6: Duyệt từng tài liệu liệu Dj trong tập tài liệu D.*

*Duyệt từng từ khóa ti trong tập từ khóa t trong từng tài liệu Dj.*

*Tính TF-IDF(Di,ti)= TF(Dj,ti)\*IDF(ti).*

Mô hình không gian vectơ, thường xuyên sử dụng TF-IDF để đánh giá các thuật ngữ và hàm cosin là hàm đo mức độ giống nhau, thể hiện là một phương thức tính độ liên quan giữa một tài liệu và một câu truy vấn tin cậy hơn. Song, phương pháp này có một nhược điểm là đánh giá các thuật ngữ độc lập với nhau, mà trong thực tế các thuật ngữ có liên quan với nhau và hiểu được điều này có thể dẫn đến việc tính độ liên quan tốt hơn.

### Ứng dụng giải thuật TF- IDF trong công cụ tìm kiếm:

Độ đo TF-IDF là một trong những độ đo được sử dụng khá nhiều trong các công cụ tìm kiếm thông tin. Độ đo TF-IDF được ứng dụng cho việc xử lý truy vấn của người dùng trong tìm kiếm cục bộ trên web site.

Quá trình xử lý câu truy vấn và tìm kiếm trên trang web không giống với những tài liệu thông thường vì trang web có một số đặc thù riêng biệt khác với các dạng tài liệu phẳng khác. Do đó, khi tiến hành thực nghiệm cần tiến hành một số bước tiền xử lý trước khi áp dụng giải thuật TF-IDF.

Công cụ tìm kiếm thông tin cục bộ trên web site được ứng dụng giải thuật TF-IDF cho xử lý trực tuyến câu truy vấn của người dùng và tài liệu tìm kiếm.Sau khi xử lý, giải thuật LPageRank sẽ được áp dụng để tối ưu trọng số của tài liệu.

Ngoài ra, trong mô hình này, cơ chế gom nhóm phiên truy cập của người dùng được sử dụng để gom các kết quả tìm kiếm vào nhóm các mẫu truy cập nhiều nhất và được người dùng quan tâm nhất nhằm hướng kết quả tìm kiếm gần với mục tiêu sử dụng của người dùng.

# SOCIAL ANNOTATION VÀ CÁC GIẢI THUẬT LIÊN QUAN

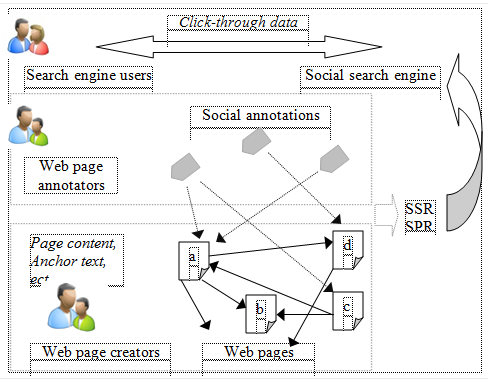
## **Ứng dụng chú thích cộng đồng (Social Annotation) cải thiện việc tìm kiếm thông tin trên mạng:**

Chú thích cộng đồng (social annatations) có nhiều tiềm năng để cải thiện việc tìm kiếm tài nguyên (Marlow et al., 2006). Tuy nhiên, các nghiên cứu đã công bố về việc sử dụng chú thích xã hội để cải thiện web search còn rất ít. Các chú thích đã được sử dụng trong nhiều năm để phân tích và mô tả tài liệu. Chúng cũng đã được sử dụng để hỗ trợ hợp tác và giao tiếp trong công việc nhóm. Những nghiên cứu gần đây đã khám phá cách sử dụng các chú thích để phân loại và truy tìm.

Các chú thích đã được sử dụng để cải thiện việc truy tìm hình ảnh và video. Nhiều hệ thống chú thích ảnh và video đã được phát triển cho các mục đích truy tìm (Bargeron et al, 1999, Sheiderman và Kang, 2000, Gertz et al, 2002., Costa et al.., Năm 2002, Bottoni et al., 2004). Các hệ thống thương mại như Google Video, YouTube, Yahoo Video, Yahoo Photo, Flickr và Delicious, cung cấp cho người dùng của họ một khả năng thêm chú thích, mà họ gọi là "thẻ", trên các ảnh, đa phương tiện và URL. Họ sử dụng các thẻ (tags) này như các chỉ mục. Quá trình này được gọi là gắn thẻ xã hội (social tagging), gắn thẻ cộng tác (collaborative tagging), đánh dấu xã hội (social bookmarking) hoặc lập chỉ mục quần chúng (mob indexing). Thông tin được sử dụng để truy tìm hình ảnh và nội dung đa phương tiện.

Trong hệ thống social search, có ba loại người dùng liên quan đó là: người tạo trang web (web page creators), người chú thích trang web (web page annotators), người sử dụng công cụ tìm kiếm (search engine users). Rõ ràng là giữa ba tập người dùng này có thể giao nhau.

* Web page creators: Tạo các trang và liên kết các trang với nhau để người dùng dễ dàng duyệt web. Họ cung cấp cơ sở cho việc tìm kiếm web.
* Web page annotators: Là những người dùng sử dụng các chú thích để tổ chức, nhớ và chia sẻ sở thích online.
* Search engine users: Sử dụng công cụ tìm kiếm để có được thông tin từ web. Họ cũng có thể trở thành người tạo chú thích nếu họ lưu và chú thích sở thích của mình từ kết quả tìm kiếm.



Hình ‑ Minh họa ba thành phần trong Social Search

Các công trình nghiên cứu trước đã cho thấy rằng cả web page creators và search engine users đóng góp rất nhiều cho web search. Những người tạo trang web không chỉ cung cấp các trang web cùng với anchor texts cho xếp hạng tương đồng mà còn cung cấp cấu trúc liên kết cho xếp hạng tĩnh từ quan điểm của những người tạo trang web (chẳng hạn như PageRank). Trong khi đó, người sử dụng công cụ tìm kiếm cũng mang lại lợi ích cho web search bằng cách cung cấp dữ liệu thông qua click, được dùng cả trong xếp hạng tương đồng (như IA) và xếp hạng tĩnh (như fRank). Ở đây, chúng ta sẽ tập trung tìm hiểu người chú thích trang web đóng góp cho web search như thế nào.

🟑**Xếp hạng tương đồng giữa truy vấn và chú thích xã hội.**

Các chú thích xã hội thường hiệu quả, tóm tắt nhiều mặt của một trang web và cung cấp một siêu dữ liệu mới cho xếp hạng tương đồng. Một cách sử dụng trực tiếp và đơn giản của các chú thích là tính toán động tương đồng dựa trên số thuật ngữ được chia sẻ giữa truy vấn và các chú thích. Đặt q = {q1,q2,...,qn} là một truy vấn bao gồm n thuật ngữ truy vấn và p A(p) = (a1,a2,...,am) là tập chú thích của trang web p, phương trình (1) chỉ ra phương pháp tính toán độ tương đồng dựa trên số thuật ngữ được chia sẻ. Lưu ý rằng simTM(q, p) được định nghĩa là 0 khi A(p) rỗng.



Tuy nhiên, chú thích của một số trang web khá thưa thớt và phương pháp tiếp cận dựa trên term-matching bị ảnh hưởng ít nhiều về vấn đề đồng nghĩa. Chẳng hạn, truy vấn và chú thích có thể có những term có ngữ nghĩa giống nhau nhưng hình thức khác nhau. Như vậy, chúng ta phải giải quyết vấn đề đồng nghĩa bằng cách khai thác cấu trúc tính đồng nghĩa của chú thích cộng đồng.

# Xây dỰng CÔNG CỤ TÌM KIẾM Ứng dỤng các thuẬt GiẢi Social Annotation, Fuzzy ArTmap, NEURAL netWork

## Giới thiệu công cụ tìm kiếm:

### Tên công cụ:

SearchEngine

### Môi trường phát triển:

* Window.
* Công cụ: Visual studio 2010, Mircosoft SQL server 2008.

## Phân tích:

### Khảo sát hiện trạng:

#### Phát biểu bài toán:

Hiện tại số lượng người sử dụng internet tăng theo từng ngày dẫn đến lượng thông tin lưu trữ cũng tăng theo, kèm sự khó khăn trong việc tìm kiếm dữ liệu đặc biệt hiện nay khi người dùng sử dụng các chức năng tìm kiếm trên một trang web hay diễn đàn lớn thường không ra kết quả cần tìm hoặc không có kết quả trong khi những dữ liệu đó vẫn tồn tại. Khi đó để thuận tiện cho người dùng các diễn đàn phải tốn chi phí mua công cụ search engine từ các công ty lớn như Google hay Yahoo…. Để không phải tốn các khoản chi phí này ta có thể xây dựng một công cụ tìm kiếm cục bộ sử dụng các thuật toán social annotation, fuzzy artmap và neural network để cho ra các kết quả có độ chính xác cao trong thời gian ngắn nhất.

#### Giải pháp xây dựng:

### Yêu cầu phi chức năng:

|  |  |
| --- | --- |
| * Lọc dữ liệu | Thiết kế hệ thống lọc dữ liệu theo mô hình Fuzzy Art Map, Neural network |
| * Đánh giá xếp hạng trang web | Xây dựng các thuật toán đánh giá xếp hạng. |

### Yêu cầu chức năng

|  |  |
| --- | --- |
| * Phân tích | Phân tích trang web, lấy dữ liệu weblog bằng cách thao tác lên từng link của trang web. |
| * Đánh giá | Đánh giá tổng quan về trang web đã được phân tích về tổng số lượng link, số lượng link lỗi… |
| * Tìm kiếm trang web | Tìm kiếm trang web theo đúng từ khóa nhập vào, vào được xếp hạng từ cao tới thấp theo các thuật toán xếp hạng có trong chương trình. |
| * Chọn các thuật toán xếp hạng | Lựa chọn thuật toán mún tối ưu khi tìm kiếm. |
| * Chọn tìm kiếm theo ngữ nghĩa | Tìm kiếm với từ khóa theo ngữ nghĩa với các kết quả gần đúng nghĩa. |
| * Phân tích nhiều trang web | Có thể phân tích nhiều trang web, theo thứ tự nhập vào. |
| * Đánh giá nhiều trang web | Có thể đánh giá nhiều trang web đã được phân tích. |

### Mô hình usecase:



## Thiết kế:

### Mô hình chương trình:

Trong mục này khóa luận trình bay một mô hình công cụ tìm kiếm mà khóa luận đã xây dựng dựa vào giải thuật LPageRank và mạng neural fuzzy ARTMAP bao gồm 4 thành phần chính:

***Web Spider (Web Crawler)*** *-* thu thập các web site, xây dựng tập từ khóa và đồ thị liên kết cho website. Công việc bao gồm rút trích mỗi trang web, trích xuất những từ khóa và xây dựng vector trọng số cho những từ khóa đó bằng phương pháp TF-IDF.

Thành phần *Web Spider* trong hệ thống thực hiện hai chức năng: (a) khởi tạo đồ thị liên kết cho website và (b) thu thập các trang HTML cho mỗi trang và đánh chỉ mục cho các trang HTML đó.

Spider cần đầu vào là một trang web bắt đầu cho việc thu thập thông tin. Giả sử trang web bắt đầu là trang chủ của một website (ví dụ như www.vnuhcm.edu.vn hoặc www.uit.edu.vn). Spider sẽ tiến hành thu thập các trang web, bằng cách trích xuất và đánh chỉ mục cho tất cả các trang HTML mà địa chỉ URL được lấy từ cùng một trang web như URL bắt đầu. Vậy là Spider bỏ qua tất cả các địa chỉ bên ngoài trong quá trình duyệt các trang web và xây dựng thành đồ thị cho các trang web.

Mỗi trang HTML thu được trong quá trình thu thập sau đó được đưa vào quy trình xây dựng vector trọng số cho từ khóa. Trong quá trình này sẽ xây dựng bộ từ vựng V = {k1, …, k­­­­M} của những từ khóa được tìm thấy trên website và sử dụng phương pháp đánh trọng số từ TF-IDF.

***Log Miner***- thu thập thông tin từ weblog và sử dụng thông tin đó để chuyển sự kết nối của website vào trong đồ thị xác suất bằng cách gán xác suất cho mỗi cạnh trong đồ thị. Từ thông tin này, LPageRank được tính cho tất cả các trang web dựa vào giải thuật LPageRank.

Thành phần *Log Miner* trong hệ thống cũng thực hiện nhiệm vụ: (a) Xóa tất cả những thông tin không thích hợp, (b) phân tích phiên truy cập, (c) hoàn thành xây dựng đồ thị xác suất cho website và (d) tính LPageRank. Đầu tiên, Log Miner lấy đầu vào là là file hoặc một tập những file log của website. File log bao gồm chuỗi những yêu cầu HTTP, file log sẽ được xóa những trang yêu cầu thất bại hoặc yêu cầu hình ảnh, bất kỳ yêu cầu nào liên quan đến robots.txt đều được thanh lọc. Dò tìm những phiên truy cập, những bước thực hiện như mục 2.2. Mỗi phiên truy cập được trích xuất từ log file sau đó được chuyển vào trong đồ thị. Mỗi cạnh (B, A) trong phiên truy cập (B, A là địa chỉ nằm trong chuỗi phiên truy cập, người dùng chuyển sang link A trong khi đang quan sát link B trong trình duyệt), chúng ta sẽ tăng trọng số của cạnh (B, A) trong đồ thị lên 1. Sau tất cả những phiên truy cập đã được phân tích, chúng ta chuẩn hóa trọng số của mỗi cạnh bằng tổng tất cả các trọng số của những cạnh trỏ ra ngoài của mỗi nút.Sau khi có được đồ thị xác suất từ bước trên, việc tính LPageRank sẽ được thực hiện. Chúng ta lấy α = 0.15 và khởi tạo giá trị LPageRank *LPRo= 1.* Chúng ta tính mỗi bước lặp như sau:



Tiến trình này dừng lại khi *LPRi*và *LPRi+1* khác nhau nhỏ hơn một số є. Vào thời điểm thực hiện, є được khởi tạo là 0.001

***Webpage Classifier***- phân lớp và đánh giá, phân lớp các trang web theo các tiêu chuẩn được đề xuất ở mục 4.1 bằng mạng neural fuzzy ARTMAP.

Thành phần *Webpage Classifier*thực hiện: (a) tính toán mỗi giá trị đánh giá tiêu chuẩn trang web và (b) những giá trị này sẽ làm đầu vào và được phân lớp bởi mạng fuzzy ARTMAP. Kết quả đầu ra của mạng sẽ cung cấp cho công cụ tìm kiếm một nhóm các trang mà trong hành vi sử dụng người dùng thường truy cập đến và được đánh giá là tốt. Chính kết quả này sẽ tăng thêm độ tin cậy của kết quả tìm kiếm trả về, hướng kết quả tìm kiếm về gần nhất với thông tin và sự mong muốn của người dùng. Kết quả của mạng fuzzy ARTMAP chỉ làm củng cố thêm trọng số xếp hạng cho trang, nó không có ý nghĩa quyết định trang đó có được tìm thấy hay không. Sau khi tiến hành phân lớp, kết quả các trang web tương tự với nhau sẽ thuộc chung một lớp pj{ak, ak+1,…} trong p lớp. Trọng số của mỗi trang trong lớp sẽ được tính bằng công thức:

với là tổng các giá trị tiêu chuẩn đánh giá trang web. là trọng số của lớp p với ai thuộc lớp p.

***Search Engin*** - nhận truy vấn từ người dùng và kết hợp bộ từ khóa cơ sở đã rút trích với giá trị LPageRank của các URL và điểm các trang web đã được đánh giá, phân lớp bằng mạng neural fuzzy ARTMAP để hiển thị theo hạng danh sách các URL khớp với truy vấn.

***Social Annotation*** – nhận truy vấn của người dùng thông qua Webpage Annotation, cùng với anchor text và cấu trúc liên kết cho xếp hạng tương đồng. Nhưng việc truy vấn và chú thích của trang web có thể có những cụm ngữ nghĩa giống nhau nhưng hình thức khác nhau.Vì thế nên sự củng cố của hai giải thuật SocialSimRank (về mặt ngữ nghĩa) và SocialPageRank (về mặt sắp xếp kết quả trả về) là cần thiết để phục vụ cho việc tối ưu tìm kiếm.

Quá trình hoạt động của công cụ tìm kiếm sẽ bao gồm 2 quá trình chính: phân tích thông tin và tìm kiếm thông tin. Quá trình phân tích với đầu vào là tập các trang web và thông tin weblog, các thành phần *Web Spider, Log Miner* và *Webpage Classifier*sẽ trích xuất các thông tin về từ khóa, trang web, các trọng số và lưu trữ lại trên đĩa cứng, sau đó khi người dùng nhập một câu truy vấn, *Search Engine, Social Annotation* sẽ thực hiện việc tìm kiếm và sắp xếp kết quả để trả về cho người dùng.

Thành phần *Web Spider*, *Log Miner*, *Webpage Classifier*của công cụ tìm kiếm làm việc off-line, chuẩn bị dữ liệu cho on-line sử dụng. *Search Engine* là thành phần on-line của công cụ tìm kiếm, có trách nhiệm nhận truy vấn từ người dùng, sử dụng công cụ xử lý ngôn ngữ tự nhiên phân tích ngữ pháp và chuyển sang biểu diễn như vector của những trang web, và tiến hành nhận những trang web có độ thích hợp cao so với truy vấn. *Social Annotation* là cũng là thành phần online của công cụ tìm kiếm, như *Search Engine* cũng nhận truy vấn từ người dùng, sử dụng công cụ xử lý về mặt ngữ nghĩa, phân tích trạng thái của trang Web và trả về những chú thích phù hợp với nhu cầu truy vấn của người dùng.

Ta có một page vector *d* và một query vector *q,* độ tương tự của chúng được tính bằng cosine của góc giữa chúng trong không gian M-chiều như sau:

*sim (d, q) = cos(d, q) =* .

Độ tương tự giữa *d* và *q* sau đó được nhân với LPageRank, trọng số phân lớp sử dụng mạng fuzzy ARTMAP của *d* để tính được điểm giống nhau cuối cùng:

*relevance (d, q) = LPR(d)ART(d)sim(d,q).*

Độ trung bình chính xác (MAP) trong Social Annotation: sử dụng chủ yếu MAP để đánh giá thao tác thực hiện tìm kiếm. Nó được đĩnh nghĩa là độ trung bình của các chính xác trên những câu truy vấn:

*Average\_precision =*

Trong đó, *p(j)* biểu thị sự chính xác trên kết quả *j*, và Δ*r(j)* là sự thay đổi quy hồi từ *j-1* đến *j.*

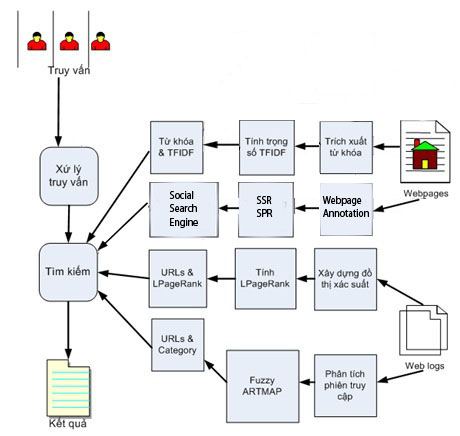
Bình thường hóa việc giảm tích lũy (NDCG): là một biện pháp thu hồi đặc biệt cho đánh giá trang web. Nó rất thích hợp đánh giá tìm kiếm trang web như tài liệu kết quả liên quan được xếp hạng nhiều hơn so với những trang được xếp hạng thấp hơn. Đối với một truy vấn q nhất định, kết quả xếp hạng được kiểm tra trong một chuyển hướng từ trên xuống, và NDCG được tính như sau:



Trong đó, *M­q*là tính toán đặc biệt thu được bậc hoàn hảo cho một NDCG với giá trị bằng 1, và mỗi *r(j)* là một số nguyên phù hợp (0 = “không liên quan” và 1 = “liên quan”) của kết quả trả về tại vị trí *j*.

Chi tiết về thiết kế của mỗi thành phần sẽ được trình bày trong mục kế tiếp của khóa luận.

### Sơ đồ luồng dữ liệu:



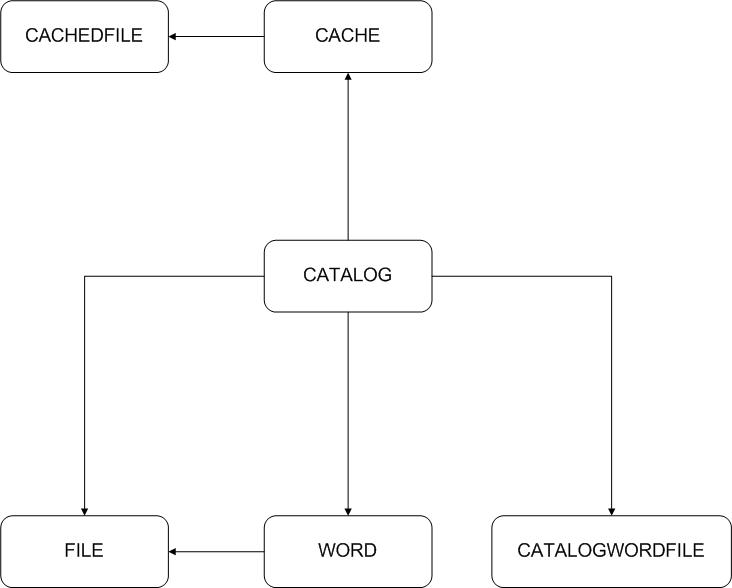
Hình ‑ Kiến trúc công cụ tìm kiếm

## Phát triển ứng dụng:

### Weblog Spider:

#### Cấu trúc dữ liệu:

Cấu trúc dữ liệu lưu giữ thông tin các trang và từ khóa tìm kiếm của chương trình được miêu tả bởi sơ đồ sau:

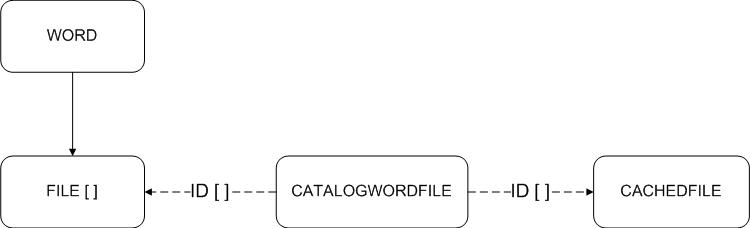


Hình ‑ Mô hình cấu trúc dữ liệu của WebSpider

Ý tưởng chính là xây dựng một tập các từ khóa, và mỗi từ khóa có chứa tham chiếu đến các trang/file mà nó xuất hiện.

Gồm 1 đối tượng Catalog chính, qua đó chứa các thông tin về từ khóa và các trang web. Đối tượng Word và CatalogWordFile dùng để lưu thông tin về các từ khóa, bao gồm cả tập các trang mà chúng xuất hiện và tập vị trí cho từng trang tương ứng, trong đó Word được dùng trong quá trình phân tích và tìm kiếm, và CatalogWordFile dùng chủ yếu trong quá trình save và load dữ liệu từ file (XML). Thông tin cơ bản của trang web sẽ được lưu trong lớp File, và toàn bộ các thông tin chi tiết của các trang sẽ được lưu trong đối tượng CachedFile với chi tiết về nội dung, các giá trị trọng số,...

Ta có sự phân loại lớp đối tượng Word và CatalogWordFile chủ yếu để phục vụ cho quá trình save và load dữ liệu XML, được xây dựng trên ý chỉ lưu ID của File chứa từ khóa và không thật sự chứa thông tin File trong từ khóa. Qua đó sẽ giảm bớt sự lập lại dữ liệu trong file XML và giảm kích thước đáng kể file dữ liệu lưu trên đĩa.



Hình ‑ Thông tin từ khóa và trang web

Chi tiết các lớp:

FILE : Thông tin về file/trang web

* ID trang web – Thông tin dùng để xác định trang web trong quá trình hoạt động, save, load dữ liệu.
* URL trang web
* Tiêu đề
* Định dạng
* Miêu tả
* Thời gian được phân tích
* Kích thước

WORD: Thông tin về từ khóa

* Nội dung từ
* Danh sách các trang có chứa từ và các vị trí cùa từ trong từng trang web tương ứng
* Tập trọng số TF-IDF đối với từng trang web

CATALOGWORDFILE: Thông tin về từ khóa, nội dung từ.

* Tập các chuỗi thông tin về các trang web có chứa từ và vị trí trong trang web. Có dạng : *xxx:aa,bb,cc,dd* với xxx là ID của trang web, aa, bb,cc là các vị trí của từ trong trang
* Tập các chuỗi thông tin về giá trị TF-IDF đối với từng trang web. Có dạng *xxx:aaa* với xxx là ID của trang web, aaa là giá trị TF-IDF của từ đối với trang web đó.

CACHE: lưu trữ tập các trang đã được phân tích với các thông tin chi tiết

* Tập các CachedFile đã được phân tích

CACHEDFILE: đối tượng lưu giữ các thông tin chi tiết về 1 trang đã được phân tích.

* ID trang web
* URL trang web
* Tập các từ khóa có trong trang
* Giá trị PageRank của trang
* Giá trị LpageRank của trang
* Các giá trị đánh giá (độ cập nhật, level, số lượng truy cập, thời gian truy cập, độ lỗi, tính phân loại)

CATALOG: lớp chính chứa thông tin về các trang web, từ khóa đã phân tích

* Tập từ khóa
* Tập các trang web
* Tập các trang cache

Các lớp lưu trữ thông tin về cấu trúc liên kết và đồ thị liên kết giữa các trang:

LINKSTRUCTURE: Lớp lưu lại cấu trúc liên kết liên kết của từng trang bao gồm tập các trang và tập các liên kết tương ứng của chúng

VERTEX : Đối tượng miêu tả đỉnh của đồ thị liên kết, chính là là 1 trang web.

EDGE : Đối tượng miêu tả cạnh của đồ thị liên kết, biểu diễn bằng 2 trang web mà có liên kết với nhau qua các đường dẫn URL.

GRAPH : Đồ thị liên kết của toàn webstie:

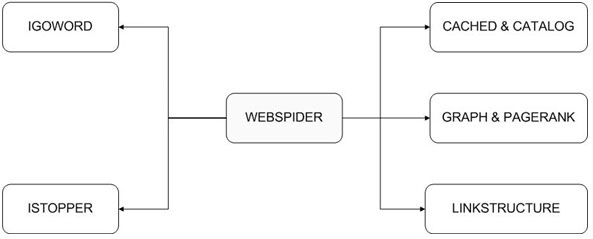
* Tập các đỉnh (trang web) có trong đồ thị.
* Tập các cạnh của đồ thị, là tập các liên kết giữa các trang web.
* Ma trận xác suất giữa các trang.

Tập các giá trị PageRank cho từng trang tương ứng

#### Web Spider:

Đối tượng WebSpider sẽ thực hiện chức năng chính là phân tích website để lấy về tập các trang và từ khóa, phục vụ cho quá trình tìm kiếm.Đây là lớp khởi tạo dữ liệu.

WebSpider sẽ bao gồm các đối tượng Cache, Catalog dùng để lưu lại tập trang web và từ khóa tìm kiếm và các giá trị trọng số của chúng, LinkStructure lưu lại cấu trúc liên kết cho từng trang và Graph lưu giữ đồ thị liên kết giữa các trang. Ngoài ra WebSpider còn sử dụng các đối tượng IGoWord, IStopper để thực hiện việc lọc các từ khóa tìm kiếm.

****

Hình ‑ Cấu trúc WebSpider

Quá trình phân tích 1 web site có thể gom thành 2 giai đoạn chính : Phân tích nội dung các trang web, tạo nên tập các trang web và từ khóa và giai đoạn tính các giá trị tìm kiếm cho từng trang (TF-IDF, PageRank, LpageRank,…).

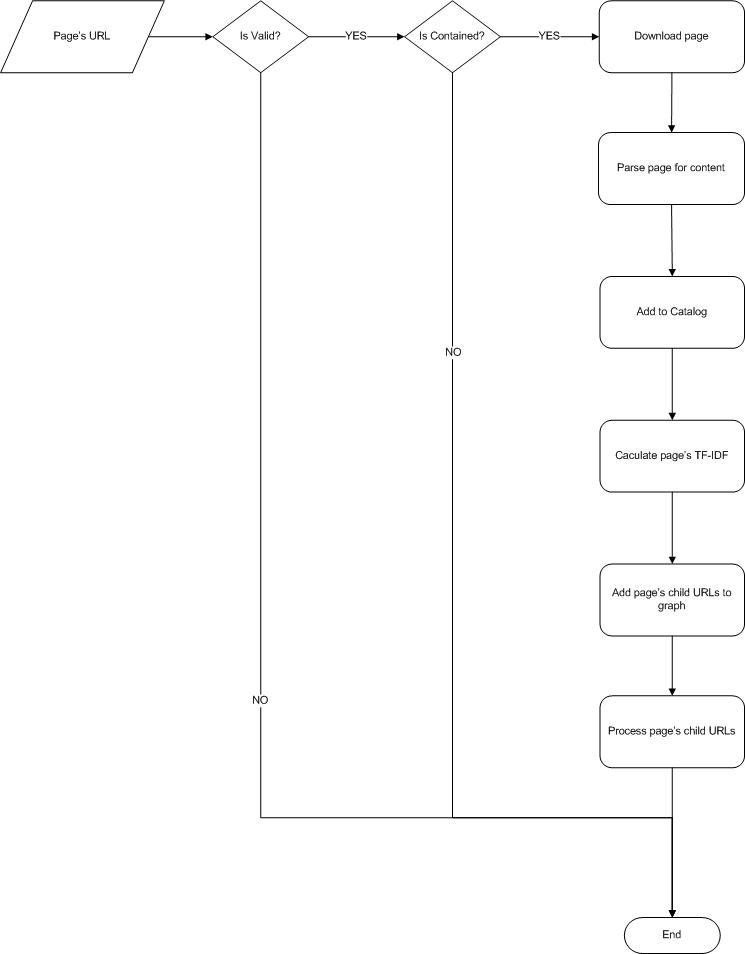
*Phân tích WebSite:*

Input của quá trình là URL ban đầu của trang web (trang chủ hoặc trang con) và output là tập các đối tượng file đã được phân tích và tập các từ khóa.

Với mỗi url đưa vào, ta thực hiện quá trình:

* Kiểm tra URL gồm kiểm tra hợp lệ bằng các bộ lọc như lọc định dạng file, tên file,… và kiểm tra URL đã được phân tích trước đó hay chưa.
* Thực hiện download file qua giao thức HTTP. Trong quá trình download, có thể kiểm tra lại server có thể dẫn đến 1 trang khác (redirect) hay không, và thực hiện lại quá trình kiểm tra ở bước 1 đối với trang được dẫn đến. Lưu lại URL của trang web để tránh lặp lại phân tích nhiều lần.
* Thực hiện phân tích nội dung file vừa download để lấy ra tập các từ khóa, URL, tiêu đề, miêu tả. Tiến hành phân tích nội dung HTML của mỗi trang web. Thực hiện loại bỏ các tag HTML, tìm và lấy về các giá trị Metadata, title nếu cần thiết, loại bỏ các giá trị rác bằng các bộ lọc StopWord, GoWord. và thực hiện tách chuỗi nội dung thành tập từ vựng cho mỗi trang. Thêm trang và tập từ khóa vào dữ liệu phân tích. Để phân tích chính xác, có thể phân loại riêng các từ khóa từ tiêu đề, miêu tả và nội dung thông thường.
* Tính toán ban đầu giá trị TF-IDF của các từ khóa trong trang.
* Lấy về cấu trúc liên kết giữa trang đang phân tích với các trang nó dẫn đến. Thêm các URL con trong trang web vào đồ thị liên kết, với đỉnh là các trang web, và cạnh là các liên kết giữa chúng. Việc này để giúp tạo nên đồ thị liên kết giữa các trang và tính toán giá trị PageRank về sau.
* Thực hiện lại quá trình với từng URL con trong trang.

Kết thúc quá trình ta đã có được tập từ khóa và trang web, là cơ sở cho quá trình tìm kiếm thông tin.



Hình ‑ Lưu đồ giải thuật TF-IDF

*Tính giá trị TF - IDF:* Trong quá trình phân tích trang web ta đã thực hiện việc tính toán giá trị TF đối với mỗi trang. Và sau quá trình phân tích toàn bộ, ta thực hiện việc tính giá trị TF-IDF hoàn chỉnh với tất cả các trang với số lượng trang, từ khóa và từ khóa trong mỗi trang đã có.

*Giá trị TF-IDF của 1 từ khóa K đối với 1 trang web P được xác định với công thức:*

*TF-IDF = TF \* IDF*

*với :*

*TF = C / T với C là số lần từ khóa K xuất hiện trong trang web W, T là tổng số từ trong trang web W*

*IDF = D / DF với D là tổng số trang web, DF là số lượng băn bản có chứa từ khóa K.*

*Tính giá trị PageRank:* với đồ thị trang web có được từ quá trình tìm kiếm, ta thực hiện việc tính toán giá trị PageRank cho từng trang với thuật toán đã được đề cập trong các phần trước.

### Log Miner:

Log Miner thực hiện chức năng phân tích các log file có được từ web server, tạo nên đồ thị liên kết và ma trận xác suất giữa các trang, và phân tích các phiên truy cập của người dùng.

1 thông tin weblog có dạng :

*127.0.0.1 [12/Jan/2010:23:14:15 +0700] "GET /uit.edu.vn/ HTTP/1.1" "-" 200 28138*

Với các thông tin về :

* IP yêu cầu trang web
* Thời gian thực hiện yêu cầu
* Trang web được yêu cầu và giao thức yêu cầu
* Referrer
* Status code

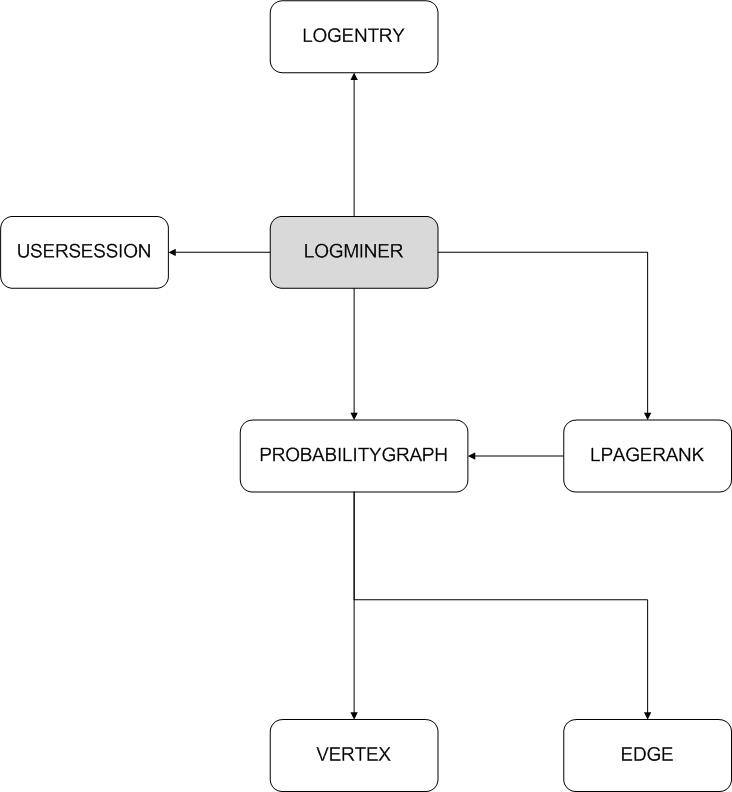
#### Cấu trúc dữ liệu:

Cấu trúc chính của Log Miner bao gồm các đối tượng dữ liệu và các đối tượng xử lý.

Trong đó 2 đối tượng xử lý chính là Log Miner – thực hiện việc phân tích các bản tin Weblog từ web server và xây dựng nên đồ thị liên kết giữa các trang; và LpageRank – thực hiện tính toán giá trị LpageRank cho các trang từ thông tin đồ thị liên kết.

Các lớp dữ liệu bao gồm : LogEntry và UserSession lưu giữ thông tin truy cập, ProbabilityGraph, Edge và Vertext lưu trữ các thông tin về đồ thị xác suất

Sơ đồ mối liên hệ giữa các lớp trong Log Miner:



Hình ‑ Quá trình phân tích website

#### Log Miner:

VERTEX: đối tượng miêu tả đỉnh của đồ thị, là các trang web.

EDGE: đối tượng biểu diễn cho cạnh của đồ thị, được hình thành bởi 2 trang web có liên kết với nhau, đỉnh trỏ đến được xem là đỉnh nguồn và đỉnh được trỏ đến được xem là đỉnh đích.

PROBABILITYGRAPH: đối tượng miêu tả đồ thị xác suất của website:

* Tập các cạnh trên đồ thị (Edge).
* Tập các đỉnh của đồ thị (Vertex).
* Ma trận xác suất giữa các đỉnh, là một ma trận vuông có chỉ mục là các đỉnh trong đồ thị, có giá trị là xác suất của của đỉnh A trỏ đến đỉnh B trong đồ thị
* Tập liên kết đến và đi từ 1 đỉnh

LPAGERANK: đối tượng thực hiện việc tính giá trị LpageRank cho các đỉnh (trang web), sử dụng một đối tượngProbabilityGraphđể lấy các thông tin về đồ thị.

LOGENTRY: đối tượng lưu trữ thông tin về 1 bản tin weblog

* IP yêu cầu trang web
* Thời gian thực hiện yêu cầu
* Trang web được yêu cầu và giao thức yêu cầu
* Referrer
* Status code

USERSESSION: biểu diễn một phiên truy cập của người dùng.

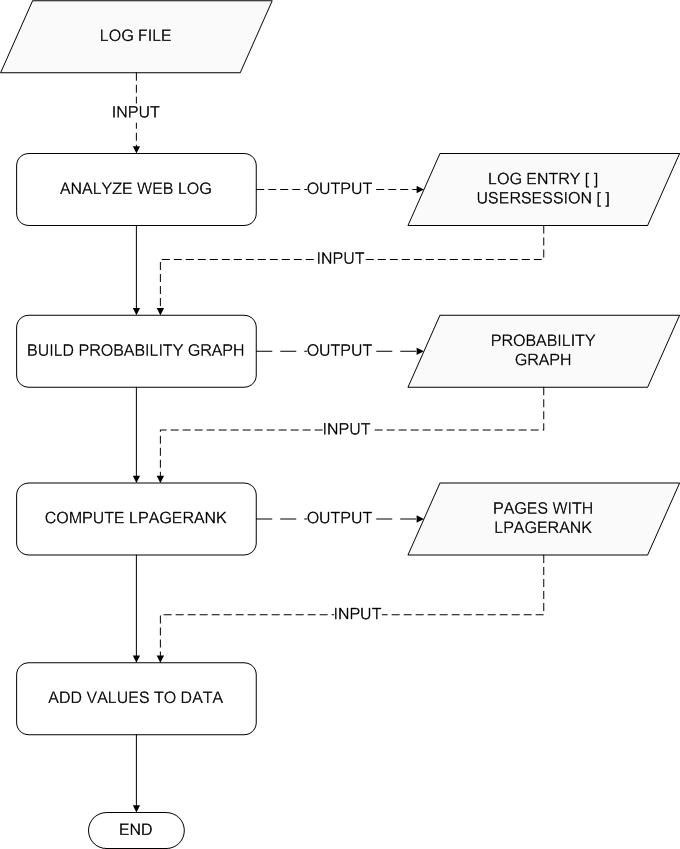
* Tập các trang web đã truy cập và thời gian truy cập tương ứng
* Tổng thời gian truy cập của phiên.

LOGMINER: lớp xử lý chính, thực hiện việc phân tích các file weblog từ web server, xây dựng nên đồ thị xác suất và tính giá trị LpageRank cho các trang web.

Dữ liệu input cho quá trình là các thông tin weblog từ web server, và dữ liệu output là các cặp trang web và giá trị LpageRank tương ứng, dữ liệu này sẽ được dùng làm 1 phần đánh giá kết quả tìm kiếm của mỗi kết quả tìm kiếm (trang web).

Quá trình hoạt động cúa Log Miner bao gồm các bước chính:

* *Phân tích thông tin weblog:*dữ liệu input là các file weblog được cung cấp từ web server, thực hiện phân tích từng dòng trong log file, lưu lại thông tin về địa chỉ IP truy cập, thời gian truy cập, phương thức truy cập, địa chỉ yêu cầu, địa chỉ trỏ đến, kết quả truy cập, thanh lọc weblog bằng cách loại bỏ những dòng server ghi lại của người dùng yêu cầu image, video, audio và những trang yêu cầu thất bại. Kết quả của bước này là tập các bản tin truy cập và phiên truy cập của người dùng. Các phiên truy cập được xác định thông qua các bản tin có IP giống nhau trong 1 thời gian liên tục.
* *Xây dựng đồ thị xác suất:* Xây dựng đồ thị xác suất từ các thông tin truy cập có được.
* *Tính giá trị LpageRank:*thực hiện tính giá trị LpageRang cho từng trang (đỉnh đồ thị) bằng phương pháp đã được trình bày ở các phần trước.
* *Lưu giá trị:* lưu lại các giá trị vào dư liệu trên đĩa, thêm giá trị LpageRank vào các CachedFile tương ứng từ quá trình phân tích Web Site.

**

Hình ‑ Sơ đồ hoạt động của Log Miner

### Webpage Classifier:

Thành phần Neural Network thực hiện việc thiết lập hệ thống Fuzzy Artmap.

Nhắc lại, các chỉ số dùng để đánh giá một trang web là :

* Điểm LpageRank
* Độ cập nhật
* Cấp độ (“độ xa” so với tên miền gốc)

#### Cấu trúc dữ liệu:

Gồm các lớp chính : Matrix – lưu trữ thông tin về các mẫu để phân lớp và kết quả quá trình phân lớp, ARTMAPNetwork – mạng neural dùng trong quá trình phân lớp, ART và ARTMAP – lớp chính thực hiện quá trình học và phân loại trang web.

MATRIX: lớp ma trận biểu diễm cho tập mẫu và kết quả quá trình phân lớp.

* Column : số cột của ma trận
* Row : số hàng của ma trận
* Values : Chuỗi 2 chiều biểu diễn các giá trị của ma trận.

Ma trận M biểu diễn tập gồm X mẫu với Y thuộc tính sẽ là một ma trận gồm X cột và Y hàng với M[x,y] là giá trị của thuộc tính y cùa mẫu x.

Ma trận M biểu diễn giá trị kết quả phân loại của X mẫu là ma trận gồm X cột và 1 hàng với M[x,1] là giá trị phân loại của mẫu x.

ARTMAPNETWORK: lớp biểu diễn mạng neural dùng trong quá trình phân lớp.

* numFeatures: số lượng thuộc tính dùng để phân lớp.
* numCategories: số lượng lớp (tốt, trung bình, xấu).
* maxNumCategories: số lượng lớp tối đa.
* weight: ma trận trọng số của mỗi lớp.
* mapField: ma trận sử dụng trong quá trình so khớp.
* vigilance : tham số kiểm tra độ giống nhau giữa 2 mẫu.
* bias : độ lệch.
* numEpochs: số vòng lặp.
* learningRate: bước học.

Trong quá trình, một mạng neural với n thuộc tính và m lớp được khởi tạo với cái giá trị mặc định sau :

*numFeatures = n*

*numCategories = 0*

*maxNumCategories = 100*

*numClasses = m*

*weight = Matrix[n, 0]*

*mapField = Matrix[0, 0]*

*vigilance = 0.75*

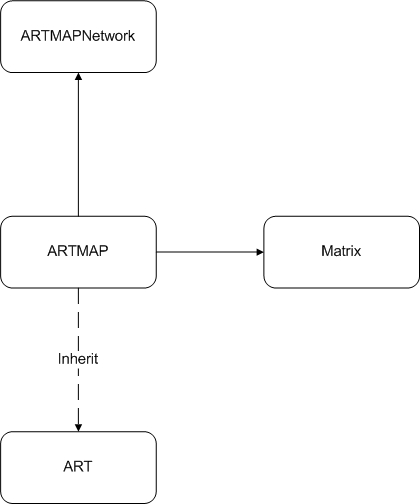
*bias = 0.000001*

*numEpochs = 100*

*learningRate = 1.0*

ART và ARTMAP: các lớp xử lý chính, thực hiện 2 việc chính :

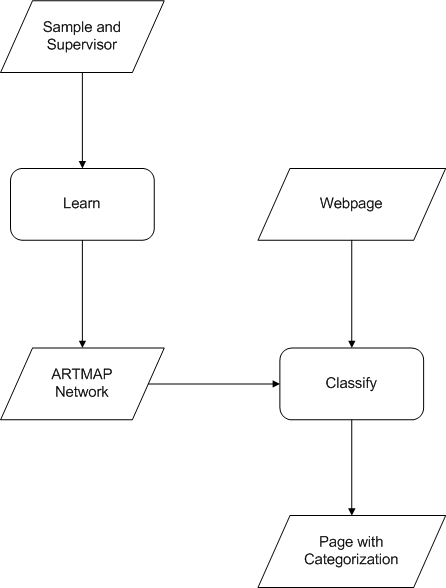
* Học từ tập mẫu và giám sát cho trước để tạo nên mạng neural phân lớp.
* Phân loại các mẫu mới từ mạng neural đã học được.



Hình ‑ Quan hệ giữa các lớp

#### Webpage Classifier:

Hoạt động của quá trình phân lớp gồm 2 quá trình chính : quá trình học có giám sát và quá trình phân lớp với mạng neural có được từ quá trình học.



Hình ‑ Hoạt động của bộ phận lớp

*Quá trình học:* thực hiện học từ mẫu và giám sát được cung cấp sẵn, tạo nên mạng neural phân lớp. Đầu vào của quá trình là 2 ma trận biểu diễn cho tập dữ liệu mẫu (mỗi dữ liệu là một tập các thuộc tính của một trang web) và tập lớp kết quả phân loại (tốt, trung bình, xấu), qua mỗi mẫu và mỗi vòng lặp của quá trình, các trọng số phân lớp và so khớp của mạng neural sẽ được cập nhật, “huấn luyện” để sinh ra một mạng neural có thể thực hiện phân loại đối với các mẫu mới.

Quá trình học được thực hiện nhiều lần đến khi không có thay đổi nào xảy ra hoặc đạt đến số vòng lặp cho trước (100).

Hoạt động của mỗi vòng lặp được miêu tả như sau:

* + Lấy từng mẫu trong tập và kết quả mong muốn tương ứng đã được cho trước.
  + Kiểm tra mapfield của mạng neural:
    - Nếu mapfield rỗng hoặc không có chứa kết quả, thêm vào tập một lớp (kết quả) mới.
    - Ngược lại thực hiện phân lớp mẫu vào 1 lớp thích hợp:
      * Kiểm tra so khớp mẫu với từng lớp:
        + Nếu khớp, cập nhật ma trận trọng số với mẫu hiện tại.
        + Ngược lại, thêm mẫu mới vào tập.

*Quá trình phân lớp :* thực hiện phân loại trang web vào một lớp thích hợp, đầu vào của quá trình là một trang web với các thông tin liên quan. Quá trình sẽ thực hiện so khớp trang web với các lớp để tìm ra lớp thích hợp nhất.

### Social Annotation

Social Annotation là lớp thực hiện việc tìm kiếm và sắp xếp các trang tìm kiếm dựa trên sự tương đồng của các chú thích từ tài nguyên có sẵn của người dùng hay từ những trang Web cung cấp.

Quá trình tìm kiếm gồm 3 bước:

* Xử lý câu lệnh truy vấn.
* Tìm kiếm thông qua xếp hạng tương đồng giữa truy vấn và chú thích.
* Sắp xếp lại bằng việc đánh giá định lượng các trang phổ biến.

Xử lý câu lệnh truy vấn: Xử lý câu lệnh tìm kiếm, xuất ra tập các từ khóa (từ sự tương đồng của các chú thích) và các tùy chỉnh tìm kiếm.

* Nhận diện các cấu trúc tìm kiếm định trước (tìm theo Tags).

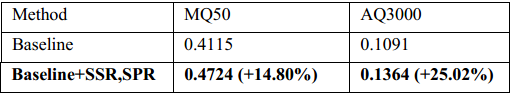
*Tìm kiếm:* thực hiện việc tìm kiếm dựa trên tập các tags từ tập dữ liệu tài nguyên của trang web đó. Kết quả bước này là tập các trang có chú thích tương đồng phù hợp với từ khóa.

*Sắp xếp kết quả:* thực hiện việc sắp xếp các kết quả tìm kiếm dựa trên các chỉ số đánh giá chú thích của từng trang Web.

**

Hình ‑ Chỉ số xếp hạng của một trang web

Bằng việc kết hợp cả SSR và SPR, chúng ta có thể nhận được một kết quả tìm kiếm tốt thông qua bảng 1, T-Test trên MAP cho thấy việc cải thiện tìm kiếm mang tính thống kê (p-value < 0.05)



Hình ‑ Sự kết hợp giữa SSR và SPR (tham khảo Optimizing Web Search Using Social Annotations – Dynamic Ranking).

🟑 Đánh giá kết quả từ SocialSimRank (SSR):

Qua một số những thử nghiệm, thuật toán SocialSimRank trong 12 vòng lặp chú thích cho ra bốn sự lựa chọn cùng với ngữ nghĩa liên quan đến nhau:

|  |  |
| --- | --- |
| **Liên quan đến công nghệ** |  |
| dublin | Siêu dữ liệu, ngữ nghĩa, tiêu chuẩn |
| debian | Phân bố, distro, ubuntu, linux |
| **Liên quan đến kinh tế** |  |
| adsense | Ý nghĩa, quảng cáo, doanh nghiệp, tiền |
| 800 | Số lượng, bảng kiểm kê, điện thoại, kinh doanh |
| **Liên quan đến vui chơi giải trí** |  |
| album | ảnh, nhiếp ảnh, bức tranh toàn cảnh |
| chat | Nói luyên thuyên, tin nhắn nhanh, truyền tin |
| **Liên quan đến thực thể (Entity)** |  |
| Einstein | Khoa học, hoài nghi, sự tiến hóa, lượng tử |
| Christian | Cống hiến, đức tín, tôn giáo, thần |

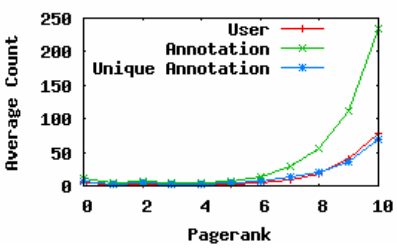
Bảng 5‑1 Đánh giá kết quả của giải thuật Social Annotation (tham khảo Optimizing Web Search Using Social Annotations).

🟑 Đánh giá kết quả từ SocialPageRank (SPR):

Qua một số thử nghiệm, thuật toán SocialPageRank sử dụng 7 vòng lặp, mỗi PageRank của trang cũng lấy từ thanh công cụ Google API trong tháng 7 năm 2006. Sau đây, chúng tôi sẽ sử dụng PageRank để mô tả cách trích của PageRank Google bằng cách mặc định.

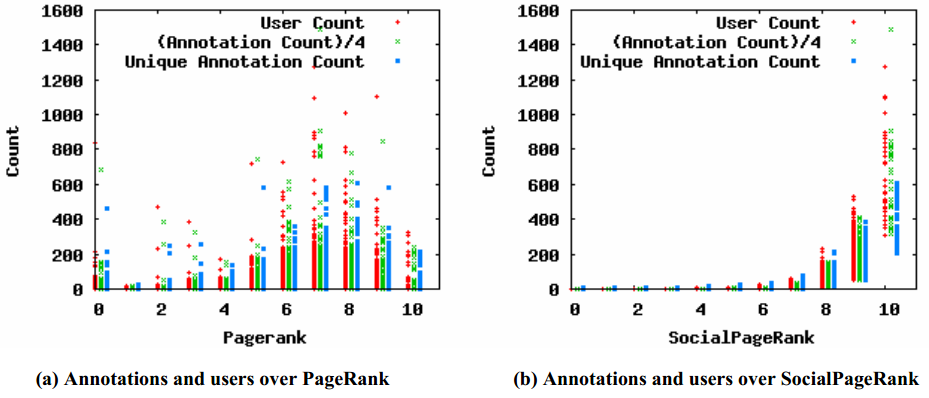
* SocialPageRank với PageRank (phân tích phân bố):

Hình 5-12 cho thấy số lượng trung bình của các chú thích và các annotator trên trang web với các giá trị PageRank khác nhau và cho ra những dòng *chú thích độc đáo* (Unique Annotation) có ý nghĩa giá trị đặc biệt khác so với những chú thích khác. Từ những thông tin này, chúng ta có thể kết luận rằng trong nhiều trường hợp, các trang với một PageRank cao hơn có khả năng được chú thích bởi nhiều người dùng hơn với các chú thích nhiều hơn.



Hình ‑ Phân bố số lượng trung bình trên PageRank (tham khảo Optimizing Web Search Using Social Annotations).

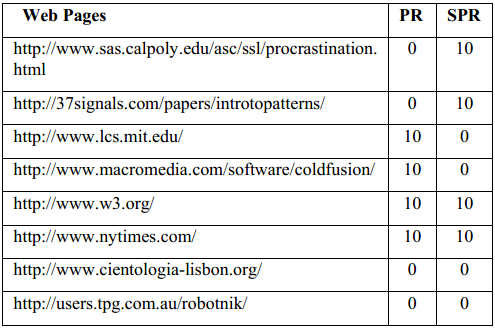
Tiếp tục sự điều tra phân bố số lượng các chú thích và annotators cho tất cả các trang thu thập được với PageRank khác nhau, được đưa ra trong hình 5-13(a) .Dễ để thấy rằng các trang với mỗi giá trị PageRank đa dạng sẽ tỉ lệ vào số lượng các chú thích và người dùng.Hình 5-13(b) cho thấy số lượng chi tiết của chú thích, người sử dụng các trang có giá trị SocialPageRank khác nhau. Dễ thấy rằng SPR thành công cho độ phổ biến của các trang trong annotators web.



Hình ‑ Những phân tích phân bố của Social Annotation (tham khảo Optimizing Web Search Using Social Annotations).

* SocialPageRank với PageRank (Case Study)

Bảng 5-2 cho thấy 8 Case Study của PageRank với SPR. Một số trang web có SRP cao nhưng lại có PageRank thấp. Từ các Case Study chúng tôi kết luận rằng những người tạo rat rang web khác với những người sử dụng nó (những người chú thích trang web), đó là sự thành công của SocialPageRank.



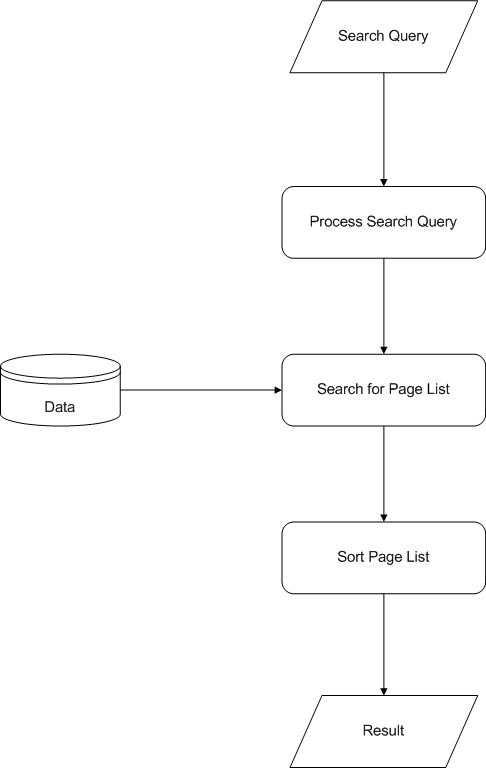
Bảng 5‑2 Case Study của SocialPageRank với PageRank (tham khảo Optimizing Web Search Using Social Annotations).

### Search Engine:

*Search Engine* là lớp thực hiện việc tìm kiếm và sắp xếp các trang tìm kiếm dựa trên tập từ khóa truy vấn và tập dữ liệu có được từ các quá trình trước đó.

Quá trình tìm kiếm gồm 3 bước chính :

* Xử lý câu lệnh truy vấn.
* Tìm kiếm tập các trang phù hợp với truy vấn.
* Sắp xếp lại tập trang kết quả dựa trên các chỉ số.



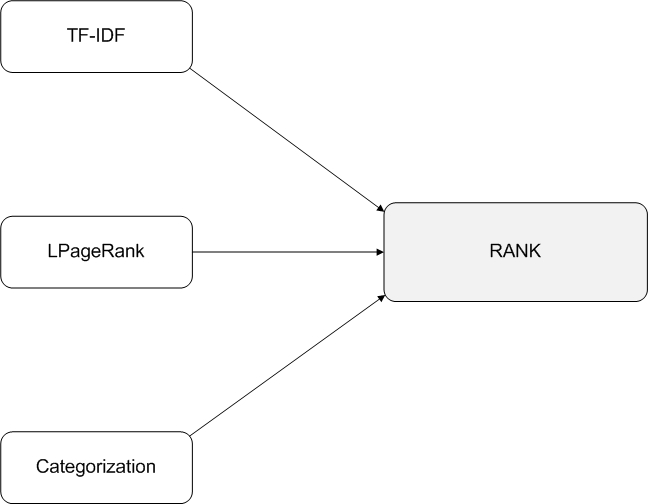
Hình ‑Quá trình tìm kiếm của Search Engine

*Xử lý câu lệnh truy vấn* : Xử lý câu lệnh tìm kiếm, xuất ra tập các từ khóa và các tùy chỉnh tìm kiếm

* Thực hiện làm sạch câu truy vấn, loại bỏ các ký tự thừa, khoảng trắng, nhận diện GoWord.
* Nhận diện các cấu trúc tìm kiếm định trước (tìm theo từ khóa, theo tiêu đề, theo URL, phương pháp tìm kiếm mong muốn,…)

*Tìm kiếm* : Thực hiện việc tìm kiếm dựa trên tập từ khóa từ tập dữ liệu. Kết quả của bước này là tập các trang có nội dung phù hợp với từ khóa. Tùy theo câu truy vấn mà có thể xử lý thêm về tiêu đề và URL của trang.

*Sắp xếp tập kết quả* : Thực hiện việc sắp xếp các kết quả tìm kiếm dựa trên các chỉ số.



Hình ‑ Chỉ số xếp hạng của một trang web

Với mỗi trang web trong tập kết quả, ta tạo một trọng số P dựa trên 3 trọng số đã tính trước đó là TF-IDF, LpageRank, Categorization (phân loại tốt xấu trang web).

P = TF-IDF + LpageRank + Categorization

Chỉ số TF-IDF là tổng của từng chỉ số TF-IDF đối với nội dung, tiêu đề và URL.

Chỉ số phân loại Categorization (C):

*C = LpageRank + (- Log10(Level)) + Update + Visit + AccessTimeInterval*

Nếu trang web được phân loại tốt :*C = C*

Nếu trang web được phân loại trung bình :*C = C / 2*

Nếu trang web được phân loại xấu :*C = C / 10*

### Lưu trữ dữ liệu:

#### Lưu trữ dưới dạng database xml:

Như đã đề cập trước đó, các quá trình phân tích web lấy từ khóa, phân tích tập weblog lấy thông tin LpageRank và phân loại trang web được thực hiện trước, sau đó tập các từ khóa, trang web và các trọng số được lưu trữ lại trên đĩa cứng và bộ nhớ cache của server. Sau cùng, Search engine sẽ dùng những thông tin có được này để thực hiện tìm kiếm.

Các dữ liệu lưu trữ trên đĩa cứng sẽ được lưu trữ dưới dạng file XML với 2 file catalog.xml và catalog-cache.xml.

CATALOG.XML:

File lưu trữ thông tin về từ khóa, với tập các từ và trang web có chứa từ khóa tương ứng và trọng số TF-IDF, tập vị trí của từ khóa trong trang web đó; và tập các trang web với các thông tin cơ bản.

Cấu trúc xml của 1 từ khóa bao gồm

* Nội dung từ khóa
* Vị trí của từ khóa trong từng trang web với dạng *fff:xx,yy,zzz…*với fff là id của trang web, xx, yy, zz là tập các vị trí của từ khóa trong trang đó
* Giá trị tf-idf của từ khóa đối với từng trang web có chứa có dạng fff:xxx với fff là id của trang web, xxx là giá trị TF-IDF của từ khóa đối với trang web đó.
* Ví dụ cấu trúc xml của 1 từ khóa :

*<w>*

*<t>word</t>*

*<ii>133:893</ii>*

*<ii>134:870</ii>*

*<ii>135:870</ii>*

*…*

*<f>105:0.0242102951160097</f>*

*<f>133:0.0262411807990304</f>*

*<f>134:0.0272019562973931</f>*

*<f>135:0.0264161220043573</f>*

*…*

*</w>*

Cấu trúc xml của 1 trang web bao gồm

* Id của trang web.
* URL.
* Định dạng file (tại liệu được phân tích có thể không phải là 1 file web).
* Thời gian trang web được phân tích.
* Độ lớn của trang web (Byte)
* Miêu tả của trang web
* Ví dụ cấu trúc xml của 1 từ khóa :

*<f*

*id="295"*

*u="http://cktd.ntt.edu.vn/index.php?view=xxx"*

*e="pdf" d="2010-08-12T03:42:15.9933022+07:00"*

*s="240358">*

*<d>DS SV tham gia tiếp sức mùa thi của trường năm...</d>*

*</f>*

CATALOG-CACHE.XML:

Lưu thông tin chi tiết về 1 trang web với các thông tin về nội dung, các trọng số đã được tính.

Cấu trúc thông tin của 1 trang web được lưu trữ trong file:

* Id của trang web.
* URL.
* Tập từ, nội dung của trang web
* Chỉ số PageRank.
* Chỉ số LpageRank.
* Cấp độ
* Tần suất truy cập
* Độ lỗi
* Phân lớp của trang (tốt, trung bình, xấu)
* Ví dụ cấu trúc xml của 1 trang web:

*<f>*

*<i>156</i>*

*<u>http://cn.ntt.edu.vn/images/xxx</u>*

*<w>Sheet1</w>*

*<w>00000000</w>*

*<w>TTCT&amp;CBT</w>*

*<w>HK1</w>*

*<w>TC</w>*

*<w>CĐ</w>*

*…*

*<r>0.27749999999999997</r>*

*<l>0</l>*

*<v>0.047091412742382273</v>*

*<s>0</s>*

*<t>0</t>*

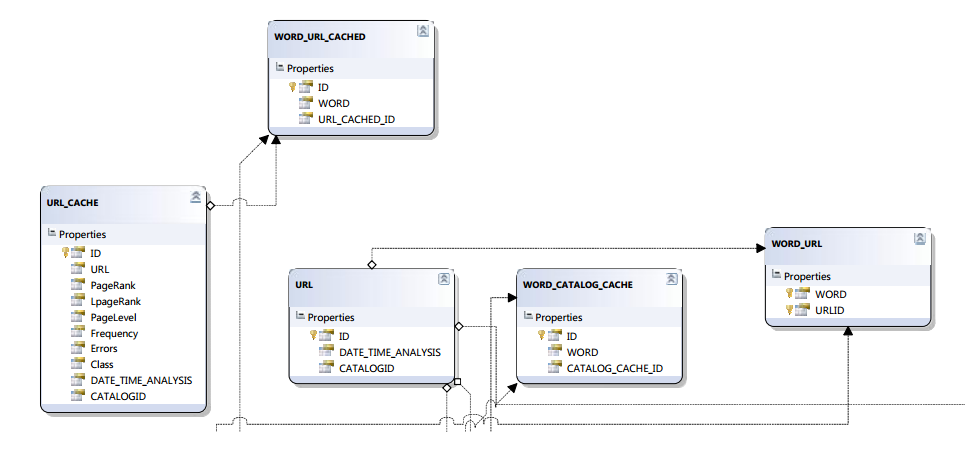
*<e>0</e>*

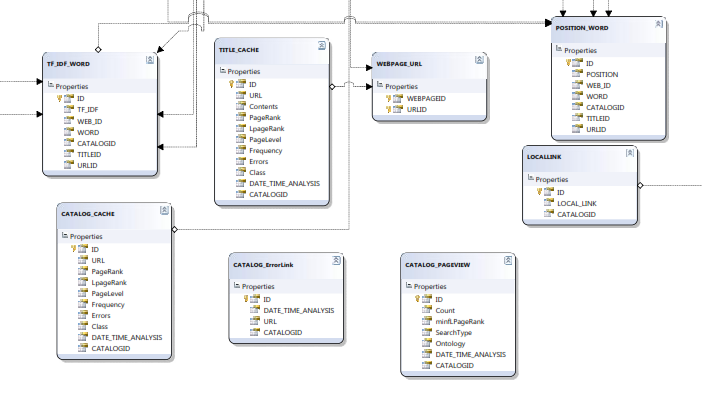
*<c>2</c>*

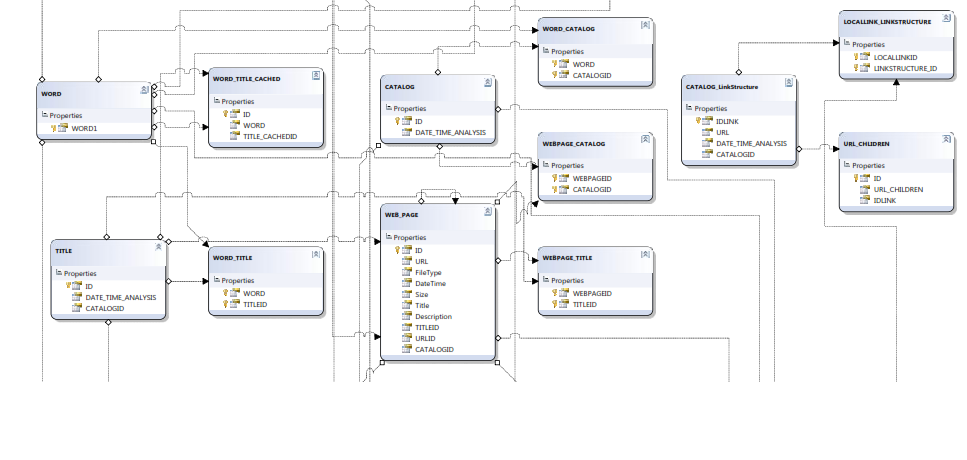
Ngoài ra, để hỗ trợ cho việc tìm kiếm, các thông tin trong tiêu đề, URL của trang web cũng được phân tích và lưu trữ (Các file catalog-Title.xml, catalog-Title-Cache.xml, catalog-Url.xml, catalog-Url-cache.xml)

#### Lưu trữ vào database SQL:

Đặt đồng thời song song với cơ sở dữ liệu xml chương trình còn hỗ trợ thêm cơ sở dữ liệu xml nhằm tối ưu các thành phần dữ liệu, ràng buộc các thuộc tính nhằm tối ưu hóa thời gian lấy dữ liệu tìm kiếm.







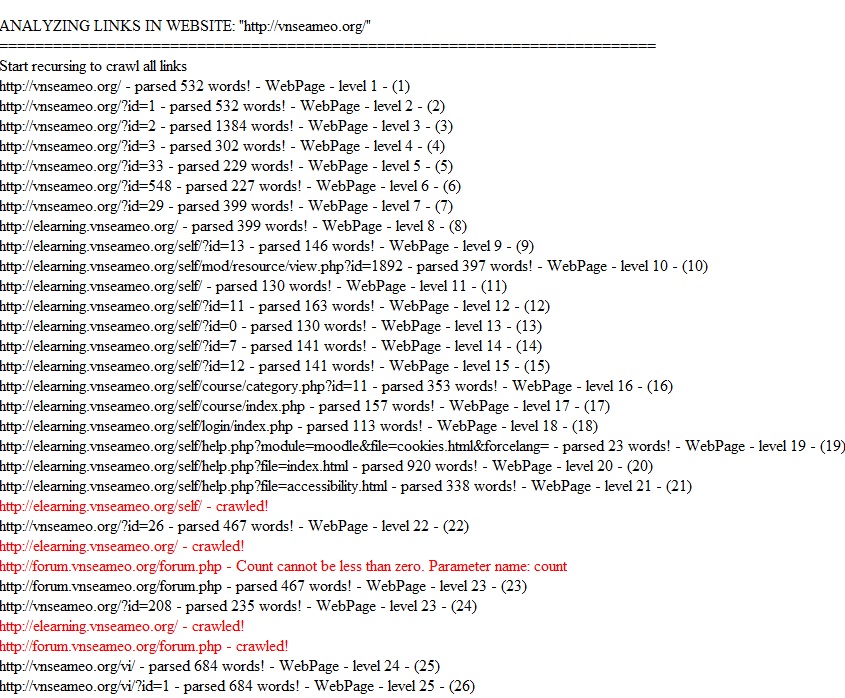
Hình ‑ Mô tả cơ sở dữ liệu

## Demo:

### Demo:

#### Phân tích trang web:

Phân tích trang web cần tích hợp công cụ search engine vào để lấy dữ liệu về database. Khi phân tích chương trình sẽ sử dụng thuật toán fuzzy ARTMAP và Neural Network để phân lớp các trang web được phân tích.



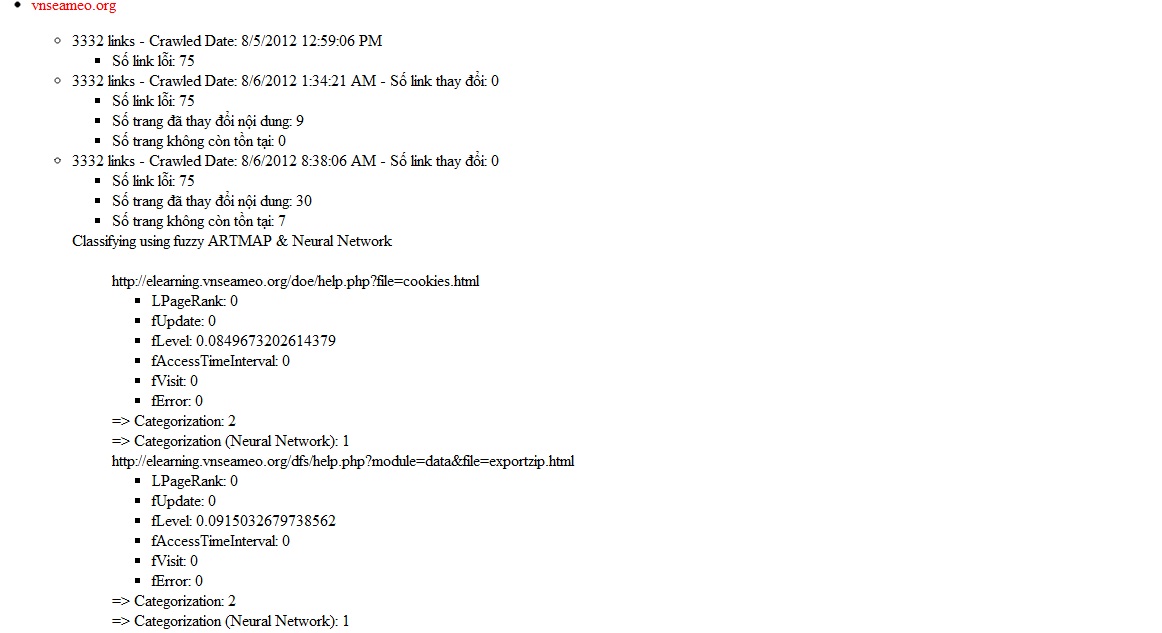
Hình ‑ Phân tích trang web



Hình ‑ Kêt quả phân tích

#### Đánh giá trang web:

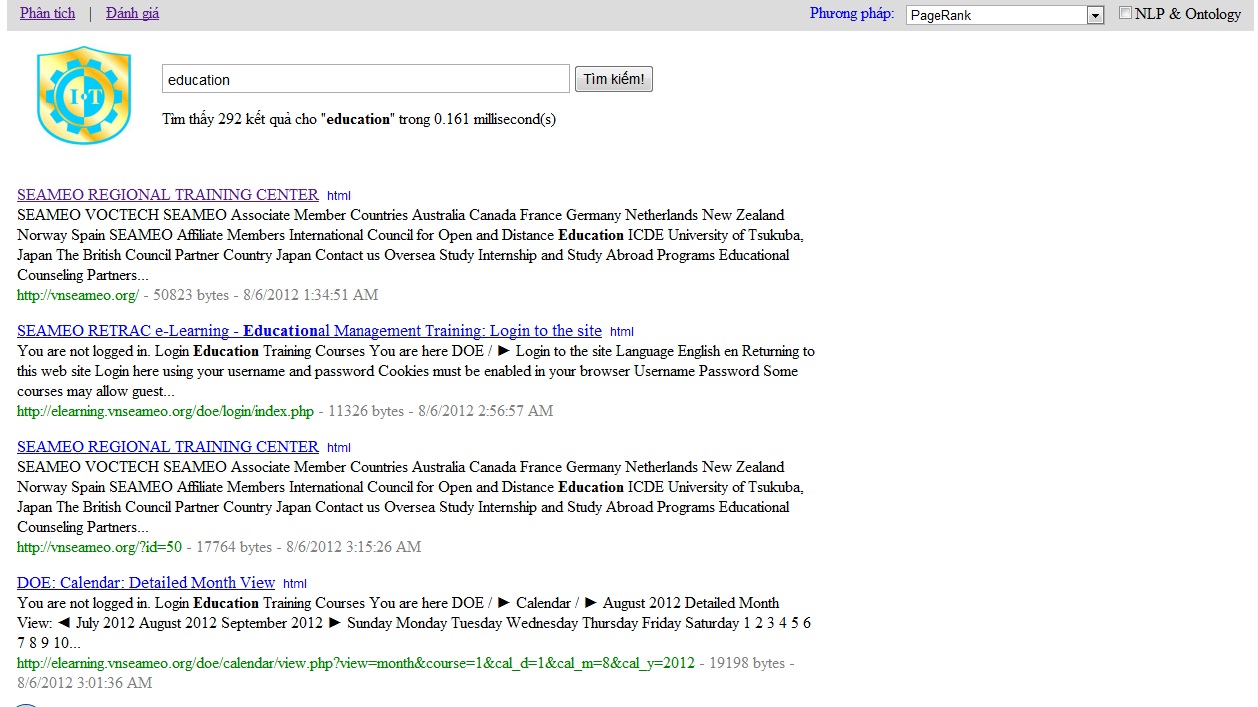
Đánh giá trang web theo các tiêu chí: tổng số link, số link lõi, số trang thay đổi nội dung, số trang không tồn tại, chi tiết các trang đã được phân lớp ở phần phân tích dữ liệu.



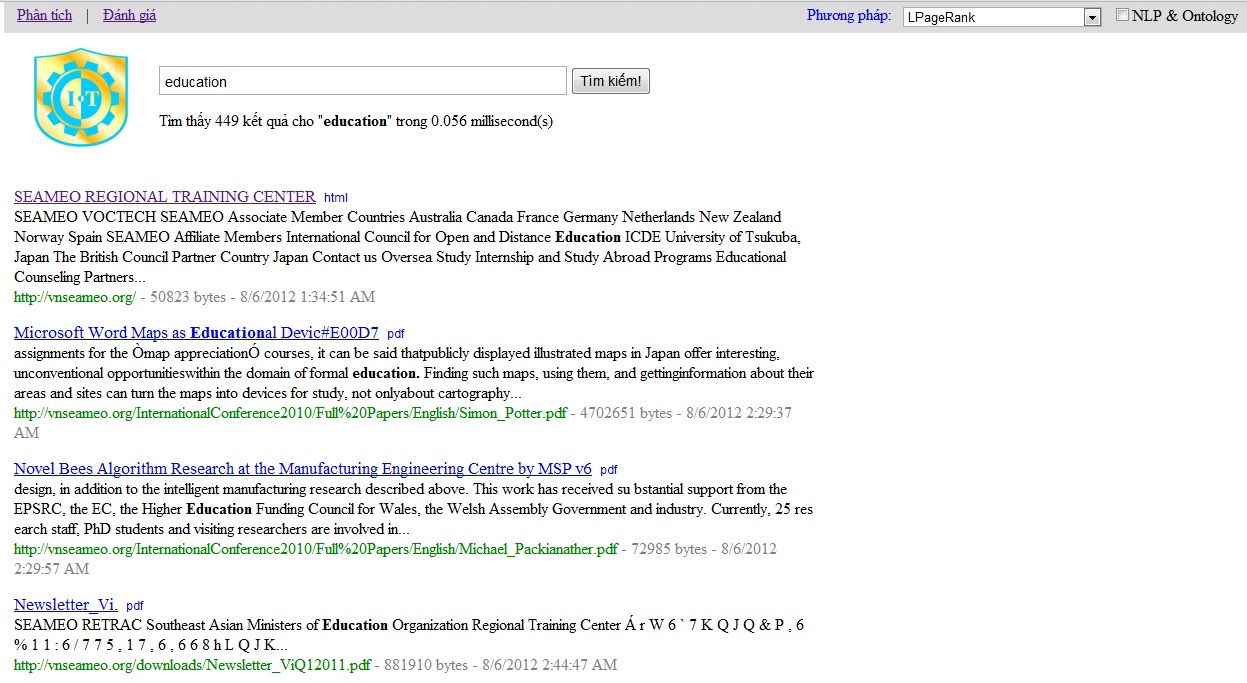
Hình ‑ Đánh giá trang web

#### Tìm kiếm:

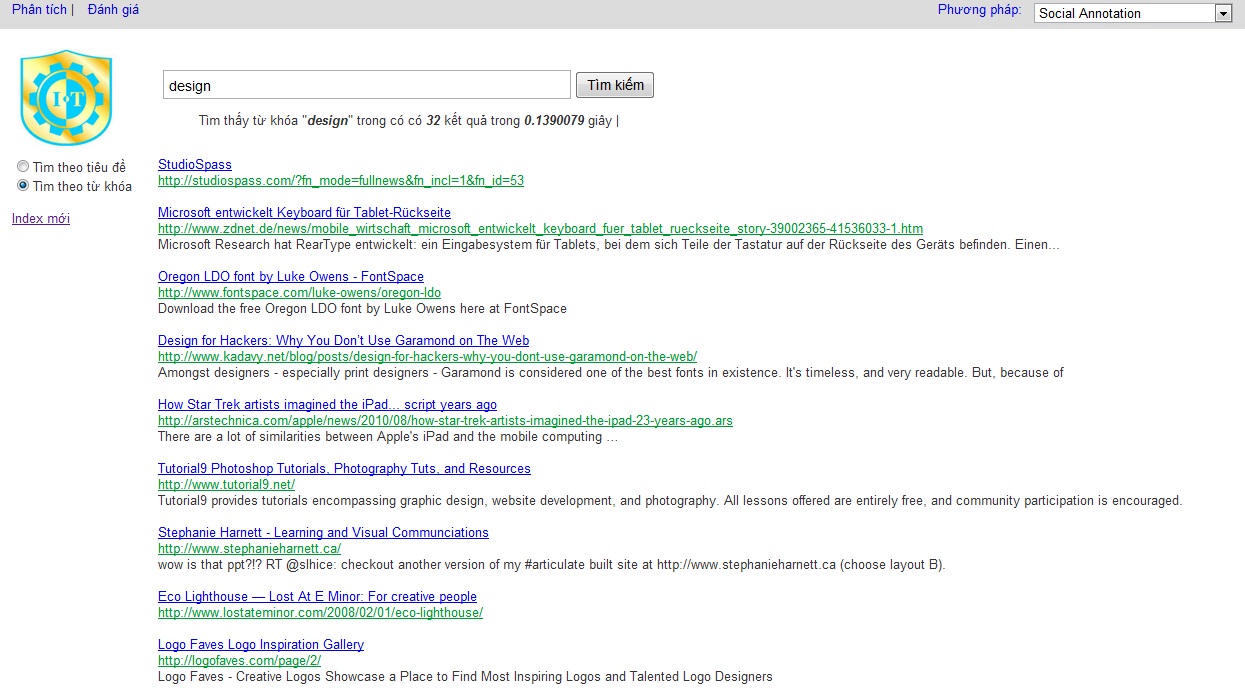
Tìm kiếm các trang web có chứa từ khóa trong dữ liệu đã được phân tích theo các thuật toán, mỗi thuật toán cho ra số lượng kết quả, thời gian khác nhau. Đồng thời sử dụng các thuật toán để sắp xếp các trang web có độ chính xác cao nhất lên đầu, tiếp theo đó là các trang web có độ tương quan cao với trang web tìm được trước đó. Tính chính xác được tính theo nội dung của từ khóa, số lượng từ khóa trong một trang web, độ quan tâm của người dùng đến trang web đó.



Hình ‑ Kết quả tìm kiếm với thuật giả PageRank



Hình ‑ Kết quả tìm kiếm với thuật giả LPageRank



Hình ‑ Kết quả tìm kiếm với thuật giả Social Annotation

### Kết quả thực hiện:

Để kiểm chứng và so sánh kết quả tìm kiếm của ứng dụng, khóa luận đã so sánh giữa các phương pháp khác nhau.Trong đó, khóa luận đã kiểm chứng độ chính xác của tài liệu tìm kiếm của phương pháp phương pháp TF-IDF và LpageRank cùng với phương pháp Social Annotation có hỗ trợ của mạng neural fuzzy ARTMAP với kết quả tìm kiếm.

Trong mô hình với mạng neural fuzzy ARTMAP, mô hình đã phản ánh được sự quan tâm của người dùng lên kết quả tìm kiếm, những trang được xếp hạng cao thường là những trang có nhiều người dùng truy cập, thời gian truy cập lâu, tần suất lỗi thấp, phản ánh được thông tin trong trang web đó là quan trọng và hướng nó gần hơn đến người dùng. Đồng thời những trang web có thông tin quan trọng nhưng do độ sâu của trang web làm cho người dùng khó có thể tiếp cận cũng được xếp hạng cao để nó có thể đến gần hơn với người dùng. Tỉ lệ trang web tìm được với từ khóa tìm kiếm ít hơn do hệ thống đã lọc những trang có từ khóa nhưng số lượng truy cập quá thấp hoặc không được truy cập và cũng làm tăng độ chính xác khi tìm kiếm.

# KẾt LuẬn

## Kết luận:

### Ưu điểm

Mô hình ứng dụng khai thác weblog cải tiến giải thuật PageRank trong tìm kiếm thông tin trên trang web cục bộ đã cho kết quả khá khả quan với sự kết hợp giữa giải thuật LPageRank và mạng neural fuzzy ARTMAP. Kết quả tìm kiếm không chỉ khắc phục sự khô cứng trong việc tìm kiếm chính xác các từ khóa trong câu truy vấn mà còn thể hiện kết quả một cách uyển chuyển sự quan tâm của người dùng lên một tài liệu tìm được. Kết quả tìm kiếm trả về cho người dùng là kết quả của việc phân tích từ khóa từ mô hình TF-IDF, kết hợp với đồ thị xác suất được xây dựng từ weblog thể hiện tần suất truy cập của người dùng lên tài liệu, kết quả từ việc đánh giá, gom nhóm các trang web của mạng neural thể hiện khả năng truy xuất của người dùng trong tương lai. Kết quả tìm kiếm đã cho thấy mô hình ứng dụng weblog trong công cụ tìm kiếm là một hướng nghiên cứu rất khả thi và có tiềm năng ứng dụng rất cao đối với các công cụ tìm kiếm trang web cục bộ.Ngoài ra, mô hình trên cũng cho thấy khả năng ứng dụng của mạng neural fuzzy ARTMAP trong việc khai thác thông tin weblog.Một lĩnh vực mà mạng neural có khả năng ứng dụng để tối ưu các phương pháp tìm kiếm thông tin phổ biến trên weblog.

Đồng thời qua đồ án ta thấy được Social Annotatio là một cách mới cho người dùng để tìm ra thông tin, tên miền,..dựa vào những social Annotation. Những giải thuật được đề xướng kết hợp với ngôn ngữ mô hình hóa, cộng thêm Fuzzy Artmap làm cho việc tìm kiếm dễ dàng và thuận lợi hơn.Qua đây cũng cho thấy rằng có thể ước lượng được sự thành thạo của người sử dụng thông qua cường độ hoạt động.Sự đánh giá trên dựa trên dữ liệu thực tế dựa trên mô hình ngôn ngữ.Trong tương lai, con người sẽ có cách nhìn tổng quan hữu ích về việc giảm thiểu những tham số trọng yếu để có thể cải thiện được tốc độ tìm kiếm.Kích thước thời gian, những mô hình thay đổi trạng thái sẽ được chú thích và phát triển.

### Hạn chế:

Phương pháp tìm kiếm theo từ khóa được tối ưu hóa bằng cách sử dụng weblog tuy đã được những kết quả khả quan bằng phương pháp đánh giá của khóa luận, những kết quả mà phương pháp này dựa vào thông tin weblog và một số thông tin đánh giá website khác, tuy phản ánh được độ quan tâm của người dùng vào các trang web mà họ mong muốn, những trang web này sẽ được xếp hạng sao cho gần với người dùng nhất nhưng điều này không mang tính trực quan cao nên đôi khi người dùng chưa nhận ra được điều đó, điều này cũng là hạn chế của công cụ tìm kiếm theo từ khóa.

### Những nét mới của khóa luận:

Trong phương pháp tối ưu hóa công cụ tìm kiếm theo từ khóa, khóa luận đã tự đề xuất các tiêu chuẩn đánh giá website và sử dụng mạng neural fuzzy ARTMAP để phân loại các trang web dựa vào tiêu chuẩn trên.

Và quan trọng hơn, khóa luận đã tự đề xuất mô hình tìm kiếm để làm thế nào có thể hướng đến người dùng, đáp ứng được nhu cầu sử dụng công cụ tìm kiếm của người dùng, làm cho người dùng cảm thấy hài lòng khi sử dụng công cụ tìm kiếm.

## Hưởng mở rộng đề tài:

Công cụ tìm kiếm luôn luôn cần phải cải tiến để ngày một đáp ứng nhu cầu của người dùng nhiều hơn.Đối với khóa luận đã đáp ứng được mục tiêu đề ra nhưng việc cải tiến cho công cụ tìm kiếm của khóa luận luôn là điều cần thiết. Một số hướng phát triển như sau:

- Xây dựng thêm cơ chế tách từ khóa và xử lý từ khóa thành kho dữ liệu cung cấp cho công cụ tìm kiếm thay cho cơ chế sử dụng giải thụât TF-IDF. Xây dựng kho dữ liệu từ khóa cho trang web để tăng độ chính xác, độ bao phủ và tốc độ xử lý của chương trình, giúp cho việc đồng bộ tập tin log trên weblog động để chương trình luôn có kết quả đánh giá tốt khả năng truy cập của người dùng lên một trang web.

**-**Phát triển thành công cụ tìm kiếm toàn cục cho tất cả các trang web tiếng Việt, và hơn nữa là tìm kiếm đa ngôn ngữ.

- Cải tiến quá trình xử lý ngôn ngữ tự nhiên để có thể nhận dạng được nhiều mẫu câu hỏi hơn, từ đó có thể đáp ứng được nhu cầu truy vấn tìm kiếm đa dạng của người sử dụng.

- Phát triển hệ thống tìm kiếm thông minh có thể trả lời những câu hỏi về toán học, hóa học, vật lý phục vụ cho nhu cầu tìm kiếm, học tập của tầng lớp tri thức như giáo viên, học sinh, sinh viên.

Danh mỤc Tài LiỆu Tham KhẢo

1. Sergey Brin and Lawrence Page, “The anatomy of a largescale hypertextual Web search engine”, *Proceedings of the seventh international conference on World Wide Web 7*, vol. 30, pp. 107-117, Apr. 1998.
2. Quing Cui and Alex Dekhtyar. “On improving local website search using web server traffic logs: a preliminary report”, *Proceedings of the 7th annual ACM international workshop on Web information and data management*, pp. 59-66, Nov. 2005.
3. Jin Zhou, Cheng Ding and Dimitrios Androutsos, “Improving Web Site Search Using Web Server Logs”, *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research*, article no. 22, Oct. 2006.
4. Jean-François Connolly, Eric Granger and Robert Sabourin, “An Adaptive Ensemble of Fuzzy ARTMAP Neural Networks for Video-Based Face Classification”**.**
5. Ben Kröse (NetherLands) & Patrick van der Smagt (Germany), “An Introduction to Neural Network**”.**
6. TANG WENG CHIN, “Feature selection for the Fuzzy Artmap Neural network using a hybrid genetic algorithm and tabu search**”.**
7. Victor Y Safronov, Thesis Director: Professor Manish Parashar. Abstract of the thesis “*Optimizing Web Using Page Rank Prefetching for Clustered Accesses*”; New Brunswick, New Jersey May-2001.
8. Fan Chung, Alexander Tsiatas – Department of Computer Science and Engineering University of California, San Diego. “*Finding and Visualizing Graph Clusters Using PageRank Optimization*” .
9. R. Andersen, F. Chung and K. Lang. Local graph partitioning using PageRank vectors. *Proceedings of the 47th Annual IEEE Symposium on Foundation of Computer Science* (FOCS 2006), 475-486.
10. F. Chung, P. Horn and A. Tsiatas. Distributing antidote using PageRank vectors. *Internet Mathematics* 6:2 (2009), 237 – 254.
11. Delicious: <http://del.icio.us>
12. A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information Retrieval in Folksonomies: Search and Ranking. In: *Proc. Of ESWC 2006*, pp.411 – 426,2006.
13. A. Muralidharan, Z. Gyongyi, Ed H. Chi. Social Annotaions in Web Search. *CHI’ 12*, May 5-10, 2011, Austin, Texas, USA.
14. D. Zhou, J. Bian, S. Zheng, H. Zha, C. Lee Giles. Exploring Social Annotations for Information Retrieval. Social Networks & Web 2.0 – Applications & Infrastructures for Web 2.0.*WWW 2008*, Bejing, China.
15. P. Mika Ontologies are us: a unified model of social networks and semantics. In: *Proc. Of ISWC 2005*.pp. 522 – 536, Nov, 2005.
16. Zanardi, V., and Capra, L. Social ranking: uncovering relevant content using tag-based recommender systems. In *Proc. ACM RecSys*, ACM Press (2008), 51.
17. Q. Cui, A. Dekhtyar. On Improving Local Website Search Using Web Server Traffic Logs: A Preliminary Report. In *Proc. WIDM’05*, Nov 5, 2005, Bremen, Germany.
18. Nguyen Dinh Thuc, Artificial intelligence, neural network and their applications. Education Publishing House, 2000.
19. Pang-Ning Tan, Vipin Kumar (2000): Modeling of Web Robot Navigational Patterns, WebKDD 2000: Web Mining for E-Commerce, Boston, MA, August 20, 2000.
20. Nikolaos Theodoros Korfiatis. The Opinion Evalution Network: Ranking Imprecise Social Interactions. In *Proc DSV – KTH*, Stockholm, Sweden, Dec 2005.