BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

NGUYỄN THỊ LAN ANH

THUẬT TOÁN HIỆU QUẢ CHO KHAI THÁC TĂNG TRƯỞNG CÁC MÔ HÌNH DUYỆT WEB

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công Nghệ Thông Tin

Mã ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: PGS.TS.Võ Đình Bảy

TP. HỒ CHÍ MINH, tháng 7 năm 2016

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học : **PGS.TS.Võ Đình Bảy**

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM ngày 10 tháng 09 năm 2016

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

TT	Họ và tên	Chức danh Hội đồng
1	TS. Trần Đức Khánh	Chủ tịch
2	PGS. TS. Quản Thành Thơ	Phản biện 1
3	TS. Phạm Thị Thiết	Phản biện 2
4	TS. Lê Văn Quốc Anh	Ủy viên
5	TS. Nguyễn Thị Thúy Loan	Ủy viên, Thư ký

Xác nhận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày 30 tháng 07 năm 2016

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: NGUYỄN THỊ LAN ANH Giới tính: Nữ

Ngày, tháng, năm sinh: 26-04-1981 Nơi sinh: Thanh Hóa

Chuyên ngành: Công nghệ thông tin MSHV: 1441860045

I- Tên đề tài:

Thuật toán hiệu quả cho khai thác tăng trưởng các mô hình duyệt Web

II- Nhiệm vụ và nội dung:

- Nghiên cứu bài toán khai thác chuỗi.
- Nghiên cứu bài toán khai thác mô hình duyệt Web, đặc biệt là mô hình có xem xét đến sự tăng trưởng.
- Cài đặt thử nghiệm.

III- Ngày giao nhiệm vụ: 20-01-2016

IV- Ngày hoàn thành nhiệm vụ: 30-07-2016

V- Cán bộ hướng dẫn: PGS.TS.Võ Đình Bảy

CÁN BỘ HƯỚNG DẪN
(Họ tên và chữ ký)

KHOA QUẢN LÝ CHUYÊN NGÀNH
(Họ tên và chữ ký)

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

Nguyễn Thị Lan Anh

LỜI CÁM ƠN

Trong suốt thời gian học tập tại trường Đại học Công nghệ TP.HCM, em đã nhận được rất nhiều sự động viên, giúp đỡ của quý Thầy cô, gia đình và bạn bè. Nhờ sự giúp đỡ ấy em mới có thể hoàn thành khóa học và luận văn này. Đặc biệt em xin tỏ lòng biết ơn sâu sắc đến Thầy PGS.TS. Võ Đình Bảy đã tận tình hướng dẫn em trong suốt quá trình làm luận văn.

Em xin chân thành cảm ơn Ban Giám Hiệu, phòng Quản lý khoa học và đào tạo sau Đại học trường Đại học Công nghệ TP. HCM đã hướng dẫn em thực hiện tốt các nội quy cũng như các thủ tục của trường trong quá trình học tập.

Em xin chân thành cảm ơn quý Thầy cô khoa Công nghệ thông tin trường Đại học Công nghệ TP.HCM đã tận tình truyền đạt những kiến thức, kinh nghiệm quý báu cho em trong quá trình học tập tại trường. Những kiến thức ấy chính là nền tảng và là hành trang giúp em tìm hiểu về lĩnh vực sáng tạo trong nghiên cứu khoa học.

Em xin chân thành cảm ơn Ban Giám Hiệu, khoa Công nghệ thông tin trường Cao đẳng Kinh tế Kỹ thuật Kiên Giang đã nhiệt tình giúp đỡ và tạo điều kiện thuận lợi nhất để em hoàn thành khóa học.

Do kiến thức còn hạn hẹp nên trong quá trình viết luận văn khó tránh khỏi thiết sót kính mong quý Thầy cô bỏ qua. Đồng thời em rất mong nhận được nhiều ý kiến đóng góp quý báu của quý Thầy cô và các bạn cùng lớp để kiến thức cũng như luận văn của em được hoàn thiện hơn.

Cuối cùng em xin kính chúc quý Thầy cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

Nguyễn Thị Lan Anh

TÓM TẮT

Khai thác Web liên quan đến việc áp dụng các kỹ thuật khai thác dữ liệu với số lượng lớn các dữ liệu liên quan đến Web nhằm cải thiện các dịch vụ Web. Khai thác mô hình duyệt Web liên quan đến việc khám phá mô hình truy cập của người sử dụng từ các bản ghi truy cập máy chủ Web. Thông tin này có thể cung cấp gợi ý định hướng cho người dùng Web đưa ra hành động thích hợp nhất khi có thể. Tuy nhiên các bản ghi Web tăng trưởng liên tục, và một số bản ghi Web có thể trở nên lỗi thời theo thời gian. Hành vi của người sử dụng có thể thay đổi khi các bản ghi Web được cập nhật, hoặc khi các cấu trúc trang Web được thay đổi. Ngoài ra, để xác định một min_sup tối thiểu hoàn hảo trong quá trình khai thác dữ liệu để tìm quy luật là rất khó khăn. Do đó, phải liên tục điều chỉnh các độ hỗ trợ tối thiểu cho đến khi kết quả khai thác dữ liệu được tìm thấy là thỏa đáng.

Bản chất của việc khai thác dữ liệu tăng trưởng là khả năng sử dụng kết quả khai thác trước đó để làm giảm quá trình không cần thiết khi nhật ký truy cập Web được cập nhật, cấu trúc trang Web được thay đổi, hoặc khi điều chỉnh min_sup. Trong luận văn này, trình bày các thuật toán khai thác mô hình duyệt Web khi CSDL được cập nhật hoặc cấu trúc trang Web thay đổi, bên cạnh đó thuật toán khai thác mô hình duyệt Web khi min_sup được điều chỉnh để khám phá các mô hình duyệt Web phù hợp với yêu cầu của người sử dụng. Thuật toán này sử dụng kết quả khai thác trước đó để tìm kiếm các mô hình duyệt Web mới như vậy tổng thời gian khai thác có thể được giảm.

ABSTRACT

Web mining involves the application of data mining techniques to the large number of web-related data to improve web services. Web traversal pattern mining involves discovering patterns of user access logs from Web Server access. This information can provide hints to guide web users make the most appropriate action when possible. However, web logs continue to grow constantly, and some web logs may become outdated over time. User behavior may change when the updated web logs, or when the site structure is changed. In addition, to determine a minimum threshold perfect support in the process of data mining to find the rule is very difficult. Therefore, we must constantly adjust the minimum threshold of support until the results of data mining can satisfactorily be found.

The Substance of the incremental data mining is the capability to use previous data mining results to reduce unnecessary process when web logs or web site structure are updated, or when the minimum support is changed. In this master thesis, I present incremental web traversal pattern mining algorithms for the mainte-nance of web traversal patterns when a database is updated or a web site structure is changed. I also present an interactive web traversal pattern mining algorithm to find all web traversal patterns when min_sup is adjusted. This algorithm utilizes previous mining results to find new web traversal patterns such that the total mining time can be reduced.

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

CSDL	: Cơ sở dữ liệu
TID	: Traversal identifier
FS	: Full scan
SS	: Selective scan
IPA	: Integrating Path traversal patterns and Association rules
MAFTP	: Maintenance of frequent traversal patterns
MFTP	: Mining frequent traversal patterns
IPA	: Integrating Path traversal patterns and Association rules
ISL	: Incremental sequence lattice Algorithm
SPADE	: Sequential pattern discovery using equivalence classes
IncSpan	: Incremental mining in sequential pattern
PrefixSpan	: Prefix-projected sequential pattern mining
KISP	: Knowledge base assisted incremental sequential pattern
KB	: Knowledge base

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH ẢNH

CHƯƠNG 1: MỞ ĐẦU

1.1 Đặt vấn đề

Trong cuộc sống ngày nay, cùng với sự phát triển nhanh chóng của các công nghệ hiện đại, phục vụ cho nhu cầu sống và làm việc của con người trong đó phải nói đến công nghệ thông tin với tốc độ phát triển từng phút. Vì vậy công nghệ thông tin đã và đang chi phối hầu như tất cả các lĩnh vực như: kỹ sư, nhà giáo, nhân viên văn phòng đến những người nông dân, .v.v.... Đối với các doanh nghiệp, cơ hội liên kết và làm ăn trên mạng ở phạm vi trong nước và quốc tế rất lớn. Website doanh nghiệp không những là cách để quảng bá thương hiệu đến bất cứ nơi đâu trên thế giới, mà còn là nơi cung cấp một lượng lớn thông tin về sản phẩm được cập nhật liên tục đến người tiêu dùng và đối tác, với chi phí thấp và phạm vi lãnh thổ không bị hạn chế. Chính vì vậy, số lượng lớn các dữ liệu dễ dàng sản xuất và thu thập từ môi trường Web bởi sự phát triển của Internet. Do đó, làm thế nào để phát hiện ra những thông tin và kiến thức bổ ích hiệu quả từ số lượng lớn những dữ liệu Web đã trở thành một chủ đề quan trọng trong thời gian gần đây của các nhà khoa học.

1.2 Lý do chọn đề tài

Khai thác Web [6, 7, 11, 12] đề cập đến việc khai thác các thông tin và kiến thức bổ ích từ một lượng lớn dữ liệu Web, trong đó có thể được sử dụng để cải thiện các dịch vụ Web. Khai thác mô hình duyệt Web [6, 7, 11, 12] có nghĩa là phát hiện hầu hết các mô hình truy cập của người sử dụng từ các bản ghi Web. Việc phát hiện ra mô hình truy cập của người sử dụng không những được sử dụng để cải thiện thiết kế trang Web (ví dụ; cung cấp hiệu quả việc truy cập giữa các đối tượng tương quan cao, tác giả thiết kế cho các trang Web tốt hơn, .v.v…), mà còn có thể giúp chúng ta định hướng và quyết định tốt hơn trong thị trường thay đổi (ví dụ; đặt quảng cáo ở những nơi lý tưởng, phân loại khách hàng hoặc hành vi phân tích tốt hơn, .v.v…).

Trong thực tế hành vi của người sử dụng thay đổi theo thời gian, chúng ta cần phải khám phá các mô hình duyệt Web ứng với những trình tự người dùng gần nhất.

Vì vậy một số trình tự người dùng cũ cần phải được xóa khỏi cơ sở dữ liệu, và các trình tự người dùng mới cần phải được bổ sung vào cơ sở dữ liệu. Nếu trình tự người dùng cũ không được xóa khỏi cơ sở dữ liệu một cách kịp thời, thì các mô hình duyệt Web phát hiện sẽ không phản ánh các hành vi sử dụng gần đây nhất. Hơn nữa, cấu trúc trang Web có thể được cập nhật, và các mô hình duyệt Web có thể được thay đổi khi trình tự người dùng mới được đưa vào (hoặc các trình tự người dùng sẽ bị xóa) từ cơ sở dữ liệu trình tự duyệt. Vì vậy, chúng ta phải tái khám phá các mô hình duyệt Web từ các phiên bản cập nhật gần đây nhất của dữ liệu trong cơ sở dữ liệu.

Trước những nhu cầu thực tiễn và cấp thiết đó tôi đã chọn đề tài "*Thuật Toán Hiệu Quả Cho Khai Thác Tăng Trưởng Của Mô hình Duyệt Web*" nhằm làm giảm quá trình không cần thiết khi nhật ký Web, cấu trúc trang Web được cập nhật, hoặc khi min_sup được điều chỉnh.

1.3 Mục tiêu, nội dung và phương pháp nghiên cứu

1.3.1 Mục tiêu của đề tài:

- Khai thác các mô hình duyệt Web trong trường hợp các trình tự sử dụng có thể được xóa khỏi cơ sở dữ liệu trình tự duyệt Web.
- Khai thác các mô hình duyệt Web khi độ hỗ trợ được điều chỉnh.
- Khai thác mô hình duyệt Web khi xóa bỏ liên kết trong cấu trúc trang Web.

Tập trung nghiên cứu xây dựng thuật toán cho khai thác tăng trưởng của mô hình duyệt Web khi cơ sở dữ liệu được cập nhật, cấu trúc trang Web dược cập nhật, hoặc khi độ hỗ trợ được điều chỉnh nhằm tìm ra mô hình duyệt Web của người dùng một cách chính xác và kịp thời trong bối cảnh cơ sở dữ liệu Web đang phát triển nhanh chóng.

1.3.2 Nội dung nghiên cứu:

Đề tài này nghiên cứu các nội dung chính như sau:

- Nghiên cứu, tìm hiểu về khai thác Web.

- Nghiên cứu dữ liệu duyệt Web.
- Nghiên cứu các giải pháp khai thác tăng trưởng và các cấu trúc dữ liệu được sử dụng cho khai thác tăng trưởng.
- Nghiên cứu cách sử dụng cấu trúc cây trong khai thác tăng trưởng.
- Dựa trên những nội dung đã tìm hiểu và nghiên cứu để xây dựng các thuật toán.
- Chạy thử trên dữ liệu thật, tối ưu và hoàn thiện thuật toán.

1.3.3 Phương pháp nghiên cứu

- Nghiên cứu và tìm kiếm tài liệu liên quan đến các từ khóa: Web mining, Incremental data mining, Traversal sequence, Web traversal pattern.
- Nghiên cứu tài liệu liên quan đến các thuật toán tăng trưởng, trình tự duyệt và trình tự duyệt Web.
- Cài đặt thuật toán.
- Dùng phương pháp thực nghiệm trên cơ sở dữ liệu trình tự duyệt Web, sau đó nhận xét kết quả và hỏi ý kiến giáo viên hướng dẫn.

1.4 Đối tượng nghiên cứu:

- Các cơ sở dữ liệu duyệt Web: MSNBC, BMSWebView.
- Các thuật toán tăng trưởng trong khai thác mô hình trình tự cụ thể là thuật toán ISL, thuật toán IncSpan.
- Các thuật toán tăng trưởng trong khai thác mô hình trình tự duyệt Web như thuật toán MAFTP và thuật toán IncWTP [14].

1.5 Phạm vi nghiên cứu:

- Nghiên cứu thuật toán sinh ứng viên trình tự duyệt Web (CandidateGen) [14] và thuật toán IncWTP [14].

1.6 Cấu trúc của luận văn

Luận văn bao gồm những nội dung sau:

- **Chương 1 - Mở đầu:** Nội dung của chương này trình bày lý do chọn đề tài, mục tiêu của đề tài, đối tượng và phạm vi nghiên cứu, nội dung nghiên cứu và phương pháp nghiên cứu của đề tài.
- **Chương 2 - Tổng quan về khai thác Web:** Nội dung của chương này giới thiệu về các đặc điểm của khai thác Web, các lĩnh vực khai thác Web, các kỹ thuật và các bài toán được đặt ra trong khai thác Web. Những khó khăn và thuận lợi của kỹ thuật khai thác Web, lịch sử giải quyết vấn đề và những tồn đọng cần nghiên cứu.
- **Chương 3 – Nghiên cứu và xây dựng thuật toán:** Nội dung chương này trình bày về các vấn đề liên quan đến khai thác mô hình duyệt Web và xây dựng các thuật toán.
- **Chương 4 – Thực nghiệm và đánh giá kết quả:** Nội dung chương này trình bày kết quả chạy thực nghiệm các thuật toán trên các bộ dữ liệu chuẩn: MSNBC, BMSWebView1.
- **Chương 5 – Kết luận và hướng phát triển:** Nội dung chương này trình bày kết quả đạt được của luận văn, những ưu và nhược điểm, hướng phát triển của đề tài.

CHƯƠNG 2: TỔNG QUAN VỀ KHAI THÁC WEB

2.1 Khai thác Web (Web mining)

Sự phát triển mạnh mẽ của mạng Internet và Intranet hiện nay là điều không ai có thể phủ nhận được bởi những lợi ích nó đem lại cho chúng ta là rất lớn. Ngày nay Intrnet đã trở thành một trong những kênh về khoa học, thông tin kinh tế, thương mại và quảng cáo. Một trong những lý do cho sự phát triển này là để duy trì một trang Web trên Internet chỉ cần một khoản chi phí thấp. Nếu So sánh với những dịch vụ khác như đăng tin hay quảng cáo trên một tờ báo hoặc tạp chí, thì một trang Web rẻ hơn rất nhiều và việc cập nhật nhanh chóng hơn tới hàng triệu người dùng khắp mọi nơi trên thế giới. Do vậy, Internet ngày càng chứng tỏ tầm quan trọng của nó trong đời sống của con người trên tất cả các lĩnh vực. Từ những điều ấy một lượng thông tin đa dạng khổng lồ đã được sinh ra, và World Wide Web đã và đang trở thành một lĩnh vực phong phú cho các nghiên cứu về khai thác dữ liệu. Những nghiên cứu về khai thác Web đang phát triển mạnh và bao gồm nhiều lĩnh vực nghiên cứu khác nhau như thu hồi thông tin (Information retrieval), trí tuệ nhân tạo (AI) và các lĩnh vực khác.

Tuy nhiên, chưa có một định nghĩa rõ ràng về khai thác Web, nhưng chúng ta có thể hiểu bản chất của *khai thác Web là kỹ thuật khai thác và phân tích những thông tin có ích từ World Wide Web*. Để hình dung rõ ràng hơn chúng ta có thể xem khai thác Web là sự kết hợp giữa khai thác dữ liệu và World Wide Web, cụ thể hơn là:

Khai thác Web = Khai thác dữ liệu + World Wide Web

Khai thác Web được sử dụng để tìm kiếm, chọn lọc ra các mô hình hoặc các tri thức hữu ích tiềm ẩn của CSDL Web khổng lồ nhằm phục vụ cho nhu cầu của người sử dụng, các nhà quản lý và các nhà nghiên cứu hiện nay.

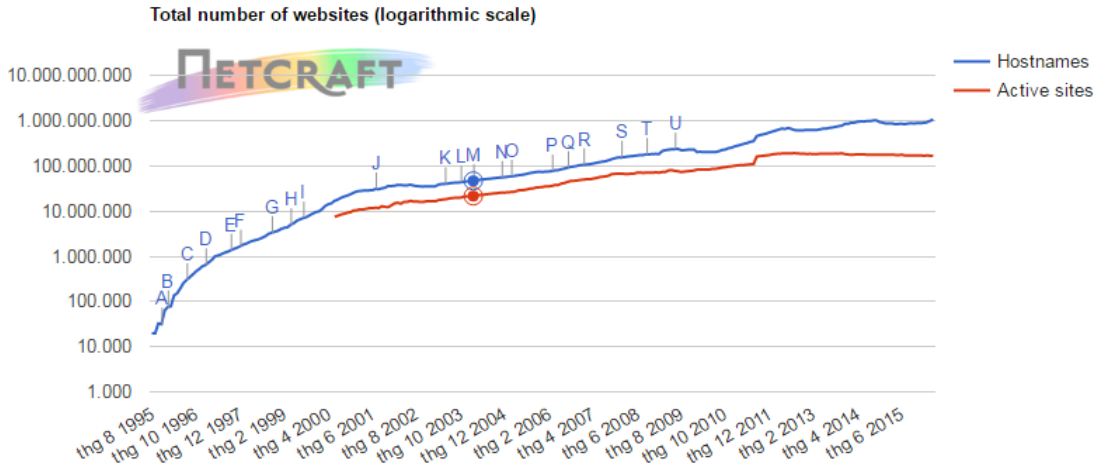
2.2 Đặc điểm của khai thác Web

Sau đây là những thách thức cũng như thuận lợi trong lĩnh vực khai thác Web.

2.2.1 Khó khăn

– **Dữ liệu Web quá lớn để tổ chức thành kho dữ liệu**

Những CSDL truyền thống thường có kính thước không quá lớn, nên được lưu trữ tập trung. Trong khi đó, kính thước Web lại rất lớn, lên đến hàng terabytes không những thay đổi liên tục, mà còn phân tán trên rất nhiều máy tính khắp nơi trên thế giới. Hình 2.1 cho số liệu thống kê tại thời điểm tháng 04/2016 cho thấy có hơn 1 tỷ Website trên Internet[1]. Trong 4 tháng đầu năm 2016 trung bình có trên 44 triệu Website mới xuất hiện. Theo kết quả thống kê vào tháng 01/2016, có hơn 60 nghìn tỷ trang Web được đánh chỉ số trên Google[2]. Kích thước trung bình của mỗi trang là 5-10KB thì tổng kích thước của các trang Web được đánh chỉ số trên Goolge là rất lớn. Bên cạnh đó, số lượng trang Web tăng rất nhanh. Như vậy việc xây dựng một kho dữ liệu để lưu trữ, sao chép hay tích hợp các dữ liệu trên Web gần như không thể.



Hình 2.1 *Thống kê số lượng Website (tháng 04/2016)* [1].

[1] http://news.netcraft.com/archives/2016/04/21/april-2016-web-server-survey.html
[2] http://expandedramblings.com/index.php/by-the-numbers-a-gigantic-list-of-google-stats-and-facts/.

– **Độ phức tạp của trang Web là rất lớn**

Trong các CSDL truyền thống dữ liệu thường đồng nhất (ngôn ngữ, định dạng, .v.v…) nhưng dữ liệu Web thì không đồng nhất. Đối với dữ liệu Web có rất nhiều loại ngôn ngữ khác nhau (về ngôn ngữ diễn tả nội dung cũng như ngôn ngữ lập trình), có rất nhiều định dạng (hình ảnh, âm thanh, PDF, HTML, .v.v…), nhiều loại từ vựng (các liên kết, số điện thoại, mã nén, địa chỉ Email, .v.v…). Tóm lại, cấu trúc các trang Web không đồng nhất. Chúng được xem như là "mộ thư viện kỹ thuật số khổng lồ". Tuy nhiên, với khối lượng tài liệu khổng lồ như thế mà không được sắp xếp theo một tiêu chuẩn nào, không theo một quy luật nào,.v.v….Đây chính là một thách thức vô cùng to lớn cho việc tìm kiếm các thông tin cần thiết trong thư viện này.

– **Web là nguồn tài nguyên mà thông tin có độ thay đổi cao**

Web không những thay đổi thông tin trong các trang Web mà số lượng của các trang Web cũng có thay đổi liên tục. Theo kết quả thống kê[3] chỉ trong tháng 04/2016 Google đã nhận được trên 87 triệu yêu cầu gỡ bỏ URL từ các chủ sở hữu bản quền và các tổ chức. Các công ty quản cáo, trung tâm phụ vụ Web luôn cập nhật trang Web của họ. Hơn nữa sự kết nối thông tin và truy cập bản ghi Web cũng được cập nhật liên tục.

– **Web phục vụ cộng đồng người dùng rộng lớn và đa dạng**

Theo thống kê[4] thì tháng 11/2015 toàn cầu có trên 3.2 tỷ người sử dụng Internet và con số này vẫn tiếp tục tăng. Mỗi người dùng có một kiến thức, nhu cầu khác nhau. Nhưng phần lớn người dùng không am hiểu nhiều về cấu trúc mạng thông tin, hoặc không biết cách tìm kiếm nên họ thường hay "lạc" trong thư viện số khổng lồ. Từ đó, dẫn đến sự nhàm chán vì mất thời gian và công sức để tìm kiếm nhưng nhận được những thông tin không hữu ích cao, thậm chí còn nhận cả những thông tin vô ích.

---

[3] https://www.google.com/transparencyreport /removals/copyright/.
[4] http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx.

– **Chỉ một phần rất nhỏ của thông tin trên Web là thực sự có ích**

Dường như ai cũng có thói quen bắt đầu việc tìm kiếm thông tin trên mạng bằng Google. Tỉ lệ người dùng mạng sử dụng Google hằng tháng trên tổng số các bộ máy tìm kiếm thống kê được là trên dưới 80 %[5].

Tuy nhiên, nếu làm một phép thử bằng một cách rất thông thường. Cần tìm tài liệu về toán học lớp 5. Xuất phát với Google một cách thông thường, trang mặc định sẽ bằng tiếng Việt. Gõ từ khoá "toán lớp 5" và bắt đầu tìm kiếm. Tóm tắt kết quả: khoảng 2.490.000 kết quả trong 0,38 giây. Lướt qua 100 kết quả đầu tiên, toàn các trang sao đi chép lại từ một vài trang uy tín hay từ vài bài báo. Hãy tưởng tượng nếu tiếp tục dò để lọc được vài bài trong mỗi 100 kết quả, trong hơn 2,9 triệu kết quả mà Google cung cấp, thì có thể đánh giá được hiệu suất cơ bản của cách tìm kiếm này.

Những đặc điểm trên cho thấy sự khác biệt lớn giữa việc tìm kiếm trong một CSDL truyền thống với CSDL Web. Chính những thách thức ấy đã thúc đẩy hoạt động nghiên cứu khai thác dữ liệu Web.

2.2.2 Thuận lợi

Bên cạnh những khó khăn cũng có một số thuận lợi cho khai thác Web do lượng thông tin trên Web rất phong phú:

+ Trang Web được cấu trúc theo quy định của ngôn ngữ định dạng. Do đó, ngoài phần nội dung như; văn bản, hình ảnh và các dữ liệu đa phương tiện khác, thì trang Web còn chứa các thẻ thi hành cấu trúc các nội dung của trang Web đó. Do đó, trang Web được coi như một loại dữ liệu bán cấu trúc. Điều đó đã tạo cho khai thác Web có một số thuận lợi nhất định.

+ Một Website không những bao gồm các trang, mà còn có các liên kết trỏ từ trang này tới trang khác. Khi tác giả tạo một liên kết từ trang của

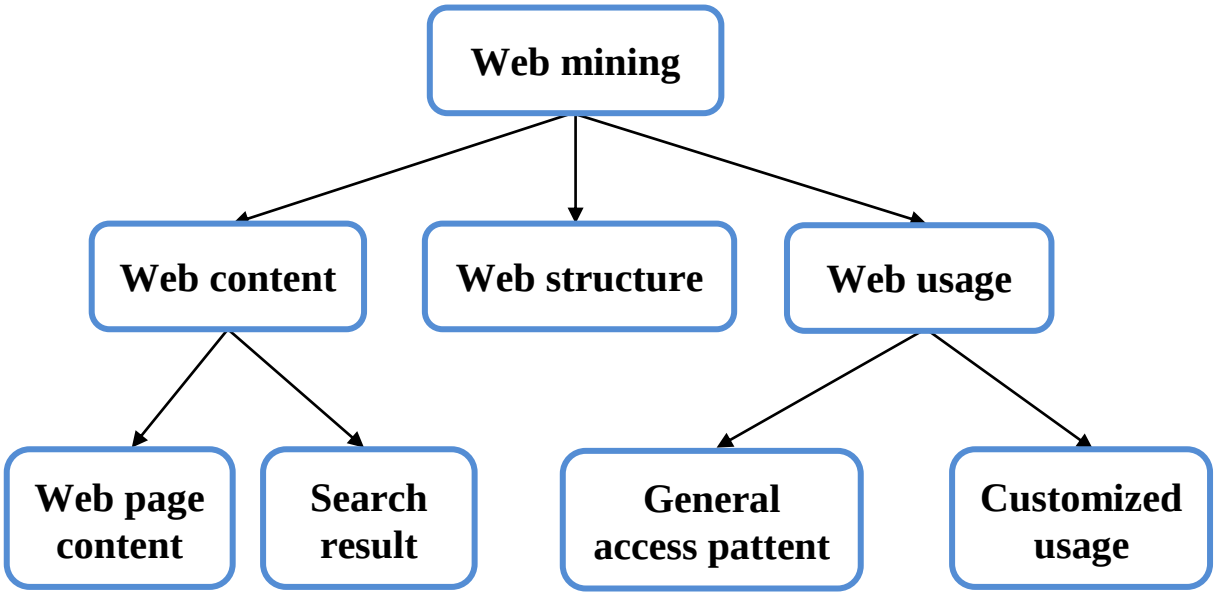
[5] http://bigseo.vn/co-phai-thi-phan-tim-kiem-cua-google-thuc-su-dang-giam-sut/

ông ta đến trang Y nào đó, nghĩa là trang Y có hữu ích với vấn đề đang đề cập. Một trang Web có nhiều liên kết trỏ đến, cho thấy trang đó quan trọng. Do đó, các thông tin liên kết trang cho một lượng lớn thông tin về mối liên quan, chất lượng và cấu trúc của nội dung trang Web. Đây cũng là nguồn tài nguyên giàu có cho khai thác Web.

+ Một máy chủ Web thường đăng ký một bản ghi đầu vào (Weblog entry) cho mỗi lần truy cập trang Web, bao gồm; địa chỉ IP, URL, timestamp. Do đó, việc phân tích dữ liệu Weblog có thể thu được nhiều thông tin hữu ích như; xu hướng truy cập Web, cấu trúc Web.

2.3 Các lĩnh vực trong khai thác Web (Web mining)

Khai thác Web cho phép chúng ta tìm kiếm các mô hình dữ liệu thông qua khai thác nội dung (Web content mining), khai thác cấu trúc (web structure mining), và khai thác sử dụng (Web usage mining). Hình 2.2 thể hiện sự phân loại các lĩnh vực nghiên cứu trong khai thác Web.



Hình 2.2 *Các lĩnh vực trong khai thác Web [2]*

2.3.1 Khai thác nội dung trang Web

Khai thác nội dung trang Web là quá trình khai thác chỉ lấy các tri thức từ nội dung của trang Web. Khai thác nội dung trang Web được chia thành hai lĩnh vực; khai thác trực tiếp nội dung của trang Web, và nâng cao khả năng tìm kiếm nội dung của các công cụ khác như máy tìm kiếm.

- *Khai thác nội dung trang Web (Web Page summarization)*: liên quan tới việc truy xuất các thông tin từ các văn bản có cấu trúc, văn bản bán cấu trúc hay các văn bản siêu liên kết. Lĩnh vực này liên quan tới việc khai thác bản thân nội dung các văn bản.
- *Tối ưu kết quả trả về (search engine result summarization)*: Tìm kiếm trong kết quả trả về. Trong các máy tìm kiếm, sau khi tìm ra các trang Web thỏa mãn yêu cầu của người dùng, tiếp theo là một công việc không kém phần quan trọng là phải sắp xếp, chọn lọc kết quả theo mức độ phù hợp với yêu cầu người dùng. Quá trình này thường sử dụng các thông tin như tiêu đề trang, content-type, URL, các liên kết trong trang Web, .v.v... để tiến hành phân lớp và đưa ra tập con các kết quả tối ưu nhất cho người dùng.

2.3.2 Khai thác cấu trúc trang Web

Nhờ vào các kết nối giữa các văn bản siêu liên kết, ngoài các thông tin ở bên trong văn bản World Wide Web còn chứa đựng nhiều thông tin khác. Ví dụ, các liên kết trỏ tới một trang Web chỉ ra mức độ quan trọng của trang Web đó, trong khi các liên kết đi ra từ một trang Web thể hiện các trang có liên quan tới chủ đề đề cập trong trang hiện tại. Thực chất việc khai thác cấu trúc Web là quá trình xử lý để lấy ra các tri thức từ cách tổ chức và liên kết giữa các tham chiếu của các trang Web.

2.3.3 Khai thác sử dụng Web

Khai thác sử dụng Web hay khai thác hồ sơ Web (Weblogs mining) là quá trình xử lý nhằm lấy ra các thông tin hữu ích trong các hồ sơ truy cập Web. Thông

thường các Web Server sẽ ghi lại và tích lũy dữ liệu về các tương tác của người dùng mỗi khi nó nhận được một yêu cầu truy cập. Việc phân tích các hồ sơ truy cập Web của các Website khác nhau sẽ dự đoán các tương tác của người dùng khi họ tương tác với Web cũng như tìm hiểu cấu trúc của Web. Từ đó, cải thiện các thiết kế của các hệ thống liên quan. Có hai chiến lược chính trong khai thác sử dụng Web là; phân tích các mô hình truy cập và phân tích các xu hướng cá nhân.

- *Phân tích các mô hình truy cập (General Access Pattern tracking)*: phân tích các hồ sơ Web để biết được các mô hình và các xu hướng truy cập. Các phân tích này có thể giúp cấu trúc lại các site trong các phân nhóm hiệu quả hơn, hoặc xác định các vị trí quảng cáo sao cho hiệu quả nhất, cũng như gắn các quảng cáo sản phẩm nhất định cho những người dùng nhất định để đạt được hiệu quả tốt nhất.
- *Phân tích các xu hướng cá nhân (Cusomized Usage tracking)*: Mục tiêu nhằm chuyên biệt hóa các Website cho các lớp đối tượng người dùng. Các thông tin được hiển thị, độ sâu của cấu trúc site và định dạng của các tài nguyên, tất cả đều có thể chuyên biệt hóa một cách tự động cho mỗi người dùng theo thời gian dựa trên các mô hình truy cập của họ.

2.4 Các bài toán được đặt ra trong khai thác Web

Các bài toán của khai thác Web đều cần bao hàm tính đặc thù của Web. Có thể chia khai thác Web thành hai loại bài toán sau:

- *Các bài toán chung của khai thác dữ liệu văn bản với việc bổ sung các yếu tố của miền ứng dụng dữ liệu Web*:
    + Các bài toán phân lớp, phân cụm và phân đoạn trong khai thác dữ liệu Web tương tự như các bài toán tương ứng trong khai phá dữ liệu văn bản, nhưng có bổ sung đặc thù của Web như nội dung trang Web có các siêu liên kết hướng tới các trang Web khác. Nhiều trường hợp dạng bài toán này được làm phù hợp với môi trường của Internet như; bài toán

phân cụm, phân lớp đối với tập các trang Web là kết quả trả về từ một máy tìm kiếm.

+ Các bài toán phát hiện ràng buộc (Constraint) và phát hiện luật kết hợp (Association Rule) không những liên quan đến các yếu tố trong nội dung văn bản mà còn liên quan đến những yếu tố đặc thù của trang Web như; sự ràng buộc của các trang Web, sự ràng buộc giữa người sử dụng với các trang Web mà họ quan tâm trong phiên làm việc, hay sự ràng buộc của nhóm người sử dụng với tập các trang Web mà mọi thành viên trong nhóm cùng quan tâm.

– *Các bài toán khai thác dữ liệu mang tính đặc thù của Web*

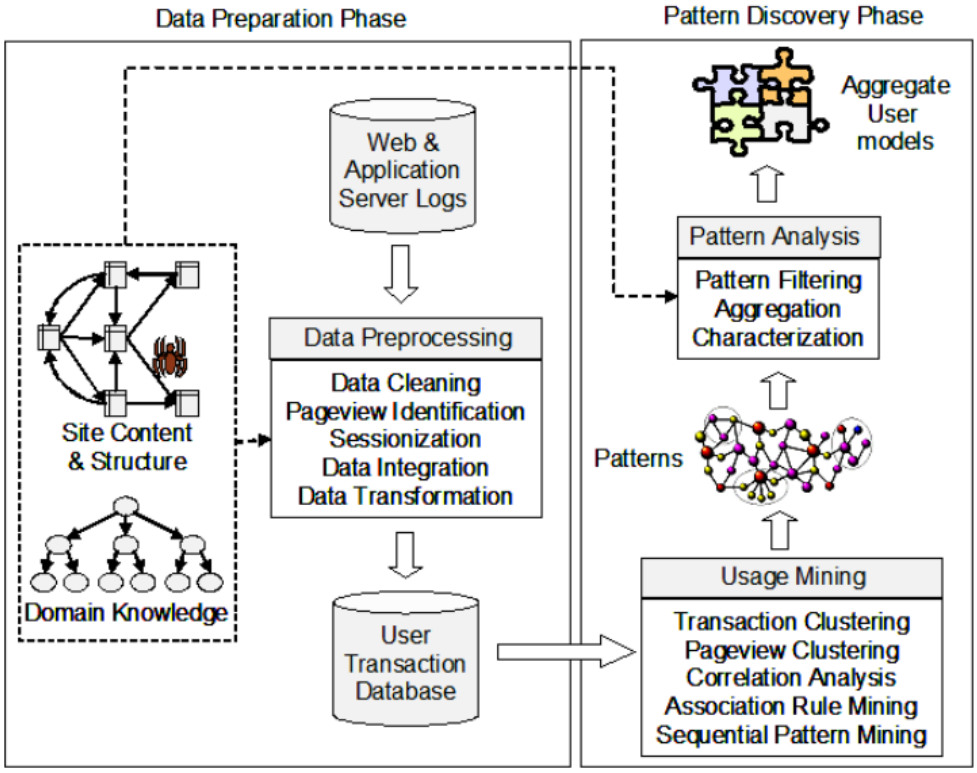
+ Bài toán dự báo (Prediction) khai thác yếu tố thời gian liên quan tới thời điểm xuất hiện trang Web để có thể dự báo xu thế về các đặc trưng như; nội dung, cấu trúc và hình thức trình bày của các trang Web xuất hiện trong thời gian tới. Khai thác trình tự sử dụng Web trong một phiên làm việc cũng là một bài toán được nhiều nhà khoa học quan tâm.
+ Các bài toán Dự đoán nhu cầu (Response prediction) và đánh giá khách hàng khai thác Web (Customer evaluation) liên quan đến đối tượng sử dụng CSDL Web.
+ Mộ số bài toán điển hình nhất của khai thác dữ liệu Web như là; tìm kiếm, phân cụm, phân lớp, trích chọn thông tin.

2.5 Khai thác sử dụng Web

Khai thác sử dụng Web là một trong ba xu thế chính của khai thác Web. Quá trình khai thác sử dụng Web được mô tả trong hình 2.3[1]. Theo Liu và đồng sự [1] thì khai thác sử dụng Web có thể chia thành ba giai đoạn liên quan và phụ thuộc vào nhau; thu thập dữ liệu và tiền xử lý dữ liệu, phát hiện mô hình và phân tích mô hình.

Trong giai đoạn thu thập và tiền xử lý dữ liệu: Các nguồn dữ liệu chính được sử dụng trong khai thác sử dụng Web là máy chủ Log Files, trong đó bao gồm các bản

ghi truy cập máy chủ Web và các bản ghi máy chủ ứng dụng, hoặc lấy từ các CSDL khách hàng. Sau đó, dữ liệu Web được làm sạch và phân chia thành một tập hợp các giao dịch sử dụng đại diện cho các hoạt động của mỗi người sử dụng trong các chuyến thăm khác nhau tới trang Web. Các nguồn khác của kiến thức như; các nội dung hoặc cấu trúc, cũng như kiến thức ngữ nghĩa từ miền ontology của trang Web (chẳng hạn như danh mục sản phẩm hoặc hệ thống phân cấp khái niệm), cũng có thể được sử dụng trong tiền xử lý hoặc để tăng cường giao dịch dữ liệu người dùng.



Hình 2.3 *Quá trình khai thác sử dụng Web* [1]

Trong giai đoạn phát hiện mô hình: Gồm các hoạt động thống kê, cơ sở dữ liệu, và máy học được thực hiện để có được mô hình ẩn phản ánh hành vi điển hình của người sử dụng, cũng như số liệu thống kê tóm tắt về tài nguyên Web, phiên, và người sử dụng.

Trong giai đoạn phân tích mô hình: Đây là giai đoạn cuối cùng của quá trình khai thác sử dụng Web, các mô hình đã phát hiện và thống kê được tiếp tục xử lý, lọc, kết quả có thể là sự tổng hợp các mô hình và được sử dụng làm đầu vào cho

các ứng dụng như công cụ giới thiệu, công cụ trực quan, phân tích Web và các công cụ tạo báo cáo. Toàn bộ quá trình được mô tả trong hình 2.3.

2.5.1 Phân tích mô hình truy cập Web

Trong khai thác sử dụng Web thì bài toán phân tích mô hình truy cập Web quan tâm đến việc khám phá các mô hình truy cập có tính phổ dụng của tập người sử dụng khi truy cập Web, tập người sử dụng này được xem là đối tượng nghiên cứu của bài toán phân tích mô hình truy cập Web.

Thông tin truy cập của người dùng được Web Server ghi nhận trong Web Server Log theo mẫu Log chung (common Log Format: CLF), hoặc mẫu Log chung mở rộng (Extended CLF: ECLF). Thông tin được lưu giữ liên quan đến phiên truy cập của người dùng bao gồm các thông tin như địa chỉ IP của máy người dùng, thời điểm bắt đầu truy cập, nhu cầu người dùng (phương thức, địa chỉ Web, giao thức), mã trạng thái đáp ứng yêu cầu (bình thường, truy cập không hoàn chỉnh, không tìm thấy,.v.v...), kích thước dữ liệu truy cập Web của người dùng.

Từ dữ liệu truy cập Web được lưu trữ tại Web Server Log, các nhà khai thác dữ liệu có thể sử dụng CSDL này để thực hiện các bài toán khai thác dữ liệu Web như; tìm mối quan hệ về nội dung giữa các trang Web (dựa vào mối liên hệ giữa các địa chỉ URL đến và trang Web), thói quen truy cập, hay xu hướng truy cập của người dùng.

Khai thác sử dụng Web hiện nay không chỉ quan tâm đến việc phát hiện các tri thức tiềm ẩn từ nội dung trang Web được lưu giữ tạm thời tại Web server, mà còn phát hiện được hành vi sử dụng Web của người dùng.

Ngoài ra thuật toán khai thác sử dụng Web cần quan tâm đến các vấn đề như; làm sao để lưu trữ các trang Web một cách có lợi nhất trong bối cảnh dung lượng bộ nhớ hạn chế bởi vùng Web-cache. Một số giải pháp điều phối cache được thi hành nhằm loại bỏ các trang Web không hữu ích ở vùng Web-cache, chúng hoạt động giống như việc loại bỏ các trang bộ nhớ của hệ điều hành.

Điển hình nhất trong phân tích mô hình truy cập Web là luật kết hợp. Thuật toán Apriori và các biến thể của thuật toán này đã được sử dụng rất phổ biến trong việc giải quyết bài toán phát hiện luật kết hợp từ Log Files.

– *Luật kết hợp:*

Cho một CSDL giao dịch D = $\{t_1,t_2,...,t_n\}$, $1 \leq i \leq n$. Trong đó mọi t_i là một giao dịch và là một tập con thuộc D.

Cho X,Y là hai tập mục (hai tập con của D). Luật kết hợp được ký hiệu là $X \rightarrow Y$, trong đó $X \cap Y = \emptyset$, thể hiện mối ràng buộc của tập mục Y theo tập mục X theo nghĩa "X kéo theo Y" ra sao về sự xuất hiện trong giao dịch. Tập mục X được gọi là xuất hiện trong giao dịch t nếu như $X \subseteq t$, và có thể được diễn giải là "mọi tên mặt hàng trong X đều xuất hiện trong phiếu giao dịch t".

Giá trị của luật kết hợp $X \rightarrow Y$ được thể hiện thông qua hai độ đo là; độ hỗ trợ $supp(X \rightarrow Y)$ và độ tin cậy $conf(X \rightarrow Y)$.

+ Độ hỗ trợ của một tập mục X (ký hiệu supp(X) được định nghĩa là:

$$Supp(X) = |\{t \in D: X \subseteq t\}|/|D|$$

+ $Supp(X \rightarrow Y) = Supp(XY)$ là tỷ lệ giao dịch có chứa $(X \cup Y)$ trong tập D.
+ $Conf(X \rightarrow Y) = supp(X \rightarrow Y)/supp(X)$ là tỷ lệ tập giao dịch có chứa $(X \cup Y)$ so với tập giao dịch có chứa X.

Theo định nghĩa ta có: $0 \leq supp(X \rightarrow Y) \leq 1$ và $0 \leq conf(X \rightarrow Y) \leq 1$. Theo quan điểm xác suất, độ hỗ trợ là xác suất xuất hiện tập mục $X \cup Y$, còn độ tin cậy là xác suất có điều kiện xuất hiện Y khi đã xuất hiện X.

Luật kết hợp $X \rightarrow Y$ được coi là một "tri thức" (hoặc mẫu có giá trị) nếu xảy ra đồng thời.

$Supp(X \rightarrow Y) \geq$ minsup và $conf(X \rightarrow Y) \geq$ minconf

với minsup và minconf là hai ngưỡng cho trước. Tập mục X có độ hỗ trợ qua ngưỡng minsup (supp(X) $\geq$ minsup) được gọi là tập phổ biến.

Mục tiêu của khai thác luật kết hợp là tìm ra tất cả các luật kết hợp có giá trị. Để giải quyết bài toán trên, trước hết cần phải tìm ra tất cả các tập phổ biến, mỗi tập phổ biến đóng vai trò của tập XY trong luật kết hợp X$\rightarrow$Y.

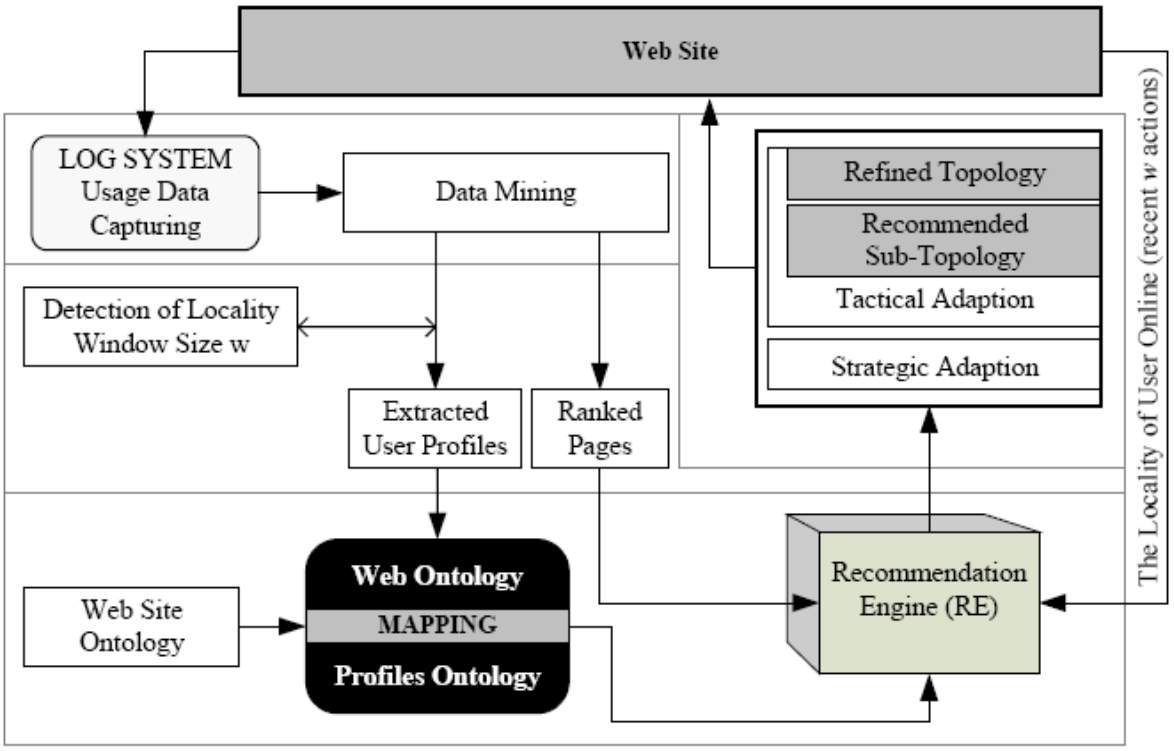
2.5.2 Phân tích xu hướng cá nhân

Phân tích xu hướng cá nhân có tính cá nhân hóa, vì thế dữ liệu cũng cần mang tính cá nhân, hoặc ở các Log Files của máy khách, hay ở CSDL khách hàng, hoặc dữ liệu thu thập online với khách hàng. Sau đây là một số nội dung phân tích xu hướng cá nhân không có CSDL khách hàng và các hệ thống tư vấn khách hàng.

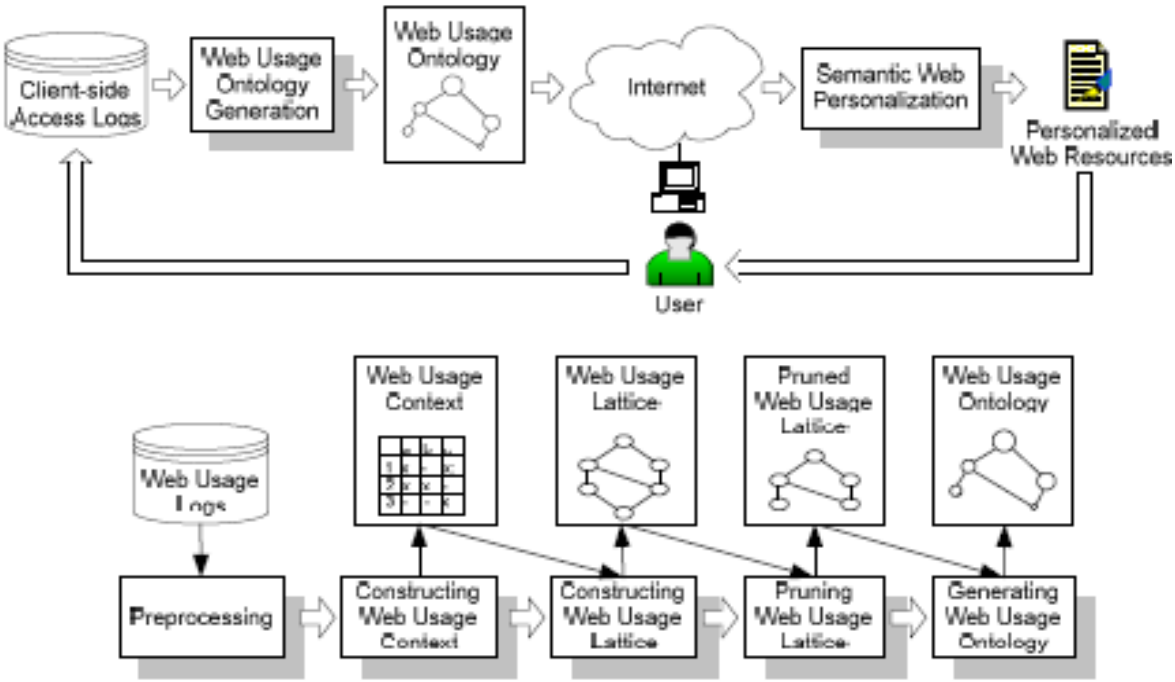
– *Phân tích xu hướng cá nhân từ máy khách*

Hình 2.4 [1] trình bày hệ thống khai phá sử dụng Web có sử dụng dữ liệu người dùng từ máy khách. Thông tin người dùng được các phần mềm hệ thống tại máy khách trích chọn và dữ liệu sử dụng hệ thống tư vấn cho từng người dùng cụ thể.

Hình 2.5 [2] trình bày hệ thống dựa theo Log Files xây dựng các ontology sử dụng Web để tư vấn người sử dụng hệ thống.



Hình 2.4 *Sinh tư vấn dựa trên trích chọn tiểu sử người dùng* [2]



Hình 2.5 *Hệ thống tư vấn hướng cá nhân: kiến trúc hệ thống (trên) và sinh ontology sử dụng Web (dưới)* [2]

Nhiều thông tin hành vi người dùng được các hệ thống khai thác nhằm khai thác trình tự hành vi của người dùng, để dự đoán hành vi tiếp theo của người dùng và chuẩn bị sẵn tài nguyên phù hợp với thao tác tiếp theo của họ.

– *Các hệ thống tư vấn khách hàng*

Một ứng dụng điển hình của khai phá dữ liệu Web là hệ thống tư vấn khách hàng. Trong hệ thống này, CSDL khách hàng lưu trữ về thông tin khách hàng đăng ký. Những thông tin này cho phép thực hiện các công việc sau:

+ Kết nối các phiên làm việc của cùng một khách hàng, điều này tạo điều kiện thuận lợi cho việc khảo sát mối quan hệ khách hàng – mặt hàng.
+ Kết nối các khách hàng có cùng một (hay nhiều) thuộc tính như: giới tính, độ tuổi, nghề nghiệp, thu nhập,.v.v…. Trong một số hệ thống các thuộc tính mô tả sở thích của khách hàng cũng được lưu trữ trong CSDL.

Theo [2] dữ liệu khách hàng có tại các máy phục vụ, máy khách hoặc tại các vị trí trung gian như máy phục vụ proxy. Ứng dụng điển hình là hệ thống tư vấn khách hàng tự động. Lọc cộng tác là cách tiếp cận chính trong hệ thống, trong đó hệ thống sử dụng các lựa chọn của các cá nhân trong quá khứ để dự báo sự chọn lựa mới và đưa ra tư vấn mới. Hai xu thế chính cho mô hình lọc này là mô hình lọc cộng tác người láng giềng gần nhất và mô hình lọc cộng tác dựa trên mô hình.

Mô hình lọc cộng tác người láng giềng gần nhất: Tư tưởng của mô hình này là đối với một người dùng A, đầu tiên tìm ra tập các người dùng gần giống A trong dữ liệu lọc. Sau đó, sử dụng lựa chọn thuộc tính của các người dùng gần giống A để dự báo chọn lựa của người dùng A đối với thuộc tính đó. Trong mô hình này các bài toán cần giải quyết là; xác định trọng số trong phương trình dự báo, thu gọn số chiều bài toán, tính toán và phân cụm.

Mô hình lọc cộng tác dựa trên mô hình: Tư tưởng của mô hình này là thừa kế các mô hình lựa chọn của người dùng trong quá khứ, xây dựng không trực tuyến

một mô hình kỳ vọng về mối liên quan giữa các mặt hàng. Sau đó, mô hình được sử dụng trực tuyến để dự báo sự lựa chọn của người dùng mới. Mô hình này chủ yếu hướng đến yêu cầu tính toán thời gian thực hiện mà không phụ thuộc vào kích thước của CSDL.

2.6 Khai thác cấu trúc Web

2.6.1 Khai thác đồ thị Web

Khai thác đồ thị Web là bài toán điển hình và kinh điển nhất của khai thác cấu trúng Web. Đồ thị Web xem như một mạng xã hội, hiện nay đối tượng nghiên cứu này rất được quan tâm. Trong một đồ thị Web thì trang Web được xem là đỉnh và một trang Web có cung tới một trang Web khác khi trong nội dung của nó có liên kết đến trang Web đó. Tùy vào bài toán mà đồ thị Web được xem xét dưới dạng có hướng hoặc không có hướng.

Điển hình nhất trong đồ thị Web là bài toán tính hạng trang Web. Hạng trang Web được sử dụng trong nhiều trường hợp như; dẫn dắt đường đi trên trang Web, những trang Web có hạng cao sẽ được dẫn đi xem trước. Trong máy tìm kiếm thì hạng trang Web được sử dụng để hiển thị kết quả tìm kiếm, nếu trang Web nào có hạng cao hơn sẽ được hiển thị trước. Ngoài ra tính hạng trang Web còn được ứng dụng trong bài toán phát hiện địa chỉ Email spam trong mạng Email, các địa chỉ Email có hạng thấp thì khả năng đó là một địa chỉ spam rất cao.

2.6.2 Khai thác cấu trúc trang Web

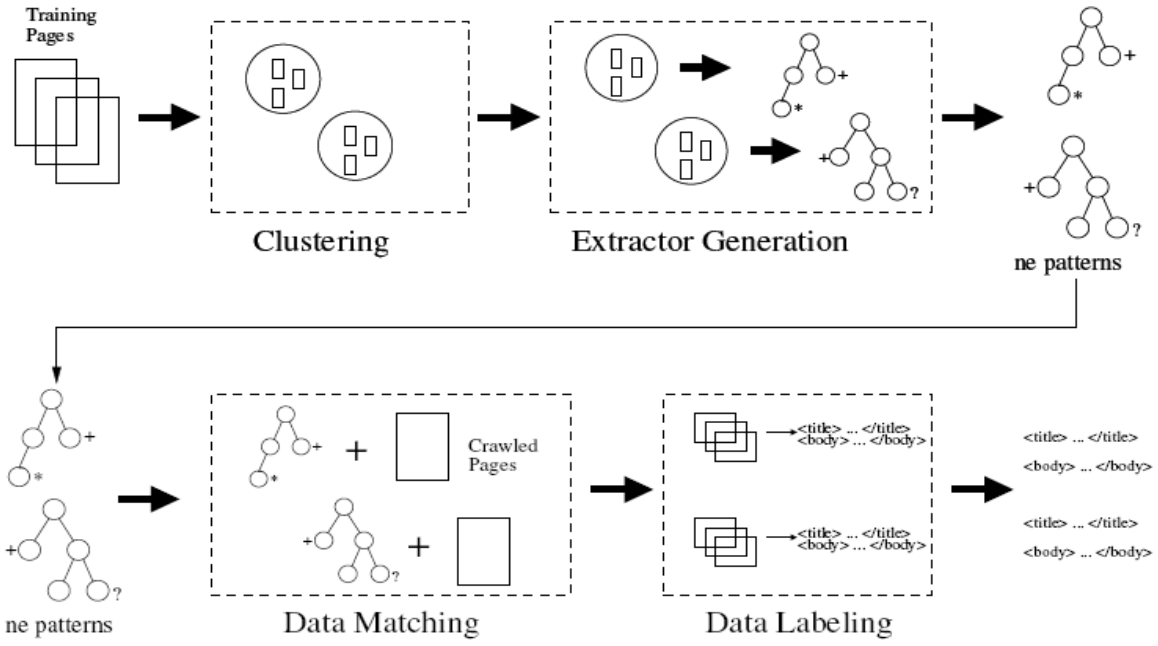
Khai thác cấu trúc trang Web là việc phát hiện các mô hình từ tập các kiến trúc trang Web. Trong trường hợp này dữ liệu là khung trang Web gồm các đối tượng; thẻ và cấu trúc giữa các thẻ. Kết quả của khai thác cấu trúc trang Web được dùng để hỗ trợ cho các bài toán khai thác dữ liệu Web khác.

Theo [2] thì bài toán trích chọn tự động tin tức trên Web theo cách tiếp cận khoảng cách cây. Cấu trúc dạng cây được chọn để biểu diễn trang Web và sử dụng độ đo chi phí chuyển đổi cây (Tree Edit Distance) để đánh giá độ tương tự về cấu trúc của các trang Web. Dựa trên thuật toán RTDM (Restricted Top – Down

Mapping) mô hình trích chọn tin tức từ cổng thông tin (portal) báo điện tử gồm các bước:

+ Dùng kỹ thuật phân cụm cấu trúc để phân cụm các trang báo điện tử (độ đo khoảng cách giữa hai trang Web là chí phí chuyển đổi cây cấu trúc trang Web).
+ Sinh các mô hình trích chọn dưới dạng cây.
+ Kiểm tra đánh giá các mô hình trích chọn dựa theo thuật toán RTMD, định giá cho các thao tác thay đỉnh (Vertex Replacement), chèn đỉnh (Vertex Insertion), và xóa bỏ đỉnh (Vertex Removal).
+ Áp dụng mô hình để trích chọn tin tức.

Hình 2.6 trình bày quá trình trích chọn tự động qua bốn bước mô tả trên [2]. Mô hình trên đã được ứng dụng thành công trong sản phẩm Viennews "*Các kênh báo điện tử trên thiết bị điện thoại di động thông minh*" đã đoạt giải ba cuộc thi Trí tuệ Việt Nam năm 2006 [2].



Hình 2.6 *Quá trình trích chọn thông tin tự động* trên Web [2]

2.7 Tổng quan về khai thác tăng trưởng mô hình duyệt Web

Kỹ thuật khai thác mô hình con đường duyệt [4, 12] khám phá các hành vi chuyển hướng cho hầu hết những người sử dụng trong môi trường Web. Các nhà thiết kế trang Web có thể sử dụng thông tin này để cải thiện thiết kế trang Web của họ, và để tăng hiệu suất trang Web. Nhiều công trình nghiên cứu đã tập trung vào lĩnh vực này như; thuật toán quét toàn bộ (FS) [4], thuật toán quét chọn lọc (SS) [4], thuật toán duy trì mô hình duyệt thường xuyên (MAFTP) [12], .v.v.... Tuy nhiên, các thuật toán này có những hạn chế trong đó họ chỉ có thể khám phá mô hình con đường duyệt đơn giản, nơi một trang không thể xuất hiện nhiều lần trong mô hình duyệt. Ngoài ra, những thuật toán này chỉ xem xét chuyển tiếp tiến trong các cơ sở dữ liệu trình tự duyệt. Do đó, các mô hình con đường duyệt đơn giản phát hiện bởi các thuật toán trên là không đủ cho một môi trường Web. Một mô hình con đường duyệt không đơn giản, chẳng hạn như mô hình duyệt Web, không chỉ chuyển tiếp tiến mà còn chuyển tiếp lùi. Thông tin này có thể nắm bắt được hành vi chuyển hướng người dùng hoàn toàn chính xác. Đây là phương pháp được trình bày trong thuật toán khai thác mô hình duyệt thường xuyên (MFTP) [17], và thuật toán tích hợp các mô hình con đường duyệt và luật kết hợp (IPA) [6, 7].

Thuật toán MAFTP [12] là một kỹ thuật tăng trưởng cải tiến cho việc duy trì các mô hình con đường duyệt được phát hiện khi các trình tự người dùng được thêm vào cơ sở dữ liệu. Thuật toán MAFTP phân vùng cơ sở dữ liệu thành các đoạn và quét các phân đoạn cơ sở dữ liệu. Đối với mỗi phân đoạn quét, ứng cử viên trình tự duyệt không phải là mô hình duyệt thường xuyên được cắt tỉa để đảm bảo trình tự duyệt thường xuyên có thể được phát hiện sớm hơn. Tuy nhiên, thuật toán MAFTP chỉ xem xét chuyển tiếp tiến trong cơ sở dữ liệu trình tự duyệt. Hơn nữa, thuật toán MAFTP chỉ giải quyết các trường hợp trình tự sử dụng được chèn; nó không thể giải quyết các trường hợp trình tự sử dụng có thể bị xóa. Khai thác mô hình trình tự [5, 8, 9, 10, 13] cũng tương tự như khai thác mô hình duyệt Web. Sự khác biệt quan trọng nhất giữa mô hình duyệt Web và các mô hình tuần tự, là mô hình duyệt Web đòi hỏi một liên kết giữa mỗi hai trang Web trong cấu trúc Web.

Đó là, phải chắc chắn rằng có một liên kết từ mỗi trang đến trang tiếp theo trong một mô hình duyệt Web. Ngoài ra, khai thác mô hình trình tự chỉ phát hiện ra mô hình trình tự từ một cơ sở dữ liệu trình tự của khách hàng. Zaki và đồng sự [10] đề xuất thuật toán khai thác mô hình trình tự tăng trưởng sử dụng dàn (ISL). Thuật toán này được dựa trên mô hình trình tự phát hiện sử dụng các lớp tương đương (SPADE) [13]. Thuật toán ISL cập nhật các cấu trúc dàn bất cứ khi nào các cơ sở dữ liệu được cập nhật. Các cấu trúc dàn giữ tất cả các mô hình trình tự, ứng cử viên trình tự, và độ hỗ trợ của nó, do đó chỉ ứng cử viên trình tự được tạo mới cần phải được tính từ cơ sở dữ liệu ban đầu nhằm nâng cao hiệu quả khai thác. Nếu các ứng cử viên trình tự có độ hỗ trợ bằng không cũng được lưu giữ trong dàn, cấu trúc dàn sẽ là quá lớn để lưu vào bộ nhớ. Thuật toán khai thác mô hình trình tự tăng trưởng là khai thác tăng trưởng trong mô hình tuần tự (IncSpan) thuật toán đã được đề xuất bởi Cheng và các đồng sự [5]. Thuật toán này được dựa trên thuật toán khai thác mô hình trình tự chiếu dựa trên tiền tố (PrefixSpan) [8, 9]. Thuật toán IncSpan sử dụng các khái niệm về cơ sở dữ liệu chiếu để khai thác đệ quy các mô hình trình tự. Tuy nhiên, cả hai thuật toán ISL và IncSpan chỉ có thể giải quyết những trường hợp mà các trình tự người dùng mới được đưa vào cơ sở dữ liệu trình tự khách hàng. Họ chỉ giải quyết việc chèn các giao dịch vào các trình tự sử dụng đã tồn tại từ trước. Trong thực tế, các trình tự sử dụng có thể xuất hiện bất cứ lúc nào trong môi trường Web. Bên cạnh đó, thuật toán ISL và IncSpan phù hợp cho khai thác mô hình trình tự.

Để cải thiện những thiếu sót, thuật toán IncWTP [14] đề xuất bởi Lee và đồng sự. Thuật toán IncWTP có thể khám phá các mô hình con đường duyệt không đơn giản, và cả hai trường hợp chèn vào (xóa từ) cơ sở dữ liệu được xem xét. Hơn nữa, chỉ có một số lượng nhỏ các ứng cử viên trình tự duyệt cần phải được tính từ cơ sở dữ liệu trình tự duyệt ban đầu trong thuật toán.

Thuật toán IncWTP sử dụng một cấu trúc dàn đặc biệt, có tên dàn mở rộng để giữ trình tự ứng cử viên có độ hỗ trợ lớn hơn không. Ngoài ra, các thứ tự của trình tự cũng được lưu giữ trong cấu trúc dàn để nâng cao hiệu quả trong quá trình thực

hiện khai thác mô hình tăng trưởng. Do đó cấu trúc dàn quá lớn để được giữ trong bộ nhớ, nên nó được lưu trong ổ cứng. Trong khi thực hiện quá trình khai thác mô hình tăng trưởng, mỗi cấp trong dàn sẽ được nạp vào bộ nhớ và duy trì bởi lượt. Nhưng quá trình này sẽ mất rất nhiều thời gian cho nhập/xuất. Ngoài ra thuật toán IncWTP cho phép các ứng viên α ở mức *k* tham gia sinh ứng viên α ở mức *k*+1 khi chúng là một mô hình duyệt Web đủ tiêu chuẩn và Support(α) ≥ min_sup. Trong đó min_sup là một ngưỡng tối thiểu người dùng chỉ định và $\alpha = <x_1, x_2, \ldots, x_l>$ là một mô hình duyệt Web đủ tiêu chuẩn, nếu có một liên kết từ x_i đến $x_i + 1$ (cho tất cả i, $1 \leq i \leq l-1$) trong một cấu trúc trang Web. Như vậy, khi ta điều chỉnh min_sup giảm thì khả năng các ứng cử viên mới sẽ được sinh ra là rất cao, và lúc này cần phải quét lại CSDL để đếm độ hỗ trợ cho ứng viên này. Hơn nữa, khi trình tự mới được thêm vào thì khả năng sinh ứng viên mới cũng rất cao.

Vì vậy trong luận văn này tôi đề xuất thuật toán WebTP sử dụng cấu trúc cây để lưu trữ các ứng viên theo cấp. Tất cả các ứng cử viên trình tự có độ hỗ trợ bằng một và các TIDs cũng được lưu trữ trên cây để nâng cao hiệu quả trong quá trình khai thác mô hình tăng trưởng. Bên cạnh đó thuật toán IntWebTP duyệt cấu trúc cây một lần để tìm ra các mô hình duyệt Web mỗi khi điều chỉnh min_sup mà không cần duyệt lại CSDL ban đầu. Và thuật toán RemoveLink để cập nhật mô hình duyệt Web khi xóa liên kết trong cấu trúc WebSite.

CHƯƠNG 3: THUẬT TOÁN KHAI THÁC MÔ HÌNH DUYỆT WEB

3.1 Các vấn đề liên quan

Khai thác mô hình duyệt Web nghĩa là phát hiện hầu hết các mô hình truy cập của người sử dụng từ các bản ghi Web. Để bắt đầu, chúng ta xác định mô hình duyệt Web như sau:

+ Cho $I = \{ x_1, x_2, \ldots, x_n\}$ là một tập hợp của tất cả các trang Web trong một Website.

+ Một trình tự duyệt $S = <w_1, w_2,..,w_m>$ ($w_i \in I$, $1 \leq i \leq m$) là một danh sách các trang Web được sắp xếp tăng dần theo thời gian duyệt, trong đó mỗi trang Web có thể xuất hiện nhiều hơn một lần trong một trình tự duyệt.

+ Chiều dài |S| của một trình tự duyệt S là tổng số của các trang Web trong S. Một trình tự duyệt với chiều dài *l* được gọi là một trình tự duyệt *l*.

Giả sử rằng có hai trình tự duyệt $\alpha = <a_1, a_2, \ldots, a_m>$ và $\beta = <b_1, b_2,.., b_n)$ với ($m \leq n$). Nếu tồn tại $i_1 < i_2 < \ldots < I_m$, như vậy mà $bi_1 = a_1$, $bi_2 = a_2$, . . . , $bi_m = a_m$, thì β bao gồm α, α là một chuỗi con của β và β là một chuỗi cha của α. Ví dụ, nếu có hai chuỗi duyệt α = <BEA> và β = <ABCEA>, thì α là một chuỗi con của β và β là một chuỗi cha của α.

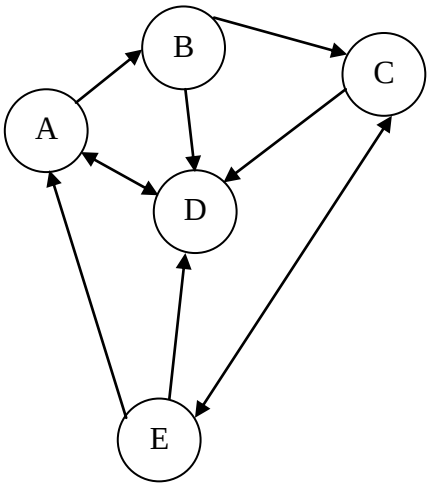
Một cơ sở dữ liệu trình tự duyệt D, như thể hiện trong Bảng 3.1, chứa một tập các bản ghi. Mỗi bản ghi bao gồm một TID và một trình tự sử dụng . Một trình tự sử dụng là một chuỗi duyệt, đó là viết tắt cho một hành vi duyệt Web thành công của người dùng nhất định. Độ hỗ trợ của một trình tự duyệt α là tỷ số giữa số trình tự duyệt có chứa α với tổng số trình tự duyệt trong D. Nó thường được ký hiệu là Support (α). Độ hỗ trợ của α là số các trình tự duyệt có chứa α.

Một chuỗi duyệt $\alpha = <x_1, x_2, \ldots, x_l>$ là một mô hình duyệt Web nếu Support(α) ≥ min_sup và có một liên kết từ x_i đến $x_i + 1$ (cho tất cả i, $1 \leq i \leq l-1$) trong một cấu trúc trang Web, trong đó min_sup là một ngưỡng tối thiểu người dùng chỉ định.

Một cấu trúc Website như được thể hiện trong hình 3.1, mỗi đỉnh là một trang Web và mỗi cung thể hiện cho mối liên kết giữa các trang Web.

Bảng 3.1 C*ơ sở dữ liệu trình tự duyệt*

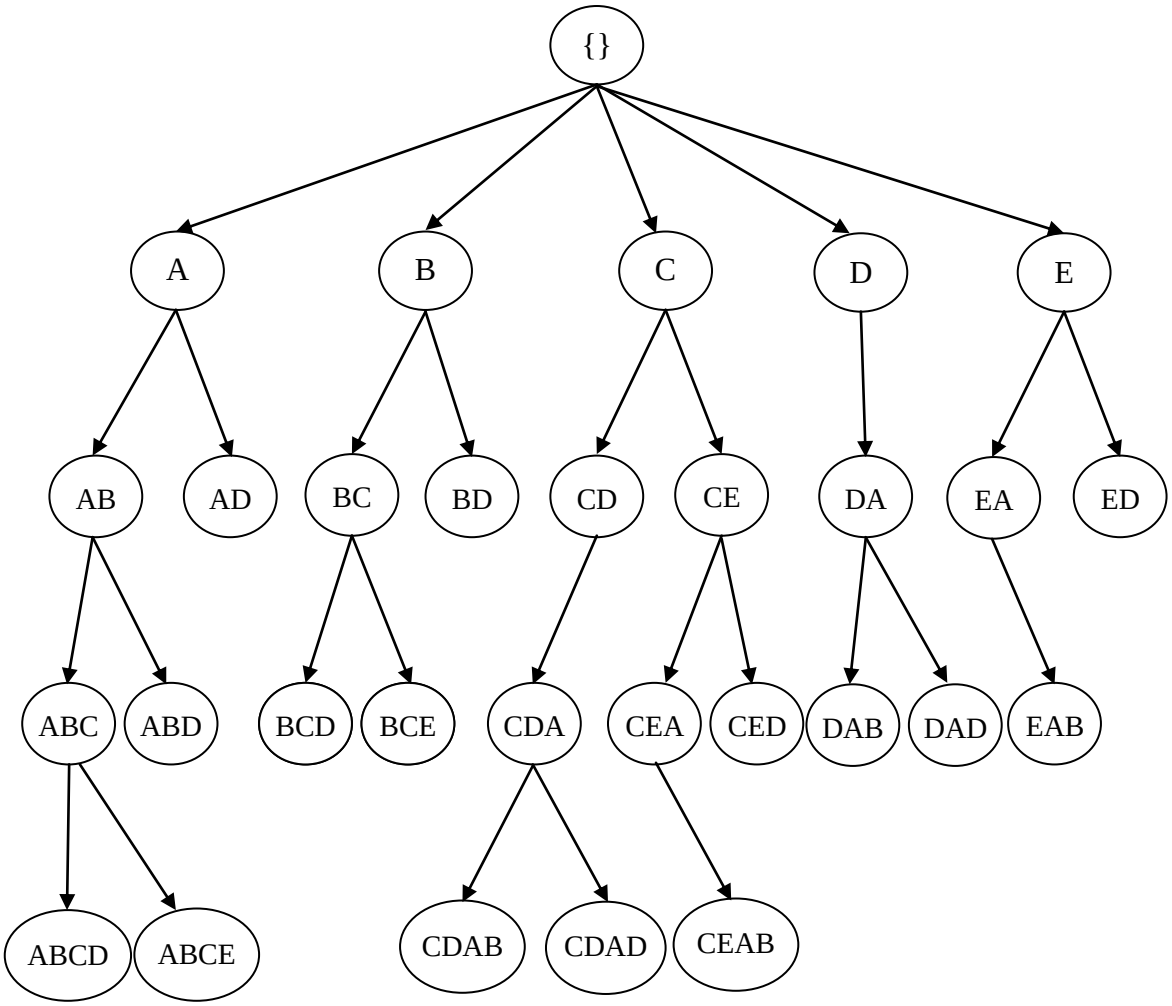
TID	User sequence
1	ABCED
2	ABCD
3	CDEAD
4	CDEAB
5	CDAB
6	ABDC



Hình 3.1: *Cấu trúc Website*

3.2 Cấu trúc dữ liệu được sử dụng cho khai thác mô hình duyệt Web

Để khai thác mô hình duyệt Web, kết quả khai thác trước đó được sử dụng để khám phá các mô hình mới, do đó thời gian khai thác có thể được giảm. Trong luận văn này, cấu trúc cây được sử dụng để lưu trữ kết quả khai thác trước đó. Hình.3.2 cho thấy cấu trúc cây cho cơ sở dữ liệu được mô tả trong Bảng 1. Khi min_sup được thiết lập đến 50 %, chỉ có mô hình duyệt Web đủ tiêu chuẩn được lưu trữ trong cấu trúc cây. Một mô hình duyệt $\alpha = <x_1, x_2, \ldots, x_l>$ là mô hình duyệt Web đủ tiêu chuẩn khi Support(α) ≥ 1 và có một liên kết từ x_i đến $x_i + 1$ (cho tất cả i, $1 \leq i \leq l$-1) trong một cấu trúc trang Web.



Hình 3.2 *Cấu trúc cây đơn giản*

Để từng bước khai thác tương tác các mô hình duyệt Web và tăng tốc độ quá trình khai thác, cấu trúc cây được mở rộng để ghi thêm thông tin. Việc mở rộng cấu trúc cây được hiển thị trong hình.3.3. Trong hình.3.3, mỗi nút chứa một trình tự duyệt Web có support lớn hơn hoặc bằng một. Nối thêm support vào phần trên của mỗi nút. Thông tin này được sử dụng để tính toán và tích lũy support như tiền khai thác mô hình gia tăng.

Ngoài ra, các TIDs trình tự duyệt cũng được thêm vào phần dưới của mỗi nút. thông tin này được sử dụng để giảm số lần quét cơ sở dữ liệu không cần thiết. Đây là sự khác nhau với một cấu trúc cây đơn giản, chúng ta đặt tất cả các ứng cử viên trình tự duyệt có độ hỗ trợ bằng một (với support được tính dựa vào min_sup) vào cấu trúc cây.

Chúng ta có thể sử dụng cấu trúc cây này để nhanh chóng tìm ra các mô hình duyệt Web khi min_sup được điều chỉnh. Ví dụ, nếu người dùng muốn điều chỉnh min_sup từ 50 % xuống 20 %, họ có thể chỉ đơn giản là đi qua các mức của cấu trúc cây và trả về các mô hình có support ≥ min_sup. Hơn nữa, nếu ai đó muốn tìm mô hình duyệt Web tối đa mà không muốn quan tâm đến các mô hình duyệt Web khác, họ chỉ cần trả lại các mô hình duyệt Web có support ≥ min_sup trong mức trên cùng của cấu trúc cây. Ví dụ, trong hình.3.2, mô hình duyệt Web <ABC>, <ABD> và <CDA> là mô hình duyệt Web tối đa.

Với các mô hình duyệt Web từ CSDL trình tự duyệt như trong Bảng 3.1. Kết quả cuối cùng được hiển thị trong hình.3.3 nơi min_sup đã được thiết lập đến 50%.

Giả sử rằng một Website có 300 trang Web nếu chúng ta giữ các ứng viên có độ hỗ trợ lớn hơn hoặc bằng 1 trên cấu trúc cây và không dựa vào cấu trúc trang Web, thì có khoảng 299 × 300 = 89.700 ứng cử viên có chiều dài 2 cần lưu giữ. Giả sử rằng ta thiết min_sup là 50 % thì có khoảng 89.700 : 2 × 1.1 = 49.333.

Phương pháp sinh ứng viên được dựa theo phương pháp được đề xuất trong [5].

Xét một dãy có *k* phần tử, gọi là *k-sequence* (nếu một phần tử xuất hiện nhiều lần trong các thành phần khác nhau của một dãy, mỗi lần xuất hiện được tính vào giá trị của *k*). Gọi L_k biểu thị tập tất cả các dãy phổ biến *k-sequence* và C_k biểu thị tập các dãy ứng viên *k-sequence*.

Cho L_{k-1} là tập tất cả các dãy phổ biến *(k-1)-sequence*, ta cần tạo ra tập cha (superset) của tập tất cả các dãy phổ biến *k-sequence*. Đầu tiên, ta định nghĩa khái niệm về một dãy con liên tục.

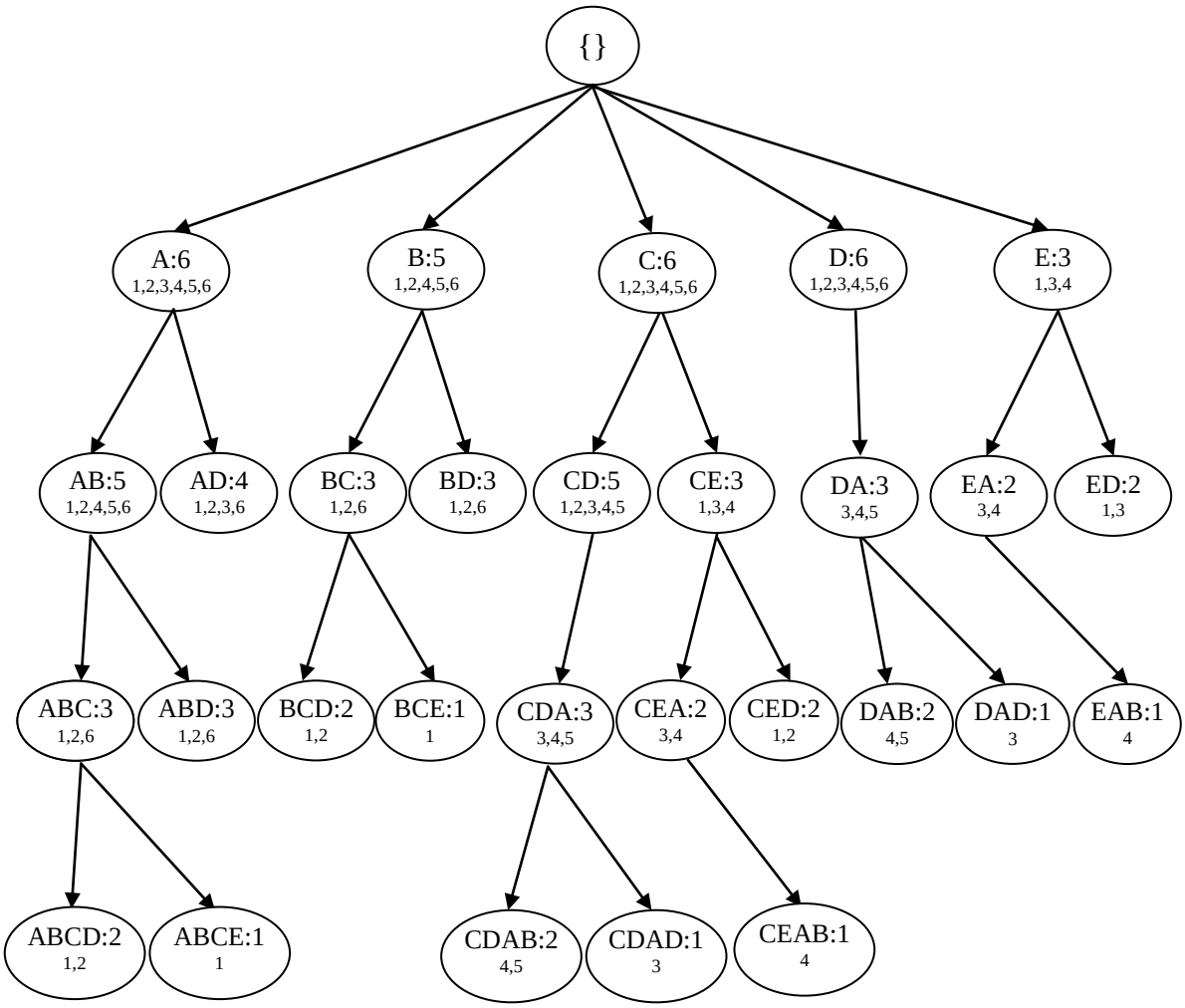
Định nghĩa: Cho dãy $s = <s_1\ s_2\ ...\ s_n>$ và một dãy con c, c là dãy con liên tục của s nếu thỏa mãn bất kỳ điều kiện nào sau đây:

1. c nhận được từ s bằng cách lược bỏ phần tử s_1 hoặc s_n.
2. c nhận được từ *s* bằng cách lược bỏ một phần tử từ thành phần s_i mà s_i có ít nhất hai phần tử.
3. c là dãy con liên tục của c', và c' là dãy con liên tục của *s*.

Nghĩa là đối với hai mô hình duyệt Web phân biệt: $s_1 = <s_1,...,s_{k\_1}>$ và $s_2 = <u_1,...,u_{k-1}>$ có thể ghép với nhau để tạo thành một trình tự duyệt *k* khi và chỉ khi $<s_2,..,s_{k-1}>$ và $<u_1,...,u_{k-2}>$ là chính xác giống nhau, hoặc $<u_2,...,u_{k\_1}>$ là chính xác giống như $<s_1,...,s_{k\_2}>$ (tức là, sau khi bỏ trang đầu tiên trong một mô hình duyệt Web s_1 và trang cuối cùng trong mô hình duyệt Web s_2, ta được trình tự duyệt (*k-2*) giống nhau). Dãy ứng viên mới được sinh là dãy s_1 được mở rộng với phần tử cuối cùng trong s_2.

***Ví dụ*:** ứng cử viên trình tự duyệt <ABCDE> có thể được tạo ra bằng cách kết hợp hai mô hình duyệt Web <ABCD> và <BCDE>.

Đối với một ứng cử viên trình tự duyệt *l* của α, được sinh ra bởi trình tự con chiều dài (*l* -1) của α. Chúng ta không cần kiểm tra tất cả các trình tự duyệt Web con với chiều dài *l*-1 của mỗi ứng cử viên trình tự duyệt *l*. Trong ví dụ trên, chúng ta không cần kiểm tra xem <ABDE>, <ACDE> và <ABCE> có phải là mô hình duyệt Web. Vì tất cả các mô hình đã lưu trữ trên cây đều là mô hình duyệt Web đủ tiêu chuẩn và được tham gia sinh ứng viên.



Hình 3.3: *cấu trúc cây mở rộng*

3.3 Thuật toán

3.3.1 Thuật toán InWebTP

Thuật toán InWebTP thực hiện việc khai thác mô hình duyệt Web khi các trình tự duyệt được thêm vào CSDL. Thuật toán làm việc từ mức đầu tiên đến mức cuối cùng của cấu trúc cây. Đối với mỗi mức thuật toán kiểm tra nếu các mô hình có trên cấu trúc cây thì thêm TID vào node và tăng support cho node đó. Nếu mô hình chưa có trong cấu trúc cây thì một node mới được tạo ra.

Thuật toán: InWebTP(InsTID, T)

Dữ liệu đầu vào: tập các giao dịch được thêm vào InsTID, cấu trúc cây T.

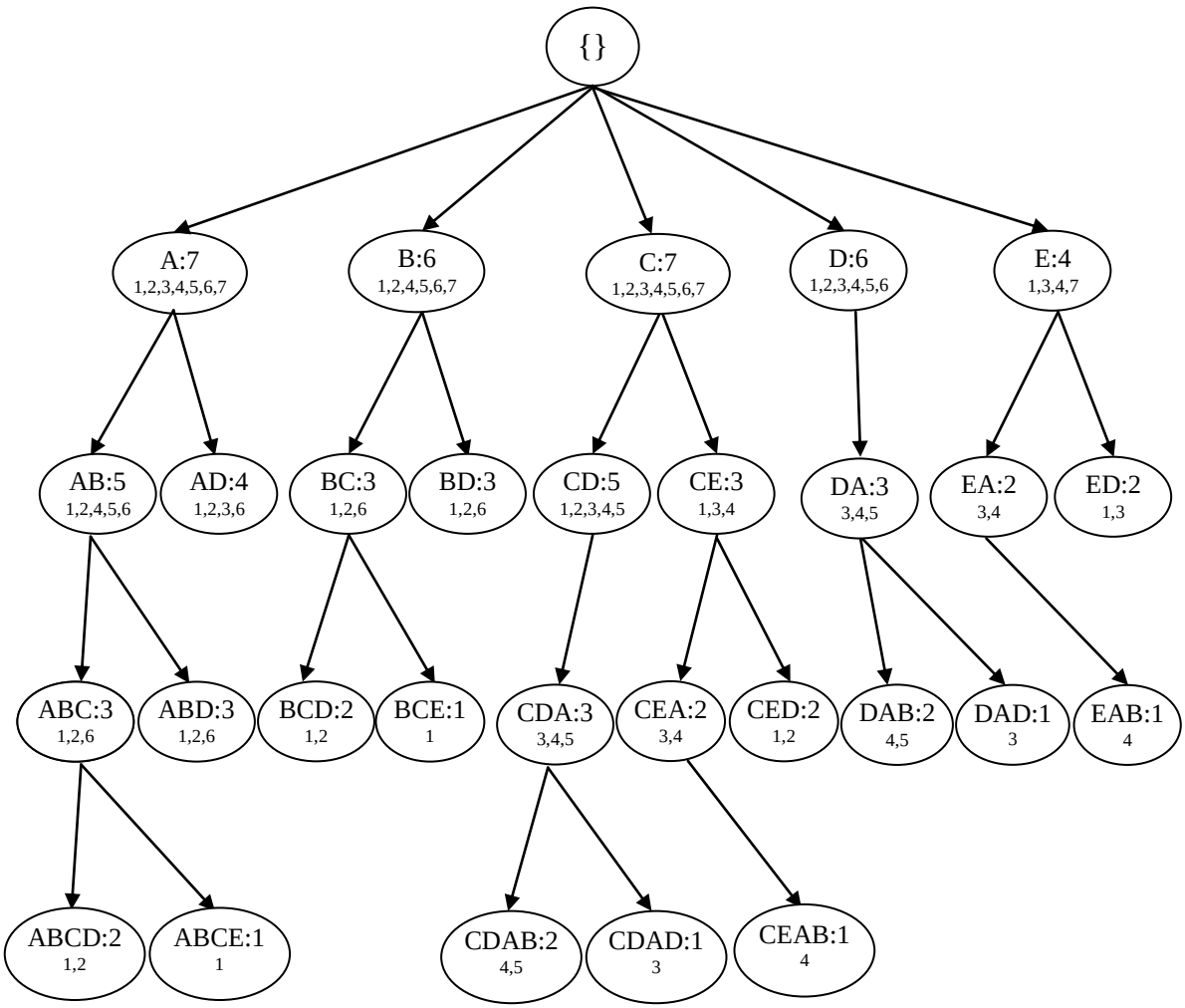
Dữ liệu ra: Tất cả các mô hình duyệt Web sau khi thêm tập InsTID.

```
int level = InsTID.getLevelCount();
for(int i = 1; i<= level; i++)
  List<Pattern> listE = patterns.getLevel(i);
   for (int j = 0; j <listE.size() ;j ++)
          Pattern seq = (Pattern)listE.get(j);
          List<ItemAbstractionPair> listitem = seq.getElements();
          int item = Integer.parseInt(listitem.get(0).getItem().toString());
          BitSet listbit = seq.getAppearingIn();
          ITNode find = T;
          for (int findlevel = 0; findlevel< i-1;findlevel++)
                if (find == null) break;
                List<ITBitSetNode> listC = find.getListChildNode();
                for (int fitem = 0; fitem < listC.size(); fitem++)
                      if( listC.get(fitem).getItem() == item)
                            find = listC.get(fitem);
                             item = Integer.parseInt(listitem.get(findlevel+1)
                             .getItem().toString());
     output all the web traversal patterns in
```

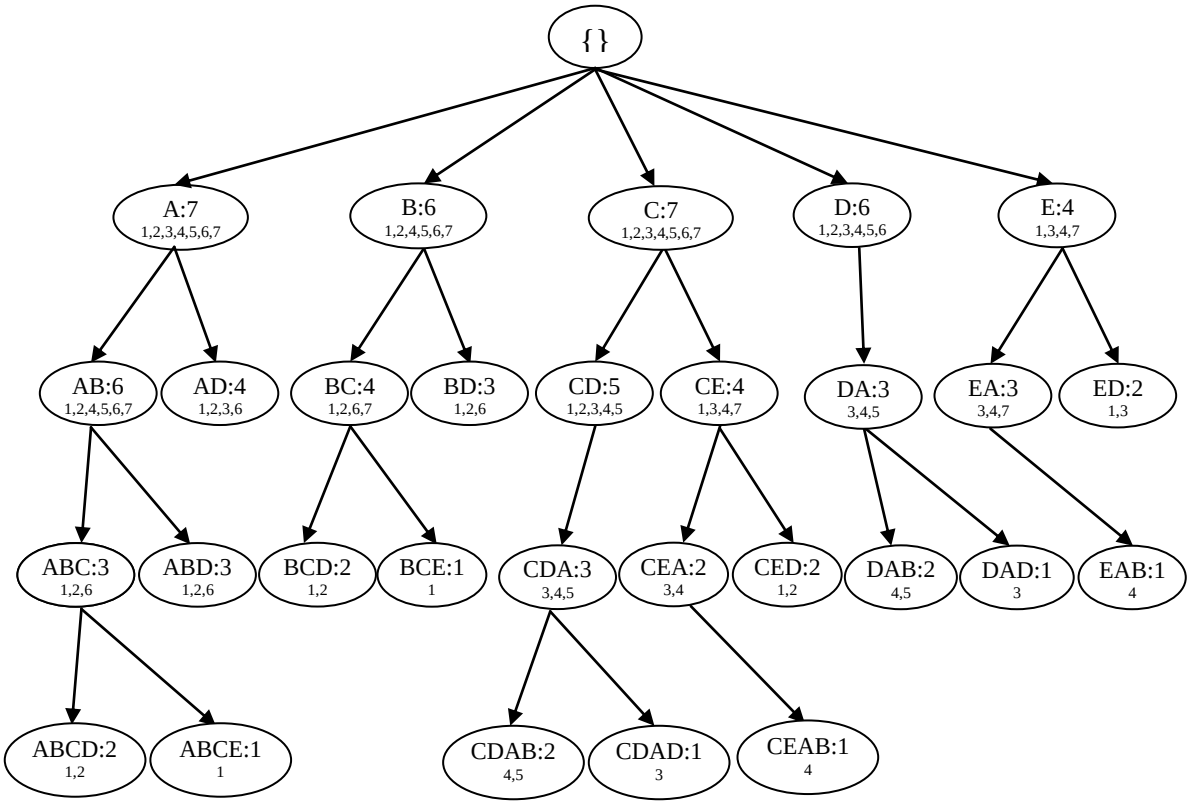
Ví dụ: Trong bảng 3.1, chúng ta thêm trình tự sử dụng (7, ABCED) như thể hiện trong Bảng 3.2. Min_sup cũng được thiết lập đến 50 %. Ở mỗi mức của cấu trúc cây các mô hình được phân bổ vào các node và support tăng.

Bảng 3.2 C*ơ sở dữ liệu trình tự duyệt sau khi thêm TID 7*

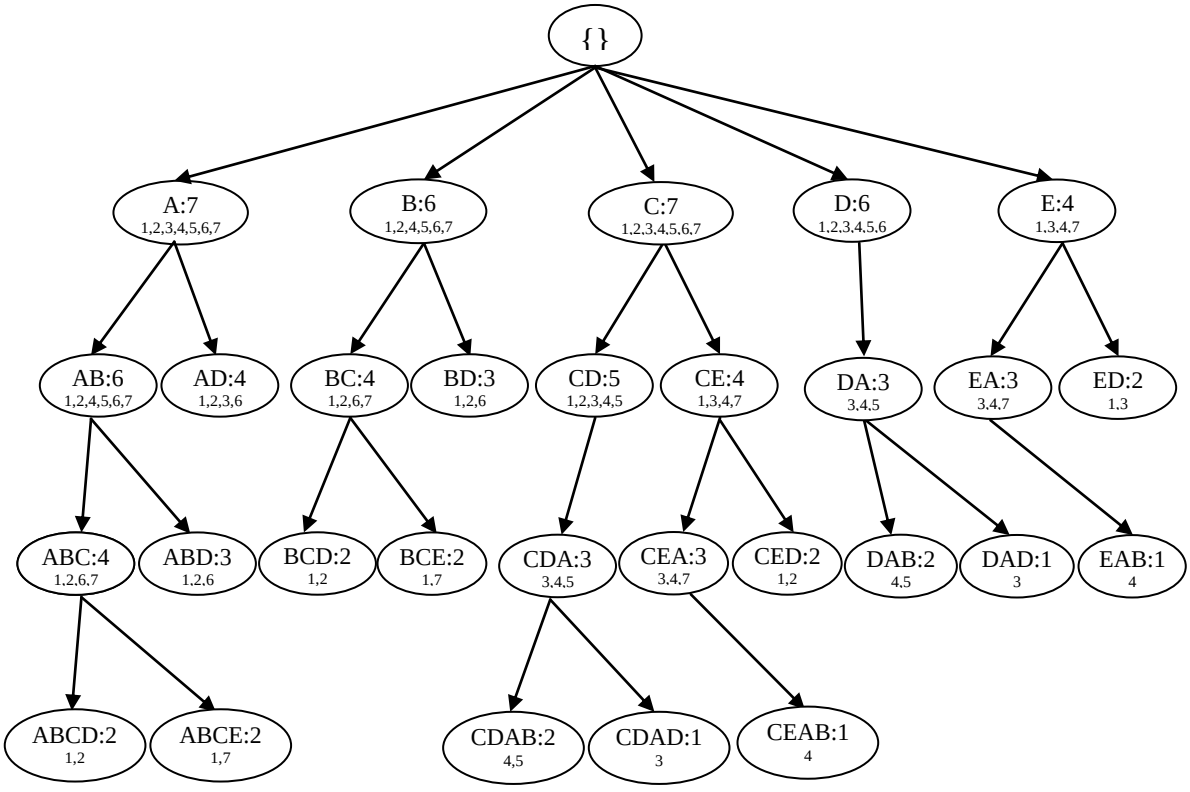
TID	User sequence
1	ABCED
2	ABCD
3	CDEAD
4	CDEAB
5	CDAB
6	ABDC
7	ABCEA



Hình 3.4: *Cập nhật cấu trúc cây sau khi xử lý ở mức1*



Hình 3.5: *Cập nhật cấu trúc cây sau khi xử lý ở mức 2*



Hình 3.6: *Cập nhật cấu trúc cây sau khi xử lý ở mức 3 và 4*

3.3.2 Thuật toán WebTP

Thuật toán WebTP thực hiện việc khai thác mô hình duyệt Web sẽ làm việc từ node con đầu tiên cho đến node cuối cùng của cấu trúc cây. Đối với mỗi node tất cả các node con của node đó được duyệt qua để kiểm tra, nếu node chứa các TIDs bị xóa thì ta lấy ra TIDs từ node đó và support của node đó sẽ được giảm. Sau khi xóa kiểm tra nếu Support của node đó bằng không thì node đó sẽ bị xóa khỏi cấu trúc cây.

Thuật toán: WebTP (DelTID, T)

Dữ liệu đầu vào: tập các giao dịch cần xóa DelTID, cấu trúc cây T.

Dữ liệu ra: Tất cả các mô hình duyệt Web sau khi xóa tập DelTID.

```
if (DelTID != null)
        for (i = 1; i <= DelTID.leght; i++)
                RemoveTreeTid (DelTID[i], T);
if (T != null)
   output all the web traversal patterns in
```

Thuật toán: RemoveTreeTid (int tid,T)

Dữ liệu đầu vào: giao dịch cần xóa tid , cấu trúc cây T

Dữ liệu ra: Tất cả các mô hình duyệt Web sau khi xóa tập DelTID.

```
        List<ITNode> child = T.getListChildNode();
        if(child != null)
                for(int i = child.size()-1; i >=0 ; i--)
                        ITNode nodeC = child.get(i);
                        nodeC.removeTid(tid);
```

***Thuật toán*:** Removetid(DelTID, T)

Dữ liệu đầu vào: giao dịch bị xóa DelTID, cấu trúc cây T.

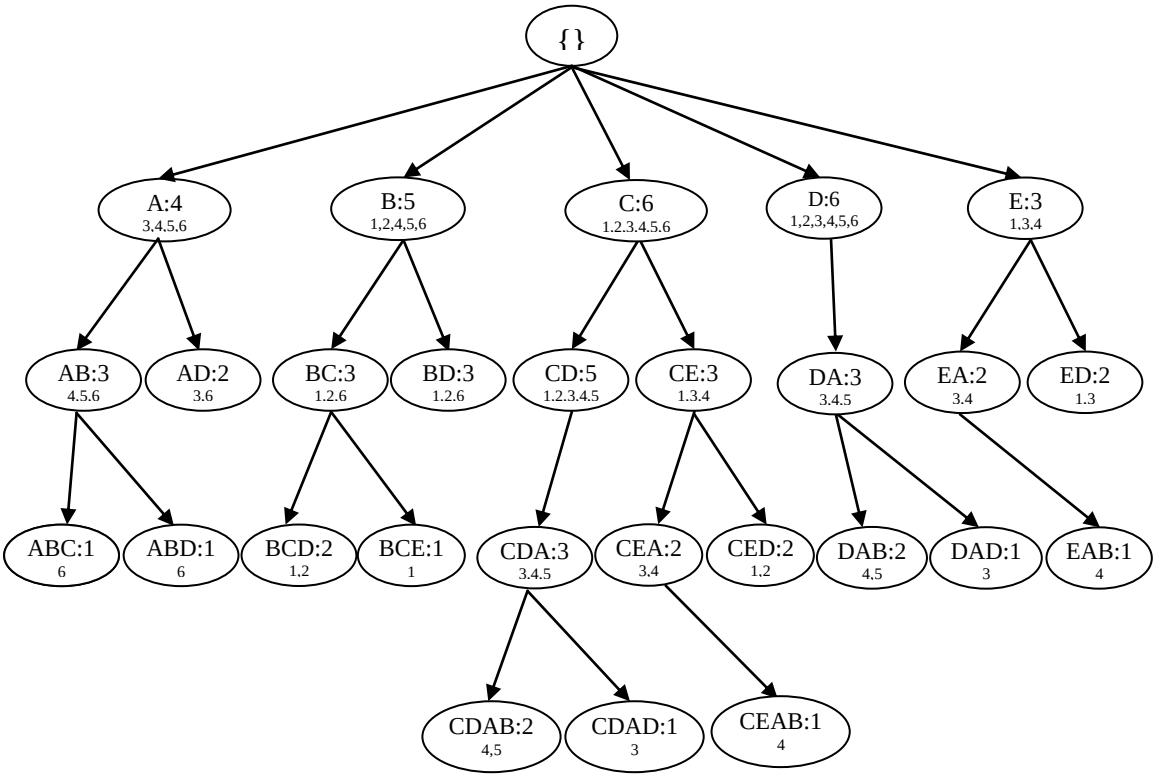
Dữ liệu ra: Tất cả các mô hình duyệt Web có support >0

```
if(T != null)
        for (int i = 0; i <= T.getListChildNode(); i ++)
            if(T.getchild(i).get(item)== tid )
                listTid = T.getchild(i).gettid;
                listTid.set(i, false);
                if( this.child != null )
                    for(int j=0; j<child.size(); j++)
                        child.get(j).removeTid(tid);
                        if (child.get(j).getSupport()==0)
                            child.get(j).deleteChildNode();
                            child.remove(j);
                            j--;
```

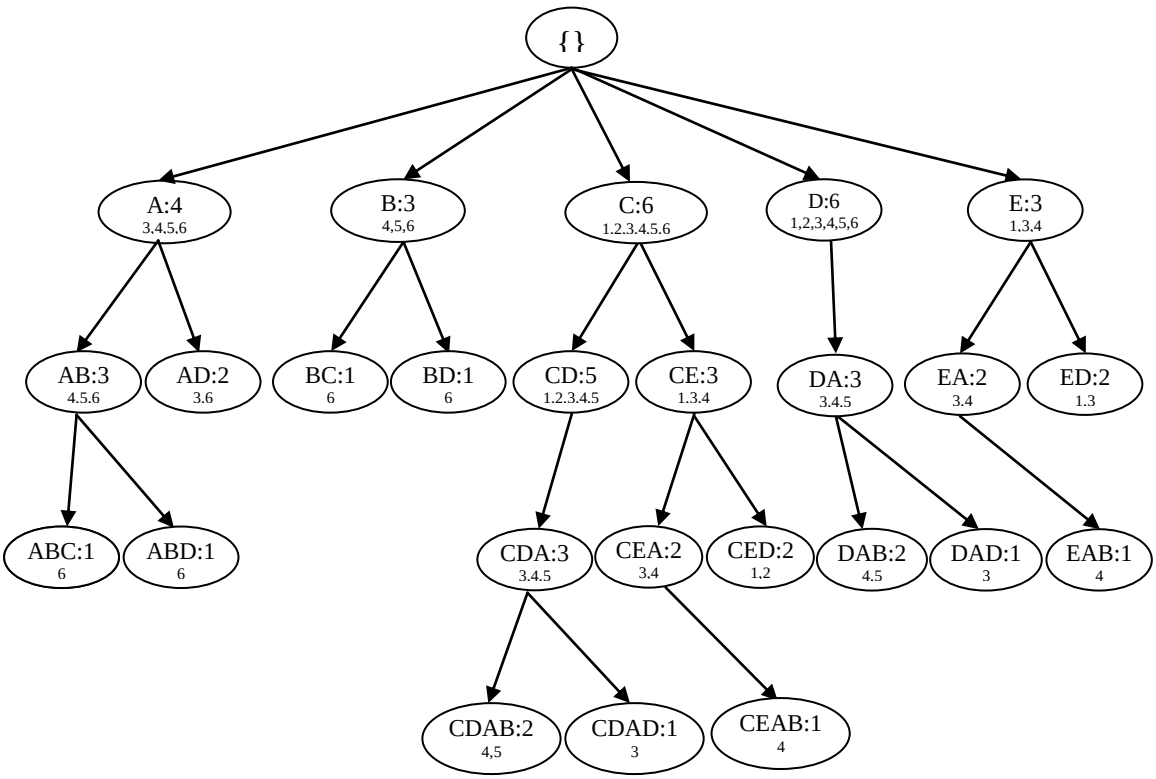
Ví dụ: Trong bảng 3.1, chúng ta xóa hai trình tự sử dụng (1 , ABCED) và (2, ABCD) như thể hiện trong Bảng 3.3. Min_sup cũng được thiết lập đến 50 %. Ở mức đầu tiên của cấu trúc cây trong hình. 3.4, TID 1 và TID 2 được xóa và support giảm ở các nút chứa TID 1 hoặc TID 2.

Bảng 3.3 C*ơ sở dữ liệu trình tự duyệt sau khi xóa TID 1 và TID 2*

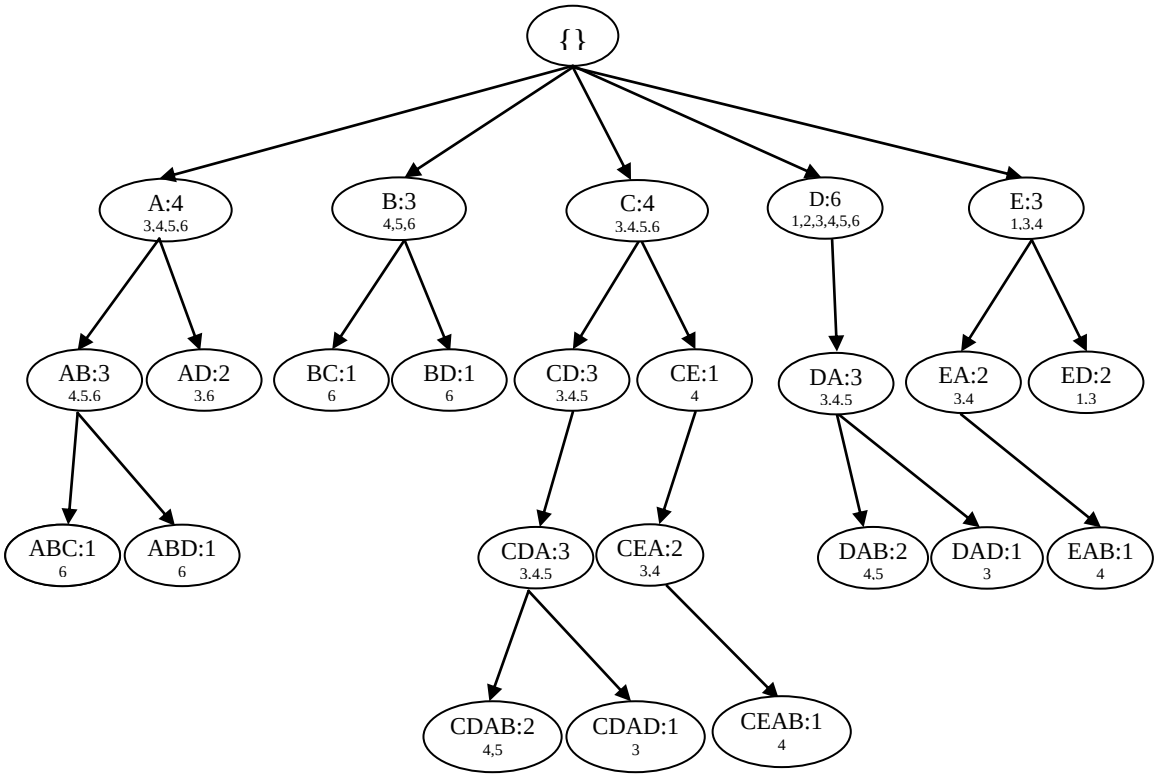
TID	User sequence
3	CDEAD
4	CDEAB
5	CDAB
6	ABDC



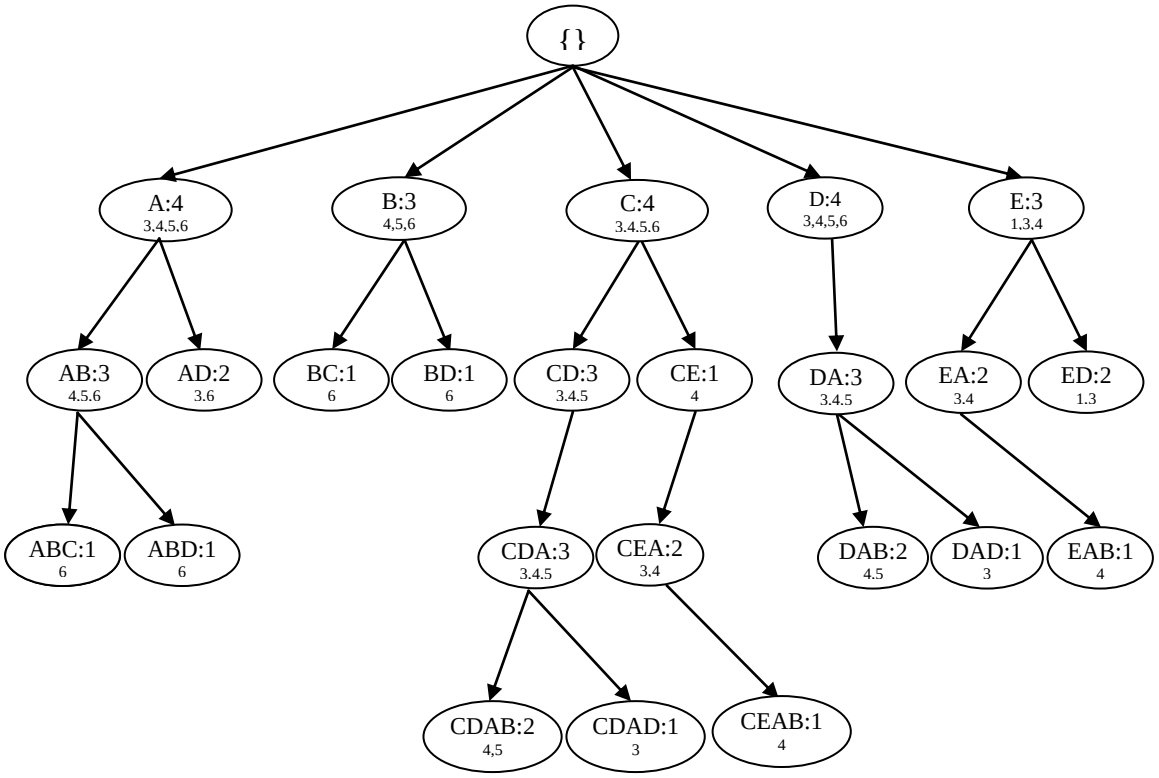
Hình 3.7: *Cập nhật cấu trúc cây sau khi xử lý ở node con A*



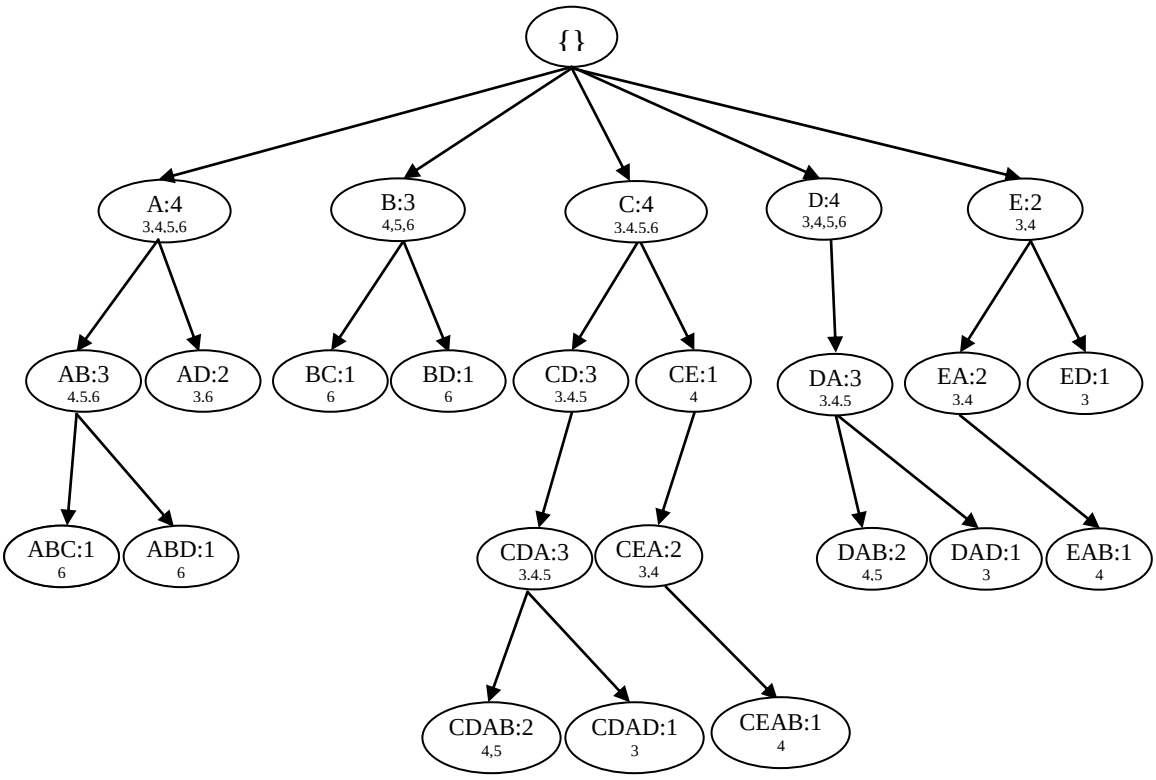
Hình 3.8: *Cập nhật cấu trúc cây sau khi xử lý ở node con B*



Hình 3.9: *Cập nhật cấu trúc cây sau khi xử lý ở node con C*



Hình 3.10: *Cập nhật cấu trúc cây sau khi xử lý ở node con D*



Hình 3.11: *Cập nhật cấu trúc cây sau khi xử lý ở node con E*

3.3.3 Thuật toán IntWebTP

Thuật toán IntWebTP phục vụ việc khai thác mô hình duyệt Web khi min_sup được điều chỉnh. Thuật toán này sẽ duyệt cấu trúc cây một lần để trả về tất cả các mô hình duyệt Web thỏa min_sup mới mà không cần phải duyệt lại CSDL ban đầu.

Thuật toán: IntWebTP (New_minsup, T)

Dữ liệu vào: Độ hỗ trợ mới New_minsup, cấu trúc cây T.

Dữ liệu ra: Tất cả các mô hình duyệt Web thỏa min_sup mới.

```
if( T != null)
    List <ITNode> child = T.getListChildNode();
    if(child != null)
        for(int i = 0; i < child.size(); i++)
            ITNode nodeC = child.get(i);
            if( nodeC.getSupport() >= support)
                if(write != null)
```

```
                    write.write(nodeC.NodetoString());
                    write.newLine();
                else
                System.out.println(nodeC.NodetoString());
            IntWebTP (write,support,nodeC);
```

Ví dụ: Ta sử dụng bảng 3.1 và cấu trúc cây ở hình 3.3 để minh họa thuật toán

Trường hợp thứ nhất: min_sup được điều chỉnh tăng từ 50 % lên 70 %. Thuật toán IntWebTP sẽ duyệt qua cấu trúc cây một lần và các mô hình duyệt Web thỏa min_sup mới sẽ được trả về (tức là nơi mà support không nhỏ hơn 4). Trong trường hợp này ta có <A>, <B>, <C>, <D>, <AB> và <CD> là mô hình duyệt Web.

Trường hợp thứ hai: min_sup được điều chỉnh giảm từ 50 % xuống 40 %. Thuật toán IntWebTP cũng sẽ duyệt qua cấu trúc cây một lần và các mô hình duyệt Web thỏa min_sup mới sẽ được trả về (tức là nơi mà support không nhỏ hơn 2). Trong trường hợp này ta có <A>, <B>, <C>, <D>, <E>, <AB>, <AD>, <BC>, <BD>, <CD>, <CE>, <DA>, <ABC>, <ABD> và <CDA> là mô hình duyệt Web.

3.3.4 Thuật toán RemoveLink

Thuật toán RemoveLink phục vụ việc khai thác mô hình duyệt Web khi một liên kiên bị xóa khỏi cấu trúc trang Web. Thuật toán này sẽ duyệt cấu trúc cây một lần để xóa các nút có chứa liên từ trang Web A đến trang Web B, sau đó trả về tất cả các mô hình duyệt Web mới.

Thuật toán: RemoveLink(T, A, B)

Dữ liệu vào: Cấu trúc cây T, Liên kết cần xóa từ trang Web A đến trang Web B.

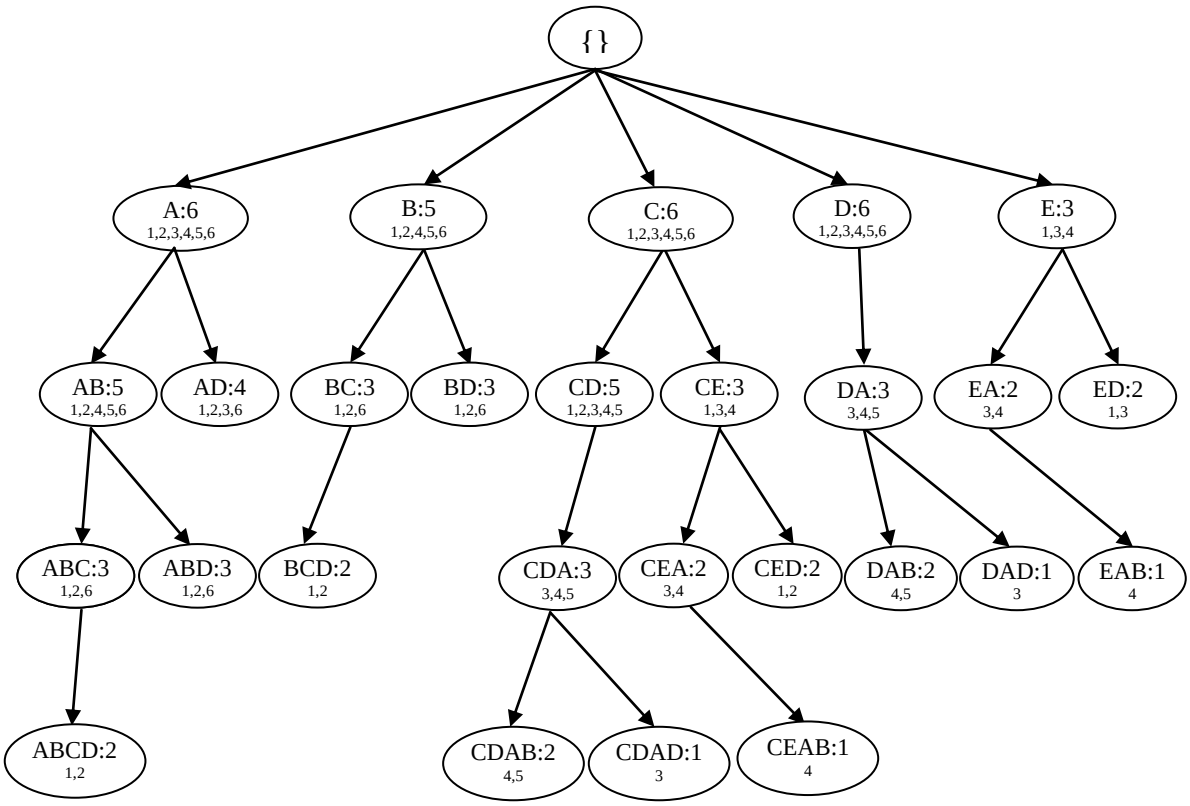
Dữ liệu ra: Tất cả các mô hình duyệt Web không chứa <AB>.

```
if (T != null)
        if (T.getListChildNode() == null) return false;
        if (T.getItem() == item1)
```

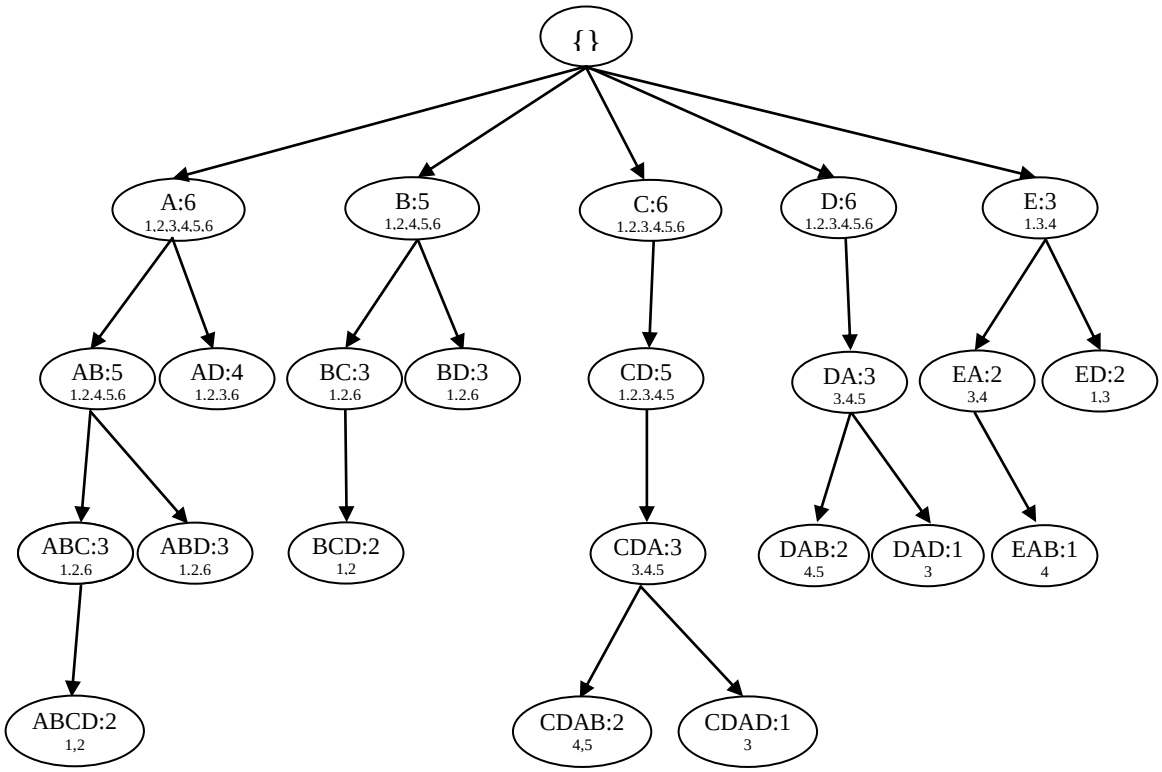
```
            ITNode mychild;
            for(int i = 0; i< T.getListChildNode().size(); i++)
                mychild = T.getListChildNode().get(i);
                if( mychild.getItem() == item2)
                    mychild.deleteChildNode();
                    T.getListChildNode().remove(i);
        else
            ITNode mychild;
            for(int i = 0; I < T.getListChildNode().size(); i++)
                mychild = T.getListChildNode().get(i);
                removeLink(mychild, item1, item2);
```

Ví dụ: Ta sử dụng bảng 3.1 và cấu trúc cây ở hình 3.3

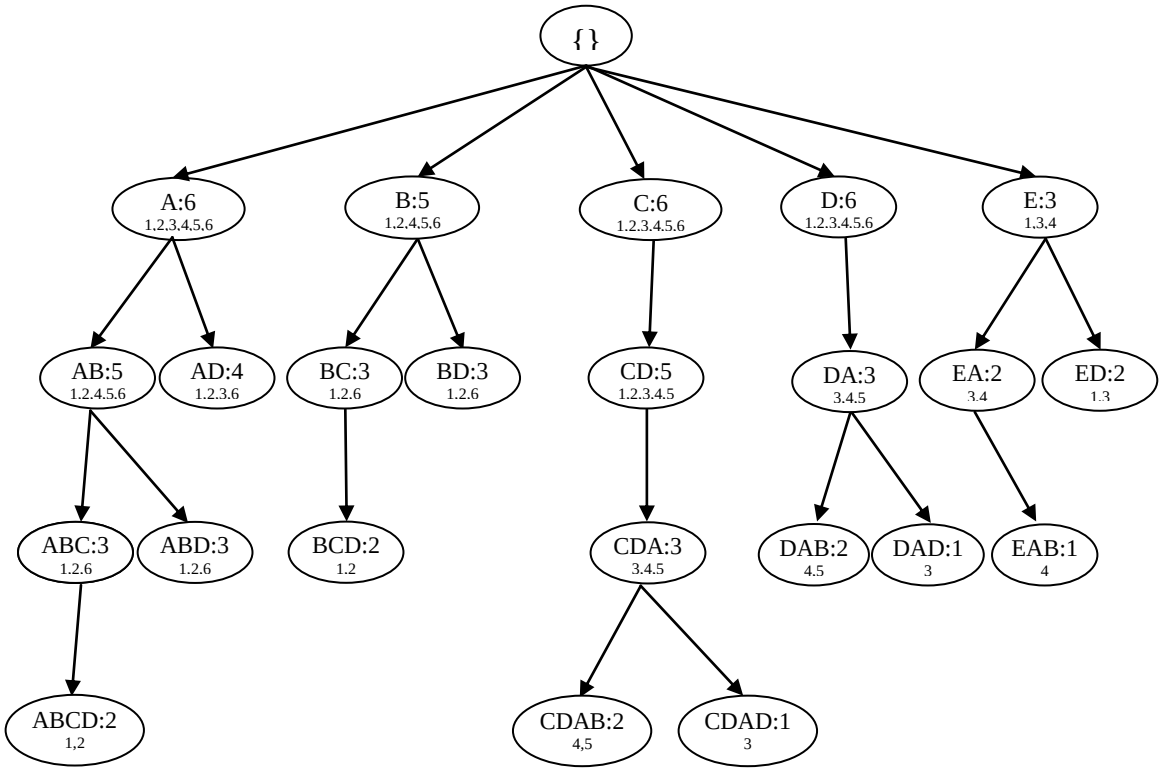
Trường hợp thứ nhất: Xóa liên kết từ trang Web C đến trang Web E. Thuật toán RemoveLink sẽ duyệt qua cấu trúc cây một lần và xóa tất cả các mô hình duyệt Web chứa <AB>. Do các node con của A, B, D, E không có mô hình chứa <AB>, nên sau khi xử lý ở các node này cấu trúc cây không thay đổi. Trong trường hợp này cấu trúc cây chỉ thay đổi khi xử lý ở node A,B,C, và <ABCE>, <BCE>, <CE>, <CEA>, <CED>, <CEAB> là các mô hình duyệt Web bị xóa khỏi cấu trúc cây.



Hình 3.12: *Cập nhật cấu trúc cây sau khi xóa liên kết từ C→E tại node A và B*

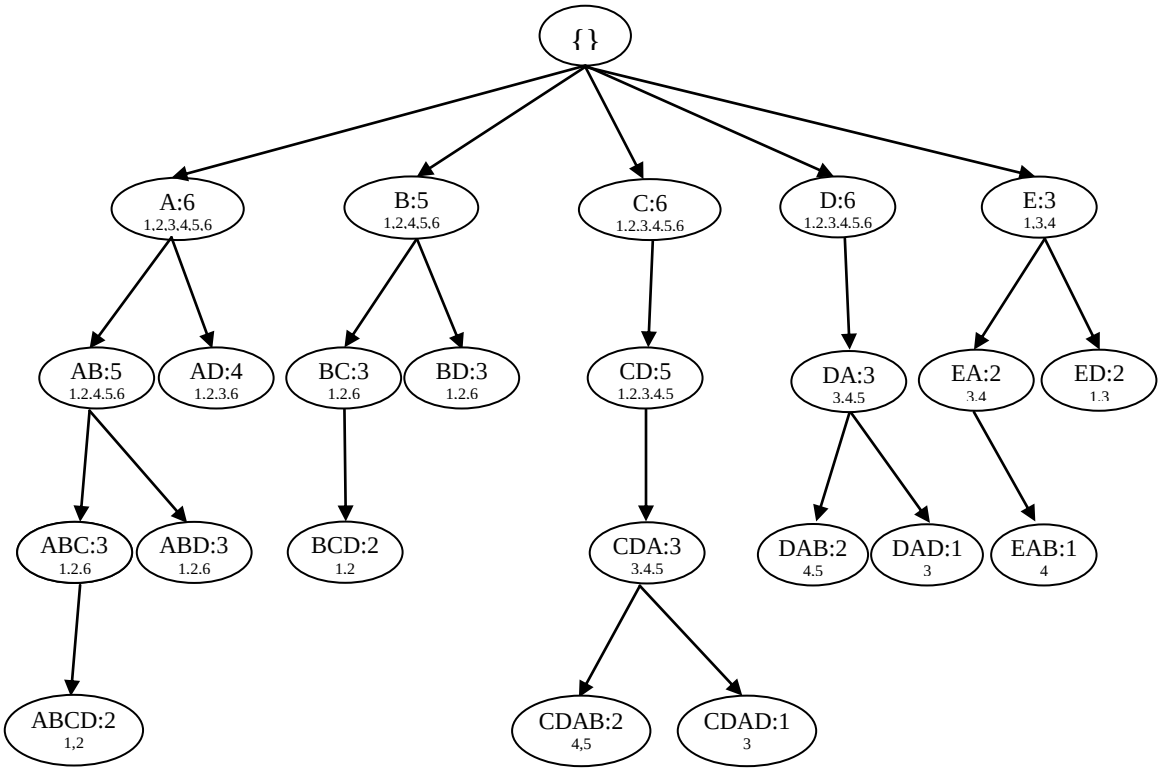


Hình 3.13: *Cập nhật cấu trúc cây sau khi xóa liên kết từ C → E tại node C*

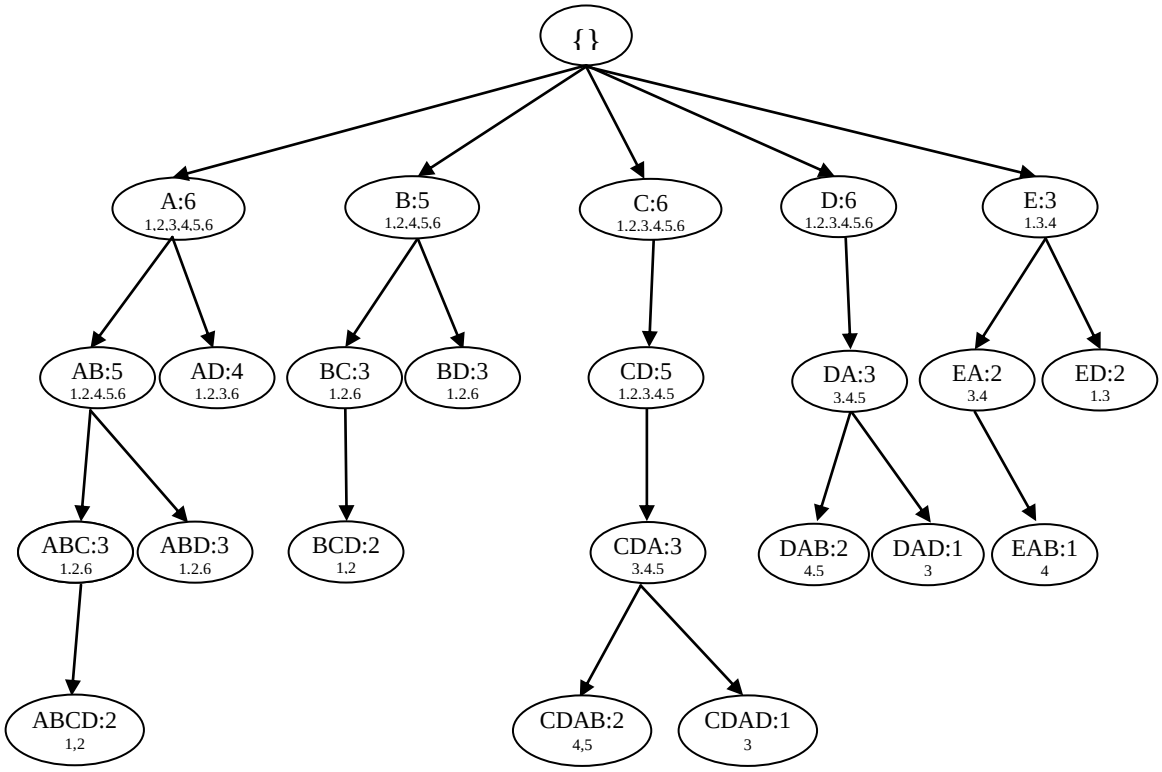


Hình 3.14: *Cập nhật cấu trúc cây sau khi xóa liên kết từ C → E tại node D và E*

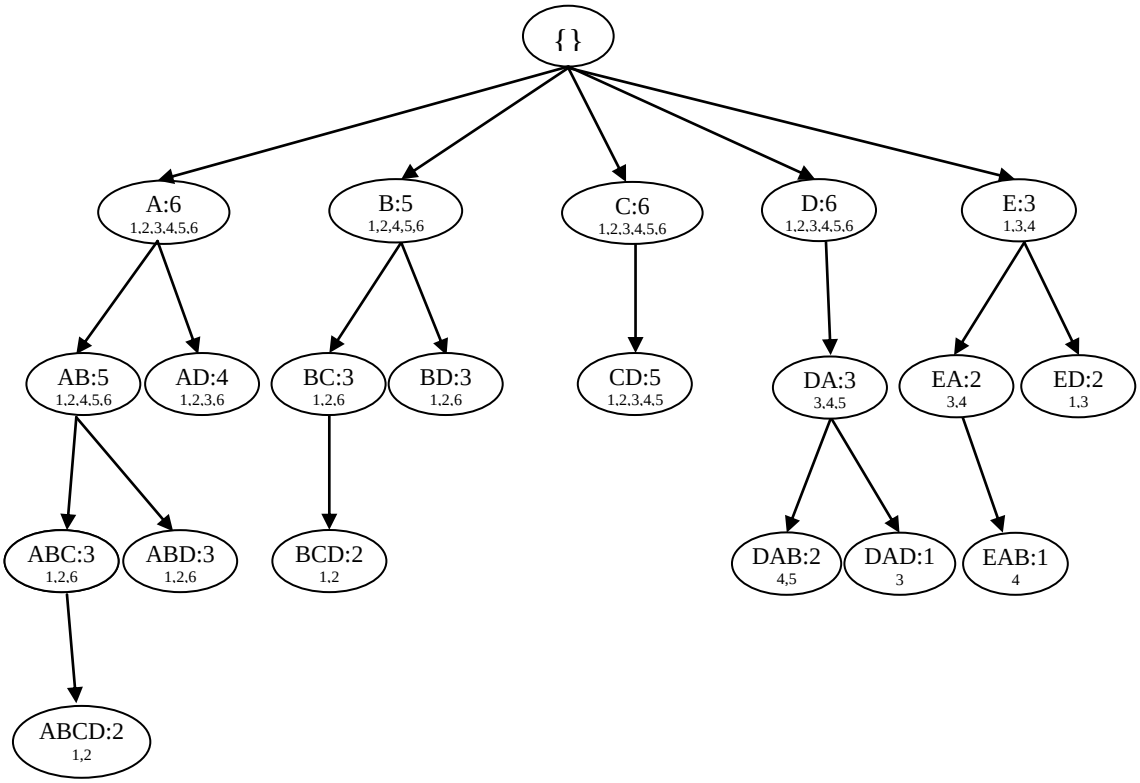
Trường hợp thứ hai: Tiếp tục xóa liên kết từ trang Web D đến trang Web A, Thuật toán RemoveLink cũng duyệt qua cấu trúc cây một lần và xóa tất cả các mô hình duyệt Web chứa <DA>. Do các node con của A, B, E không có mô hình chứa <DA>, nên sau khi xử lý ở các node này cấu trúc cây không thay đổi. Trong trường hợp này cấu trúc cây chỉ thay đổi khi xử lý ở node C, node D, và <CDA>, <CDAB>, <CDAD>, <DA>, <DAB>, <DAD> là các mô hình duyệt Web bị xóa khỏi cấu trúc cây.



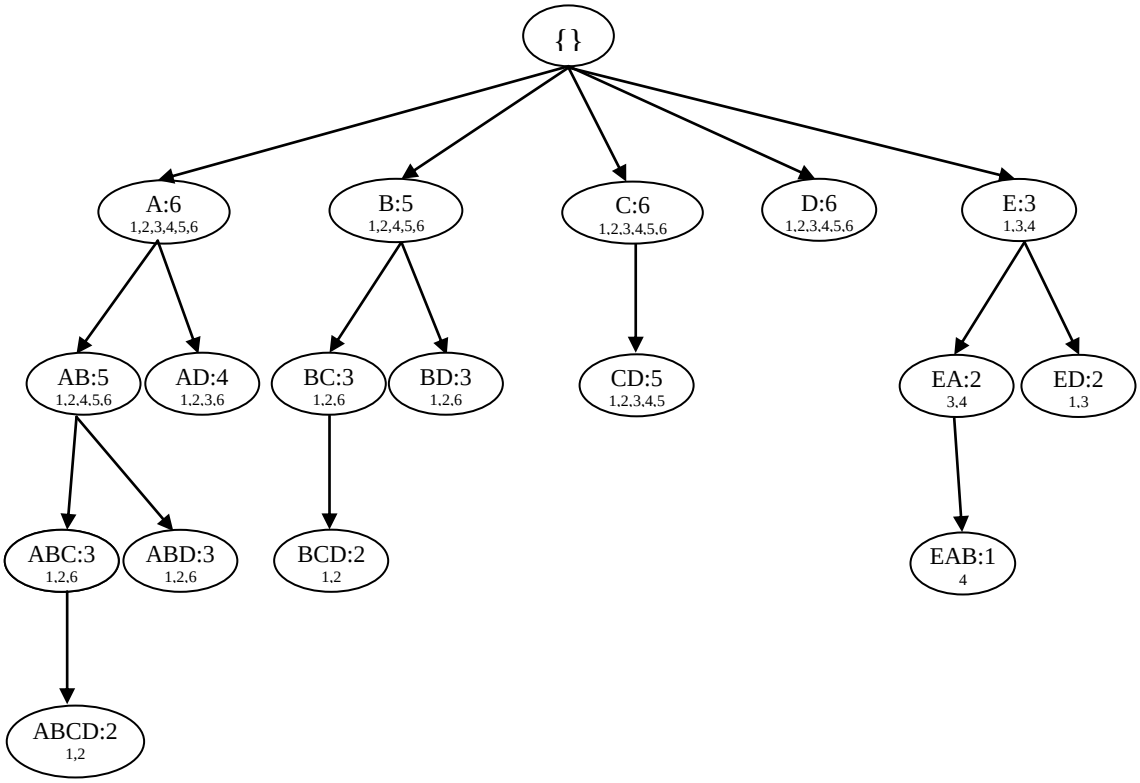
Hình 3.15: *Cập nhật cấu trúc cây sau khi xóa liên kết từ D → A tại node A*



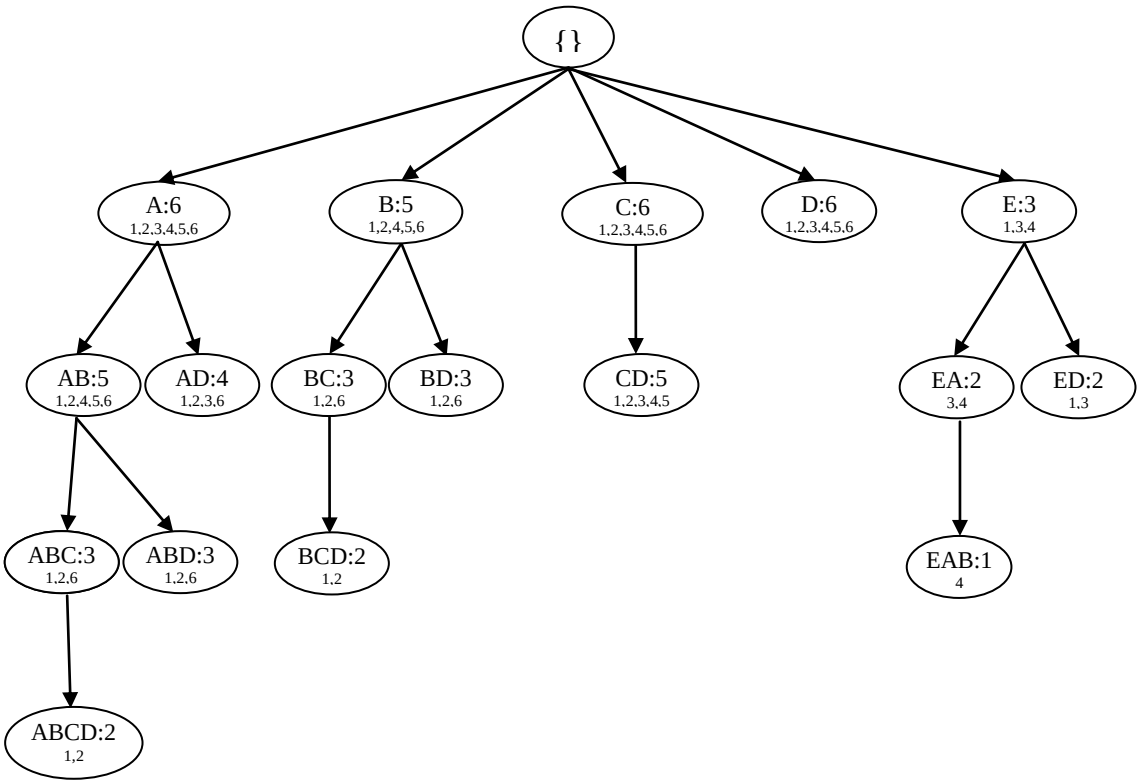
Hình 3.16: *Cập nhật cấu trúc cây sau khi xóa liên kết từ D → A tại node B*



Hình 3.17: *Cập nhật cấu trúc cây sau khi xóa liên kết từ D → A tại node C*



Hình 3.18: *Cập nhật cấu trúc cây sau khi xử xóa liên kết từ D → A tại node D*



Hình 3.19: *Cập nhật cấu trúc cây sau khi xử xóa liên kết từ D → A tại node E*

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Môi trường thực nghiệm

❖ **Phần cứng**

Cấu hình máy tính:

- Processor là Intel(R) Core(TM) i5-2410M CPU @ 2.30 GHz (4CPUs), ~ 2.3GHz
- Memory: 4 GB RAM

❖ **Phần mềm**

- Hệ điều hành Windows 7 Home Premium 64-bit (6.1, Build 7601).
- Ngôn ngữ Java.

4.2 Giới thiệu cơ sở dữ liệu thực nghiệm

4.2.1 thực nghiệm thứ nhất

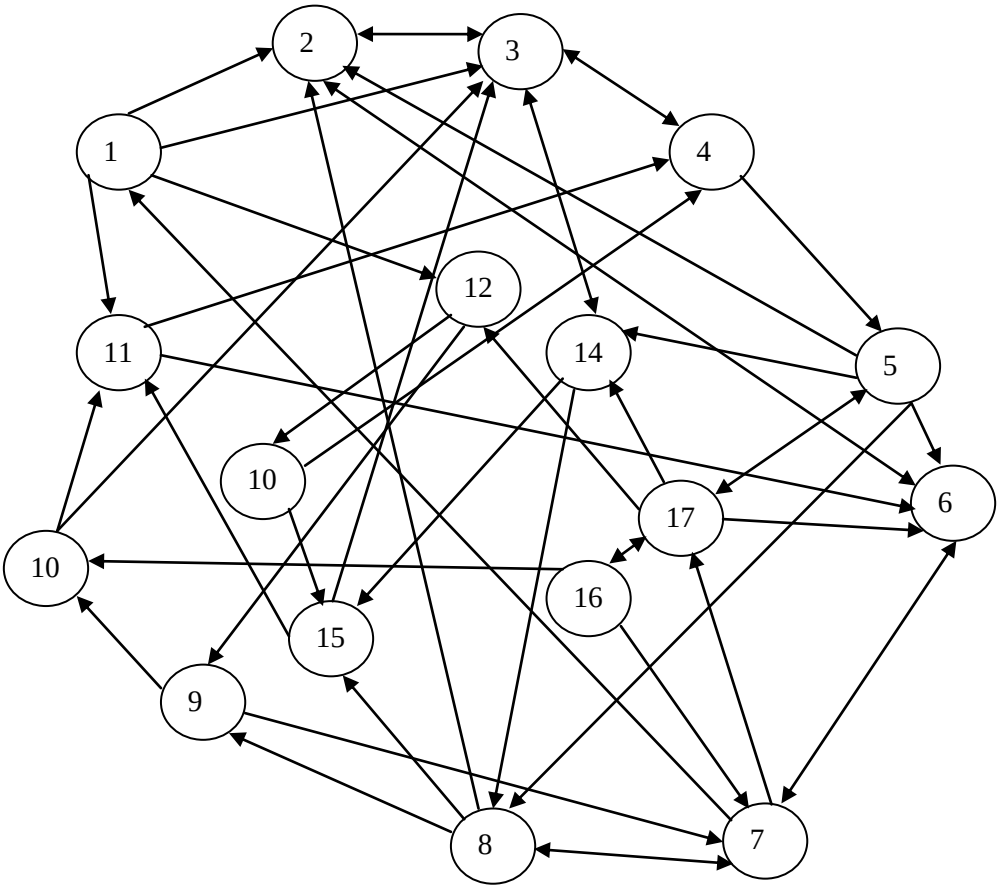
Trong thực nghiệm thứ nhất, chạy thuật toán WebTP, thuật toán IntWebTP, thuật toán RemoveLink với bộ dữ liệu MSNBC.txt bao gồm 31,790 trình tự duyệt Web, và 17 trang Web, mật độ trung bình của mỗi trình tự là 13.33 Items, và mật độ trung bình của trang Web trong mỗi trình tự là 5.33. Với cấu trúc trang Web được tổ chức như hình 3.7.

Từ dữ liệu ban đầu thêm lần lượt 1K, 2K, 3K, 4K, 5K,6K, 7K, 8Kvới min_sup được thiết lập lần lượt 5 %, 3 %, 1 %. Biểu đồ 4.1 thể hiện thời gian thực hiện thuật toán WebTP được tính bằng giây.

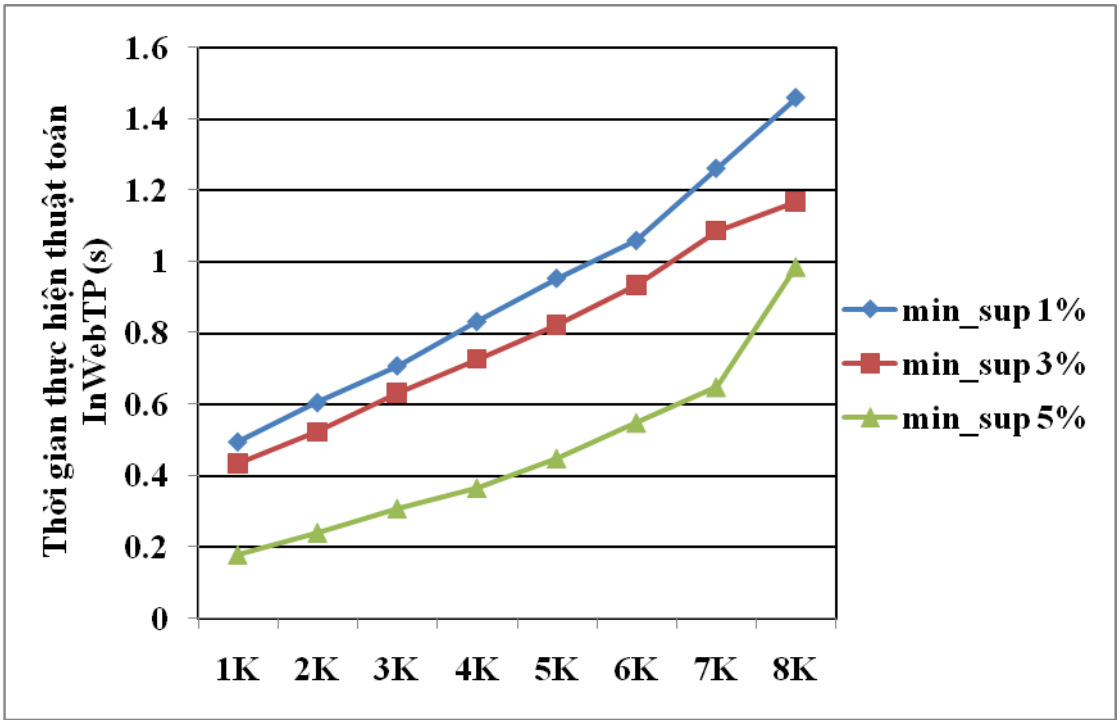
Từ dữ liệu ban đầu xóa lần lượt 1K, 2K, 3K, 4K, 5K,6K, 7K, 8Kvới min_sup được thiết lập lần lượt 5 %, 3 %, 1 %. Biểu đồ 4.1 thể hiện thời gian thực hiện thuật toán WebTP được tính bằng giây.

Từ CSDL ban đầu lần lượt chia ra các bộ 30K, 20K và 10K với min_sup được thiết lập lần lượt 2.5 %, 2 %, 1.5 %, 1 %, 0.5 %. Biểu đồ 4.2 thể hiện thời gian thực hiện thuật toán RemoveLink được tính bằng giây.

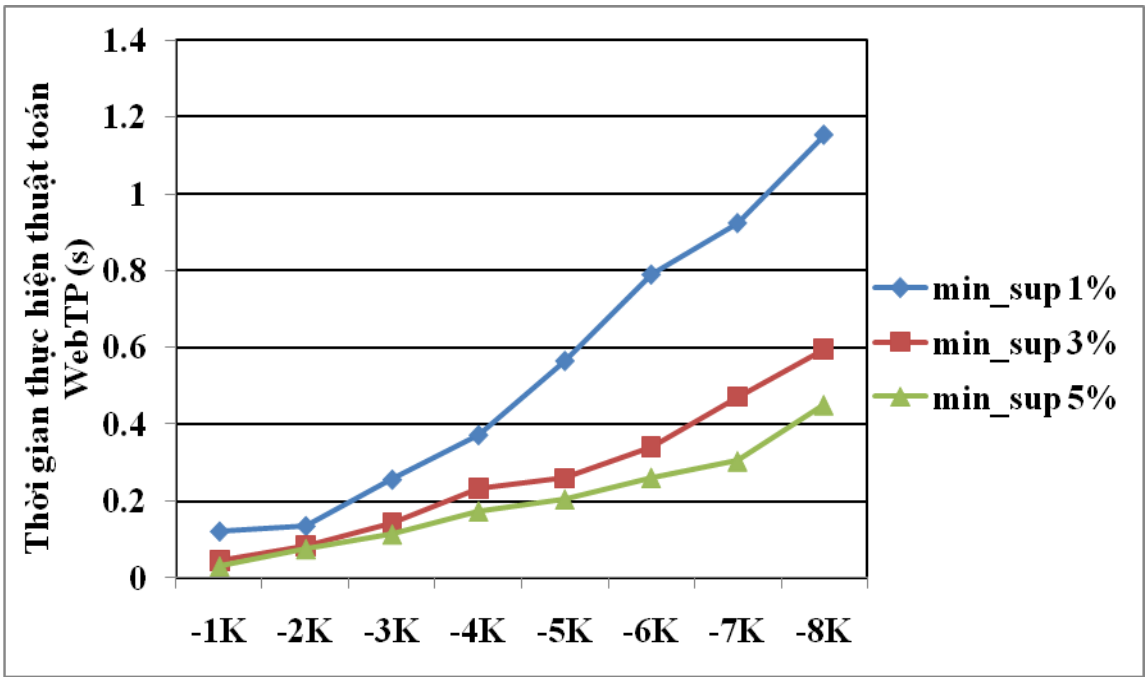
Từ CSDL ban đầu lần lượt chia ra các bộ 30K, 20K và 10K với min_sup được thiết lập lần lượt 2.5 %, 2 %, 1.5 %, 1 %, 0.5 %. Biểu đồ 4.3 thể hiện thời gian thực hiện thuật toán IntWebTP được tính bằng giây.



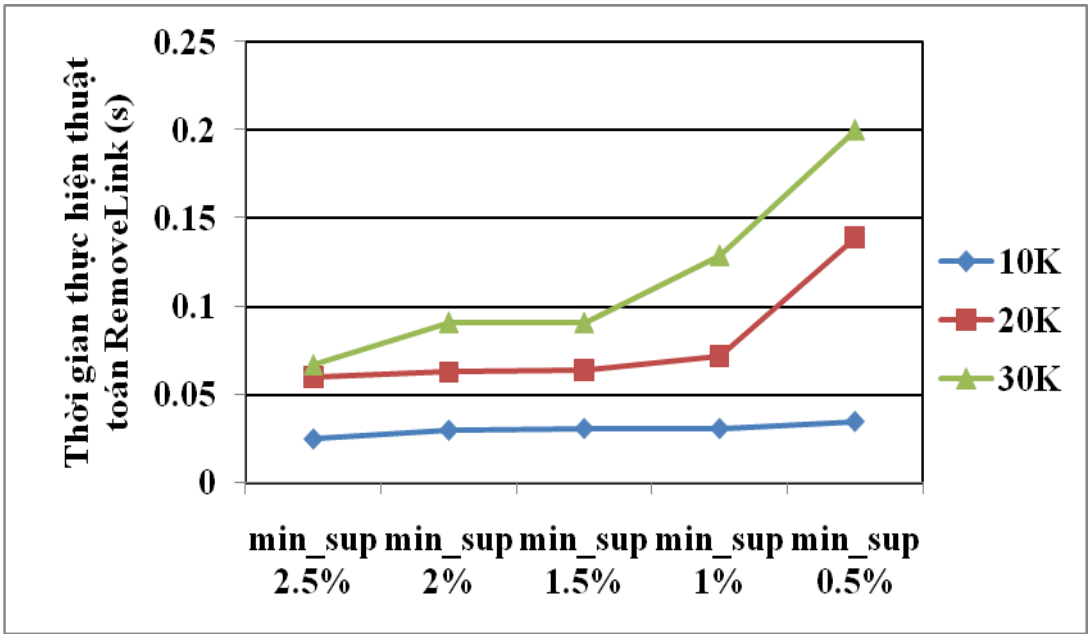
Hình 4.1: *Cấu trúc Website gồm 17 trang Web*



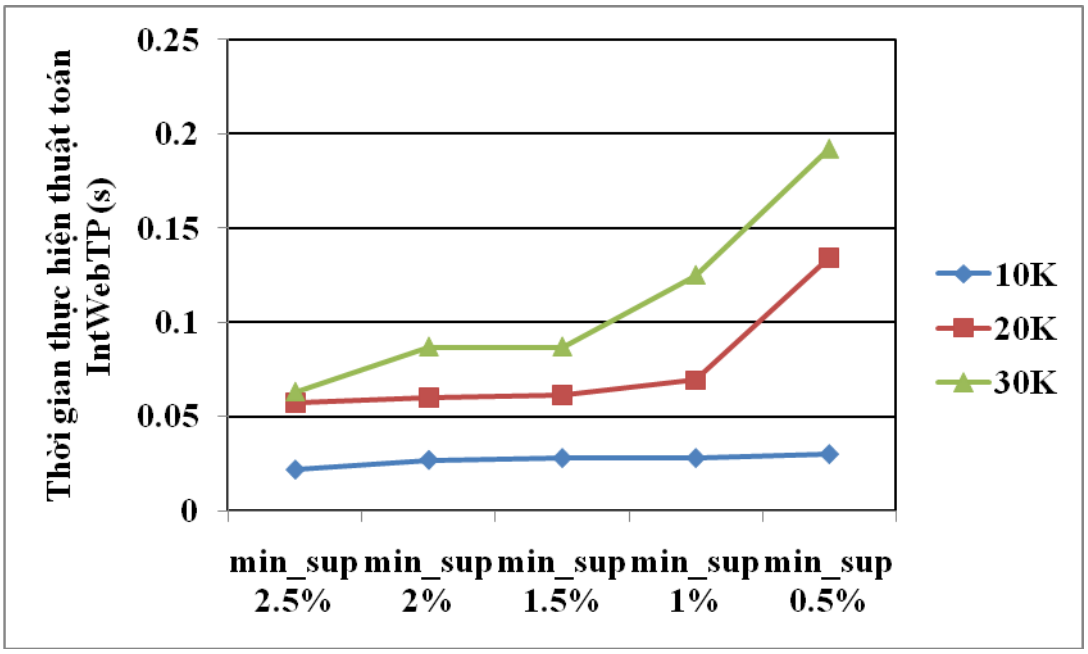
Hình 4.2: *Biểu đồ thời gian thực hiện thuật toán InWebTP khi các TIDs thêm tăng*



Hình 4.3: *Biểu đồ thời gian thực hiện thuật toán WebTP khi xóa các TIDs tăng*



Hình 4.4: *Biểu đồ thời gian thực hiện thuật toán RemoveLink*



Hình 4.5: *Biểu đồ thời gian thực hiện thuật toán IntWebTP*

4.2.2 Thực nghiệm thứ hai

Trong thực nghiệm thứ hai, chạy thuật toán WebTP, thuật toán IntWebTP, thuật toán RemoveLink với bộ dữ liệu BMSWebView1.txt bao gồm 59,601 trình

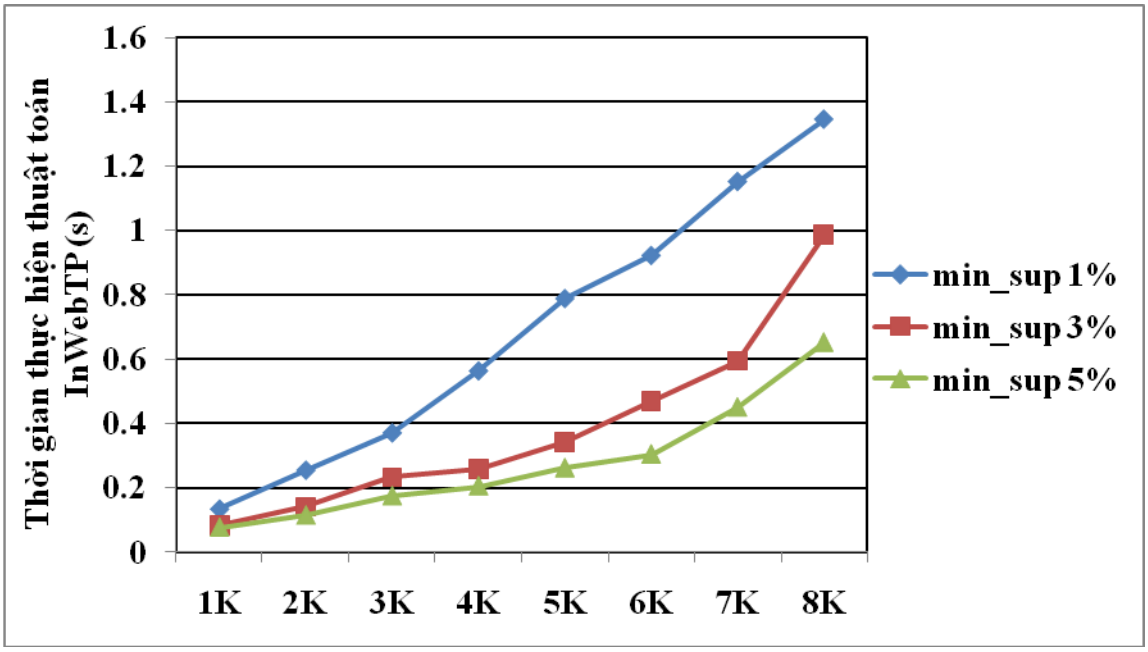
tự duyệt Web, và 497 trang Web, mật độ trung bình của mỗi trình tự là 2.42 Items, có một số trình tự dài bao gồm hơn 20 Items.

Từ CSDL ban đầu thêm lần lượt 1K, 2K, 3K, 4K, 5K, 6K, 7K và -8K với min_sup được thiết lập lần lượt 5 %, 3 %, 1 %. Biểu đồ 4.4 thể hiện thời gian thực hiện thuật toán InWebTP được tính bằng giây.

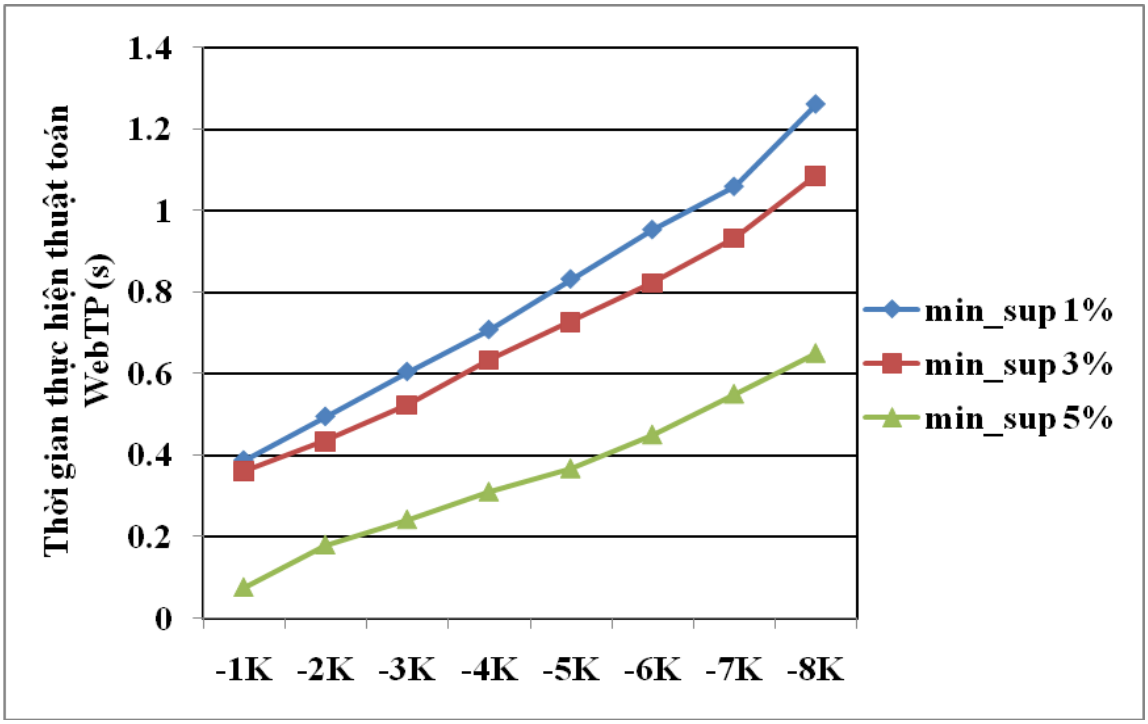
Từ CSDL ban đầu xóa lần lượt -1K, -2K, -3K, -4K, -5K, -6K, -7K và -8K với min_sup được thiết lập lần lượt 5 %, 3 %, 1 %. Biểu đồ 4.4 thể hiện thời gian thực hiện thuật toán WebTP được tính bằng giây.

Từ CSDL ban đầu lần lượt chia ra các bộ 50K, 30K và 10K với min_sup được thiết lập lần lượt 2 %, 1.5 %, 1 %, 0.5 %. Biểu đồ 4.6 thể hiện thời gian thực hiện thuật toán RemoveLink được tính bằng giây.

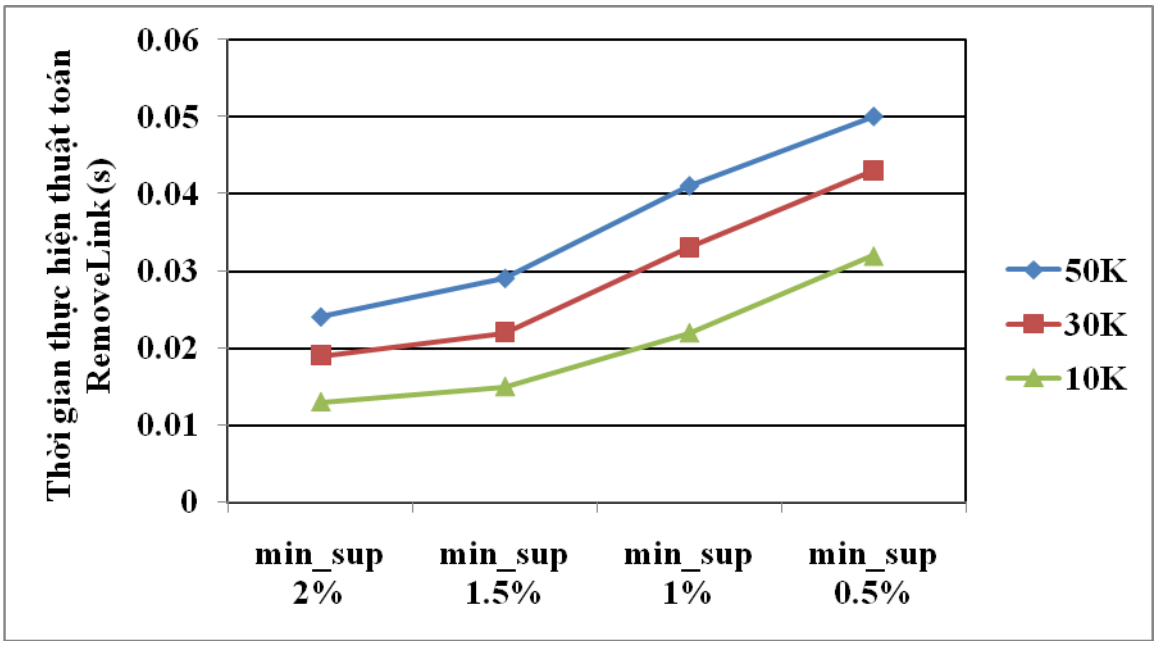
Từ CSDL ban đầu lần lượt chia ra các bộ 50K, 30K và 10K với min_sup được thiết lập lần lượt 2.5 %, 2 %, 1.5 %, 1 %, 0.5 %, 1 %. Biểu đồ 4.5 thể hiện thời gian thực hiện thuật toán IntWebTP khi điều chỉnh min_sup được tính bằng giây.



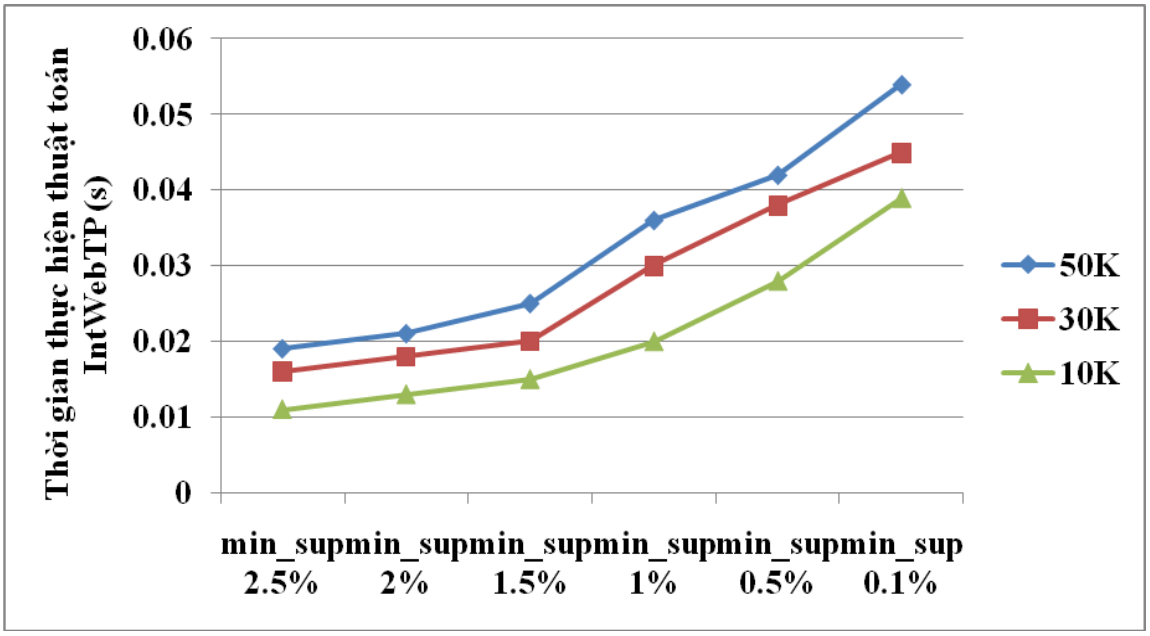
Hình 4.6: Biểu đồ thời gian thực hiện thuật toán InWebTP khi các TIDs thêm tăng



Hình 4.7: *Biểu đồ thời gian thực hiện thuật toán WebTP khi các TIDs bị xóa tăng*



Hình 4.8: *Biểu đồ thời gian thực hiện thuật toán RemoveLink*



Hình 4.9: Biểu đồ thời gian thực hiện thuật toán IntWebTP khi điều chỉnh min_sup giảm dần

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Luận văn đã hoàn thành các mục tiêu, nội dung và phương pháp nghiên cứu được đề ra. Bên cạnh đó, luận văn cũng thực nghiệm các thuật toán WebTP, thuật toán IntWebTP và thuật toán RemoveLink trên các bộ dữ liệu chuẩn để đánh giá hiệu quả của của các thuật toán.

Với đề tài: "*Thuật Toán Hiệu Quả Cho Khai Thác Tăng Trưởng Của Mô hình Duyệt Web*" tuy chưa giải quyết tất cả các vấn đề tồn đọng, song nếu có thời gian phát triển và hoàn thiện hơn thì đề tài này sẽ giúp ích rất nhiều trong việc khai thác sử dụng Web. Tuy nhiên luận văn đã đóng góp một số nội dung cho lĩnh vực khai thác sự dụng Web, cụ thể là khai thác mô hình truy cập Web như sau:

- Nghiên cứu thuật toán IncWPT, thuật toán ISL, thuật toán IncSpan, và thuật toán sinh ứng viên trình tự duyệt Web (CandidateGen).
- Nghiên cứu phương pháp sinh ứng viên của các mô hình duyệt Web dựa vào cấu trúc trang Web.
- Nghiên cứu các bài toán được đặt ra trong khai thác Web nói chung, và cụ thể là bài toán khai thác mô hình duyệt Web.
- Thực nghiệm khảo sát thời gian thực hiện các thuật thuật toán trên các bộ dữ liệu chuẩn.

5.2 Nhận xét

❖ **Ưu điểm:**

- Luận văn được trình bày một cách khoa học và có hệ thống những kiến thức hiểu biết của bản thân, có tham khảo các tài liệu về các vấn đề có liên quan đến nội dung tìm hiểu, nghiên cứu.
- Luận văn đã trình bày chi tiết các thuật toán và ví dụ cụ thể cho từng thuật toán.

- Chạy thực nghiệm các thuật toán trên các bộ dữ liệu: MSNBC[6], BMSWebView[6].

❖ **Nhược điểm**

- Số lượng các trang Web và các trình tự người dùng tham gia vào các trang Web trên thế giới sẽ tiếp tục phát triển. Vì vậy, các cấu trúc cây có thể trở nên quá lớn để được nạp vào bộ nhớ.
- Luận văn được thực hiện trong một thời gian ngắn nên không thể tránh khỏi những sai sót, rất mong được sự đóng góp ý kiến của các thầy cô để bài luận văn được hoàn thiện hơn.

5.3 Hướng phát triển

Chạy thực nghiệm thuật toán trên bộ dữ liệu đủ lớn để kiểm tra không gian lưu trữ của cấu trúc cây, từ đó tìm giải pháp tốt nhất để phân vùng cấu trúc cây, để có thể giảm bớt không gian lưu trữ cần thiết, lần lượt cho phép tất cả các thông tin cho mỗi phân vùng để dễ dàng nạp vào bộ nhớ.

[6] http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

Tài liệu tham khảo

Sách:

[1] B. Liu (2011). Web Data Ming. 2nd ed., Springer Publishing Company, Heidelperg Dordrecht London New York.

[2] Hà Quang Thụy, Phan Xuân Hiếu, Đoàn Sơn, Nguyễn Trí Thành, Nguyễn Thu Trang, Nguyễn Cẩm Tú.(eds) (2009). Giáo trình khai phá dữ liệu Web. Nhà xuất bản giáo dục, Việt Nam.

Bài báo:

[3] B. Vo, T. Le, T.-P. Hong, B. Le, Fast Updated Frequent-itemset Lattice for Transaction Deletion. Data and Knowledge Engineering, 2015, pp. 96-97, 78-89.

[4] M.S.Chen, J.S.Park, P.S.Yu, Effcient data mining for path traversal patterns in a web environment, IEEE Trans. Knowl. Data Eng. 10 (2) (1998) 209–221.

[5] H.Cheng, X.Yan, J.Han, IncSpan: Incremental mining of sequential patterns in large database, in: Proceedings of International Conference on Knowledge Discovery and Data Mining, 2004, pp. 527–532.

[6] Y.S. Lee, S.J. Yen, G.H. Tu, M.C. Hsieh, Web usage mining: Integrating path traversal patterns and association rules, in: Proceedings of International Conference on Informatics, Cybernetics, and Systems, 2003, pp. 1464–1469.

[7] Y.S. Lee, S.J. Yen, G.H.Tu, M.C.Hsieh, Mining traveling and purchasing behaviors of customers in electronic commerce environment, in: Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004, pp. 227–230.

[8] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, PrefixSpan: Mining sequential patterns effciently by prefix-projected pattern growth, in: Proceeding of International Conference on Data Engineering, 2001, pp. 215–224.

[9] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, Mining sequential patterns by pattern-growth: The PrefixSpan approach, IEEE Trans. Know. Data Eng. 16 (10) (2004) 1–17.

[10] S.Parthasarathy, M.J. Zaki, M. Ogihara, S. Dwarkadas, Incremental and interactive sequence mining, in: Proceedings of International Conference on Information and Knowledge Management, 1999, pp. 251–258.

[11] S.J. Yen, An effcient approach for analyzing user behaviors in a web-based training environment, Int. J. Dist. Edu. Technol. 1 (4) (2003) 55–71.

[12] S.J. Yen, Y.S. Lee, C.W. Cho, Effcient approach for the maintenance of path traversal patterns, in: Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004, 207–214.

[13] M. Zaki, SPADE: An effcient algorithm for mining frequent sequences, Mach. Learn. 40 (1–2) (2001) 31–60.

[14] S.J. Yen, Y.S. Lee, Incremental and interactive mining of web traversal patterns, Inform. Sci. 178 (2008) 287–306.