TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**LỚP CỬ NHÂN TÀI NĂNG**

NGUYỄN HOÀN – HOÀNG XUÂN THẢO

NGHIÊN CỨU VÀ PHÁT TRIỂN

hỆ thỐng rút trích ontology

tỪ web

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

TP.HCM, 2010

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**LỚP CỬ NHÂN TÀI NĂNG**

NGUYỄN HOÀN 0612109

HOÀNG XUÂN THẢO 0612416

NGHIÊN CỨU VÀ PHÁT TRIỂN

hỆ thỐng rút trích ontology

tỪ web

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN TIN HỌC

GIÁO VIÊN HƯỚNG DẪN

TS.TrẦn Minh TriẾt

NIÊN KHÓA 2006– 2010

|  |
| --- |
| **NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.  TpHCM, ngày …… tháng …… năm 2010  Giáo viên hướng dẫn |
| **NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.  TpHCM, ngày …… tháng …… năm 2010  Giáo viên phản biện |

LỜI CÁM ƠN

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Học Khoa Học Tự Nhiên, Tp.HCM đã tạo điều kiện tốt cho chúng em thực hiện đề tài này.

Chúng em xin chân thành cảm ơn Thầy Trần Minh Triết, là người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt thời gian thực hiện đề tài. Chúng em cũng xin cảm ơn Thầy Lương Vĩ Minh, Thầy Nguyễn Đức Huy đã có những trao đổi, những chỉ dẫn giúp chúng em giải quyết các vấn đề và hoàn thiện đề tài.

Chúng em cũng xin gửi lời cảm ơn sâu sắc đến quý Thầy Cô trong Khoa đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quí báu trong những năm học vừa qua.

Chúng em xin gửi lòng biết ơn sâu sắc đến Ba, Mẹ, các anh chị và bạn bè đã ủng hộ, giúp đỡ và động viên chúng em trong những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu.

Mặc dù chúng em đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý Thầy Cô và các bạn.

Nhóm thực hiện

Nguyễn Hoàn & Hoàng Xuân Thảo

ĐỀ CƯƠNG CHI TIẾT

|  |
| --- |
| **Tên Đề Tài**: Nghiên cứu và phát triển hệ thống rút trích ontology từ web |
| **Giáo viên hướng dẫn**: TS. Trần Minh Triết |
| **Thời gian thực hiện**: từ ngày 25/02 /2010 đến ngày 15/07 /2010 |
| **Sinh viên thực hiện:**  Nguyễn Hoàn (0612109) – Hoàng Xuân Thảo (0612416) |
| **Loại đề tài:** Nghiên cứu lý thuyết, giải pháp kỹ thuật và xây dựng framework hiện thực hóa |
| **Nội Dung Đề Tài** (mô tả chi tiết nội dung đề tài, yêu cầu, phương pháp thực hiện, kết quả đạt được, …):   * Nghiên cứu tổng quan về ontology cũng như các hướng tiếp cận trong việc xây dựng ontology, trong đó đặc biệt quan tâm đến hướng tiếp cận rút trích ontology từ dữ liệu web * Phân tích quy trình 6 bước để rút trích ontology do nhóm tác giả Du C. Timon, Li Feng, và King Irwin đề xuất năm 2009. * Xây dựng Ontology Extractor Framework để hiện thực hóa quy trình này và đề xuất những cải tiến minh họa cho framework * Hiện thực hóa các module cụ thể để triển khai vào framework này. |

|  |  |
| --- | --- |
| Kế Hoạch Thực Hiện:  25/02/2010-15/03/2010: Tìm hiểu ontology.  16/03/2010-30/03/2010: Tìm hiểu các phương pháp xây dựng ontology.  01/04/2010-15/04/2010: Nghiên cứu và phân tích quy trình 6 bước rút trích ontology từ dữ liệu web.  16/04/2010-29/04/2010: Đưa ra kiến trúc tổng quát cho hệ thống  02/05/2010-20/05/2010: Đưa ra các giải pháp cho các vấn đề và lựa chọn giải pháp thích hợp để xây dựng framework theo kiến trúc đã đưa ra.  21/05/2010-15/06/2010: Đưa ra các giải pháp cho các vấn đề và lựa chọn giải pháp thích hợp để hiện thực hóa các module cụ thể để sử dụng trong framework.  15/06/2010-30/06/2010: Xây dựng hoàn chỉnh framework.  01/07/2010-15/07/2010: Tiến hành chạy thử nghiệm. | |
| **Xác nhận của GVHD**  **TS. Trần Minh Triết** | **Ngày tháng năm** 2010  **Nhóm SV Thực hiện**  **Nguyễn Hoàn – Hoàng Xuân Thảo** |

MỤC LỤC

[Chương 1 Mở đầu 1](#_Toc267996375)

[1.1. Giới thiệu ontology 1](#_Toc267996376)

[1.2. Các hướng tiếp cận xây dựng ontology 2](#_Toc267996377)

[1.3. Mục tiêu đề tài 3](#_Toc267996378)

[1.4. Nội dung luận văn 4](#_Toc267996379)

[Chương 2 Tổng quan Ontology 5](#_Toc267996380)

[2.1. Giới thiệu 5](#_Toc267996381)

[2.2. Định nghĩa 5](#_Toc267996382)

[2.3. Phân loại 6](#_Toc267996383)

[2.4. Ngôn ngữ biểu diễn ontology 7](#_Toc267996384)

[2.5. Ứng dụng 9](#_Toc267996385)

[2.5.1. Tìm Kiếm Thông Tin và Quản Lý Tri Thức 9](#_Toc267996386)

[2.5.2. Thương mại Điện tử 11](#_Toc267996387)

[2.5.3. Web ngữ nghĩa 12](#_Toc267996388)

[2.6. Kết luận 18](#_Toc267996389)

[Chương 3 Các hướng tiếp cận trong việc xây dựng ontology 19](#_Toc267996390)

[3.1. Các nguồn dữ liệu dùng để xây dựng ontology 19](#_Toc267996391)

[3.2. Xây dựng ontology 20](#_Toc267996392)

[3.3. Phương pháp dựa trên việc xử lý ngôn ngữ tự nhiên 20](#_Toc267996393)

[3.4. Phương pháp dựa vào thống kê 22](#_Toc267996394)

[3.5. Phương pháp máy học 25](#_Toc267996395)

[3.6. Phương pháp kết hợp 27](#_Toc267996396)

[Chương 4 Quy trình rút trích ontology từ WWW 29](#_Toc267996397)

[4.1. Mở đầu 29](#_Toc267996398)

[4.2. Một số giả định 30](#_Toc267996399)

[4.3. Quy trình 6 bước rút trích ontology từ WWW 31](#_Toc267996400)

[4.4. Chuẩn bị 32](#_Toc267996401)

[4.5. Biến đổi 34](#_Toc267996402)

[4.6. Gom cụm 37](#_Toc267996403)

[4.7. Nhận diện 38](#_Toc267996404)

[4.8. Liên kết 39](#_Toc267996405)

[4.9. Tinh chỉnh 40](#_Toc267996406)

[4.10. Kết luận 40](#_Toc267996407)

[Chương 5 Ontology Extractor Framework rút trích ontology từ WWW 41](#_Toc267996408)

[5.1. Kiến trúc hệ thống 41](#_Toc267996409)

[5.2. Phân hệ Chuẩn bị (Preparation) 44](#_Toc267996410)

[5.2.1. Kiến trúc phân hệ 44](#_Toc267996411)

[5.2.2. Tải các trang web về lưu trữ ngoại tuyến 46](#_Toc267996412)

[5.2.3. Loại bỏ trang web không hợp lệ 48](#_Toc267996413)

[5.2.4. Chuẩn hoá các trang web 49](#_Toc267996414)

[5.3. Phân hệ Biến đổi (Transformation) 51](#_Toc267996415)

[5.3.1. Kiến trúc phân hệ 51](#_Toc267996416)

[5.3.2. Gom nhóm các trang web dựa trên đường dẫn gốc 54](#_Toc267996417)

[5.3.3. Tinh chỉnh số lượng các trang web trong một nhóm 56](#_Toc267996418)

[5.3.4. Loại trừ các thành phần lặp giữa 2 trang web 56](#_Toc267996419)

[5.3.5. Loại trừ các phần trùng nhau giữa các trang web trong cùng một nhóm 60](#_Toc267996420)

[5.3.6. Lưu trữ các trang web vào hệ quản trị cơ sở dữ liệu 61](#_Toc267996421)

[5.4. Phân hệ Gom cụm (Instance Clustering) 63](#_Toc267996422)

[5.4.1. Kiến trúc phân hệ 63](#_Toc267996423)

[5.4.2. Lựa chọn các từ khoá 66](#_Toc267996424)

[5.4.3. Xây dựng vector đặc trưng 67](#_Toc267996425)

[5.4.4. Gom cụm các trang web dựa trên độ tương đồng giữa các vector đặc trưng 68](#_Toc267996426)

[5.5. Phân hệ Nhận diện (Recognition) 69](#_Toc267996427)

[5.5.1. Kiến trúc phân hệ 69](#_Toc267996428)

[5.5.2. Nhận diện đặc trưng cụm 71](#_Toc267996429)

[5.6. Phân hệ Liên kết (Refinement) 72](#_Toc267996430)

[5.6.1. Kiến trúc phân hệ 72](#_Toc267996431)

[5.6.2. Xây dựng đồ thị mối quan hệ giữa các khái niệm 73](#_Toc267996432)

[5.7. Phân hệ Tinh chỉnh (Revision) 75](#_Toc267996433)

[5.7.1. Kiến trúc phân hệ 75](#_Toc267996434)

[Chương 6 Kết luận 77](#_Toc267996435)

[6.1. Các kết quả đạt được 77](#_Toc267996436)

[6.2. Hướng phát triển của đề tài 78](#_Toc267996437)

DANH MỤC CÁC HÌNH

[Hình 1‑1 Một ontology trong lĩnh vực về Rượu 1](#_Toc269161682)

[Hình 2‑1 Đồ thị của một phần ontology WordNet [16] 7](#_Toc269161683)

[Hình 2‑2 Các ngôn ngữ đánh dấu ontology [21] 8](#_Toc269161684)

[Hình 2‑3 Ba lĩnh vực ứng dụng của ontology [8] 9](#_Toc269161685)

[Hình 2‑4 Kiến trúc chung của hệ thống OntoBroker [28] 10](#_Toc269161686)

[Hình 2‑5 Một đoạn trang web được gán nhãn bằng OntoBroker [28] 11](#_Toc269161687)

[Hình 2‑6 Market place sử dụng ontology [10] 12](#_Toc269161688)

[Hình 2‑7 Mô hình phương pháp kết hợp web service [39] 16](#_Toc269161689)

[Hình 3‑1 Kiến trúc của Hasti [45] 22](#_Toc269161690)

[Hình 3‑2 Thiết kế chung của phương pháp [46] 23](#_Toc269161691)

[Hình 3‑3 Kiến trúc chung của hệ thống CRCTOL [49] 24](#_Toc269161692)

[Hình 3‑4 Một phần đồ thị hai phía sinh ra từ hệ thống [50] 25](#_Toc269161693)

[Hình 3‑5 Quy trình thu nhận Ontology [55] 28](#_Toc269161694)

[Hình 4‑1 Kiến trúc tổng quát của hệ thống rút trích ontology từ web [2] 31](#_Toc269161695)

[Hình 4‑2 Quy trình 6 bước rút trích ontology từ WWW [2] 32](#_Toc269161696)

[Hình 4‑3 Phân nhóm dựa trên chuỗi đường dẫn 36](#_Toc269161697)

[Hình 5‑1 Kiến trúc tổng quát Ontology Extractor Framework 41](#_Toc269161698)

[Hình 5‑2 Kiến trúc các phân hệ chính của Ontology Extractor Framework 42](#_Toc269161699)

[Hình 5‑3 Giao diện Phân hệ Chuẩn bị 44](#_Toc269161700)

[Hình 5‑4 Kiến trúc Phân hệ Chuẩn bị 45](#_Toc269161701)

[Hình 5‑5 Mô hình xử lý tổng quát Phân hệ Chuẩn bị 46](#_Toc269161702)

[Hình 5‑6 Giao diện phân hệ Biến đổi 51](#_Toc269161703)

[Hình 5‑7 Kiến trúc Phân hệ Biến đổi 52](#_Toc269161704)

[Hình 5‑8 Mô hình xử lý tổng quát Phân hệ Biến đổi 54](#_Toc269161705)

[Hình 5‑9 Các đường dẫn gốc được phân nhóm 55](#_Toc269161706)

[Hình 5‑10 Cây đường dẫn 55](#_Toc269161707)

[Hình 5‑11 So sánh và đánh dấu thành phần lặp giữa 2 cây cấu trúc HTML 58](#_Toc269161708)

[Hình 5‑12 Tính độ tương đồng cây cấu trúc HTML 59](#_Toc269161709)

[Hình 5‑13 Xpath 62](#_Toc269161710)

[Hình 5‑14 Giao diện phân hệ Gom cụm 63](#_Toc269161711)

[Hình 5‑15 Kiến trúc Phân hệ Gom cụm 64](#_Toc269161712)

[Hình 5‑16 Mô hình xử lý tổng quát Phân hệ Gom cụm 66](#_Toc269161713)

[Hình 5‑17 Giao diện phân hệ Nhận diện 69](#_Toc269161714)

[Hình 5‑18 Kiến trúc Phân hệ Nhận diện 69](#_Toc269161715)

[Hình 5‑19 Mô hình xử lý tổng quát Phân hệ Nhận diện 70](#_Toc269161716)

[Hình 5‑20 Giao diện phân hệ Liên kết 72](#_Toc269161717)

[Hình 5‑21 User Control phân hệ Liên kết 72](#_Toc269161718)

[Hình 5‑22 Mô hình xử lý tổng quát Phân hệ Liên kết 72](#_Toc269161719)

[Hình 5‑23 Giao diện phân hệ Tinh chỉnh 75](#_Toc269161720)

[Hình 5‑24 User Control phân hệ Tinh chỉnh 75](#_Toc269161721)

[Hình 5‑25 Mô hình xử lý tổng quát Phân hệ Tinh chỉnh 76](#_Toc269161722)

DANH MỤC CÁC BẢNG

[Bảng 3‑1 Tóm tắt các công trình nghiên cứu có liên quan 20](#_Toc269161665)

[Bảng 4‑1 Phân loại văn bản cùng các thẻ tương ứng 37](#_Toc269161666)

[Bảng 5‑1 Các lớp chính cấu thành phân hệ Chuẩn bị 46](#_Toc269161667)

[Bảng 5‑2 Các bộ lọc trang web 49](#_Toc269161668)

[Bảng 5‑3 Các lớp chính cấu thành phân hệ Biến đổi 53](#_Toc269161669)

[Bảng 5‑4 Các lớp chính cấu thành phân hệ Gom cụm 65](#_Toc269161670)

[Bảng 5‑5 Các bộ lọc từ khóa 67](#_Toc269161671)

[Bảng 5‑6 Các lớp chính cấu thành phân hệ Nhận diện 70](#_Toc269161672)

TÓM TẮT KHÓA LUẬN

Với sự xuất hiện mạnh mẽ của web ngữ nghĩa, khái niệm ontology ngày càng được đề cập đến nhiều hơn. Nhu cầu sử dụng ontology tăng cao không chỉ trong lĩnh vực web ngữ nghĩa mà còn trong nhiều lĩnh vực khác cần có nguồn thông tin giàu ngữ nghĩa do ontology đem lại phục vụ cho nhu cầu thực hiện các suy diễn và các thuật toán một cách tốt hơn. Do nhu cầu cao như vậy nên các nghiên cứu về phương pháp xây dựng ontology nhận được nhiều sự quan tâm trong giới nghiên cứu. Những thực tế này cho thấy nhu cầu xây dựng nên các hệ thống rút trích ontology một cách bán tự động hoặc tự động trở nên rất cấp thiết. Do đó mục tiêu của khóa luận này nhằm nghiên cứu về các vấn đề liên quan đến ontology và các phương pháp rút trích nhằm xây dựng một framework để hiện thực hóa một quy trình rút trích ontology.

Nội dung của đề tài này tập trung vào việc *Nghiên cứu và phát triển hệ thống rút trích ontology từ web*. Ngoài việc nghiên cứu các vấn đề liên quan đến ontology, khóa luận còn tập trung vào phân tích quy trình rút trích ontology từ web và xây dựng framework để hiện thực hóa quy trình này. Khóa luận còn xây dựng các plugin cụ thế để sử dụng trên framework này.

Nội dung khóa luận bao gồm 6 chương:

**Chương** 1**:** Mở đầu

**Chương** 2**:** Tổng quan Ontology

**Chương** 3**:** Các hướng tiếp cận trong việc xây dựng Ontology

**Chương** 4**:** Quy trình rút trích Ontology từ WWW

**Chương** 5**:** Ontology Extractor Framework rút trích ontology từ WWW

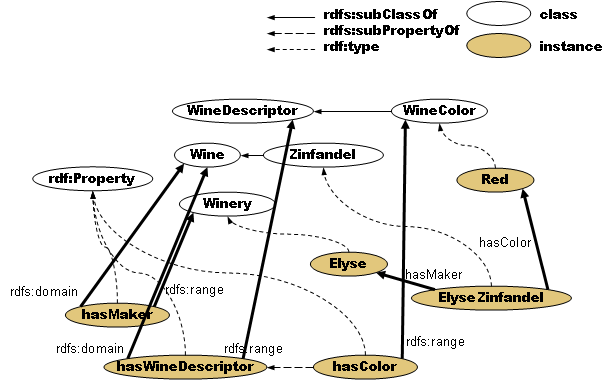
**Chương** 6**:** Kết luận

# Mở đầu

* Nội dung Chương 1 trình bày sơ lược khái niệm ontology cũng như nhu cầu cần thiết phải có ontology. Đồng thời Chương 1 cũng nêu lên mục tiêu, nội dung và ý nghĩa của đề tài.

## Giới thiệu ontology

Khái niệm “ontology” có nguồn gốc ban đầu không phải từ lĩnh vực Trí Tuệ Nhân Tạo, đây là khái niệm có nguồn gốc xuất hiện từ lĩnh vực Triết học liên quan đến ngành học về sự tồn tại. Từ những năm 1970, các nhà nghiên cứu nhận thức được việc thu thập tri thức là công việc quan trọng trong việc xây dựng các hệ thống tri thức lớn, và cũng cho rằng họ có thể xây dựng ra các ontology mới như là các mô hình tính toán phục vụ cho các dạng nhất định trong việc suy diễn tự động. Sau đó, khái niệm này được sử dụng trong lĩnh vực Trí Tuệ Nhân Tạo từ những năm 1980 [[1](#Gru09)] dùng để chỉ lý thuyết về thế giới được mô hình hóa và cũng để chỉ một trong những thành phần của những hệ thống tri thức. Khái niệm “ontology” trong Trí Tuệ Nhân Tạo được sử dụng ngày nay có thể được xem như là một thể hiện dạng chính quy của tri thức dưới dạng tập hợp các khái niệm trong một lĩnh vực và các mối quan hệ giữa các khái niệm này.



Hình 1‑1 Một ontology trong lĩnh vực về Rượu

*(Nguồn: http://www-kasm.nii.ac.jp/~koide/SWCLOS*2*/Manual/FIGURES/WineRDF.PNG)*

Nhu cầu ban đầu cần có ontology là để cung cấp các nguồn thông tin giàu ngữ nghĩa mà máy tính có thể xử lý và thao tác được, đồng thời vẫn có thể dùng ontology để chia sẻ tri thức giữa người với người và với các hệ thống khác. Hiện nay nhu cầu về ontology ngày càng tăng cao, và ontology không những phục vụ cho nhu cầu chia sẻ tri thức đơn thuần mà còn được áp dụng vào nhiều lĩnh vực khác nhau như các hệ thống Quản Lý Tri Thức, Thương Mại Điện Tử, Web Ngữ Nghĩa, Cơ Sở Dữ Liệu hay các hệ thống bảo mật, cung cấp nguồn thông tin giàu ngữ nghĩa giúp cho các hệ thống thực hiện các tác vụ với kết quả tốt hơn. Ví dụ: Ontology được tổ chức W3C đưa vào làm một trong những nền tảng xây dựng Web Ngữ Nghĩa. Ontology còn có thể dùng để gán nhãn lại các trang web, các web service hay các nguồn dữ liệu khác trên internet nhằm tăng tính hiệu quả trong việc truy xuất, tìm kiếm và khám phá dữ liệu. Ngoài ra, người ta còn sử dụng Ontology trong các hệ thống chính sách bảo mật khác nhau. Việc sử dụng ontology giúp cho các xử lý và thuật toán trong các hệ thống thuộc các lĩnh vực khác nhau có thêm các thông tin khác giàu ngữ nghĩa nhằm đưa ra kết quả tốt hơn.

## Các hướng tiếp cận xây dựng ontology

Do nhu cầu sử dụng ontology ngày càng cao, nên việc đưa ra các phương pháp khác nhau để xây dựng ontology một cách tự động hoặc bán tự động là hết sức cần thiết. Các phương pháp này giúp giảm bớt chi phí về thời gian và công sức so với việc xây dựng các ontology một cách thủ công. Nhưng mặt khác, chất lượng của các ontology thu được từ những phương pháp này phụ thuộc khá nhiều vào các thuật giải được sử dụng, nguồn dữ liệu mà thuật giải sử dụng, cũng như từng lĩnh vực mà phương pháp được áp dụng vào.

Có nhiều phương pháp xây dựng ontology đã được các tác giả nghiên cứu và phát triển. Một trong những hướng xây dựng ontology là rút trích ontology từ các nguồn dữ liệu khác nhau. Các phương pháp rút trích ontology sử dụng nhiều cách thức khác nhau từ phương pháp máy học, xử lý ngôn ngữ tự nhiên cho đến thống kê. Trong đó, phương pháp rút trích ontology dựa trên xử lý ngôn ngữ tự nhiên rút trích các khái niệm bằng cách phân tích từ vựng, cú pháp của các văn bản thuộc về một lĩnh vực nào đó, sau đó dựa vào mối quan hệ cú pháp và từ vựng để xây dựng nên mối quan hệ về mặt ngữ nghĩa giữa các khái niệm. Phương pháp rút trích ontology dựa vào việc thống kê sẽ tiến hành thống kê trên các nguồn dữ liệu để rút trích ontology. Các phương pháp sử dụng việc học máy sẽ khai thác các nguồn dữ liệu nhằm rút ra các đặc trưng của dữ liệu, các khuôn mẫu cũng như các tập luật phục vụ cho việc rút trích ontology.

Một trong những hướng tiếp cận đáng quan tâm là rút trích ontology từ dữ liệu web. Các nguồn dữ liệu được dùng trong việc rút trích ontology khá đa dạng, từ dữ liệu dạng văn bản, dữ liệu quan hệ trong các cơ sở dữ liệu quan hệ, cho đến dữ liệu từ web. Trong đó nguồn dữ liệu từ web có lợi thế là nguồn thông tin phong phú, đa dạng, và có sẵn trên internet. Đây là một kho thông tin rất lớn. Trong các phương pháp rút trích ontology, nhóm tác giả Du C. Timon, Li Feng, và King Irwin đã đưa ra một quy trình chung trong việc rút trích ontology từ dữ liệu web [[2](#Tim09)]. Nhóm tác giả đưa ra một mô hình chung bao gồm 6 bước nhưng trong đó nhóm tác giả đã không trình bày hết các bước hiện thực hóa quy trình đó và đồng thời việc hiện thực hóa quy trình này dẫn đến những vấn đề kỹ thuật khác. Quy trình này có mức độ tổng quát cao, có thể được áp dụng trong nhiều lĩnh vực. Ngoài ra, quy trình này còn có nhiều khả năng tùy biến về sau, nên mục tiêu của đề tài này được đặt ra như sau.

## Mục tiêu đề tài

Nghiên cứu tổng quan về ontology cũng như các hướng tiếp cận trong việc xây dựng ontology, trong đó đặc biệt quan tâm đến hướng tiếp cận rút trích ontology từ dữ liệu web; phân tích quy trình [[2](#Tim09)] do nhóm tác giả Du C. Timon, Li Feng, và King Irwin đề xuất; xây dựng Ontology Extractor Framework để hiện thực hóa quy trình này và đề xuất những cải tiến minh họa cho framework; hiện thực hóa các module cụ thể để triển khai vào framework này.

## Nội dung luận văn

Luận văn bao gồm 6 chương, nội dung chính từng chương như sau:

**Chương** 1**:** Mở đầu

Trình bày sơ lược khái niệm ontology, vai trò của ontology trong các lĩnh vực khác nhau, và các hướng tiếp cận khác nhau để xây dựng ra các ontology

**Chương** 2**:** Tổng quan Ontology

Trình bày sơ lược nguồn gốc của khái niệm ontology cũng như các định nghĩa khác nhau về ontology của các nhóm tác giả. Ngoài ra, nội dung chương còn đề cập đến các loại ontology khác nhau cũng như các cách biểu diễn ontology và ứng dụng cụ thể của những ontology vào các lĩnh vực khác nhau.

**Chương** 3**:** Các hướng tiếp cận trong việc xây dựng Ontology

Trình bày các nguồn dữ liệu được sử dụng trong việc xây dựng ontology cũng như tổng quan các phương pháp xây dựng nên ontology. Nội dung chương còn để cập đến tổng quát các phương pháp cụ thể để xây dựng ontology của các công trình nghiên cứu khác nhau.

**Chương** 4**:** Quy trình rút trích Ontology từ WWW

Nội dung của Chương 4 trình bày, tóm tắt và phân tích phương pháp rút trích onotlogy từ WWW được các tác giả Du C. Timon, Li Feng, và King Irwin đề xuất [[2](#Tim09)]. Dựa trên phương pháp này, chúng em đã nghiên cứu và đề xuất một số giải pháp nhằm hiện thực hóa và cải tiến quy trình rút trích ontology từ WWW. Những giải pháp hiện thực hóa và cải tiến này được chúng em trình bày chi tiết trong chương 5.

**Chương** 5**:** Ontology Extractor Framework rút trích ontology từ WWW

Nội dung chương 5 trình bày hệ thống framework được nhóm phát triển để hiện thực hóa quy trình rút trích ontology từ WWW đã được trình bày ở chương 4. Các vấn đề và giải pháp khi xây dựng framework và hiện thực hóa quy trình cũng được trình bày trong chương này.

**Chương** 6**:** Kết luận và hướng phát triển

Nội dung của Chương 6 trình bày các kết quả đã đạt được và hướng phát triển của đề tài.

# Tổng quan Ontology

* Nội dung chương 2 giới thiệu ontology cũng như cách phân loại ontology và các ứng dụng của nó:
  + Giới thiệu Ontology
  + Định nghĩa
  + Phân loại Ontology
  + Ứng dụng

## Giới thiệu

Ontology trở thành một lĩnh vực nghiên cứu phổ biến có mặt trong nhiều lĩnh vực từ xử lý ngôn ngữ tự nhiên, công nghệ tri thức, các hệ thống trao đổi, tích hợp thông tin cho đến biểu diễn và quản lý tri thức. Ontology cung cấp và chia sẻ tri thức về một lĩnh vực, giúp dễ dàng giao tiếp giữa người và các hệ thống ứng dụng. Ontology được xây dựng để cung cấp các nguồn thông tin có ngữ nghĩa mà máy tính có thể xử lý và thao tác được nhưng đồng thời vẫn có thể giao tiếp được giữa người và phần mềm.

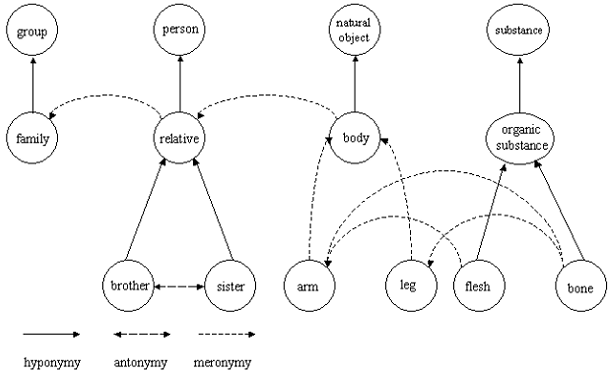
## Định nghĩa

Ontology là một thuật ngữ có nguồn gốc từ Triết học diễn tả các thực thể tồn tại trong tự nhiên và các mối quan hệ giữa chúng. Ontology xuất hiện trong Công nghệ Thông tin trong lĩnh vực Trí Tuệ Nhân Tạo nhằm giải quyết vấn đề về chia sẻ và tái sử dụng tri thức. Trong nhiều năm qua, đã xuất hiện nhiều định nghĩa khác nhau về ontology như định nghĩa của Neches [[3](#Rob91)], Ehrig Marc [[4](#Mar05)], Sure York [[5](#Yor05)], Guarino và Giaretta [[6](#Gua95)] và Bernaras Amaia [[7](#Ama96)]. Trong số những định nghĩa, định nghĩa của Gruber [[8](#Gru93)] được chúng em đánh giá là diễn tả đúng nhất về ontology: "**An ontology is an explicit specification of a conceptualization.**" Ngoài ra, một định nghĩa khác rõ ràng hơn do nhóm tác giả Studer và các đồng sự [[9](#Rud98)] đề ra dựa vào định nghĩa trên, đó là: "**An ontology is a formal, explicit specification of a shared conceptualisation.**" Trong đó, các tác giả đã giải thích cụ thể như sau: "**Conceptualisation**" chỉ đến một mô hình trừu tượng của vài hiện tượng nào đó, dùng để định danh các khái niệm có liên quan đến hiện tượng này. "**Explicit**" chỉ các khái niệm và các ràng buộc được sử dụng đã được định nghĩa rõ ràng. "**Formal**" đề cập đến vấn đề máy có thể hiểu và thao tác được trên ontology. Và cuối cùng "**Shared**" diễn tả ontology thể hiện tri thức nhưng không giới hạn trong vài cá nhân mà được chấp nhận rộng rãi trong một nhóm. Một ontology sẽ cung cấp bộ "từ vựng" các thuật ngữ và quan hệ dùng để mô hình hóa một lĩnh vực nào đó.

## Phân loại

Tùy theo mức độ tổng quát của ontology, ontology có thể được phân thành những loại như sau [[10](#Die01)]:

* Domain ontology diễn tả tri thức của một lĩnh vực cụ thể nào đó (ví dụ: ontology về y khoa: MeSH [[11](#MeS)], GALEN [[12](#GAL)] hay ontology về sinh học: Gene Ontology [[13](#Gen)], OBO [[14](#OBO)]). Những ontology này cung cấp từ vựng về những khái niệm trong một lĩnh vực và quan hệ giữa chúng.
* Metadata ontology cung cấp từ vựng dùng để mô tả nội dung của các nguồn thông tin trực tuyến. (ví dụ ontology Dublin Core [[15](#SWe95)]).
* Generic hay Common sense ontology hướng đến thể hiện tri thức chung, cung cấp các ý niệm và khái niệm cơ bản về thời gian, không gian, trạng thái, sự kiện, … Do đó, các ontology này có thể sử dụng giữa các lĩnh vực khác nhau. Trong đó có thể kể đến WordNet [[16](#Fel98)]: bộ ontology này được xây dựng nhằm mục đích mô tả tiếng Anh bằng cách mô tả từng thuật ngữ trong tiếng Anh cùng các quan hệ cơ bản giữa chúng như đồng nghĩa, phản nghĩa, … Ngoài ra còn có thể kể đến CYC [[17](#Dou95)]: ontology này mô tả chi tiết các tri thức như không gian, thời gian, và cung cấp các quan hệ giữa chúng.



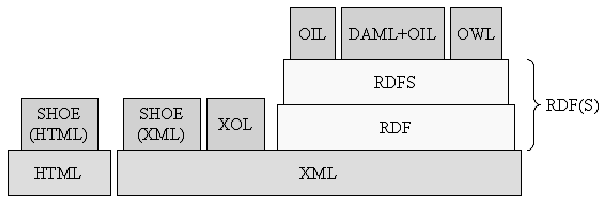
Hình 2‑1 Đồ thị của một phần ontology WordNet [[16](#Fel98)]

* Representational ontology không đại diện cho một lĩnh vực nào cụ thể. Những ontology này cung cấp những thực thể được dùng để đại diện mà không báo nó đại diện cho cái gì. Một trong những ontology thuộc loại này là Frame Ontology của Gruber [[8](#Gru93)], ontology này định nghĩa những khái niệm như là frame, slot, và các ràng buộc slot cho phép biểu diễn tri thức theo hướng đối tượng hoặc theo frame-based.
* Những loại ontology khác được gọi là method và task ontology. Task ontology [[18](#Die97)] cung cấp các thuật ngữ cụ thể cho những tác vụ cụ thể và method ontology cung cấp các thuật ngữ cụ thể cho các phương pháp giải quyết vấn đề cụ thể (Problem Solving Method - PSM) [[19](#Rud)].

## Ngôn ngữ biểu diễn ontology

Từ những năm 90, nhiều ngôn ngữ ontology đã ra đời, những ngôn ngữ này xuất phát từ lĩnh vực Trí Tuệ Nhân Tạo. Chúng được gọi là những ngôn ngữ truyền thống (Traditional Ontology Language) để phân biệt với những ngôn ngữ mới hơn ra đời sau thuộc về nhóm ngôn ngữ đánh dấu ontology (Ontology Markup Language). CycL [[20](#Dou89)] là một trong những ngôn ngữ ra đời đầu tiên phục vụ cho việc xây dựng ontology Cyc [[17](#Dou95)].

Với sự phát triển nhanh chóng của Internet, những ngôn ngữ ontology với nền tảng là ngôn ngữ web ra đời. Cú pháp của chúng dựa vào các ngôn ngữ đánh dấu (markup language) đã tồn tại sẵn đó là HTML, hay XML (những ngôn ngữ đánh dấu này vốn ra đời với mục đích không phải là thể hiện ontology mà được dùng để thể hiện dữ liệu và trao đổi dữ liệu).



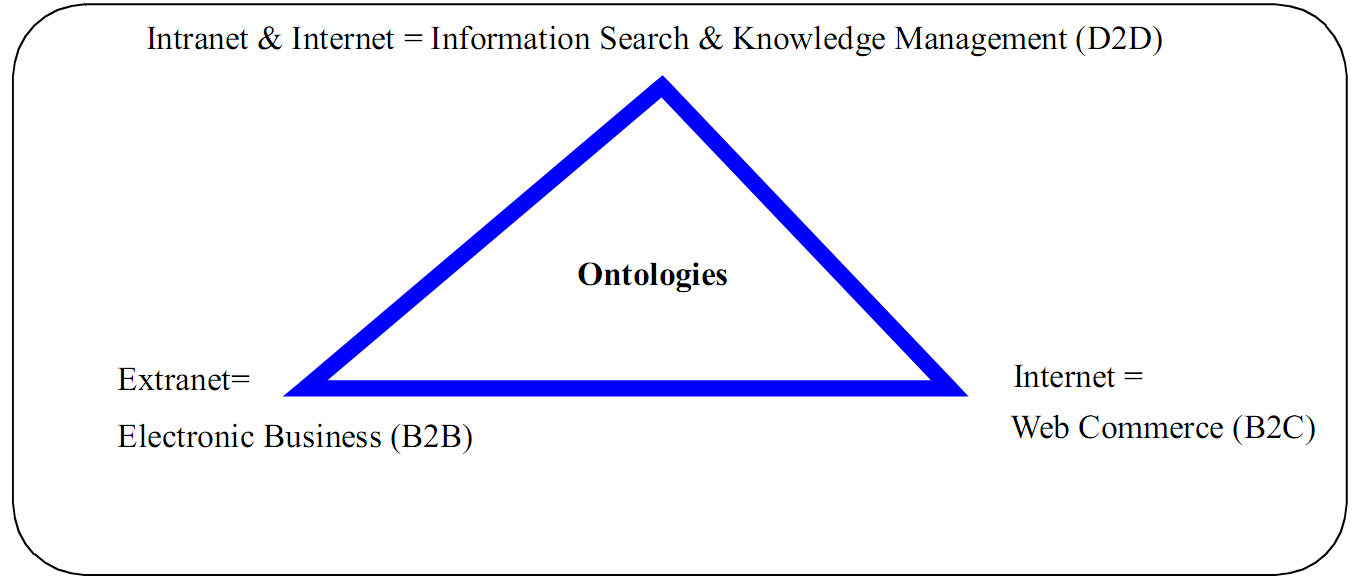
Hình 2‑2 Các ngôn ngữ đánh dấu ontology [[21](#Asu04)]

SHOE do nhóm tác giả Luke và Hefin [[22](#Sea00)] đề xuất có thể xem như là ngôn ngữ đánh dấu ontology đầu tiên. Ngôn ngữ này được xây dựng dựa trên HTML, nhưng sử dụng các thẻ khác nhờ đó cho phép thêm ontology vào các tài liệu HTML (các thẻ này không được định nghĩa trong ngôn ngữ HTML nên những gì thêm vào sẽ không hiện lên trên trình duyệt web). Sau này SHOE được chuyển qua sử dụng trên nền XML.

RDF được đề xuất bởi Lassila và Swick [[23](#Ora99)], ngôn ngữ này được phát triển tại W3C (World Wide Web Consortium). Đây là ngôn ngữ tạo ra các siêu dữ liệu (metadata) để mô tả các tài nguyên web. Sau đó, RDF Schema do Brickley và Guha [[24](#Dan03)] đưa ra như là một bản mở rộng của RDF.

Dựa trên RDF, lần lượt ba ngôn ngữ khác xuất hiện như là những ngôn ngữ mở rộng của RDF: OIL, DAML+OIL và OWL. OIL ra đời năm 2000 do Horrocks và các đồng sự [[25](#Ian00)] đề xuất, sau đó Horrocks và van Harmelen đưa ra DAML+OIL [[26](#Ian01)] vào năm 2001. Cuối cùng là OWL, ra đời vào năm 2003 do Dean và Schreiber [[27](#Mik03)] đề xuất. Đây là kết quả của nhóm nghiên cứu Web-Ontology (WebOnt) do W3C thành lập vào năm 2001, mục tiêu của nhóm là nghiên cứu và phát triển một ngôn ngữ đánh dấu mới dành cho web ngữ nghĩa. OWL có hầu hết các chức năng của DAML+OIL.

## Ứng dụng



Hình 2‑3 Ba lĩnh vực ứng dụng của ontology [[8](#Gru93)]

### Tìm Kiếm Thông Tin và Quản Lý Tri Thức

#### Nhược điểm trong các kỹ thuật tìm kiếm thông tin hiện tại

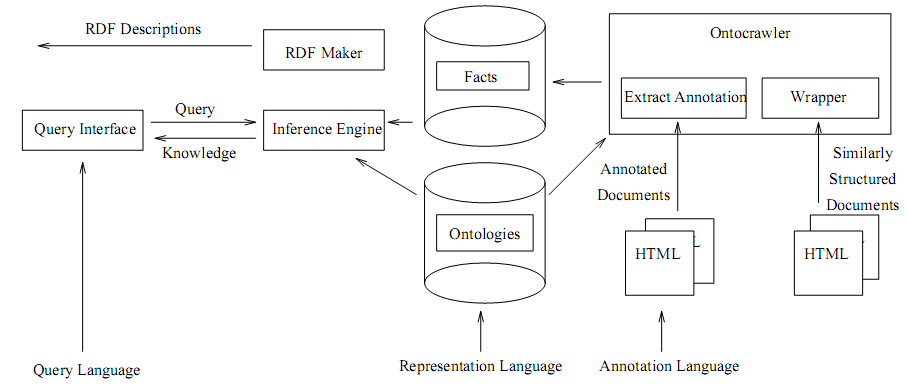
Với những nguồn thông tin khổng lồ hiện tại, như là hệ thống thông tin trên mạng Internet, việc tìm kiếm thông tin cần thiết trở nên khá khó khăn. Mặc dù với sự xuất hiện của các công cụ tìm kiếm lớn nhưng hầu hết chúng đều là các máy tìm kiếm sử dụng từ khóa (keyword-based search engine). Nhược điểm của các máy tìm kiếm này là các kết quả trả về thường chứa rất nhiều kết quả không liên quan đến nội dung tìm kiếm do chủ yếu chúng dựa vào sự xuất hiện của từ khóa trong các văn bản trên Internet.

Ngoài ra, các kết quả thu được từ các máy tìm kiếm này thường là một danh sách các liên kết, các thông tin văn bản hoặc hình ảnh được gán cho liên kết đó. Điều này đòi hỏi người truy vấn phải duyệt qua thông tin để lấy ra được thông tin họ cần. Hơn nữa, kết quả của các truy vấn trên Internet khó có thể được sử dụng một cách trực tiếp ở các chương trình khác.

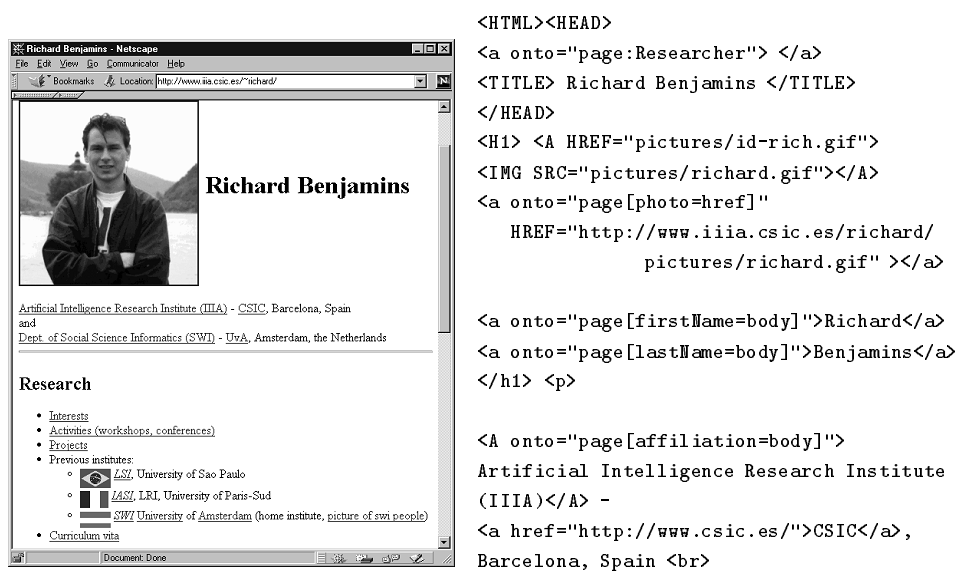
Mặt khác các công cụ tìm kiếm hiện tại không thực hiện việc suy diễn nên không đưa ra được các đề nghị đến các vấn đề có liên quan đến vấn đề đang được truy vấn.

#### Giải pháp

OntoBroker [[28](#Ste99)], [[29](#Die98)] sử dụng ontology để gán nhãn cho trang web, thực hiện các truy vấn và đưa ra các câu trả lời. Các câu trả lời do hệ thống đưa ra dựa trên ngôn ngữ có cú pháp được định nghĩa rõ ràng và có ngữ nghĩa, giúp cho các hệ thống tự động khác có thể dễ dàng sử dụng các kết quả này. Các kết quả truy vấn được còn được thực hiện thông qua việc suy diễn dựa vào ngữ nghĩa và các yếu tố khác.



Hình 2‑4 Kiến trúc chung của hệ thống OntoBroker [[28](#Ste99)]



Hình 2‑5 Một đoạn trang web được gán nhãn bằng OntoBroker [[28](#Ste99)]

Ngoài OntoBroker, còn có các hệ thống khác được phát triển dựa trên OntoBroker như On2broker [[30](#Die99)], IBROW [[31](#VBe99)].

### Thương mại Điện tử

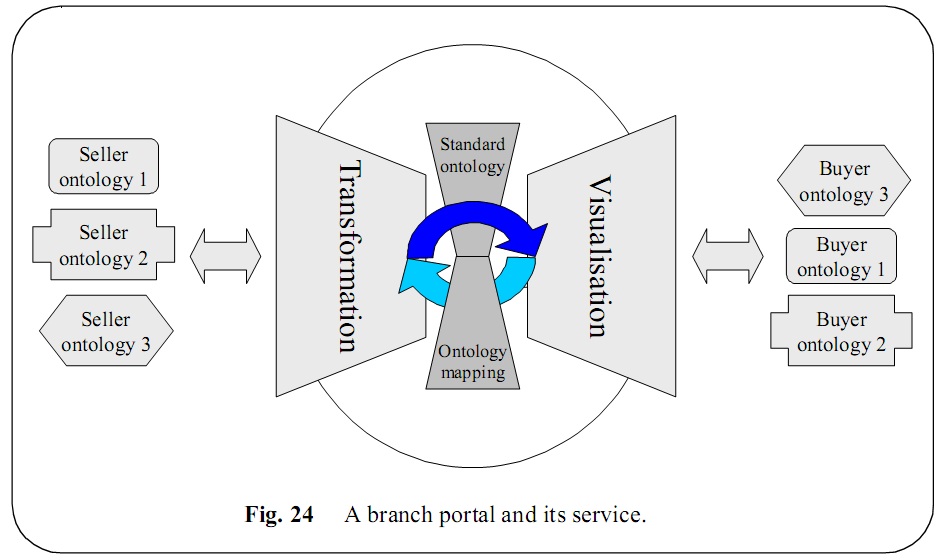
#### Các vấn đề hiện tại trong việc trao đổi thông tin trong lĩnh vực Thương mại Điện tử

Với sự phát triển của Internet, lĩnh vực Thương Mại Điện Tử trở thành một lĩnh vực kinh doanh quan trọng và phát triển không ngừng. Những thuận lợi mà Thương mại Điện tử đem lại cho hoạt động kinh doanh đã dẫn đến sự xuất hiện của hàng loạt cửa hàng trực tuyến, các website bán hàng. Và giờ đây thao tác của người dùng là tìm kiếm cửa hàng trực tuyến nào có bán sản phẩm mà họ quan tâm. Nhưng việc duyệt qua các website này tốn khá nhiều thời gian và công sức trong khi chỉ duyệt qua được số ít các lời mời hàng thực sự. Vì lý do đó các hệ thống tự động xuất hiện giúp cho người dùng tìm kiếm và so sánh giá cả các mặt hàng giữa các cửa hàng khác nhau. Cách thức mà các hệ thống này sử dụng "shopbot" duyệt qua các cửa hàng trực tuyến và xây dựng các "wrapper", được viết khác nhau cho từng cửa hàng trực tuyến cụ thể. Các wrapper này sử dụng phương pháp tìm kiếm dựa vào từ khóa để tìm kiếm sản phẩm mà người dùng quan tâm và thực hiện biến đổi định dạng để thể hiện lên một trang web tổng hợp chung. Việc sử dụng các wrapper này gặp phải các nhược điểm:

* Chi phí xây dựng các wrapper này không nhỏ, và khi các cửa hàng thay đổi cách thức trình bày thì wrapper phải được sửa chữa.
* Nội dung được wrapper rút ra không đầy đủ.

#### Giải pháp

Giải pháp được đưa ra là sử dụng ontology để mô tả các sản phẩm khác nhau và được ứng dụng vào việc định vị và tìm kiếm sản phẩm tự động với các thông tin có sẵn. Ở đây ontology đóng vai trò chuẩn hóa các nhóm mặt hàng. Ngoài ra, ontology còn có công dụng giúp cho các hệ thống tự động giao tiếp với nhau dễ dàng. Các trang web hoạt động như là cổng thông tin chung, có nhiệm vụ thực hiện các biến đổi trên ontology giữa bên bán và bên mua, một số trang web là Chemdex ([www.Chemdex.com](http://www.Chemdex.com)),  PaperExchange ([www.paperexchange.com](http://www.paperexchange.com)) và VerticalNet ([www.verticalnet.com](http://www.verticalnet.com)).



Hình 2‑6 Market place sử dụng ontology [[10](#Die01)]

### Web ngữ nghĩa

Web truyền thống đã thay đổi và phát triển nhiều, trở thành nguồn thông tin lớn nhất cũng như là phương tiện chia sẻ và trao đổi thông tin thuận tiện nhất hiện nay. Nền tảng Web hiện tại cung cấp nguồn thông tin rất lớn nhưng chỉ hỗ trợ cho con người xử lý. Điều đó gây nhiều khó khăn trong việc tìm kiếm, rút trích và quản lý thông tin đối với con người. Để giải quyết vấn đề này, Berners-Lee Tim, Hendler James, và Lassila Ora đề xuất ra Web ngữ nghĩa (Semantic Web) [[32](#Tim01)]. Web ngữ nghĩa được định nghĩa như là sự mở rộng của Web hiện tại mà trong đó thông tin được định nghĩa rõ ràng, giúp cho máy tính và con người cộng tác làm việc tốt hơn. Web ngữ nghĩa cho phép diễn tả ngữ nghĩa của dữ liệu một cách tường minh để máy tính có thể hiểu được.

Một trong những nền tảng xây dựng lên Web ngữ nghĩa đó chính là ontology. Như đã đề cập ở 2.2, ontology cung cấp các thuật ngữ cùng các mối quan hệ giữa chúng theo một cách mà cả con người lẫn máy tính đều có thể hiểu được. Các thuật ngữ thường là một từ trong ngôn ngữ tự nhiên nên con người có thể hiểu dễ dàng. Các quan hệ ngữ nghĩa cũng có thể được hiểu một cách dễ dàng, ví dụ như quan hệ “is-a” giữa hai khái niệm với nhau, diễn tả khái niệm này tổng quát hơn khái niệm kia, giống như Con Người thì tổng quát hơn Sinh Viên. Các mối quan hệ này được định nghĩa rõ ràng giúp cho máy tính có thể thực hiện các suy luận trên đây như cách con người thực hiện.

Một số các ứng dụng của ontology trong web ngữ nghĩa:

1. Gán nhãn ngữ nghĩa và hệ thống Rút Trích Thông Tin (Information Extraction) dựa trên ontology

Gán nhãn ngữ nghĩa cho một tài liệu nào đó tức là thêm vào đó những thông tin khác nhau nhằm tạo ra các phương thức tiếp cận thông tin mới hoặc làm giàu thêm những phương pháp sẵn có. Cách gán nhãn phụ thuộc vào tri thức khám phá được từ tài liệu đó thông qua hệ thống Rút Trích Thông Tin. Những thông tin này còn có thể kết nối đến ontology nhằm cung cấp ngữ nghĩa và quan hệ. Một số các hệ thống bao gồm:

Hệ thống Magpie [[33](#Joh04)] do Domingue và Dzbor giới thiệu vào năm 2004, thực hiện việc gán nhãn các trang web bằng metadata một cách tự động với phương pháp so khớp văn bản với các thể hiện cụ thể của các khái niệm trong ontology. Mục tiêu của hệ thống này là giúp cung cấp những cách nhìn cụ thể và cá nhân hóa theo người dùng cụ thể về những trang web. Bởi vì những người dùng khác nhau sẽ có mức độ nhận thức cũng như nền tảng kiến thức khác nhau, cũng như sự quen thuộc của người dùng đối với thông tin được thể hiện là không giống nhau. Nhược điểm chính của hệ thống Magpie là hệ thống chỉ có thể so khớp với các thể hiện đã có sẵn trong ontology chứ không thể thực hiện với các thể hiện mới xuất hiện.

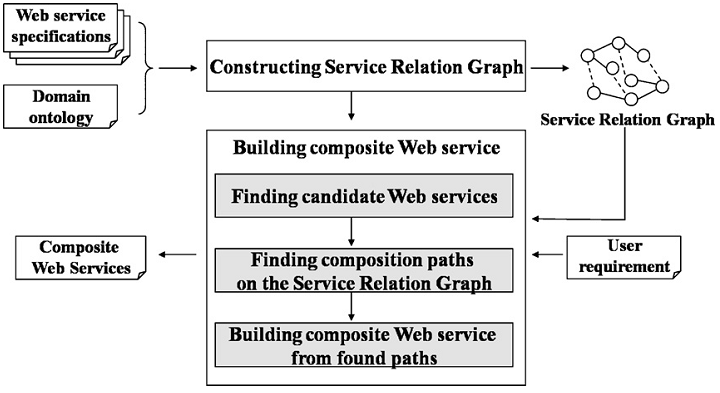
Hệ thống PANKOW [[34](#Phi04)] (Pattern-based Annotation through Knowledgeon the Web) do Cimiano cùng các đồng sự đề xuất, tự động phân loại các thể hiện từ văn bản theo một ontology đã cho sẵn. Hệ thống dùng các khuôn mẫu có dạng <INTANCE><CONCEPT> (ví dụ: the Palace hotel) và <INTANCE> is a<CONCEPT> (ví dụ: Palace is a hotel) cho quá trình phân loại của mình. Các khuôn mẫu này được xây dựng bằng cách kết hợp tất cả các tên riêng trong đoạn văn bản với các khái niệm trong một ontology đã cho sẵn. Mỗi kết hợp này được kiểm tra sử dụng Google. Điểm mạnh của hệ thống là không cần qua quá trình xử lý ngôn ngữ tự nhiên, cũng như quá trình huấn luyện. Mọi thông tin đều có sẵn trên web. Nhưng đây cũng chính là điểm yếu của hệ thống, vì phương pháp này không quan tâm đến ngữ cảnh mà các thể hiện xuất hiện vì vậy làm cho việc phân loại các thể hiện gặp khó khăn khi gặp các thể hiện có cùng tên nhưng thuộc về các lớp khác nhau trong các ngữ cảnh khác nhau (ví dụ: Paris có thể là người, cũng có thể là thành phố, …).

Năm 2003, Dill và các đồng sự đề xuất hệ thống SemTag [[35](#Ste03)] có khả năng thực hiện việc gán nhãn ngữ nghĩa trên phạm vi lớn với việc sử dụng TAP ontology [[36](#RGu)]. Đầu tiên, hệ thống tiến hành gán nhãn tất cả các thể hiện có thể có được đề cập trong TAP ontology. Sau đó, trong quá trình khử nhập nhằng, SemTag sử dụng mô hình không gian-vector để gán chính xác các lớp khái niệm hoặc để loại bỏ những thể hiện không thực sự liên quan đến một lớp trong TAP.

1. Tìm kiếm và tích hợp các web service ngữ nghĩa sử dụng ontology

Sự phát triển nhanh chóng của các web service ngữ nghĩa dẫn đến nhu cầu cần phải có các cơ chế khám phá web service. Các web service ngữ nghĩa được mô tả bằng các ontology khác nhau. Vì vậy cần thiết có một hệ thống khám phá web service ngữ nghĩa sử dụng các ontology khác nhau. Nhóm tác giả Ngan Duy Le và Goh Angela [[37](#LeN05)] đã đề xuất phương pháp tìm kiếm web service ngữ nghĩa trên các ontology khác nhau được viết trên nhiều ngôn ngữ ontology (DAML-S và OWL-S). Người dùng sẽ đưa ra mô tả về web service mà họ mong muốn, hệ thống sẽ tìm kiếm trong cơ sở dữ liệu các hồ sơ mô tả về các web service ngữ nghĩa mà nhà cung cấp đưa ra, sau đó tiến hành so sánh và trả ra web service thỏa mãn yêu cầu người dùng. Cách thức đánh giá hai khái niệm được dùng để mô tả web service trong hai ontology khác nhau được nhóm tác giả đưa ra là tính toán độ tương đồng về ngữ nghĩa của chúng dựa theo phương pháp của Marc Ehrig và York Sure [[38](#Mar04)].

Một nhu cầu khác đồng thời xuất hiện với sự phát triển nhanh chóng của web service chính là nhu cầu kết hợp các web service lại với nhau để tạo thành web service mới thỏa mãn công việc của người dùng. Các web service được mô tả bằng ontology, và các hệ thống dựa vào mô tả này để tìm kiếm ontology phù hợp và kết hợp chúng lại. Nhóm tác giả Shin Dong-Hoon, Lee Kyong-Ho, và Suda Tatsuya đề xuất phương pháp kết hợp web service tự động dựa vào chức năng của chúng [[39](#Don09)]. Phương pháp của nhóm tác giả này đề xuất bao gồm ba bước: tạo ra danh sách ứng viên các web service, tìm đường đi trên đồ thị service, và kết hợp các web service.



Hình 2‑7 Mô hình phương pháp kết hợp web service [[39](#Don09)]

1. Ứng dụng các kỹ thuật của web ngữ nghĩa trong các lĩnh vực khác có sử dụng ontology

Ontology là một trong những thành phần quan trọng đóng vai trò nền tảng của web ngữ nghĩa. Ngoài việc được ứng dụng trong khoa học thông tin ontology còn được nghiên cứu để phục vụ nhu cầu trao đổi thông tin. Việc đưa ontology vào web ngữ nghĩa giúp cho thông tin có sẵn trên web được máy tính đọc và hiểu tự động dễ dàng hơn. Ngày nay các công nghệ về web ngữ nghĩa đã được ứng dụng sang các lĩnh vực khác như:

***Hệ thống quản lý truy cập và bảo mật thông tin***

Các kỹ thuật, phương pháp và kết quả được dùng trong lĩnh vực web ngữ nghĩa được sử dụng vào trong các hệ thống quản lý việc truy cập trên mạng hoặc bảo mật thông tin: trong đó những hệ thống trực tuyến định nghĩa sẵn các chính sách để quy định quyền truy cập (access control) đối với từng nhóm người dùng cũng như các quy luật về việc truy cập. Ví dụ: một số policy như giới hạn thời gian sử dụng, số lượng download một tài nguyên hay truy cập vào một vùng máy hoặc ip nào đó. Những policy này cũng được định nghĩa hoặc đặc tả thông qua các ngôn ngữ đặc tả như là XACML (eXtensible Access Control Markup Language) hoặc SAML (Security Assertion Markup Language). Khi người sử dụng muốn dùng các hệ thống đó, họ phải cung cấp một số thông tin, không chỉ là những thông tin định danh mà còn có thể là các thông tin cá nhân khác như là: độ tuổi, nghề nghiệp, địa chỉ…. Với những policy và những thông tin do người dùng được cung cấp thì hệ thống phải so khớp chúng để xem những thông tin được cung cấp này có phù hợp với các policy này không. Do đó xuất hiện nhu cầu phải tiến hành việc đối sánh các khái niệm trong thông tin do người dùng cung cấp và các khái niệm trong các policy đã có. Ví dụ policy đòi hỏi định danh là Username, trong khi người dùng cung cấp định danh là Account. Khi đó hai khái niệm này phải khớp với nhau, chứ không phải là so khớp phần thông tin nội dung bên dưới.

***Hệ thống phần mềm thích nghi***

Trong các hệ thống phần mềm thích nghi, việc tích hợp các ứng dụng dựa trên nền tảng SOA để phát triển các hệ thống phần mềm sử dụng các thành phần khác nhau được cung cấp từ bên ngoài dưới dạng là COM, Web Service, WebAPI, … tuy nhiên cùng một công dụng có thể được cung cấp bởi nhiều nguồn, hoặc là trong quá trình sử dụng có một hay vài thành phần nào đó gặp lỗi. Do đó, hệ thống cần một giải pháp phù hợp với nhu cầu của nó, vì vậy việc gán nhãn ngữ nghĩa cho các chức năng, phương thức, hàm của các web service hay WebAPI để cho hệ thống khác dễ dàng khám phá và lắp ghép tự động các thành phần này lại với nhau.

Những ví dụ trên cho thấy những công nghệ, kỹ thuật và các kết quả đạt được trong web ngữ nghĩa được ứng dụng vào rất nhiều lĩnh vực khác nhau. Từ đó cho thấy nhu cầu cấp thiết phải có ontology.

Để sử dụng ontology thì các khái niệm trong ontology phải được gán nhãn ngữ nghĩa trong các thành phần thông tin khác nhau của một hệ thống để phù hợp với nhu cầu xử lý. Thông tin sau khi đã được gán nhãn giúp cho máy tính xử lý hiệu quả hơn, không phải xử lý trên văn bản thô về mặt ngữ nghĩa. Ví dụ:

* Web service có phần mô tả có thể được viết bằng ngôn ngữ tự nhiên, muốn *hệ thống tự thích ứng* có thể khám phá ra web service này, hệ thống có thể sử dụng các giải pháp như: (1) tiến hành xử lý ngôn ngữ tự nhiên trên văn bản mô tả web service này, (2) khi người dùng tạo ra web service họ đã gán nhãn ngữ nghĩa để mô tả cho web service này. Khi đó hệ thống sẽ xử lý hiệu quả hơn khi sử dụng phương án (2) so với phải xử lý ngôn ngữ tự nhiên ở phương án (1).
* Trong việc xử lý các câu truy vấn, ví dụ câu truy vấn sau trong một hệ thống truy vấn thông tin về địa điểm: “Các quán ăn trong phạm vi 100m gần trường Đại học Khoa Học Tự Nhiên”. Khi đó một hệ thống gán nhãn tự động sẽ giúp hiểu tốt hơn câu truy vấn so với khi xử lý câu truy vấn là một chuỗi ngôn ngữ chưa có nhãn.

## Kết luận

Ontology đang trở thành một hướng nghiên cứu phổ biến trong nhiều lĩnh vực khác nhau. Việc áp dụng ontology vào những hệ thống khác nhau giúp tăng khả năng xử lý và tính hiệu quả của hệ thống. Trong số đó, ontology đã trở thành một trong những nền móng trong lĩnh vực web ngữ nghĩa. Những công nghệ, những kỹ thuật cũng như các thành quả đạt được trong lĩnh vực nghiên cứu về web ngữ nghĩa cũng như ontology đã và đang được ứng dụng trong nhiều lĩnh vực khác nhau. Vì nhu cầu tăng cao của các ứng dụng có sử dụng ontology, nên xuất hiện ngày càng nhiều các nghiên cứu liên quan đến việc xây dựng ontology. Tổng quan các phương pháp này sẽ được đề cập thêm ở chương tiếp theo.

# Các hướng tiếp cận trong việc xây dựng ontology

* Nội dung Chương 3 đề cập đến các hướng tiếp cận khác nhau để rút trích ontology.
  + Các nguồn dữ liệu dùng để xây dựng ontology
  + Tổng quan việc xây dựng ontology
  + Phương pháp phân tích ngôn ngữ
  + Phương pháp dựa vào xác suất
  + Phương pháp máy học
  + Phương pháp kết hợp

## Các nguồn dữ liệu dùng để xây dựng ontology

Các hệ thống xây dựng ontology có thể sử dụng dữ liệu từ nhiều nguồn khác nhau, các nguồn này có thể được phân chia thành các loại sau đây [[40](#Sha03)]:

* Dữ liệu có cấu trúc: Hệ thống xây dựng lên các ontology dựa vào các dữ liệu có cấu trúc như từ database schema [[41](#Kas99)], từ những ontology đã có sẵn [[42](#Wil00)], từ những cơ sở tri thức [[43](#Lea00)] và từ các mạng từ vựng như WordNet.
* Dữ liệu bán cấu trúc: đây cũng là một nguồn khác mà các hệ thống thường sử dụng, bao gồm các từ điển, các văn bản HTML và XML.
* Dữ liệu không có cấu trúc: đây là nguồn dữ liệu khó rút trích tri thức nhất. Các hệ thống xây dựng ontology phải thực hiện các công đoạn xử lý ngôn ngữ tự nhiên trên các văn bản này để khám phá ra các khái niệm và các quan hệ. Dữ liệu dạng này bao gồm các văn bản viết trên ngôn ngữ tự nhiên hoặc các văn bản lấy từ web.

## Xây dựng ontology

Rút trích ontology là một trong những thao tác trên ontology. Việc rút trích nhằm lấy ra những yếu tố (các khái niệm) từ các nguồn khác nhau, và tạo thành ontology. Việc xây dựng một ontology một cách thủ công tốn rất nhiều thời gian và công sức. Vì vậy nhu cầu cần có những phương pháp xây dựng ontology tự động hoặc bán tự động xuất hiện, và các phương pháp rút trích ontology được đưa ra để đáp ứng yêu cầu này. Các phương pháp rút trích ontology sử dụng nhiều cách khác nhau trải dài từ các phương pháp máy học, xử lý ngôn ngữ tự nhiên cho đến thống kê.

|  |  |
| --- | --- |
| Tác giả | Phương pháp |
| Faure David và Poibeau Thierry [[44](#Dav00)] | XLNNTN |
| Shamsfard vàAbdollahzadeh [[45](#Meh04)] | XLNNTN |
| Agirre Eneko và đồng sự [[46](#Ene00)] | Thống kê |
| Faatz Andreas và Steinmetz Ralf [[47](#And02)] | Thống kê |
| Heyer và đồng sự [[48](#Ger01)] | Thống kê |
| Jiang Xing và Tan Ah-Hwee [[49](#Xin05)] | Thống kê |
| Maddi và đồng sự [[50](#Gov01)] | Thống kê |
| Buttler David, Liu Ling, và Pu Calton [[51](#AFu01)] | Máy học |
| Valter, Giansalvatore, và Paolo [[52](#Val01)] | Máy học |
| Hasan, Srinivas, và Saravanakumar [[53](#Has04)] | Máy học |
| Han Hyoil và Elmasri Ramez [[54](#Hyo04)] | Máy học |
| Jörg-Uwe, Raphael, và Alexander [[55](#Jör00)] | Kết hợp |
| Du C. Timon, Li Feng, và King Irwin [[2](#Tim09)] | Máy học |

Bảng 3‑1 Tóm tắt các công trình nghiên cứu có liên quan

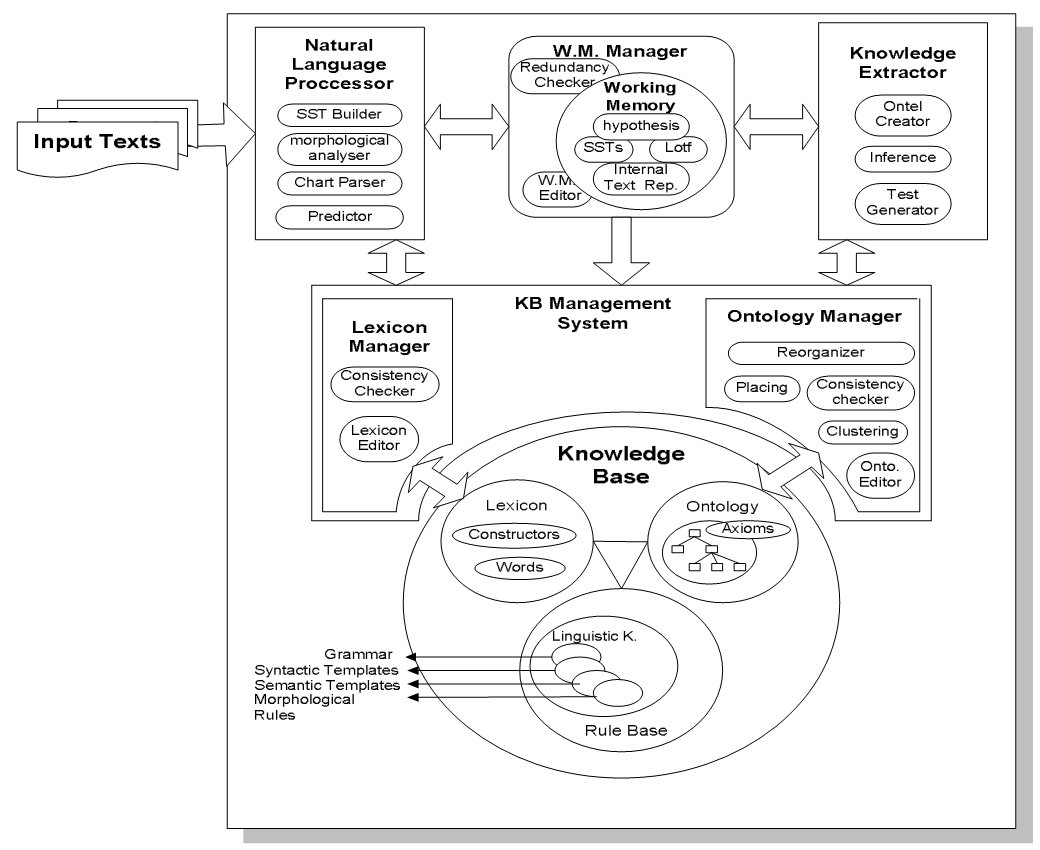
## Phương pháp dựa trên việc xử lý ngôn ngữ tự nhiên

Hệ thống ASIUM [[44](#Dav00)] được Faure David và Poibeau Thierry đề xuất sẽ tự động rút ra được từ các phần văn bản thuộc về một lĩnh vực nào đó các khung cú pháp (syntactic frame) có dạng: <verb><preposition | role: head noun>\*. Các “head noun” này sẽ được chọn lọc để tạo thành các lớp cơ bản và ASIUM tập hợp chúng lại để tạo thành các khái niệm bằng phương pháp gom cụm và các khái niệm sẽ được gán nhãn bởi chuyên gia.

Cũng dựa vào nền tảng xử lý ngôn ngữ tự nhiên, các tác giả Mehrnoush và Ahmad sử dụng một ontology đã được xây dựng thủ công từ trước để làm nhân (kernel) (nhân này chứa các khái niệm, quan hệ và các thao tác cơ bản), và sau đó xây dựng ontology dựa vào việc hiểu văn bản tự động.

Hệ thống Hasti [[45](#Meh04)] do hai tác giả này đề xuất thực hiện việc xử lý các văn bản tiếng Persia, độc lập với lĩnh vực và chỉ cần sử dụng nhân có kích thước nhỏ. Văn bản được đưa qua hệ thống xử lý ngôn ngữ tự nhiên cho tiếng Persia để phân tích. Hệ thống này sử dụng các khuôn mẫu ngữ nghĩa (semantic template) để hiểu được văn bản đã phân tích và thực hiện các suy diễn để rút trích tri thức theo hai cấp độ: cấp độ câu và cấp độ đoạn văn. Các khái niệm mới được tìm thấy sẽ được hệ thống đưa vào *ontology nhân*, và các thể hiện của các khái niệm đã có sẵn trong *ontology nhân* sẽ được hệ thống đánh nhãn. Do đó phương pháp này xây dựng được ontology bao gồm các khái niệm và quan hệ đồng thời chứa cả các thể hiện của các khái niệm đó. Phương pháp mà hệ thống sử dụng để thêm khái niệm mới vào ontology đã có là phương pháp gom cụm. Hệ thống còn sử dụng các heuristic khác nhau để khử nhập nhằng và chọn ứng viên tốt hơn.

Hệ thống này có thể áp dụng cho nhiều ngôn ngữ khác nhau, chỉ cần thay đổi bộ ngữ pháp, các luật biến đổi và các khuôn mẫu ngữ nghĩa. Sự chính xác của hệ thống này phụ thuộc nhiều vào việc xử lý ngôn ngữ tự nhiên.



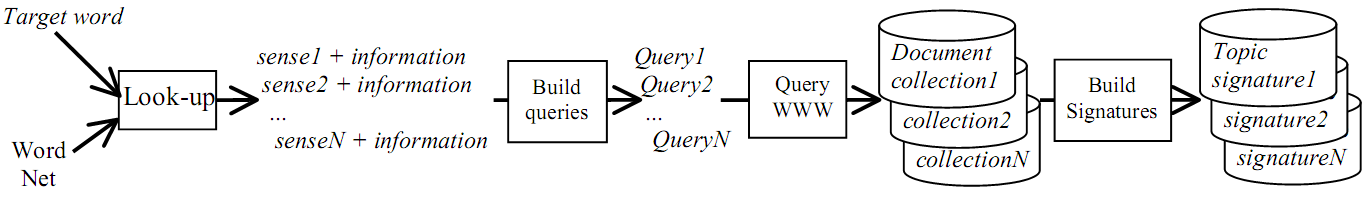
Hình 3‑1 Kiến trúc của Hasti [[45](#Meh04)]

## Phương pháp dựa vào thống kê

Agirre Eneko và các đồng sự [[46](#Ene00)] sử dụng các văn bản trên web để làm giàu ontology đã có sẵn. Ontology được nhóm tác giả sử dụng ở đây là WordNet [[16](#Fel98)]. WordNet thiếu các quan hệ giữa các nét nghĩa cùng một chủ đề. Ví dụ: farm-chicken, spoon-dinner là những nét nghĩa cùng một chủ đề với nhau. Nhóm tác giả liên kết các khái niệm có cùng chủ đề trong WordNet dựa vào tập hợp tài liệu trên web, giúp thêm quan hệ còn thiếu cho các khái niệm có sẵn trong WordNet.

Từ WordNet thu được các nét nghĩa và các thông tin khác có liên quan đến nét nghĩa đó như từ đồng nghĩa, phản nghĩa, … và từ các thông tin này các câu truy vấn sẽ được xây dựng cho từng nét nghĩa nhằm loại bỏ những tài liệu có khả năng thuộc về nhiều hơn một nét nghĩa. Từ những truy vấn này, hệ thống sẽ tìm kiếm trên Internet thông qua các máy tìm kiếm để thu được các tài liệu thỏa những câu truy vấn này, sau đó tiến hành thống kê trên những tài liệu này để tạo thành các topic signature. Các nét nghĩa trong WordNet sẽ được gom cụm dựa trên topic signature của nó.

Phương pháp do nhóm tác giả đưa ra giúp giải quyết vấn đề gom nhóm các nét nghĩa có cùng chủ đề lại với nhau (trong WordNet).



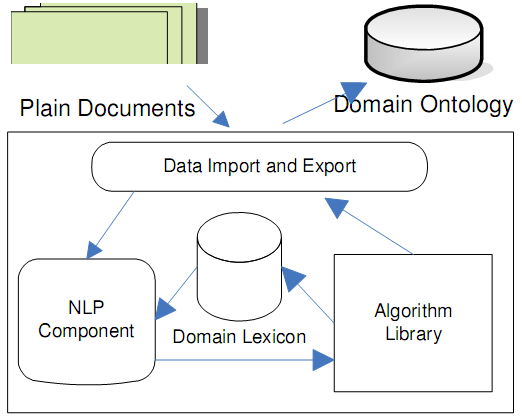
Hình 3‑2 Thiết kế chung của phương pháp [[46](#Ene00)]

Ở một hướng tiếp cận khác, tác giả Faatz Andreas và Steinmetz Ralf [[47](#And02)] cũng sử dụng các tài liệu thu được từ web để làm giàu ontology có sẵn (ở đây nhóm tác giả sử dụng ontology thuộc về lĩnh vực y khoa) và đưa ra một phương pháp bán tự động với sự trợ giúp của chuyên gia về ontology (ontology engineer). Hệ thống sẽ sử dụng ngữ liệu thu được từ web thông qua cỗ máy tìm kiếm Google để lập ra một tập hợp các khái niệm ứng viên và sau đó tính toán sự tương đồng của chúng với các khái niệm đã có sẵn trong ontology làm nhân ban đầu.

Heyer Gerhard và các đồng sự [[48](#Ger01)] sử dụng phương pháp thống kê dựa trên ngữ liệu lớn để rút trích ra các quan hệ ngữ nghĩa từ những văn bản không có cấu trúc. Điểm khác biệt ở đây là họ thống kê sự xuất hiện đồng thời của các cặp từ và đưa ra độ đo mức độ quan trọng của một cặp từ (significance measure). Độ đo này được tính như sau: gọi a, b là số lượng các câu chứa từ A và từ B, k là số lượng các câu chứa cùng lúc cả từ A lẫn từ B, và n là tổng số lượng câu. Đặt x=ab/n, nhóm tác giả định nghĩa độ đo mức độ quan trọng của cặp từ A và B:

Bằng cách giữ nguyên một từ trong cặp từ, một danh sách các cặp từ xuất hiện đồng thời với từ được cố định được sinh ra và danh sách này được sắp xếp theo thứ tự độ đo quan trọng của nó với từ được cố định, từ đó có thể rút ra các quan hệ giữa các từ đó với từ được cố định. Nhóm tác giả này đề xuất ra nhiều phương án khác nhau để nhận diện được những quan hệ này.

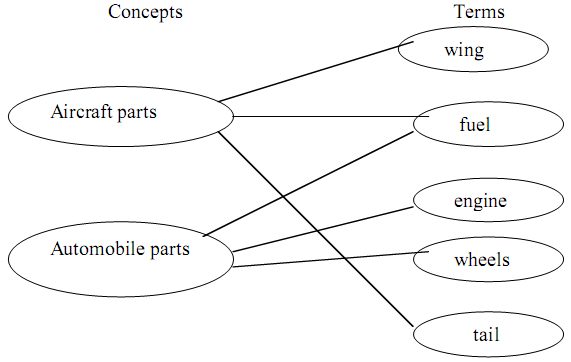
Hệ thống được các tác giả Jiang Xing và Tan Ah-Hweeđưa ra là CRCTOL [[49](#Xin05)], sử dụng phương pháp phân tích toàn bộ văn bản kết hợp với việc thống kê và các phương pháp xử lý ngôn ngữ tự nhiên trên các văn bản thuộc về một lĩnh vực nào đó cụ thể. Sau khi đi qua bộ xử lý ngôn ngữ, các thuật ngữ (term) được lọc ra và sau đó tạo thành một danh sách các thuật ngữ ứng viên cho lĩnh vực đó, các thuật ngữ này sẽ được thống kê và xét với ngưỡng. Mối quan hệ ngữ nghĩa giữa các khái niệm là một bộ <Khái niệm1, Quan hệ, Khái niệm2> thì trong các văn bản ngôn ngữ bình thường có bộ <Danh từ1, Động từ, Danh từ2> trong đó Danh từ1 và Danh từ2 là những thuật ngữ đồng thời cũng là các thể hiện của các Khái niệm tương ứng trong ontology. Sau đó hệ thống sử dụng các Động từ để rút ra mối quan hệ giữa các Khái niệm.



Hình 3‑3 Kiến trúc chung của hệ thống CRCTOL [[49](#Xin05)]

Hệ thống do Maddi Reddy Govind và các đồng sự [[50](#Gov01)] phát triển, khai thác từ tập hợp các văn bản thuộc về cùng một lĩnh vực và rút trích ontology theo phương pháp thống kê bằng cách thống kê tần số xuất hiện các từ.

Nhóm tác giả sử dụng phương pháp thống kê Latent Semantic Indexing (LSI) để biểu diễn một văn bản bằng những khái niệm. Ontology được xây dựng thành một đồ thị hai phía, trong đó một phía là các khái niệm và phía còn lại là các term (thuộc về một khái niệm nào đó).



Hình 3‑4 Một phần đồ thị hai phía sinh ra từ hệ thống [[50](#Gov01)]

## Phương pháp máy học

Việc rút trích các khái niệm từ các nguồn tài nguyên web mà không cần dùng thêm các nguồn dữ liệu bổ sung khác phụ thuộc nhiều vào việc rút trích các đối tượng từ các nguồn tài nguyên web đó. Các phương pháp sau đây thực hiện việc rút trích các đối tượng từ các trang web bằng phương pháp máy học.

Phương pháp do nhóm tác giả Buttler David, Liu Ling, và Pu Calton [[51](#AFu01)] đề xuất là duyệt qua văn bản HTML để xây dựng cây cấu trúc các thẻ của trang này cùng với các thông số thống kê cần thiết có liên quan. Từ những thông tin tính toán này, hệ thống Omini sẽ định vị được cây con của cây cấu trúc thỏa một số điều kiện để được xem là ứng viên chứa các đối tượng cần quan tâm.

Sau đó hệ thống sẽ duyệt qua cây con này, và tìm kiếm thẻ nào được dùng làm thẻ phân cách các đối tượng riêng lẻ với nhau và với các thông tin khác dựa vào một vài heuristic. Việc còn lại là kết hợp các heuristic như thế nào để đem lại hiệu quả cao nhất do các heuristic này không phải lúc nào cũng định ra được thẻ ứng viên tốt nhất.

Crescenzi Valter, Mecca Giansalvatore, và Merialdo Paolo đưa ra hệ thống RoadRunner [[52](#Val01)], hệ thống này sẽ phát sinh tự động các wrapper (được dùng để rút trích tự động các đối tượng) tùy vào từng trang web cụ thể bằng cách so sánh các trang web HTML với nhau để xem sự giống và khác nhau của chúng.

Hệ thống RoadRunner dựa vào một cặp trang web, trong đó chọn một trong số đó làm wrapper ban đầu, rồi dần dần làm mịn wrapper này bằng việc so sánh với trang còn lại (gọi là các mẫu) để xem sự khác biệt và giống nhau nào giữa chúng.

Davulcu Hasan, Vadrevu Srinivas, và Nagarajan Saravanakumar [[53](#Has04)] xây dựng hệ thống OntoMiner, hệ thống này sẽ nhận vào các trang web thuộc về cùng lĩnh vực và từ đó xây dựng cây phân cấp ngữ nghĩa cho trang web đó, mà trong đó các node là các khái niệm. Sau đó hệ thống sẽ tiến hành khai thác trên cây này để tìm ra các khái niệm chính cho lĩnh vực hiện tại cũng như các quan hệ giữa các khái niệm này.

Phương án do nhóm tác giả Han Hyoil và Elmasri Ramez [[54](#Hyo04)] đề xuất là tìm cách khám phá cấu trúc trang web bằng phương pháp Inductive Logic Programming (ILP), để rút ra được các luật có liên quan nhằm nhận biết các khái niệm từ cấu trúc của trang web. Đầu tiên các trang HTML được đánh nhãn bằng bộ POS tagger và EER (Extended Entity Relationship) tagger. Ví dụ đoạn HTML sau:

<td>

<b>Instructor</b><br>

Prof. John Smith<br>

CCB 138<br>

Phone: 404 894-2222<br></td>

Sau khi được đưa qua bộ đánh nhãn POS và EER, thu được:

<td>

<b>**<EERTAG><\*E**4**><#entNo=**20062**>**Instructor**/NNP**

**<\*E**4**></ERRTAG>**</b><br>

Prof**/NNP**.**/.** John**/NNP** Smith**/NNP**<br>

CCB**/NNP** 138**/CD**<br>

**<EERTAG><\*A**44**><#attNo=**2102**>**Phone**/NN<\*A**44**><EERTAG>**:**/:** 404**/CD** 894**/CD**-**/:**2222**/CD**<br></td>

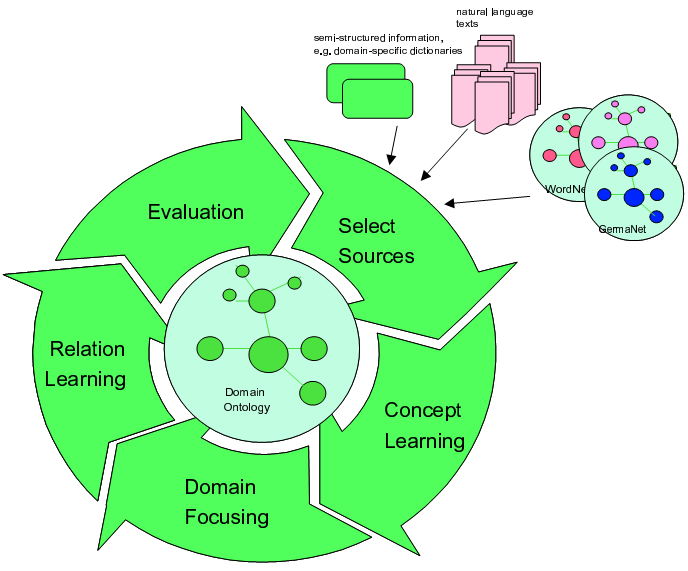
Các trang web đã được gán nhãn này sau đó được dùng để tạo thành cây ngữ nghĩa (Semantic Tree). Các đặc trưng được rút từ cây này và làm đầu vào cho Progol (hệ thống ILP) để học các pattern về quan hệ giữa các khái niệm.

Nhóm tác giả Du C. Timon, Li Feng, và King Irwin [[2](#Tim09)] đề xuất phương pháp rút trích ontology từ website một cách bán tự động bằng phương pháp máy học thông qua một quy trình bao gồm 6 bước: Chuẩn bị, Biến đổi, Gom cụm, Nhận diện, Liên kết và Tinh chỉnh. Các trang web của một website được tải về và thực hiện các biến đổi để chuẩn hóa. Sau đó chúng được gom cụm dựa trên độ tương đồng giữa các *vector đặc trưng* của chúng. Mỗi cụm sau đó được nhận diện *đặc trưng cụm* bằng cách rút ra vector đặc trưng tổng của cụm đó thông qua quá trình Nhận diện, đặc trưng của cụm cũng chính là các ứng viên cho các *khái niệm* được rút trích ra để tạo thành ontology. Ở bước Liên kết, *mối quan hệ* giữa các cụm được gán dựa trên các đường dẫn giữa các trang web trong cụm. Cuối cùng, việc tinh chỉnh ontology rút trích được từ các bước trên được thực hiện bởi một chuyên gia xử lý ontology ở bước Tinh chỉnh.

## Phương pháp kết hợp

Phương pháp do nhóm tác giả Kietz Jörg-Uwe, Volz Raphael, và Maedche D. Alexander [[55](#Jör00)] đề xuất đưa ra một quy trình xây dựng ontology bán tự động. Quy trình này bắt đầu bằng việc chọn ra một ontology làm nhân, có thể là ontology tổng quát, các mạng ngữ nghĩa (như WordNet [[16](#Fel98)], Germanet [[56](#Bir97)], ...) hoặc là ontology liên quan đến lĩnh vực đang được quan tâm. Đồng thời, hệ thống chọn ra các văn bản về lĩnh vực đang được quan tâm cần để sử dụng cho việc rút trích các thực thể của lĩnh vực đó. Các khái niệm thu được từ những văn bản này được dùng để làm giàu cho *ontology nhân*, nhưng vẫn còn khá nhiều khái niệm trong ontology này không thuộc về lĩnh vực đang quan tâm, do đó chúng phải được loại bỏ đi.

Ngoài các quan hệ có sẵn trong *ontology nhân*, hệ thống sẽ học thêm các quan hệ khác do các khái niệm mới sinh ra. Phương pháp được sử dụng bao gồm thống kê sự xuất hiện đồng thời của các khái niệm, hoặc sử dụng các pattern để nhận biết các quan hệ. Quá trình này được lặp lại để ontology ngày càng hoàn thiện hơn.



Hình 3‑5 Quy trình thu nhận Ontology [[55](#Jör00)]

# Quy trình rút trích ontology từ WWW

* Nội dung của Chương 4 trình bày, tóm tắt và phân tích phương pháp rút trích onotlogy từ WWW được các tác giả Du C. Timon, Li Feng, và King Irwin đề xuất [[2](#Tim09)]. Dựa trên phương pháp này, chúng em đã nghiên cứu và đề xuất một số giải pháp nhằm hiện thực hóa và cải tiến quy trình rút trích ontology từ WWW. Những giải pháp hiện thực hóa và cải tiến này được chúng em trình bày chi tiết trong Chương 5.

## Mở đầu

Sau quá trình khảo sát và phân tích phương pháp rút trích ontology từ WWW của nhóm tác giả Du C. Timon, Li Feng, và King Irwin, chúng em có một số nhận xét tổng quát mang tính đúc kết về phương pháp mà nhóm tác giả đề xuất. Đầu tiên, phương pháp này khai thác đặc điểm cấu trúc và nội dung văn bản bên trong các thẻ của trang web. Các trang web này cùng thuộc về một Website hướng ontology được chọn làm nguồn dữ liệu đầu vào cho phương pháp. Hơn nữa, các trang web này được tải mới hoàn toàn và chưa được gán nhãn hay định danh, do đó, phương pháp của nhóm tác giả là phương pháp học máy không giám sát. Ngoài ra, phương pháp này không những rút trích được ontology có các khái niệm bám sát nội dung thông tin của Website mà còn giải quyết được một trong các bài toán ứng dụng ontology đã được nhắc đến ở mục 2.5, đó là gán nhãn khái niệm cho các trang web của Website. Vì vậy, ngoài việc sử dụng ontology rút trích được cho các ứng dụng trong cùng lĩnh vực thông tin liên quan hay trong việc kết nối, giao tiếp giữa các lĩnh vực thông tin khác, việc sử dụng ontology này cho các tác vụ trên chính Website được chọn sẽ càng phù hợp và hiệu quả hơn. Trên cơ sở phân tích các ưu điểm của phương pháp này mà chúng em quyết định hiện thực hoá và cải tiến quy trình rút trích ontology từ WWW dựa trên quy trình 6 bước của nhóm tác giả.

## Một số giả định

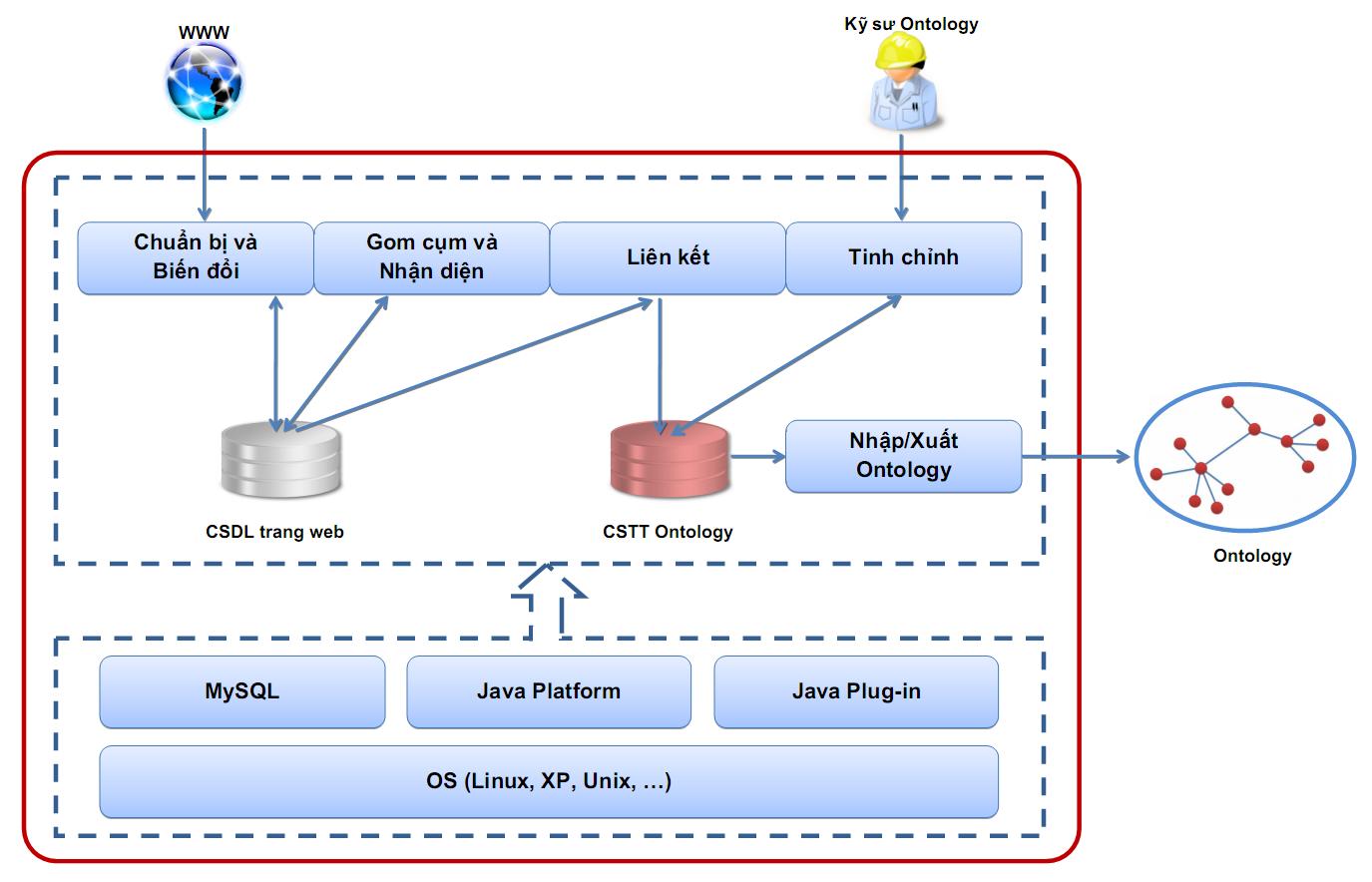
Để việc rút trích ontology được thuận lợi, một số giả định sau được đặt ra mà vẫn không làm sai lệch mục tiêu và phạm vi chính của phương pháp:

1. Website được sử dụng phải thể hiện một chủ đề nhất định: Nhờ giả định này, các khái niệm được rút trích từ Website sẽ cùng liên quan đến một chủ đề. Điều này hoàn toàn phù hợp với định nghĩa ontology ở mục 2.2 và đảm bảo chất lượng ontology kết quả của phương pháp.
2. Các trang web được viết bằng HTML: Phương pháp rút trích ontology từ WWW dựa vào cấu trúc các thẻ HTML để đánh giá và thống kê tần số xuất hiện các *từ khóa*, từ đó rút trích được các *khái niệm*.
3. Các trang web không phải là các trang web ẩn: Nghĩa là người dùng cũng như các chương trình tự động có thể truy xuất các trang web này trực tiếp mà không phải nhập tài khoản hay mật khẩu [[57](#Mic01)].
4. Các trang web phải chứa văn bản: Để rút trích được khái niệm từ các trang web, điều cần thiết là các trang web này phải chứa văn bản. Nguồn được sử dụng trong việc rút trích các *khái niệm* không phải là hình ảnh, nhạc hay phim vì nằm ngoại phạm vi nghiên cứu của quy trình được đề xuất.
5. Ngôn ngữ của Website là tiếng Anh: Việc xử lý ngôn ngữ cũng nằm ngoài phạm vi của quy trình được đề xuất, nhóm tác giả nhấn mạnh việc tránh các vấn đề liên quan đến ngôn ngữ. Do vậy, tiếng Anh là ngôn ngữ đã có nhiều công trình nghiên cứu hoàn chỉnh nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên đã được chọn.

* Nhận xét

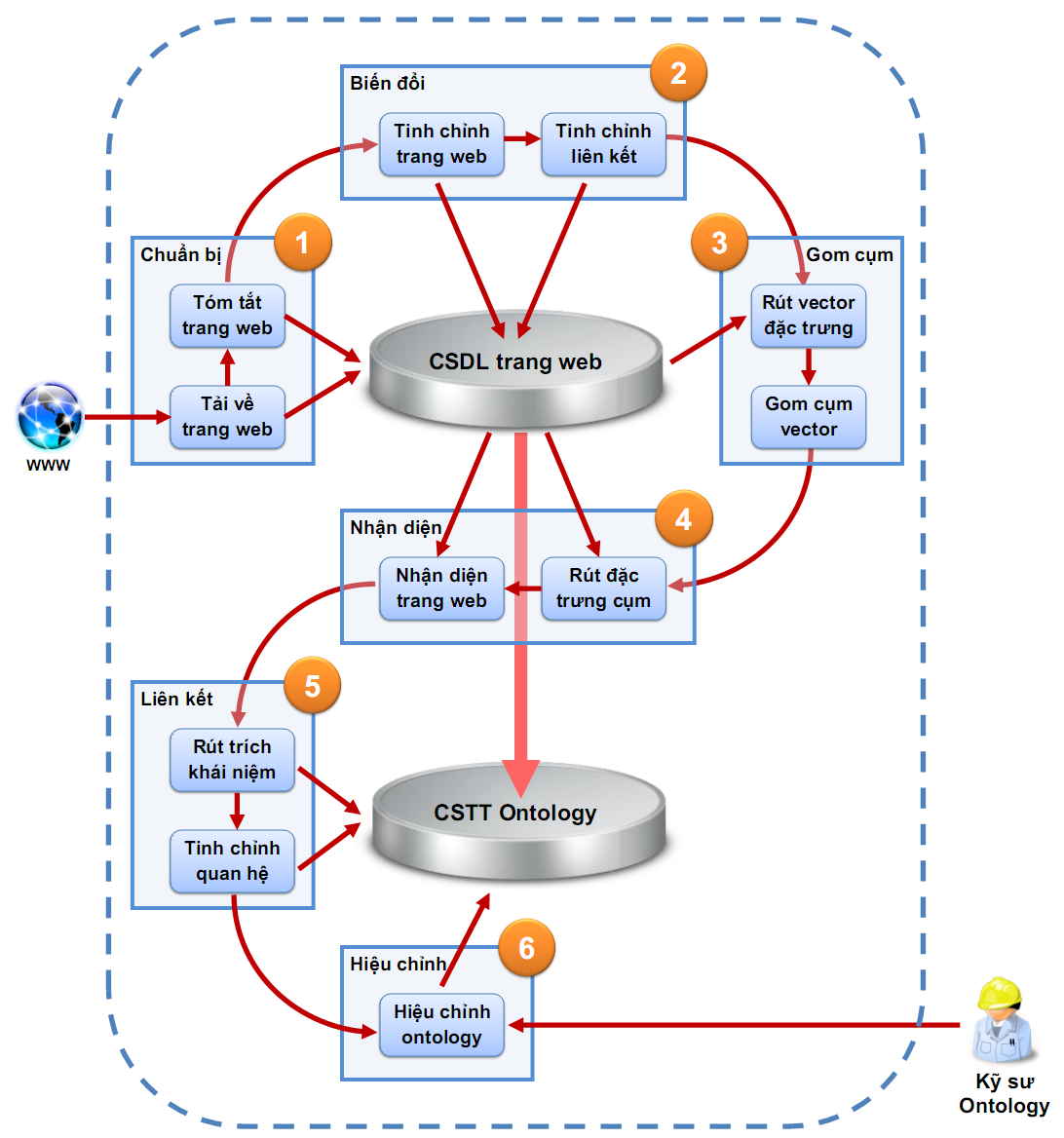
Trong số các giả định trên, chúng em nhận thấy giả định thứ 5 là không cần thiết, vì khảo sát thực tế trên toàn bộ quy trình 6 bước rút trích ontology, phương pháp chỉ giải quyết vấn đề liên quan đến ngôn ngữ khi xây dựng *vector đặc trưng* (mục 4.6), trong đó *từ khoá* trong văn bản được tính toán tần số xuất hiện và gán điểm. Tuy nhiên, vấn đề này hoàn toàn có thể giải quyết bằng phương pháp *So khớp từ dài nhất* (Longest matching) mà không phân biệt loại ngôn ngữ được xử lý. Chi tiết của giải pháp so khớp này sẽ được trình bày chi tiết ở mục 5.4.3.

## Quy trình 6 bước rút trích ontology từ WWW



Hình 4‑1 Kiến trúc tổng quát của hệ thống rút trích ontology từ web [[2](#Tim09)]

Phương pháp được nhóm tác giả đề nghị gồm 6 bước chính: *Chuẩn bị, Biến đổi, Gom cụm, Nhận diện, Liên kết* và *Tinh chỉnh*. Hình 4‑1 trình bày kiến trúc tổng quát của phương pháp do nhóm tác giả đề nghị. Hình 4‑2 biểu diễn trình tự quy trình 6 bước rút trích ontology từ WWW. Đầu tiên các trang web thuộc về một Website được tải về, loại bỏ những thẻ không phù hợp và được lưu trữ dưới dạng chuẩn hóa [[51](#AFu01)] và mô tả tóm tắt sử dụng những từ khoá ở bước *Chuẩn bị*. Bước *Biến đổi* thực hiện việc tinh chỉnh trang web bằng cách loại bỏ các *thành phần lặp* và xử lý các *đường dẫn*. Tiếp theo mỗi trang web được biểu diễn bằng một vector đại diện thể hiện đặc trưng nội dung của trang web đó, gọi là *vector đặc trưng*. Các trang web này sau đó được gom cụm dựa trên độ tương đồng giữa các *vector đặc trưng* của chúng ở bước *Gom cụm*. Mỗi cụm sau đó được nhận diện *đặc trưng cụm* bằng cách rút ra *vector đặc trưng cụm* của cụm đó thông qua quá trình *Nhận diện*. Ở bước *Liên kết*, *mối quan hệ* giữa các cụm được gán và tinh chỉnh dựa trên các *đường dẫn* giữa các trang web trong cụm. Cuối cùng, việc tinh chỉnh ontology được thực hiện bởi một chuyên gia xử lý ontology ở bước *Tinh chỉnh*. Chi tiết quy trình 6 bước này sẽ được trình bày ở các phần tiếp sau.



Hình 4‑2 Quy trình 6 bước rút trích ontology từ WWW [[2](#Tim09)]

## Chuẩn bị

Ở bước này, bộ dữ liệu cho toàn bộ phương pháp được xây dựng là tập các trang web thỏa các yêu cầu sau:

1. Cùng thuộc về một Website: chỉ những trang web thuộc về cùng một Website mới thể hiện cùng một chủ đề mà phương pháp quan tâm. Hai trang web được cho là cùng thuộc về một Website khi và chỉ khi đường dẫn gốc của chúng có cùng phần lĩnh vực.
2. Chỉ chứa văn bản: Trong các trang web thì phần duy nhất được quan tâm chính là văn bản và các thẻ chứa các văn bản đó. Do đó, để giảm kích thước lưu trữ các trang web, các thẻ HTML không chứa văn bản như , , sẽ được loại bỏ.
3. Được lưu trữ dưới dạng chuẩn hóa: Nhằm thuận lợi cho quá trình phân tích và xử lý cấu trúc thẻ của trang web, các trang web cần được chuyển đổi và lưu trữ dưới dạng chuẩn hóa. Dạng chuẩn hóa được nhóm tác giả chọn là ngôn ngữ XHTML.
4. Được gán nhãn từ khóa: Mỗi trang web được gán nhãn bằng một số từ khóa, nghĩa là tóm tắt nội dung trang web sử dụng một số từ khoá của trang web. Ở đây, nhóm tác giả sử dụng 3 nguồn từ khóa: phần văn bản trong các đường dẫn đến trang web, những từ khóa có trong các thẻ và tiêu đề trong thẻ của trang web.

* Nhận xét

Trong quá trình hiện thực hoá, chúng em nhận thấy việc lựa chọn từ khoá từ 3 nguồn để gán nhãn trang web của các tác giả là không cần thiết. Phần văn bản trong các *đường dẫn* đến trang web thường không chứa những từ khoá liên quan đến trang web như “xem tiếp”, “chi tiết” hay “xem thêm”…chính những từ khoá này có thể sẽ gây nhiễu nội dung gán nhãn của trang web. Vì vậy, chúng em chỉ lựa chọn những từ khoá từ thẻ và . Các từ khoá này sẽ có nội dung bám sát đặc trưng thông tin của trang web nhất.

Trên đây là các tiêu chuẩn của các trang web được lưu trữ trong kho dữ liệu gốc. Việc tuân thủ các tiêu chuẩn này sẽ giúp cho việc xử lý sau này được thuận lợi và chính xác hơn. Chi tiết việc hiện thực hóa bước *Chuẩn bị* được chúng em trình bày ở mục 5.2. Trong quá trình hiện thực hoá, một số vấn đề cụ thể khác cũng sẽ được giải quyết như tải các trang web về lưu trữ ngoại tuyến (mục 5.2.2), loại bỏ các trang web không hợp lệ (mục 5.2.3), chuẩn hoá các trang web (mục 5.2.4).

## Biến đổi

Các trang web chỉ chứa văn bản được duyệt và loại trừ các phần không phù hợp như đường dẫn hỏng hoặc các *thành phần lặp* như quảng cáo, nội dung bản quyền Website hay bảng danh mục Website...được lặp đi lặp lại qua các trang web. Các đường dẫn hỏng có thể được loại bỏ dễ dàng bằng cách kiểm tra sự tồn tại của trang web đích đến. Riêng việc loại bỏ các *thành phần lặp* thì không đơn giản. Trong phương pháp gốc, nhóm tác giải loại bỏ các *thành phần lặp* qua 2 bước:

* Gom nhóm các trang web có cùng cấu trúc. Các *thành phần lặp* thường xuất hiện giữa các trang web trong cùng một nhóm. Theo đó, người ta xây dựng tập các *chuỗi đường dẫn* của mỗi trang web. Một *tập chuỗi đường dẫn*  là một danh sách các *chuỗi đường dẫn* từ trang web đến trang web ,. C*huỗi đường dẫn* có thể được định nghĩa theo 2 cách như sau:
  + Là danh sách liên tiếp các đường dẫn trong đó , và .
  + Là danh sách liên tiếp các trang web trong đó , và .

Theo đó, 2 trang web được xem là cùng thuộc về một nhóm cấu trúc khi và chỉ khi tồn tại một *chuỗi đường dẫn* chung giữa hai trang web, nghĩa là .

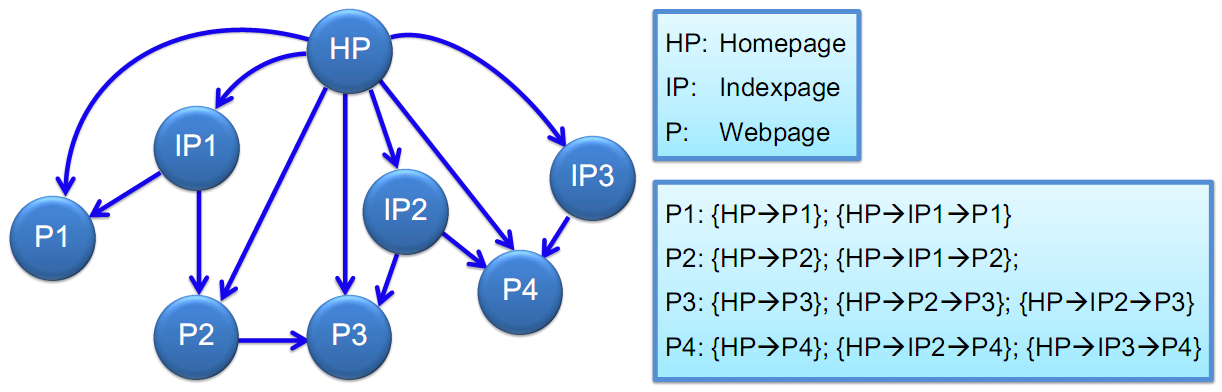
* Các trang web được chia thành 5 vùng là trên, dưới, trái, phải (phần biên) và giữa (phần trung tâm). Trong đó, nhóm phần biên thường là các *thành phần lặp.* Các *thành phần lặp* được loại bỏ bằng cách so sánh các phần biên tương ứng của các trang web trong cùng một nhóm, nếu phần biên nào trùng nhau thì sẽ được xóa khỏi trang web và các đường dẫn thuộc về phần biên này cũng được xóa khỏi cơ sở dữ liệu.
* Nhận xét

Sau khi khảo sát thực tế bước *Biến đổi*, chúng em nhận thấy phương pháp phân nhóm và loại bỏ *thành phần lặp* của các tác giả không thực sự hiệu quả và chỉ mang tính chất cục bộ, không tổng quát cho mọi Website. Trong đó :

* Phân nhóm

Phương pháp phân nhóm dựa trên *chuỗi đường dẫn* của nhóm tác giả thực sự không hiệu quả. Đối với các Website lớn với số lượng đường dẫn lên đến hàng triệu thì phương pháp trên vừa không hiệu quả về thời gian mà kết quả phân nhóm cũng không chính xác.

Trường hợp xấu nhất có thể xảy ra của phương pháp phân nhóm dựa vào *chuỗi đường dẫn* thì tất cả các trang web sẽ được phân vào một nhóm duy nhất như sẽ được trình bày ngay sau đây. Như vậy, kết quả rút trích ontology sẽ bị ảnh hưởng nghiêm trọng bởi các *thành phần lặp* đã không được loại bỏ triệt để.



Hình 4‑3 Phân nhóm dựa trên chuỗi đường dẫn

Hình 4‑3 trình bày ví dụ mô phỏng một tập hợp các trang web trong cùng một Website. Trong đó, HP là trang chủ, IP là các trang web chỉ mục của các chuyên đề khác nhau thuộc Website và P là các trang web tin tức. Dễ thấy, 2 trang web P1, P2 nên được phân vào cùng một nhóm và tương tự với P3, P4 vào một nhóm khác. Tuy nhiên, với phương pháp sử dụng *chuỗi đường dẫn* được đề nghị bởi nhóm tác giả, cả 4 trang web tin tức đều có một *chuỗi đường dẫn* chung trực tiếp từ trang chủ (HP). Như vậy, thay vì gom thành 2 nhóm khác nhau, 4 trang web lại được phân vào cùng một nhóm.

Từ phân tích trên, chúng em đề nghị một phương pháp mới gom nhóm các trang web có cùng cấu trúc dựa trên *đường dẫn* của mỗi trang web. Chi tiết của phương pháp được trình bày ở mục 5.3.2.

* Loại bỏ thành phần lặp

Phương pháp chia trang web thành 5 phân vùng để loại *thành phần lặp* chỉ mang tính chất cục bộ, không thể áp dụng cho mọi Website, vì mỗi Website khác nhau sẽ có thiết kế giao diện khác nhau. Từ đó, chúng em đề nghị phương pháp khác so sánh và tính toán độ tương đồng về cấu trúc của 2 trang web dựa trên việc phân tích cấu trúc thẻ của 2 trang web. Giải pháp loại bỏ *thành phần lặp* do chúng em đề nghị được trình bày chi tiết ở mục 5.3.4.

Ngoài ra, một số vấn đề khác cần được giải quyết khi hiện thực hóa bước *Biến đổi* cũng được chúng em trình bày chi tiết ở chương sau như loại bỏ các thành phần lặp giữa các trang web trong cùng một nhóm (mục 5.3.5), giảm số lượng trang web trong cùng một nhóm (mục 5.3.3) và lưu trữ các trang web vào hệ quản trị cơ sở dữ liệu (mục 5.3.6).

## Gom cụm

Mỗi trang web được biểu diễn bằng một *vector đặc trưng*. Mỗi vector này lượng hóa nội dung chứa bên trong trang web bằng cách gán điểm cho mỗi *từ khóa* xuất hiện trong trang web dựa vào loại thẻ HTML chứa *từ khóa* đó, như vậy, mỗi chiều của *vector đặc trưng* chính là giá trị điểm của *từ khóa* tương ứng. Sau khi xây dựng *vector đặc trưng* cho mỗi trang web, người ta tiến hành gom cụm các trang web dựa trên độ tương đồng giữa các *vector đặc trưng* này.

|  |  |  |
| --- | --- | --- |
| STT | Loại văn bản | Thẻ tương ứng |
| 1 | **linkText** | A (Thẻ chứa đường dẫn) |
| 2 | **pageText** | TITLE, META |
| 3 | **sectionText** | H1, H2, H3, H4, H5, H6 |
| 4 | **emphasizedText** | B, BIG, EM, I, STRONG, U |
| 5 | **plainText** | Các thẻ còn lại |

Bảng 4‑1 Phân loại văn bản cùng các thẻ tương ứng

Nhóm tác giả chia văn bản trong trang web thành 5 loại cụ thể với các phần tử HTML tương ứng (Bảng 3‑1). Mỗi loại văn bản có một giá trị trọng số được lưu trữ bằng *vector trọng số* , *vector trọng số* này được điều chỉnh bởi chuyên gia ontology. Mỗi *từ khóa* được tính tần số xuất hiện trong mỗi loại văn bản, người ta thu được *vector tần số* của từ khóa . Điểm của mỗi từ khóa được tính qua 3 bước như sau:

* : Tần số xuất hiện tính theo trọng số của từ khóa thứ i.
* : Điểm của *từ khóa* thứ i tính theo tỷ lệ với tổng tần số của các *từ khóa* của trang web.
* : Điểm của *từ khóa* thứ i tính theo tỷ lệ với chiều dài của *vector đặc trưng* tạm thời. Đây cũng chính là giá trị điểm cuối cùng của từ khóa.

Độ tương đồng giữa các vector đặc trưng được tính bằng giá trị góc tạo bởi hai vector đặc trưng:

* Nhận xét

Để so sánh độ tương đồng giữa hai trang web dựa vào góc xen giữa hai *vector đặc trưng*, hai vector này phải có cùng số chiều. Tuy nhiên, các *vector đặc trưng* có số chiều khác nhau tùy theo số lượng *từ khóa* của trang web. Do đó, sau khi xây dựng các *vector đặc trưng*, tất cả *vector đặc trưng* cần được biến đổi về cùng số chiều và các chiều này phải **tương ứng nhau** về nội dung *từ khóa*. Bước biến đổi *vector đặc trưng* này là cần thiết để có thể thực hiện quá trình gom cụm. Tuy nhiên, trong phương pháp của nhóm tác giả hoàn toàn không nhắc đến chi tiết này, vì vậy chúng em đề nghị phương pháp xây dựng *vector đặc trưng* mới dựa trên phương pháp của nhóm tác giả đồng thời vẫn đảm bảo được yêu cầu *cùng chiều* và *tương ứng từ khóa* bằng việc lựa chọn tập *từ khoá ứng viên* triển vọng ở mục 5.4.2. Ngoài ra, việc hiện thực hoá bước *Gom cụm* còn cần giải quyết một số vấn đề như hiện thực hóa việc xây dựng *vector đặc trưng* (mục 5.4.3) và gom cụm các *vector đặc trưng* (mục 5.4.4).

## Nhận diện

Ngoài các trang web đã được phân cụm, vẫn còn tồn tại những trang web chưa được phân vào bất cứ cụm nào. Những cụm có sẵn được nhận diện *đặc trưng cụm* bằng cách tìm ra vector đại diện của cụm đó. Các trang web chưa được phân cụm sẽ được phân vào các cụm có sẵn dựa vào độ tương đồng giữa vector của trang web và vector đại diện của cụm.

* Nhận xét

Kết quả quan trọng nhất của bước *Nhận diện* chính là *đặc trưng cụm*. Từ *đặc trưng cụm* sẽ xây dựng được danh sách các *từ khóa ứng viên* cho các *khái niệm*. Tuy nhiên, nhóm tác giả không nhắc đến chi tiết để rút trích *đặc trưng cụm*. Khi áp dụng vào thực tế, chúng em nhận diện *đặc trưng cụm* bằng cách xây dựng *vector trung bình* của các *vector đặc trưng* con trong cụm. Chi tiết phương pháp xây dựng vector đặc trưng cụm và xử lý các phần tử ngoại lai được trình bày chi tiết ở mục 5.5.

## Liên kết

Tới đây, hầu hết các trang web đã được phân cụm, và danh sách các *từ khoá ứng viên* của các *khái niệm* có đã được rút trích dựa vào các *đặc trưng cụm*. *Mối quan hệ* giữa các *khái niệm* được gán thông qua việc xem xét các đường dẫn giữa các trang web trong các cụm. Theo đó, hai cụm - đại diện cho hai *khái niệm* - được cho là có quan hệ với nhau khi có một *đường dẫn* giữa hai trang web thuộc về hai cụm đó. Một *mối quan hệ*  bao gồm 2 *khái niệm*  và và mối quan hệ giữa 2 khái niệm đó.

Ở đây, nhóm tác giả có 2 nhận định quan trọng nhằm xác định *mối quan hệ* giữa các *khái niệm* thông qua các *đường dẫn*:

* Các mối quan hệ có tính chất *duy nhất* và *đối xứng*. Nếu tồn tại và thì 2 *mối quan hệ* này là tương đương nhau, và chỉ duy nhất một *mối quan hệ* được giữ lại.
* Các mối quan hệ có tính chất bắc cầu. Nếu tồn tại , và thì sẽ bị xóa, vì hoàn toàn có thể được suy ra từ 2 mối quan hệ còn lại.
* Nhận xét

Từ 2 nhận định của nhóm tác giả, có thể thấy được các *mối quan hệ* giữa các cụm hầu như đã được xác định một cách khá rõ ràng. Tuy nhiên, một lượng lớn mối quan hệ đã bị loại bỏ một cách lãng phí do phải tuân thủ các tính chất *duy nhất, đối xứng* và *bắc cầu*. Chúng em đề nghị giải pháp tận dụng những *mối quan hệ* này nhằm đánh trọng số cho các *mối quan hệ* được lưu giữ lại. Nhờ vào các *mối quan hệ* được đánh trọng số, người chuyên gia ontology sẽ có nhiều thông tin hơn trong việc tinh chỉnh các mối quan hệ. Giải pháp này sẽ được trình bày chi tiết ở mục 5.6.

## Tinh chỉnh

Vai trò của người chuyên gia về ontology trong phương pháp rút trích ontology từ WWW là rất quan trọng. Vì đây chỉ là một phương pháp bán tự động, người chuyên gia phải tham gia giám sát toàn bộ quá trình rút trích ontology. Công việc của người chuyên gia chính là tinh chỉnh các tham số của phương pháp, các tham số này sẽ được đề cập rõ hơn ở suốt các mục trong Chương 5 và được thống kê cụ thể ở mục 5.7. Ngoài ra, người chuyên gia còn đóng vai trò tinh chỉnh, chỉnh sửa và lựa chọn các *khái niệm* và *mối quan hệ* từ tập những *từ khóa ứng viên* sao cho phù hợp với mục đích sử dụng của ontology nhất. Chi tiết các tham số cần được điều chỉnh bởi chuyên gia ontology sẽ được trình bày chi tiết khi được hiện thực hoá ở mục 5.7.

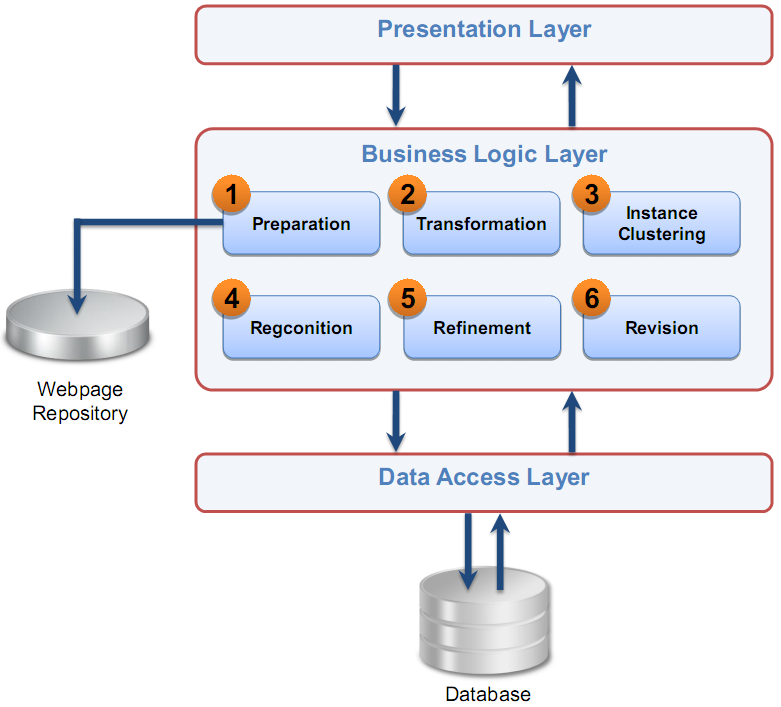
## Kết luận

Trong chương 4 này, chúng em đã trình bày, tóm tắt và phân tích quy trình rút trích ontology từ WWW của nhóm tác giả Du C. Timon, Li Feng, và King Irwin đề xuất [[2](#Tim09)]. Quy trình này gồm 6 bước Chuẩn bị, Biến đổi, Gom cụm, Nhận Diện, Liên kết và Tinh chỉnh. Dữ liệu đầu vào cho quy trình là các trang web của cùng một Website, và kết quả có được là một ontology. Dựa trên việc phân tích quy trình này, chúng em xây dựng framework nhằm hiện thực hóa và hỗ trợ khả năng sử dụng tùy biến cao trong việc áp dụng và kiểm thử quy trình này. Chi tiết kiến trúc Ontology Extractor Framework được trình bày ở chương tiếp theo đây.

# Ontology Extractor Framework rút trích ontology từ WWW

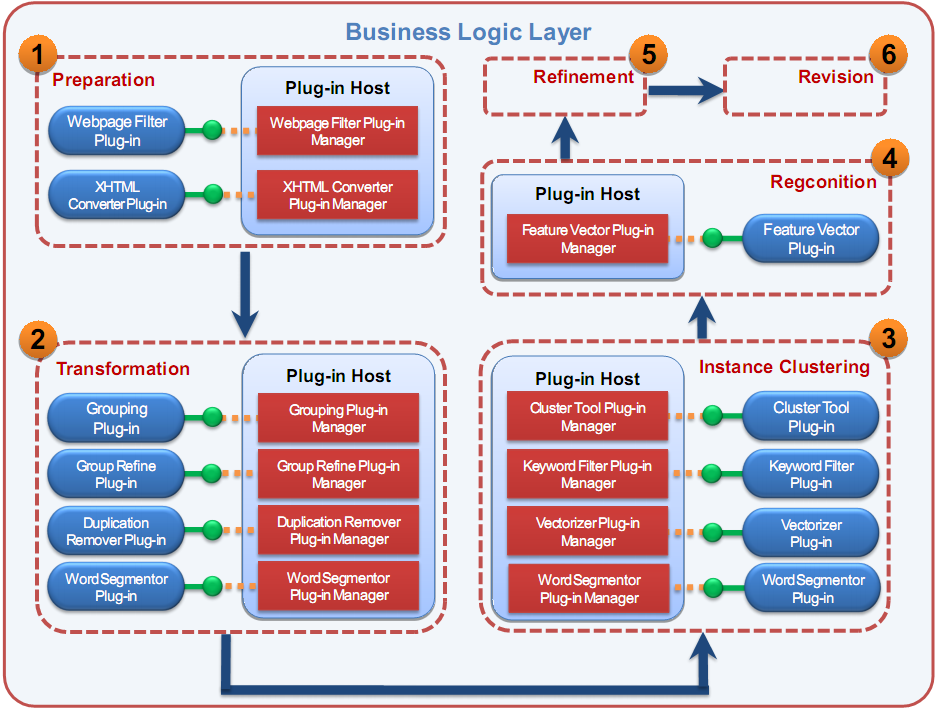
* Nội dung Chương 5 trình bày hệ thống framework được nhóm phát triển để hiện thực hóa quy trình rút trích ontology từ WWW đã được trình bày ở Chương 4. Các vấn đề và giải pháp khi xây dựng framework và hiện thực hóa quy trình cũng được trình bày trong chương này.

## Kiến trúc hệ thống



Hình 5‑1 Kiến trúc tổng quát Ontology Extractor Framework

Hình 5‑1 trình bày kiến trúc tổng quát của framework rút trích ontology từ WWW. Hệ thống được thiết kế theo mô hình 3 lớp, trong đó các phân hệ xử lý chính nằm ở tầng giữa gồm 6 phân hệ: Phân hệ Chuẩn bị (Preparation), Phân hệ Chuyển đổi (Transformation), Phân hệ Gom cụm (Instance Clustering), Phân hệ Nhận diện (Regconition), Phân hệ Liên kết (Refinement) và Phân hệ Tinh chỉnh (Revision). Kiến trúc chi tiết của các phân hệ chính được trình bày ở Hình 5‑2.



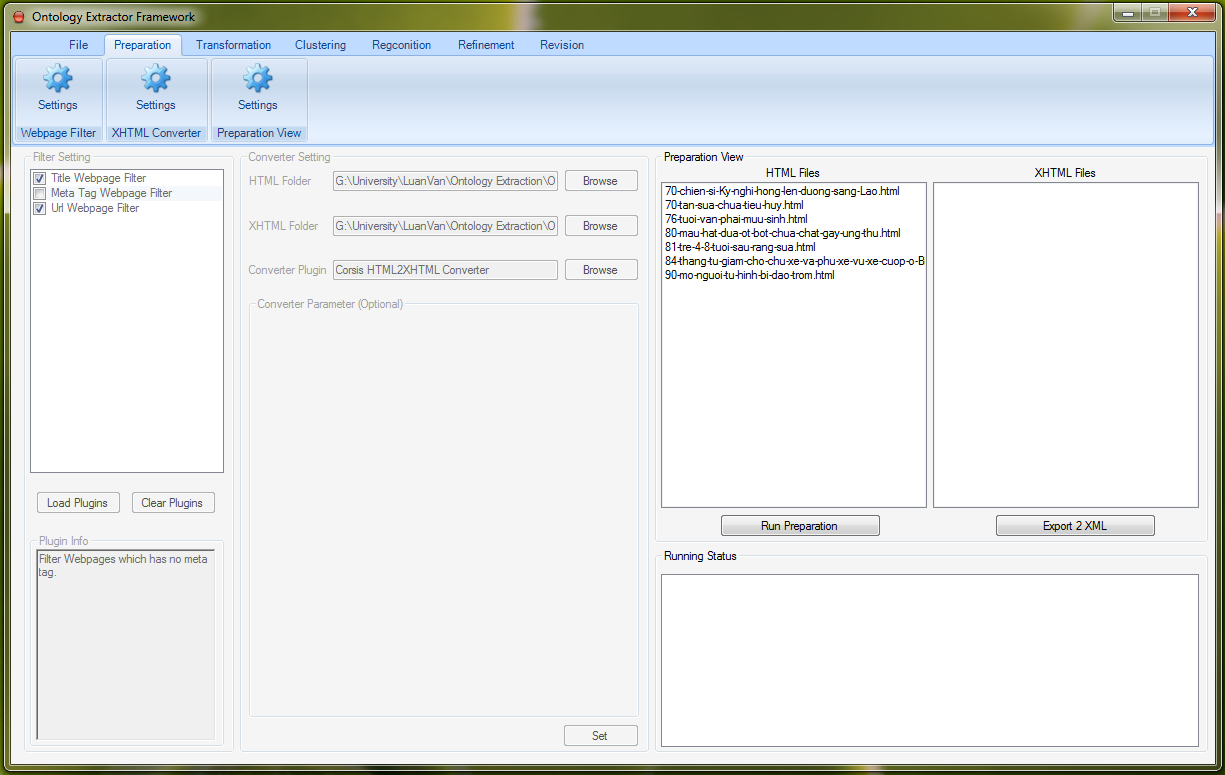
Hình 5‑2 Kiến trúc các phân hệ chính của Ontology Extractor Framework

Hình 5‑2 trình bày kiến trúc các phân hệ chính của framework rút trích ontology từ WWW. Trong đó:

1. Phân hệ Chuẩn bị (Preparation): hiện thực hóa bước *Chuẩn bị* nhằm xử lý các trang web làm dữ liệu đầu vào cho toàn bộ hệ thống. Trong đó thành phần chính sử dụng kiến trúc Plug-in hỗ trợ khả năng tùy biến cho 2 tác vụ chính của bước này là loại bỏ trang web không phù hợp và chuẩn hóa trang web.
2. Phân hệ Biến đổi (Transformation): hiện thực hóa bước *Biến đổi* với các tác vụ là gom nhóm các trang web theo cấu trúc, loại bỏ các trang web dư thừa và các thành phần lặp. Các tác vụ chính này được quản lý bằng cách sử dụng kiến trúc Plug-in hỗ trợ khả năng tùy biến cao. Ngoài ra, trong phân hệ này còn hỗ trợ tác vụ lưu trữ các trang web vào cơ sở dữ liệu quan hệ (mục 5.3.6).
3. Phân hệ Gom cụm (Instance Clustering): hiện thực hóa bước *Gom cụm* có nhiệm vụ chính là gom cụm các trang web dựa trên độ tương đồng giữa các *vector đặc trưng* của trang web. Phân hệ này cũng sử dụng kiến trúc Plug-in quản lý các tác vụ chính bao gồm chọn *từ khóa* làm chiều của các *vector đặc trưng*, xây dựng *vector đặc trưng* và gom cụm.
4. Phân hệ Nhận diện (Regconition): hiện thực hóa bước *Nhận diện* có hỗ trợ khả năng tùy biến trong việc rút trích đặc trưng cụm sử dụng kiến trúc Plug-in.
5. Phân hệ Liên kết (Refinement): hiện thực hóa bước *Liên kết* nhằm thực hiện tác vụ xác định *mối quan hệ* giữa các *khái niệm* - được đại diện bằng một cụm.
6. Phân hệ Tinh chỉnh (Revision): hiện thực hóa bước *Tinh chỉnh* nhằm hỗ trợ chuyên gia ontology trong việc lựa chọn các *từ khóa ứng viên* của các *khái niệm*.

Chi tiết về kiến trúc cũng như các vấn đề và giải pháp trong quá trình hiện thực hóa của từng phân hệ được chúng em trình bày trong các phần tiếp sau đây.

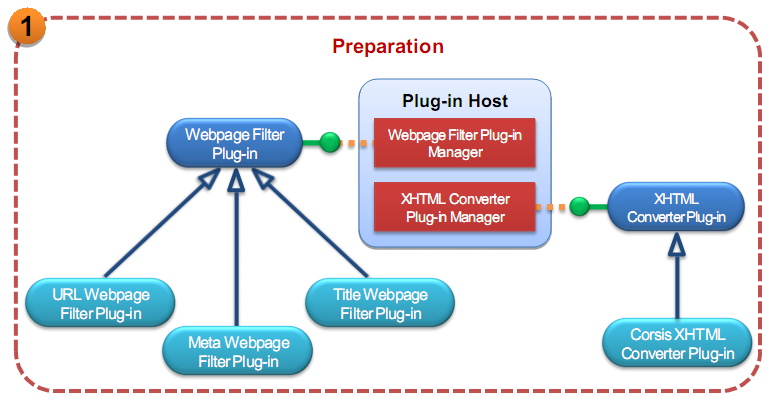
## Phân hệ Chuẩn bị (Preparation)



Hình ‑ Giao diện Phân hệ Chuẩn bị

### Kiến trúc phân hệ

Phân hệ Chuẩn bị hiện thực hóa bước đầu tiên trong quy trình 6 bước nhằm xử lý các trang web đầu vào cho toàn bộ hệ thống. Các trang web này cần được xử lý sao cho thỏa những yêu cầu như phải cùng thuộc về một Website, chỉ chứa văn bản, được lưu trữ dưới dạng chuẩn hoá và được gán nhãn từ khoá. Riêng việc gán nhãn từ khoá các trang web, nguồn từ khoá được lựa chọn từ thẻ meta hoặc tách từ tiêu đề của trang web. Các từ khoá này bám sát nội dung thông tin trang web hơn hẳn so với phần văn bản của các đường dẫn đến trang web vốn thể hiện những nội dung thông tin không liên quan. Đây cũng chính là một trong những điểm khác biệt giữa việc hiện thực hóa quy trình của chúng em và nội dung lý thuyết được các tác giả đề nghị. Kiến trúc của phân hệ sử dụng cơ chế plug-in được trình bày ở Hình 5‑4.



Hình 5‑4 Kiến trúc Phân hệ Chuẩn bị

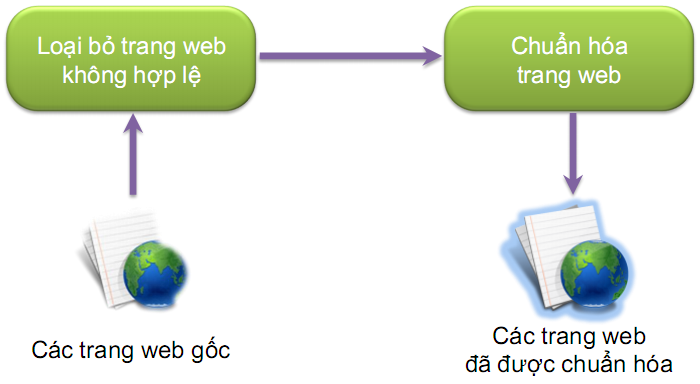
Hình 5‑4 mô tả kiến trúc của phân hệ Chuẩn bị. Phân hệ gồm một host quản lý các plug-in và có 2 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho mỗi loại plugin:

* Webpage Filter Plug-in: đóng vai trò là bộ lọc trang web không thỏa yêu cầu. (3 thể hiện)
* XHTML Converter Plug-in: là công cụ chẩn hóa trang web. (1 thể hiện)

Bảng 5‑1 Các lớp chính cấu thành phân hệ Chuẩn bị trình bày các lớp đối tượng chính cấu thành phân hệ Chuẩn bị.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | IPreparationHost | Interface mô tả host quản lý các plug-in trong phân hệ Chuẩn bị |
| 2 | PreparationHost | Thể hiện host của phân hệ Chuẩn bị |
| 3 | IwebpageFilter | Interface mô tả plug-in bộ lọc trang web |
| 4 | MetaWebpageFilterPlugin | Bộ lọc các trang web không chứa meta data |
| 5 | UrlWebpageFilterPlugin | Bộ lọc các trang web không chứa đường dẫn gốc |
| 6 | TitleWebpageFilterPlugin | Bộ lọc các trang web không chứa tiêu đề |
| 7 | IXHTMLConverter | Interface mô tả plug-in chuẩn hóa trang web |
| 8 | CorsisXHTMLConverterPlugin | Plug-in chuẩn hóa trang web sử dụng thư viện Corsis chuyển đổi sang định dạng XHTML |

Bảng 5‑1 Các lớp chính cấu thành phân hệ Chuẩn bị



Hình 5‑5 Mô hình xử lý tổng quát Phân hệ Chuẩn bị

Hình 5‑5 mô tả quy trình xử lý tổng quát của phân hệ Chuẩn bị. Đầu tiên, các trang web gốc được đưa qua bộ lọc để loại bỏ những trang web không phù hợp như trang web không có meta data hoặc không có tiêu đề (5.2.3). Sau đó, chúng được chuẩn hóa bằng cách chuyển đổi định dạng sang XHTML (5.2.4). Kết quả của phân hệ này là tập các trang web thỏa yêu cầu của người sử dụng và đã được chuẩn hóa.

### Tải các trang web về lưu trữ ngoại tuyến

* Vấn đề:

Do mục đích của hệ thống là rút trích ontology từ một *Website mục tiêu*, nên đối tượng xử lý chính của phương pháp là các trang web của Website đó. Mặt khác, số lượng các trang web là rất lớn: việc xử lý ở đây mang tính chất hàng loạt, nghĩa là toàn bộ các trang web phải luôn sẵn sàng cho việc xử lý. Do đó, để đảm bảo tính liên tục, không bị ngắt quãng của hệ thống, từ đó phát sinh nhu cầu tải toàn bộ các trang web thuộc về cùng một Website để lưu trữ ngoại tuyến. Việc tải các trang web này tạo ra tập các trang web gốc được phân hệ Chuẩn bị sử dụng như bộ dữ liệu đầu vào.

* Giải pháp:

Giải pháp được đề ra là sử dụng một công cụ crawler nhằm tải toàn bộ các trang web của Website về lưu trữ trên máy tính. Hiện nay, có rất nhiều công cụ crawler như: OfflineExplorer, Teleport, HTTrack....Có 2 hướng tiếp cận cho việc sử dụng công cụ crawler như sau:

* Sử dụng công cụ crawler mã nguồn mở: có thể sử dụng các chương trình mã nguồn mở có sẵn và chỉnh sửa, bổ sung thêm những chức năng cần thiết phù hợp với đặc thù công việc.
* Sử dụng công cụ crawler có bản quyền: với các phần mềm này, không thể chỉnh sửa mã nguồn mà chỉ có thể sử dụng các tùy chọn có sẵn của chương trình nhằm chuẩn bị một cách thuận lợi nhất cho công việc.

Lựa chọn của nhóm là sử dụng Offline Explorer (OE) – một crawler có bản quyền của Metaproducts. OE có những ưu điểm sau:

* Tùy chọn loại tập tin, nội dung tập tin được tải: ởđây, tùy chọn được quan tâm là *chỉ tải văn bản (text)* của các trang web.
* Multi Thread: OE hỗ trợ tải nhiều trang web cùng một lúc, điều này giúp tăng tốc độ thu thập dữ liệu cho hệ thống.
* Tùy chọn Server: OE hỗ trợ chỉ tải những trang web có đường dẫn (URL) thuộc về cùng một server. Tùy chọn này rất phù hợp với mục đích của đề tài, đó là rút trích ontology từ những trang web thuộc cùng một *lĩnh vực*.
* Lưu giữ được đường dẫn (URL) gốc: một nhu cầu thiết yếu khác đó là phải lưu giữ được đường dẫn gốc của các trang web. Đường dẫn gốc hỗ trợ hệ thống phân loại trang web được hiệu quả hơn.
* Tạo ra một phiên bản*“phẳng”*của Website: OE hỗ trợ kết xuất tất cả các trang web vào một thư mục đồng thời chỉnh sửa tất cả các đường dẫn (href) nhằm đảm bảo mối liên kết giữa tất cả các trang web như ban đầu. Điều này tạo thuận lợi cho quá trình xử lý và duyệt qua các trang web.

Phiên bản được chúng em sử dụng trong cài đặt là Offline Explorer Enterprise 5.4.2958.

### Loại bỏ trang web không hợp lệ

* Vấn đề

Do đặc thù của công cụ crawler, một số trang web không phù hợp vẫn được tải về như các trang web thuộc Website khác được gọi gián tiếp từ *Website mục tiêu*, các trang web không chứa dữ liệu meta hoặc các trang web không thể lưu trữ được đường dẫn gốc.

Mặt khác, số lượng trang web được lưu trữ và xử lý là rất lớn, vì vậy hệ thống cần được tối ưu hoá về thời gian và không gian lưu trữ. Chính vì vậy, hệ thống cần lọc bỏ những trang web không phù hợp nhằm giảm số lượng trang web sẽ được xử lý ở những bước tiếp theo cũng như loại bỏ những trang web không thỏa mãn yêu cầu xử lý của các tác vụ thuộc những phân hệ tiếp theo.

* Giải pháp

Việc sử dụng các công cụ crawler khác nhau sẽ phát sinh những trang web không phù hợp khác nhau đối với nhóm những Website khác nhau, vì thế hệ thống phải có khả năng hỗ trợ người dùng loại bỏ những trang web không phù hợp theo những tiêu chí khác nhau dành cho các nhóm Website khác nhau.

Giải pháp được đề ra là xây dựng một bộ lọc dữ liệu hỗ trợ các tuỳ chọn có sẵn để người dùng có thể loại bỏ những trang web không phù hợp. Ngoài ra, bộ lọc này còn có khả năng mở rộng và phát triển thêm các tuỳ chọn mới sau này. Vì vậy, kiến trúc Plug-in được sử dụng nhằm hỗ trợ khả năng tùy biến trong việc lựa chọn các bộ lọc. Bảng sau đây trình bày các bộ lọc mà chúng em đã xây dựng và triển khai.

|  |  |  |
| --- | --- | --- |
| STT | Bộ lọc trang web | Mô tả |
| 1 | MetaWebpageFilterPlugin | Bộ lọc các trang web không chứa meta data |
| 2 | UrlWebpageFilterPlugin | Bộ lọc các trang web không chứa đường dẫn gốc |
| 3 | TitleWebpageFilterPlugin | Bộ lọc các trang web không chứa tiêu đề |

Bảng 5‑2 Các bộ lọc trang web

### Chuẩn hoá các trang web

* Vấn đề:

Phương pháp của đề tài tập trung vào việc phân tích, xử lý nội dung các trang web dựa vào cấu trúc các thẻ. Phần lớn các Website Việt Nam hiện nay đều được viết bằng HTML không chuẩn hóa, vì vậy gây rất nhiều khó khăn hoặc thậm chí ảnh hưởng đến chất lượng đầu ra của hệ thống.

* Giải pháp:

Một trong những ngôn ngữ đánh dấu (Markup Language) được tổ chức W3C khuyến khích các nhà phát triển Website sử dụng là XHTML. Kết hợp những ưu điểm vượt trội của XML vào HTML, XHTML ra đời. XHTML mang những tính chất chuẩn mực của XML (<http://www.w3schools.com/xhtml/default.asp>) như:

* Các thẻ XHTML phải được lồng nhau đúng cách.
* Các thẻ XHTML mở phải luôn có thẻ đóng đi kèm.
* Tên các thẻ XHTML phải được viết bằng chữ thường.
* Mỗi văn bản XHTML phải có một phần tử gốc duy nhất.

Vì thế, nhu cầu của chúng em chính là chuyển đổi các trang web nguồn được viết bằng HTML thành XHTML chuẩn; điều này sẽ giúp ích cho quá trình xử lý và phân tích sau này. Bước chuyển đổi phải tuân thủ những yêu cầu như sau:

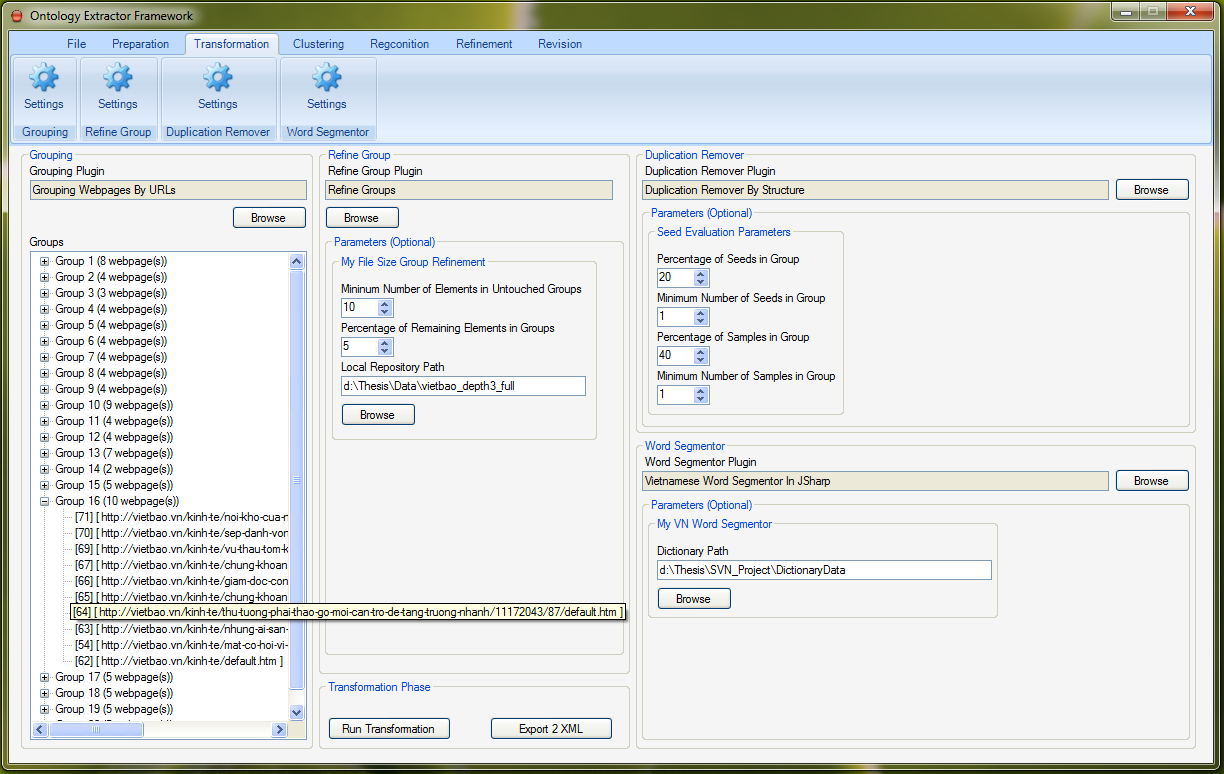
* Cách sửa lỗi các trang web phải giữ lại và đảm bảo được *mối tương quan đúng* giữa các thẻ.
* Các trang web có lỗi giống nhau phải được chuyển đổi giống nhau. Như vậy, các trang có cấu trúc thẻ giống nhau sau bước chuyển đổi vẫn giữ nguyên được mối liên quan về cấu trúc này.

Ngoài ra, công nghệ phát triển Website ngày nay là xây dựng hàng loạt các trang web theo một khuôn mẫu (template) hoặc bộ khung (layout) có sẵn. Như vậy, các lỗi định dạng HTML thiếu chuẩn mực thường mang tính *hàng loạt*. Từ nhận định trên, việc nhóm thực hiện việc chuyển đổi từ HTML sang XHTML cho tất cả các trang web có được mà vẫn đảm bảo sự tương quan cấu trúc các thẻ ban đầu.

Hiện nay có khá nhiều công cụ chuyển đổi định dạng HTML sang XHTML,và công cụ opensource Corsis (<http://sourceforge.net/projects/corsis/>) được chúng em lựa chọn để thực hiện việc chuyển đổi định dạng các trang web từ HTML sang XHTML chuẩn hoá.

Phân hệ Chuẩn bị có hỗ trợ tùy biến plug-in cho công cụ chuẩn hóa các trang web. Để có thể sử dụng công cụ Corsis thực hiện tác vụ chuẩn hóa trang web, chúng em xây dựng plug-in CorsisXHTMLConverterPlugin sử dụng dll của Corsis.

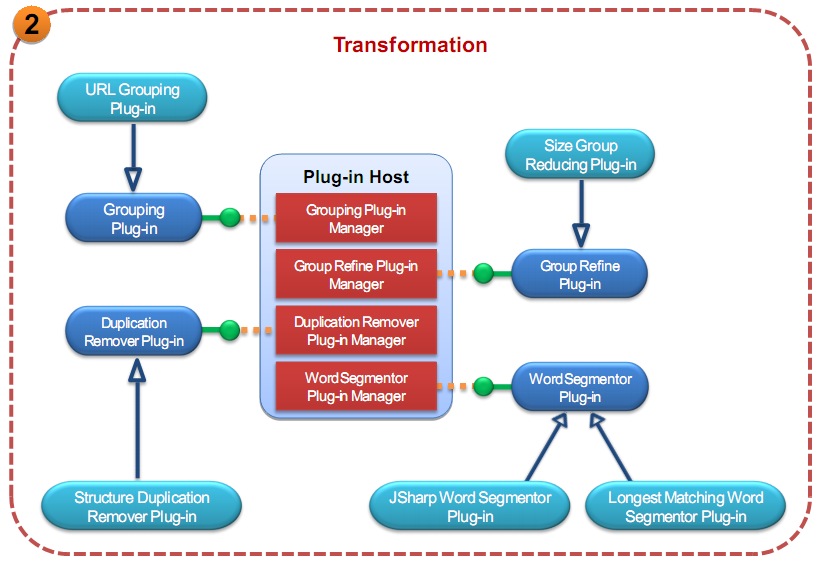
## Phân hệ Biến đổi (Transformation)



Hình ‑ Giao diện phân hệ Biến đổi

### Kiến trúc phân hệ

Các *thành phần lặp* là các phần được lặp đi lặp lại qua các trang web như quảng cáo, thông tin bản quyền Website hay thanh danh mục Website…Các *thành phần lặp* này không mang đặc trưng riêng của mỗi trang web, trong khi mục đích của phương pháp là rút trích ontology từ việc phân tích đặc trưng của các trang web, vì vậy các *thành phần lặp* này cần được loại bỏ trước khi tiến hành các bước phân tích và thống kê tiếp theo. Các *thành phần lặp* thường lặp lại ở các vị trí giống nhau giữa các trang web trong cùng một nhóm có cùng cấu trúc HTML. Và việc loại bỏ các *thành phần lặp* chính là nhiệm vụ chính của phân hệ này. Kiến trúc của phân hệ sử dụng cơ chế plug-in được trình bày ở Hình 5‑7.



Hình 5‑7 Kiến trúc Phân hệ Biến đổi

Hình 5‑7 mô tả kiến trúc của phân hệ Chuẩn bị. Phân hệ gồm một host quản lý các plug-in và có 4 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho mỗi loại plugin:

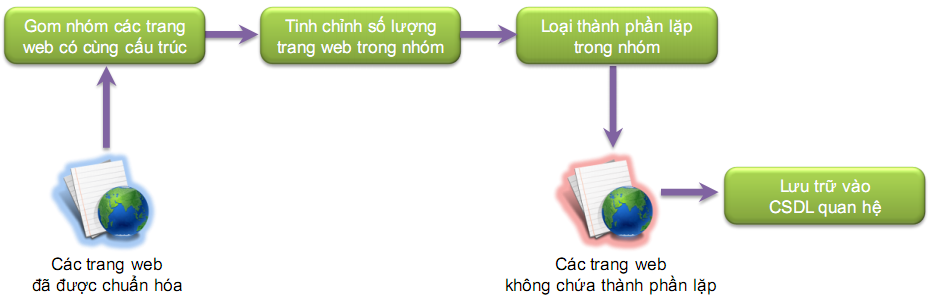
* Grouping Plug-in: gom nhóm các trang web tương đồng về cấu trúc.  
  (1 thể hiện)
* Group Refine Plug-in: tinh chỉnh số lượng trang web của nhóm.  
  (1 thể hiện)
* Duplication Remover Plug-in: loại trừ thành phần lặp giữa các trang web trong cùng một nhóm. (1 thể hiện)
* Word Segmentor Plug-in: tách từ tiếng Việt. (2 thể hiện)

Bảng 5‑3 Các lớp chính cấu thành phân hệ Biến đổi trình bày các lớp đối tượng chính cấu thành phân hệ Biến đổi.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | ItransformationHost | Interface mô tả host quản lý các plug-in trong phân hệ Biến đổi |
| 2 | TransformationHost | Thể hiện host của phân hệ Biến đổi |
| 3 | IwebpageGrouping | Interface mô tả plug-in gom nhóm các trang web |
| 4 | UrlGroupingPlugin | Plugin gom nhóm các trang web dựa trên đường dẫn gốc |
| 5 | IgroupReducing | Interface mô tả plug-in tinh chỉnh số lượng nhóm |
| 6 | SizeGroupReducingPlugin | Plugin tinh chỉnh số lượng nhóm dựa trên kích thước trang web |
| 7 | IduplicationRemover | Interface mô tả plug-in loại thành phần lặp |
| 8 | StructureDuplicationRemoverPlugin | Plugin loại thành phần lặp dựa trên sự tương đồng về cấu trúc trang web |
| 9 | IwordSegmentor | Interface mô tả plug-in tách từ tiếng Việt |
| 10 | JsharpWordSegmentorPlugin | Plug-in tách từ tiếng Việt sử dụng bộ thư viện Jsharp |
| 11 | LongestMatchingWordSegmentorPlugin | Plug-in tách từ tiếng Việt bằng phương pháp so khớp từ dài nhất |

Bảng 5‑3 Các lớp chính cấu thành phân hệ Biến đổi

Hình 5‑8 mô tả quy trình bước *chuẩn bị*, đầu tiên các trang web được gom nhóm dựa sự giống nhau về cấu trúc (5.3.2). Các nhóm này sau đó được tinh chỉnh số lượng các trang web trong nhóm (5.3.3). Kế tiếp, lần lượt các nhóm được loại bỏ *thành phần lặp* giữa các trang web trong nhóm (5.3.4 và 5.3.5). Cuối cùng, các trang web được lưu trữ vào cơ sở dữ liệu quan hệ (5.3.6).



Hình 5‑8 Mô hình xử lý tổng quát Phân hệ Biến đổi

### Gom nhóm các trang web dựa trên đường dẫn gốc

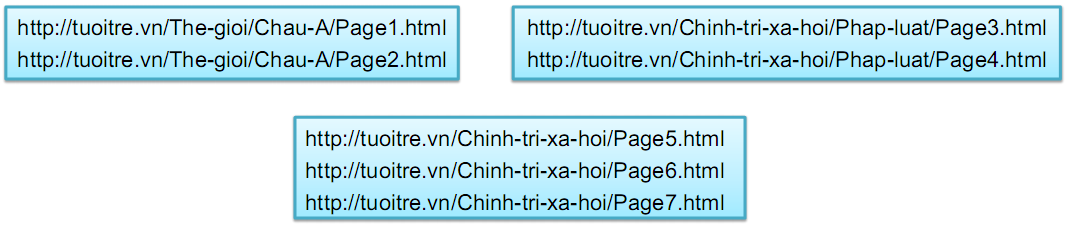
* Vấn đề

Phương pháp gom nhóm các trang web sử dụng *chuỗi đường dẫn* do các tác giả đề nghị đã được phân tích và chỉ ra những hạn chế ở mục 4.5, từ đó chúng em đề ra phương pháp gom nhóm dựa trên đường dẫn (URL) của trang web. Đường dẫn phản ánh sự lưu trữ theo phân loại nội dung, chuyên mục của các trang web trên máy chủ. Mặt khác, các trang web thuộc về cùng một chuyên mục thường được tạo ra bởi cùng một cấu trúc khuôn mẫu (template) cho nên sẽ có nhiều sự tương đồng về cấu trúc HTML. Ý tưởng phân nhóm các trang web dựa trên đường dẫn được xây dựng trên nền tảng những nhận xét và phân tích trên.

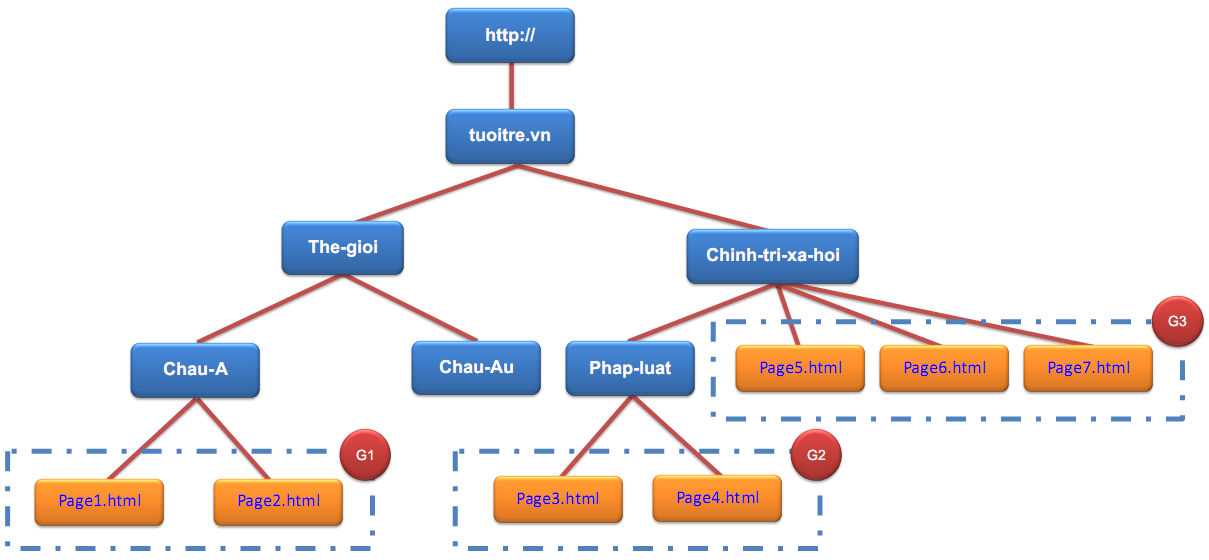
* Giải pháp

Chúng em sử dụng các thành phần của đường dẫn được phân tách bởi các dấu “**/**” để xây dựng nên cây đường dẫn như Hình 5‑10. Trong đó, mỗi một thành phần đường dẫn được xem như là một node trên cây và lá là tên các trang web. Theo đó, các trang web được đại diện bằng các node lá trên cây đường dẫn.

Dễ thấy, các trang web có phần đường dẫn giống nhau sẽ có phần tên thuộc về cùng một node cha. Như vậy, các trang web được xem cùng thuộc một nhóm khi chúng cùng thuộc về một node cha trên *cây đường dẫn*.



Hình 5‑9 Các đường dẫn gốc được phân nhóm



Hình 5‑10 Cây đường dẫn

Hình 5‑10 biểu dẫn *cây đường dẫn* xây dựng từ tập 7 đường dẫn ở Hình 5‑9. Trong đó, các trang *Page*1*.html* và *Page*2*.html* cùng thuộc về chuyên mục *Châu Á* trong nhánh *Thế giới* được phân vào *Group* 1. Các trang *Page*5*.html, Page*6*.html* và *Page*7*.html* thuộc về chuyên mục *Chính trị xã hội* nói chung được phân vào *Group* 3. Và các trang *Page*3*.html, Page*4*.html* thuộc về chuyên mục *Pháp luật* được phân thành một nhóm *Group* 2.

Phân hệ Biến đổi hỗ trợ việc tùy biến phương pháp gom nhóm các trang web có cùng cấu trúc. Trong quá trình hiện thực hóa, chúng em đã triển khai plug-in gom nhóm các trang web dựa vào đường dẫn gốc của chúng UrlGroupingPlugin.

### Tinh chỉnh số lượng các trang web trong một nhóm

* Vấn đề

Sau khi phân nhóm các trang web, khả năng tồn tại một số nhóm có số lượng phần tử không cân đối như có quá nhiều hoặc quá ít các trang web thuộc về nhóm. Các nhóm này gây khó khăn cho các bước xử lý tiếp theo. Từ đây xuất phát nhu cầu cần tinh chỉnh các nhóm này sao cho thuận lợi và phù hợp với các xử lý sau đó.

* Giải pháp

Phân hệ này hỗ trợ việc tùy biến phương pháp tinh chỉnh số lượng các trang web trong cùng một nhóm. Trong quá trình hiện thực hóa, chúng em đã triển khai plug-in tinh chỉnh nhóm các trang web SizeGroupReducingPlugin. Trong đó, phương pháp tinh chỉnh nhóm dựa trên đặc thù của phương pháp gom nhóm mà chúng em phát triển:

* Các trang web trong cùng nhóm thuộc về cùng một chuyên mục nên có nội dung tương tự nhau. Vì vậy, phương pháp tinh chỉnh sẽ chỉ giữ lại các trang web có kích thước lớn nhất trong nhóm. Nghĩa là, chỉ các trang web chứa nhiều nội dung thông tin của một chuyên mục sẽ được giữ lại.
* Các nhóm chỉ có một phần tử sẽ được gộp với nhóm gần nó nhất. Vì vậy, không tồn tại các nhóm có một phần tử. Do đó, các phương pháp tinh chỉnh sẽ bỏ qua các nhóm có số phần tử ít nhất là 2. Vì với một nhóm có 2 phần tử, các *thành phần lặp* vẫn có thể được loại trừ.

### Loại trừ các thành phần lặp giữa 2 trang web

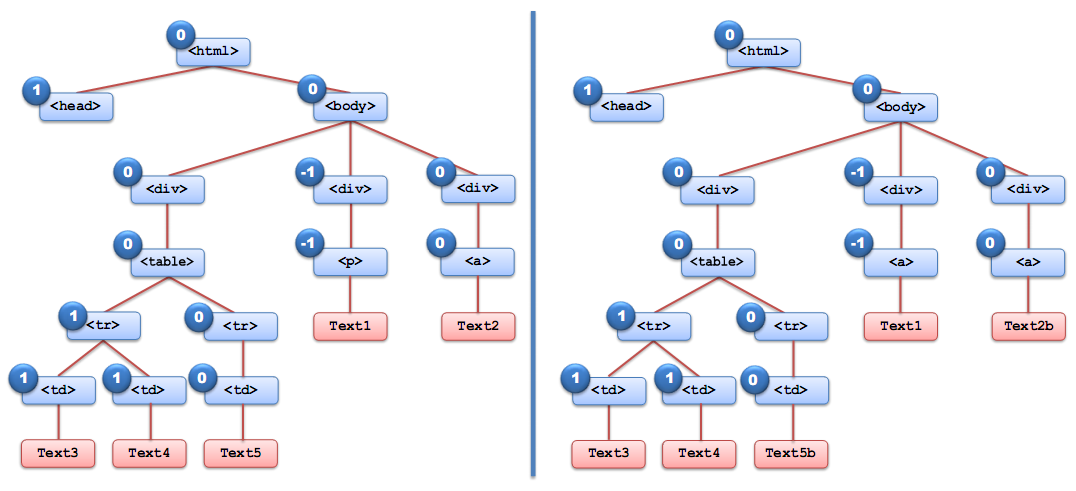
* Vấn đề:

Như đã được phân tích ở mục 4.5, phương pháp so sánh và loại *thành phần lặp* của nhóm tác giả bằng cách chia trang web thành 5 phần trên, dưới, trái phải (vùng biên) và giữa (vùng giữa) là không tổng quát và thiếu tính chính xác. Từ đó, chúng em có nhu cầu đề ra phương pháp so sánh độ tương đồng cấu trúc nhằm loại bỏ một cách chính xác hơn các *thành phần lặp*.

* Giải pháp

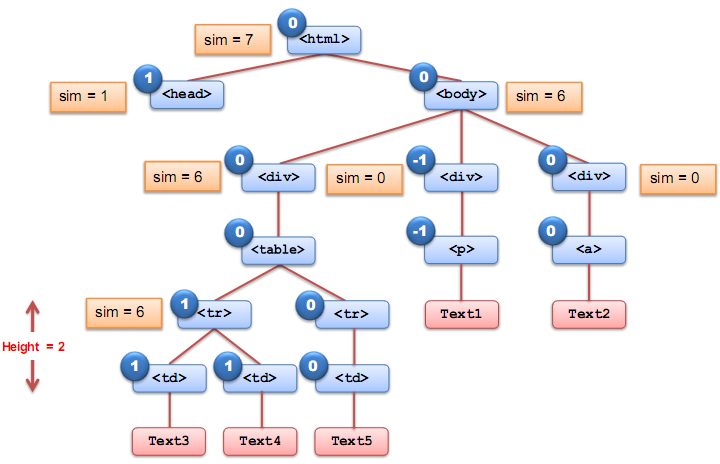
Như đã được phân tích ở mục 5.3.6, cấu trúc thẻ của một trang web hoàn toàn có thể được khôi phục từ xpath của các thẻ chứa văn bản trong trang web. Theo đó, trên cây cấu trúc của một trang web gồm nhiều node tương ứng với các thẻ trong trang web. Một trang web được xem như một node gồm nhiều con. Để đánh dấu độ tương đồng cấu trúc giữa 2 trang web, chúng em sử dụng một số luật sau:

* Hai node được xem như không tương đồng về cấu trúc khi và chỉ khi chúng có số lượng con khác nhau hoặc các con không tương ứng về tên.
* Hai node được xem như tương đồng về cấu trúc khi và chỉ khi chúng có cùng số lượng con và các con hoàn toàn tương ứng giống nhau về tên.
* Hai node được xem như là giống nhau hoàn toàn (*thành phần lặp*) khi và chỉ khi:
  + Tất cả node cha của chúng đều tương đồng về cấu trúc.
  + Có cùng số lượng con và tất cả các con đều giống nhau hoàn toàn về tên lẫn nội dung văn bản.



Hình 5‑11 So sánh và đánh dấu thành phần lặp giữa 2 cây cấu trúc HTML

Theo đó, các node không tương đồng về cấu trúc được đánh dấu -1, tương đồng về cấu trúc được đánh dấu 0 và giống nhau hoàn toàn được đánh dấu 1. Hình 5‑11 là một ví dụ đánh dấu *thành phần lặp* giữa 2 cây cấu trúc. Trên đó, có nhánh thành phần lặp được gán giá trị 1 vì có 2 thẻ con tương đồng nhau về tên và nội dung *Text*3 và *Text*4. Độ tương đồng của cây cấu trúc được tính dựa trên những chỉ số này sẽ được trình bày ngay tiếp sau đây.



Hình 5‑12 Tính độ tương đồng cây cấu trúc HTML

Hình 5‑12 mô tả cách tính độ tương đồng cấu trúc của một văn bản HTML. Độ tương đồng của cây cấu trúc HTML được tính dựa trên các nhánh lặp giữa 2 trang web. Theo đó, có 2 yếu tố ảnh hưởng đến độ tương đồng:

* Chiều cao của nhánh lặp: số tầng phân cấp từ node gốc của nhánh lặp đến lá sâu nhất.
* Trọng lượng của nhánh lặp: tổng số node kể cả lá có trong nhánh lặp đó.

Các luật để tính độ tương đồng của 2 nhánh cấu trúc HTML được áp dụng là:

* Nếu node gốc của nhánh được đánh dấu -1 thì độ tương đồng được gán bằng 0.
* Nếu node gốc của nhánh được đánh dấu 0 thì độ tương đồng được gán bằng tổng độ tương đồng các nhánh con.
* Nếu node gốc của nhánh được đánh dấu 1 và cha của node này được đánh dấu 1 thì độ tương đồng được tính bằng trọng lượng của nhánh (tổng số node con của nhánh này).
* Nếu node gốc của nhánh được đánh dấu 1 và cha của node này được đánh dấu 0 thì độ tương đồng được tính bằng tích chiều cao và trọng lượng của nhánh.

### Loại trừ các phần trùng nhau giữa các trang web trong cùng một nhóm.

* Vấn đề

Sau khi phân nhóm các trang web theo độ tương đồng cấu trúc, vấn đề đặt ra là lựa chọn phương thức loại bỏ *thành phần lặp* giữa các trang web trong nhóm sao cho hiệu quả nhất. Nghĩa là tổng độ tương đồng các *thành phần lặp* được loại bỏ là nhiều nhất.

* Giải pháp

Giải pháp của chúng em là chọn ra một trang web làm *hạt giống* (seed) để so sánh độ tương đồng với tất cả các trang web còn lại bằng cách xét một số ngẫu nhiên các trang web làm ứng viên *hạt giống* trên một tập ngẫu nhiên các mẫu thử trong nhóm. Trang web được lựa chọn làm *hạt giống* cần thỏa những tiêu chí sau:

* Trung bình độ tương đồng của ứng viên với các mẫu thử nằm trong nhóm các ứng viên có tổng độ tương đồng lớn nhất. Tiêu chí này đảm bảo tổng số thành phần lặp bị loại bỏ là lớn nhất.
* Phân bố các độ tương đồng giữa ứng viên với các mẫu thử là ổn định nhất dựa vào độ lệch chuẩn (Standard Deviation). Tiêu chí này đảm bảo các thành phần lặp được loại bỏ khỏi các trang web là đồng đều nhất.

### Lưu trữ các trang web vào hệ quản trị cơ sở dữ liệu

* Vấn đề:

Với yêu cầu phân tích và xử lý cấu trúc các trang HTML - được cấu thành bởi các thẻ và phần nội dung văn bản kèm theo thẻ. Với các Website lớn, mỗi trang web có chứa rất nhiều thẻ, trung bình một trang web chứa xấp xỉ hơn 500 thẻ mà số lượng trang web được lưu trữ lên đến hàng chục ngàn. Vì vậy, việc lưu trữ tất cả các thẻ này vào cơ sở dữ liệu thực sự là một gánh nặng.

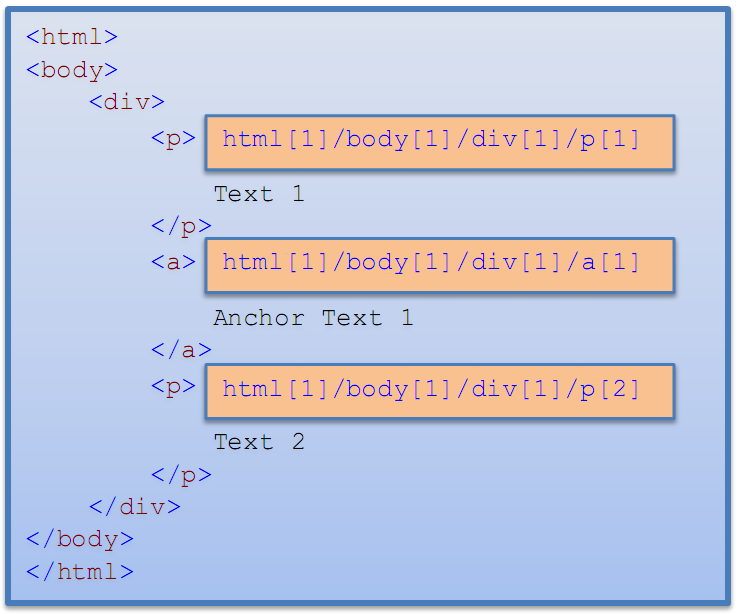
* Giải pháp:

Dựa trên đặc thù yêu cầu của phương pháp, giải pháp cho vấn đề trên phải thỏa mãn 2 yêu cầu thiết yếu sau:

* Có khả năng xây dựng lại cấu trúc thẻ của trang web như ban đầu với số lượng record lưu trữ là ít nhất có thể.
* Phải đảm bảo lưu trữ được tất cả các phần văn bản của trang web.

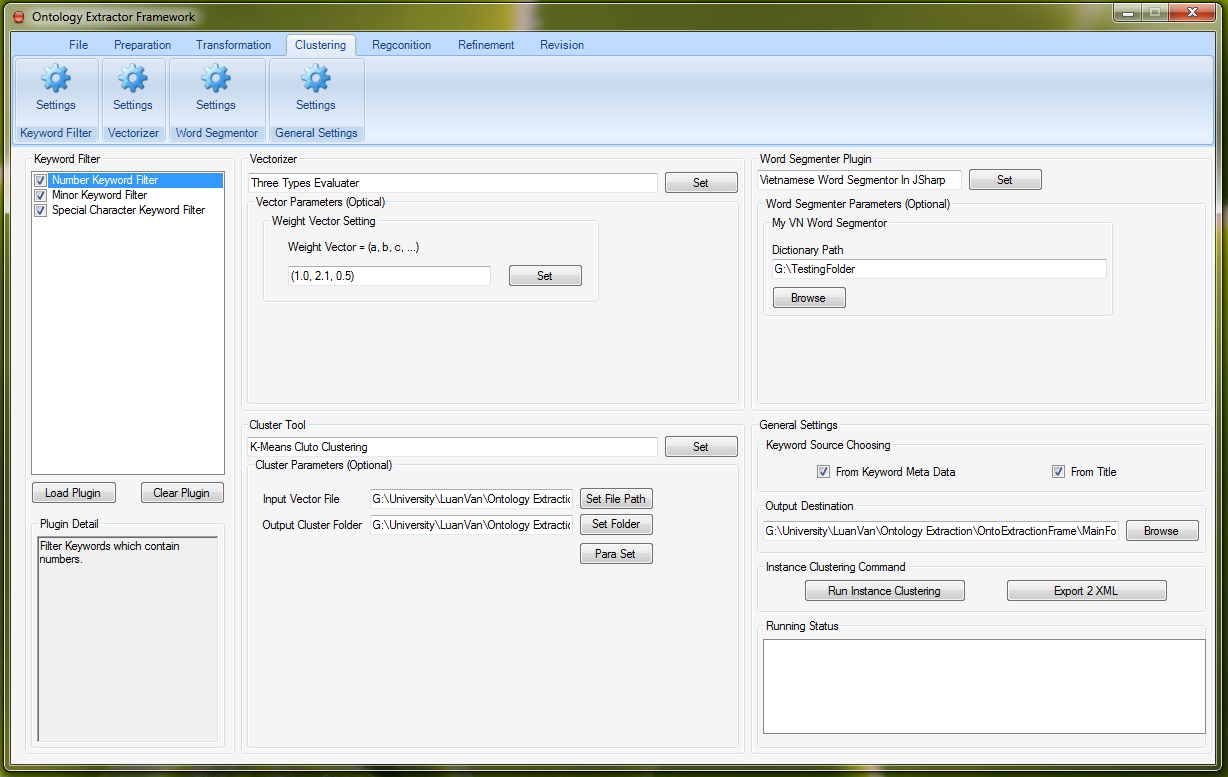
Từ nhận xét các phần văn bản luôn là phần cuối cùng trong một nhánh thẻ của một trang HTML, đặt xpath của một thẻ là đường đi từ thẻ root đến thẻ đó, thẻ root của một văn bản HTML thường là . Giải pháp được chúng em đề ra là chỉ cần lưu trữ xpath của các thẻ có chứa văn bản kèm theo. Như vậy cấu trúc của một trang web có thể hoàn toàn được khôi phục dựa vào xpath của tất cả các thẻ chứa văn bản. Việc lưu trữ tất cả các thẻ chứa văn bản đảm bảo lưu trữ được tất cả các phần văn bản của trang web, tạo tiền đề cho việc thống kê tần số xuất hiện các từ khóa được đầy đủ và trọn vẹn.

Hình 5‑13 cho thấy cách lưu trữ các xpath của một văn bản HTML, trong đó chỉ những thẻ chứa văn bản mới được lưu trữ xpath. Mỗi xpath gồm một dãy các thẻ thể hiện đường đi từ thẻ *root* đến thẻ đang được xét. Mỗi thẻ trong dãy xpath có thêm chỉ số thể hiện thứ tự xuất hiện của mình trong số các thẻ đồng cấp đồng loại. Khi lưu trữ trong cơ sở dữ liệu, các xpath của một văn bản HTML được lưu tuần tự liên tiếp nhau, điều này đảm bảo cho quá trình khôi phục cấu trúc thẻ của văn bản dựa vào tập các xpath này.



Hình 5‑13 Xpath

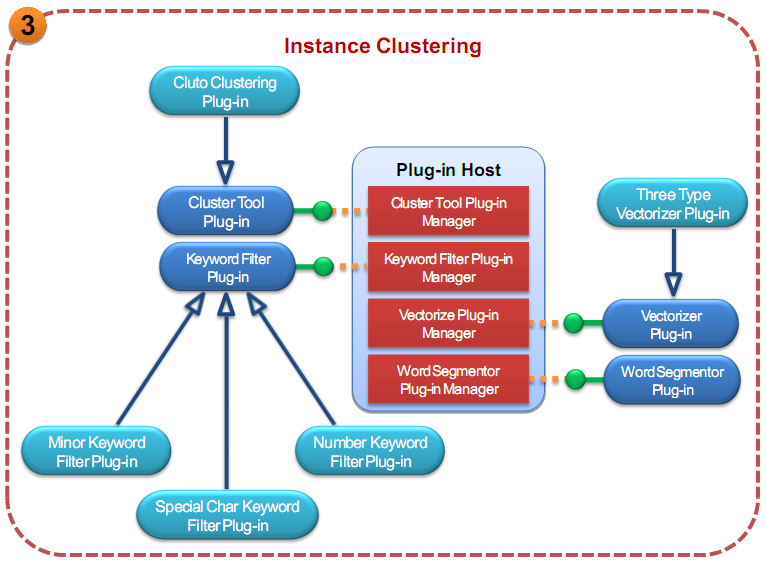
## Phân hệ Gom cụm (Instance Clustering)



Hình ‑ Giao diện phân hệ Gom cụm

### Kiến trúc phân hệ

Các trang web sau khi được lọc bỏ các thành phần không cần thiết ở bước *Phân hệ* Biến đổi được biểu diễn bằng các *vector đặc trưng* phản ánh nội dung bên trong của các trang web. Nhiệm vụ chính của phân hệ Gom cụm chính là tiến hành gom cụm các trang web dựa vào độ tương đồng về nội dung thông qua *vector đặc trưng*. Kiến trúc của phân hệ sử dụng cơ chế plug-in được trình bày ở Hình 5‑15.



Hình 5‑15 Kiến trúc Phân hệ Gom cụm

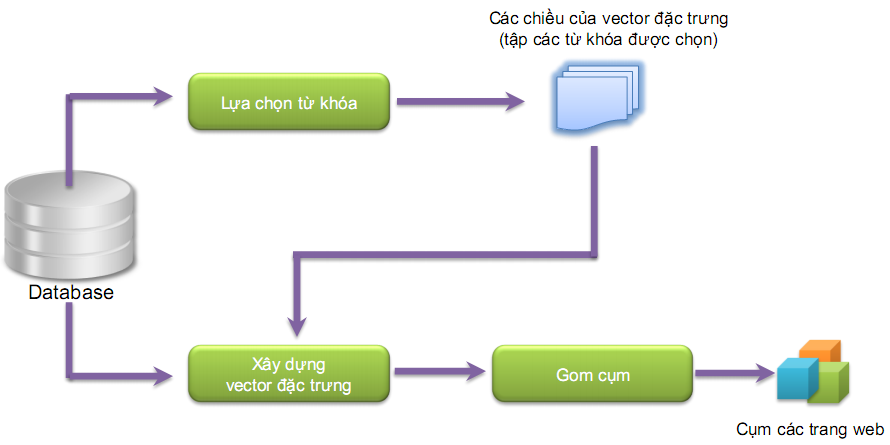
Hình 5‑15 mô tả kiến trúc của phân hệ Chuẩn bị. Phân hệ gồm một host quản lý các plug-in và có 4 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho mỗi loại plugin:

* Cluster Tool Plug-in: hỗ trợ công cụ gom cụm các vector đặc trưng. (1 thể hiện)
* Keyword Filter Plug-in: bộ lọc các từ khóa. (3 thể hiện)
* Vectorizer Plug-in: xây dựng vector đặc trưng cho các trang web. (1 thể hiện)
* Word Segmentor Plug-in: công cụ tách từ tiếng Việt. (2 thể hiện: tái sử dụng của phân hệ Biến đổi)

Bảng 5‑4 trình bày các lớp đối tượng chính cấu thành phân hệ Chuẩn bị.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | IinstanceClusteringHost | Interface mô tả host quản lý các plug-in trong phân hệ Gom Cụm |
| 2 | InstanceClusteringHost | Thể hiện host của phân hệ Gom cụm |
| 3 | Iclustering | Interface mô tả plugin gom cụm các vector đặc trưng |
| 4 | ClutoClusteringPlugin | Plugin gom cụm sử dụng thư viện Cluto |
| 5 | Ivectorizer | Interface mô tả plugin xây dựng vector đặc trưng |
| 6 | ThreeTypeVectorizerPlugin | Plugin xây dựng vector đặc trưng trên cơ sở phân loại văn bản thành 3 loại |
| 7 | IkeywordFilter | Interface mô tả plugin bộ lọc từ khóa |
| 8 | NumberKeywordFilterPlugin | Bộ lọc các các từ khóa có chứa chữ số |
| 9 | SpecialCharKeywordFilterPlugin | Bộ lọc các từ khóa có chứa ký tự đặc biệt |
| 10 | MinorKeywordFilterPlugin | Bộ lọc các từ khóa có tần số xuất hiện thấp |

Bảng 5‑4 Các lớp chính cấu thành phân hệ Gom cụm



Hình 5‑16 Mô hình xử lý tổng quát Phân hệ Gom cụm

Như đã được phân tích ở mục 4.6, các tác giả không nhắc đến cách xây dựng các vector đặc trưng *cùng chiều* và *tương ứng từ khoá*, vì thế chúng em đề nghị giải pháp cho vấn đề này bằng cách chọn ra một tập các từ khoá ứng viên trong số tất cả các từ khoá đã được gán nhãn cho các trang web. Tập các từ khoá ứng viên này được dùng làm các chiều chuẩn chung của *vector đặc trưng* cho tất cả các trang web. Chi tiết phương pháp chọn từ khoá ứng viên được mô tả ở mục 0.

Hình 5‑16 cho thấy, trước khi thực hiện việc xây dựng vector, bộ lọc từ khoá làm nhiệm vụ lựa chọn ra tập các từ khoá ứng viên làm số chiều chuẩn chung cho tất cả các vector đặc trưng. Cuối cùng, các trang web được gom cụm thông qua việc gom cụm các vector đại diện của chúng.

### Lựa chọn các từ khoá

* Vấn đề

Giải pháp được đề ra nhằm đảm bảo yêu cầu các vector đặc trưng phải *cùng chiều* và *tương ứng từ khoá* là chọn ra một tập các từ khoá ứng viên. Vì tổng số lượng từ khoá được lưu trữ là rất lớn, nên chỉ những từ khoá nào có triển vọng nhất mới được chọn.

* Giải pháp

Giải pháp của chúng em là xây dựng bộ lọc các từ khoá, với các tiêu chí động nhằm hỗ trợ người chuyên gia ontology lựa chọn được tập các ứng viên ưng ý nhất. Một số tiêu chí đã được chúng em xây dựng được trình bày ở bảng:

* Chỉ lấy từ khoá là chữ.
* Chỉ lấy từ khoá không có ký tự đặc biệt.
* Chỉ lấy từ khoá có tần số xuất hiện cao.

|  |  |  |
| --- | --- | --- |
| STT | Bộ lọc từ khóa | Mô tả |
| 1 | NumberKeywordFilterPlugin | Bộ lọc các các từ khóa có chứa chữ số |
| 2 | SpecialCharKeywordFilterPlugin | Bộ lọc các từ khóa có chứa ký tự đặc biệt |
| 3 | MinorKeywordFilterPlugin | Bộ lọc các từ khóa có tần số xuất hiện thấp |

Bảng 5‑5 Các bộ lọc từ khóa

Trong đó, bộ lọc các từ khoá có tần số xuất hiện cao đã được chúng em xây dựng và triển khai bằng cách thống kê số trang web chứa từ khoá này. Chỉ những từ khoá có số lần xuất hiện trong các trang web lớn hơn một ngưỡng ε mới được chọn. Ngưỡng này được điều chỉnh bởi người chuyên gia ontology như đã được đề cập ở mục 5.7.

Phương pháp xây dựng vector đặc trưng của các trang web được hiện thực hoá và cải tiến dựa trên phương pháp của nhóm tác giả như đã được đề cập ở mục 4.6. Trong đó, giá trị các chiều của *vector trọng số* cần được điều chỉnh tuỳ vào người chuyên gia ontology như sẽ được đề cập ở mục 5.7.

### Xây dựng vector đặc trưng

* Vấn đề

Phương pháp được sử dụng trong luận văn này dựa trên tần số xuất hiện của các từ khóa (chọn từ thẻ , ) trong phần văn bản nội dung của trang web. Do đó, nhu cầu tách từ phần văn bản chính nhằm thống kê tần số là một trong những vấn đề cần được giải quyết trong bước *gom cụm* này.

* Giải pháp

Một trong những phương pháp tách từ tiếng Việt được xem là hiệu quả và nhanh nhất hiện này là phương pháp so khớp từ dài nhất (Longest Matching). Theo đó, để tách một phần văn bản, bộ từ điển có sẵn được sử dụng và đánh dấu các từ trong phần văn bản theo các từ có trong từ điển với tiêu chí ưu tiên các từ dài nhất trước. Lần lượt đánh dấu tất cả các từ có thể có, phần còn lại được phân biệt bởi các từ đã được đánh dấu được xem như một từ. Từ đây, có hai hướng tiếp cận:

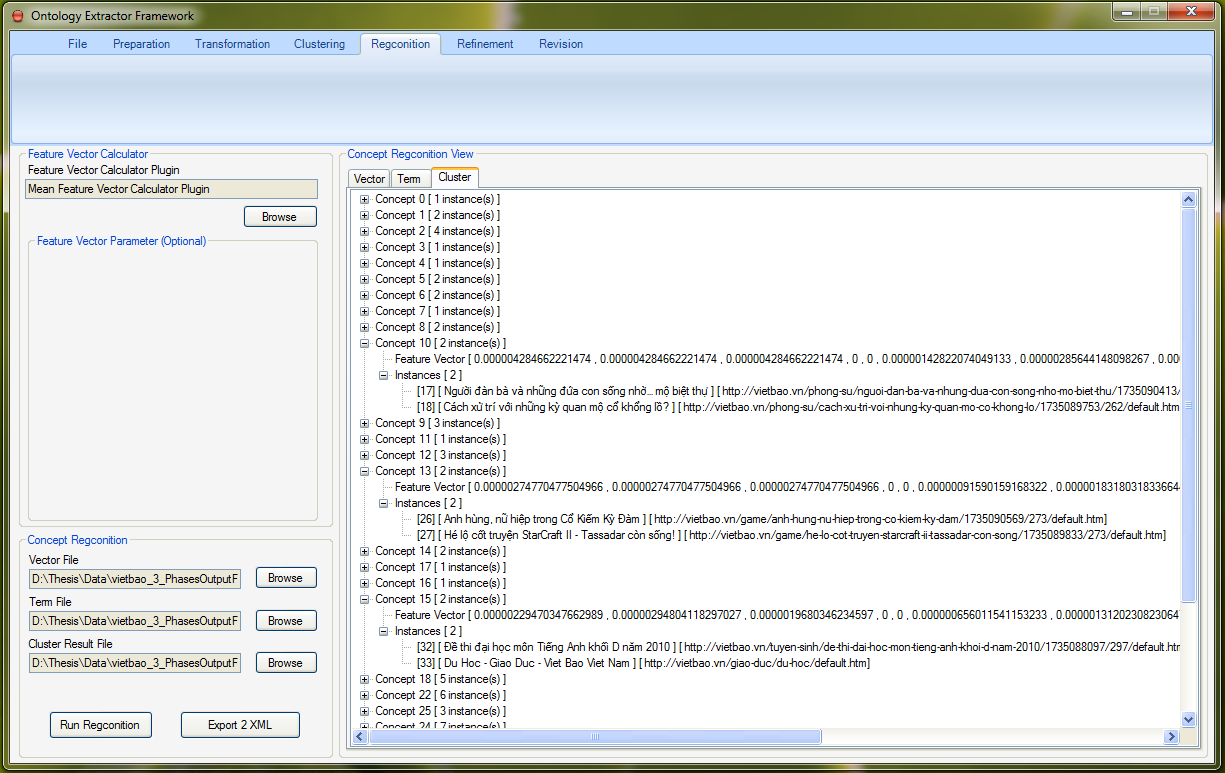
* Sử dụng bộ từ điển chung, được xây dựng sẵn một cách phân biệt với hệ thống.
* Sử dụng tập các từ khóa ứng viên được chọn của Website làm từ điển.

Trong thực tế hiện thực hoá, chúng em nhận thấy việc sử dụng tập các từ khoá ứng viên được chọn của Website làm từ điển mang lại kết quả tốt hơn, vì các từ tách được chính xác với các từ khoá, do đó kết quả tính toán tần số sẽ tốt hơn.

### Gom cụm các trang web dựa trên độ tương đồng giữa các vector đặc trưng

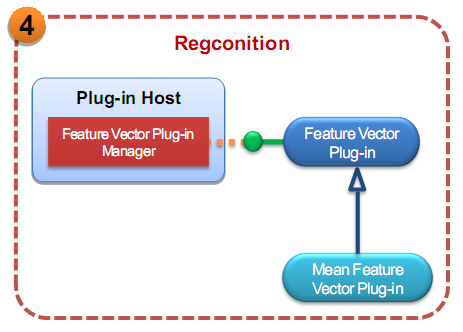
Đặc thù của việc gom cụm các vector đặc trưng là số chiều rất lớn. Công cụ CLUTO là một công cụ gom cụm miễn phí để thực hiện gom cụm các vector (<http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview>). Một trong các ưu điểm của CLUTO mà chúng em quan tâm là CLUTO có thể thực hiện gom cụm trên các phần tử có số chiều rất lớn. Điều này giải quyết được vấn đề gom cụm các vector đặc trưng có số chiều lớn của phương pháp. Vì thế, chúng em lựa chọn CLUTO như là giải pháp cho vấn đề gom cụm.

## Phân hệ Nhận diện (Recognition)



Hình ‑ Giao diện phân hệ Nhận diện

### Kiến trúc phân hệ



Hình 5‑18 Kiến trúc Phân hệ Nhận diện

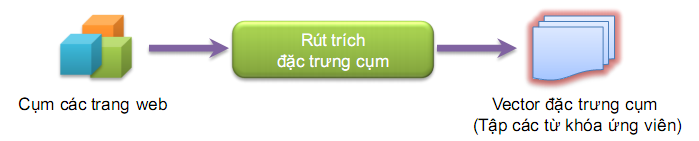
Hình 5‑18 mô tả kiến trúc của phân hệ Nhận diện. Phân hệ gồm một host quản lý các plug-in và có 1 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho loại plugin này:

* Feature Vector Plug-in: rút trích đặc trưng cụm. (1 thể hiện)

Bảng 5‑3 Các lớp chính cấu thành phân hệ Biến đổi trình bày các lớp đối tượng chính cấu thành phân hệ Biến đổi.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | IregconitionHost | Interface mô tả host quản lý các plug-in trong phân hệ Nhận diện |
| 2 | RegconitionHost | Thể hiện host của phân hệ Nhận diện |
| 3 | IfeatureVectorCalculator | Interface mô tả plugin rút trích đặc trưng cụm |
| 4 | MeanFeatureVectorPlugin | Plugin rút trích đặc trưng cụm sử dụng trung bình cộng |

Bảng 5‑6 Các lớp chính cấu thành phân hệ Nhận diện



Hình 5‑19 Mô hình xử lý tổng quát Phân hệ Nhận diện

Hình 5‑22 mô tả 2 bước của quá trình *nhận diện*. Trong đó, kết quả quan trọng nhất của bước *nhận diện* chính là *vector đặc trưng cụm*. Mỗi cụm tương ứng với một *khái niệm*. Từ *vector đặc trưng cụm*, các từ khoá ứng viên của *khái niệm* được rút ra. Trong đó, các chiều của *vector đặc trưng cụm* có giá trị lớn hơn giá trị ε do chuyên gia ontology điều chỉnh (mục 5.7) sẽ được rút làm *từ khoá ứng viên* của *khái niệm*.

### Nhận diện đặc trưng cụm

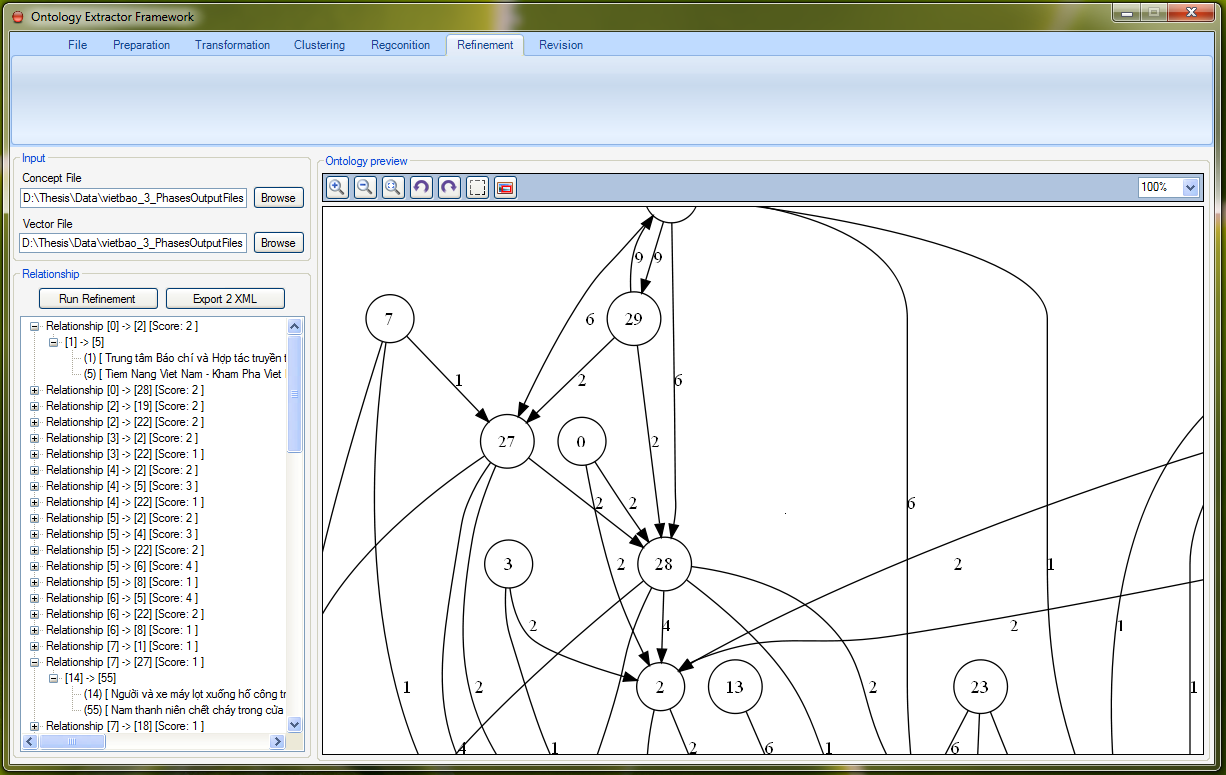
* Vấn đề

Mỗi cụm gồm nhiều trang web tương đồng về nội dung. Mỗi trang web được đại diện bằng một vector đặc trưng, vì vậy *đặc trưng cụm* chính là một vector *cùng chiều* và *tương ứng từ khóa* với các vector thành phần.

* Giải pháp

Mỗi phần tử được biểu diễn bằng một vector n chiều. Giải pháp được chúng em đề nghị và triển khai trong thực tế là rút trích các vector đại diện cho các cụm. Trong đó, giá trị mỗi chiều của vector đại diện là trung bình cộng (mean) các giá trị tại chiều tương ứng của các vector phần tử.

## Phân hệ Liên kết (Refinement)



Hình ‑ Giao diện phân hệ Liên kết

### Kiến trúc phân hệ



Hình 5‑21 User Control phân hệ Liên kết

Kiến trúc phân hệ chỉ gồm một User Control là RefineUC đóng vai trò hỗ trợ giao diện trong việc xây dựng đồ thị các *mối quan hệ*.



Hình 5‑22 Mô hình xử lý tổng quát Phân hệ Liên kết

Ở bước *Phân hệ* Nhận diện, các *khái niệm* đã xác định được các từ khoá ứng viên tương ứng. Hình 5‑22 mô tả 2 bước thực hiện quá trình *Liên kết*. Đầu tiên, các *mối quan hệ* giữa các *khái niệm* được xác định và gán trọng số. Kết quả của bước này thu được một đồ thị với các node là các *khái niệm* được đại diện bằng tập các từ khoá ứng viên. Các cạnh của đồ thị có hướng và được gán trọng số. Sau đó, đồ thị được tinh chỉnh nhằm thu được kết quả chính xác và thuận lợi hơn cho người chuyên gia ontology ở bước Phân hệ Tinh chỉnh tiếp theo.

### Xây dựng đồ thị mối quan hệ giữa các khái niệm

* Vấn đề

Mỗi *khái niệm* được đại diện bằng một cụm mà các phần tử trong đó là các trang web. Vấn đề đặt ra là cần xác định các *mối quan hệ* giữa các *khái niệm*. Hai trang web được xem là có quan hệ khi có đường dẫn từ trang này qua trang kia. Như vậy, *mối quan hệ* giữa hai *khái niệm* được xác định bằng đường dẫn giữa các trang web thuộc về 2 *khái niệm* đó. Vấn đề đặt ra chính là các *khái niệm* và *mối quan hệ* này chỉ đang ở dạng thô sơ, cần được biểu diễn sao cho có thể hỗ trợ tối đa nhất cho việc quan sát, xử lý và tinh chỉnh sau này của chuyên gia ontology.

* Giải pháp

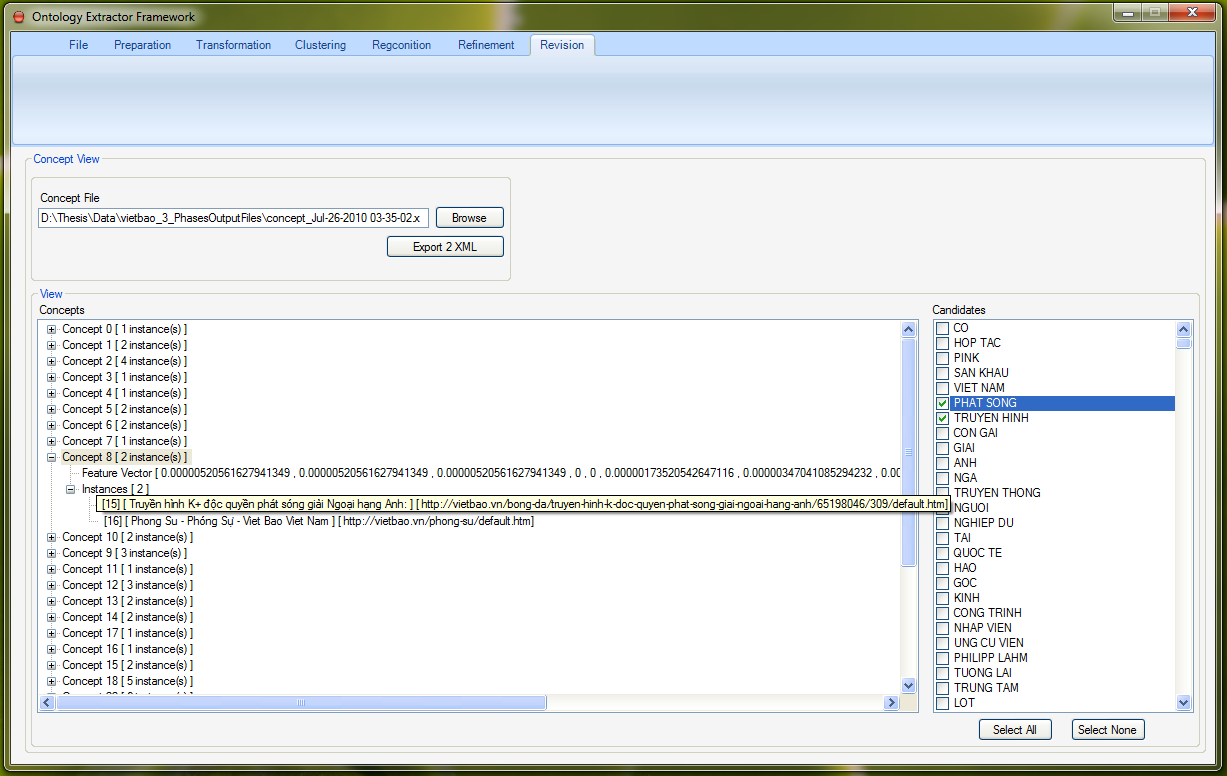
Giải pháp được đặt ra là biểu diễn các *khái niệm* và *mối quan hệ* dưới dạng đồ thị. Đồ thị mối quan hệ do chúng em đề nghị xây dựng mang những đặc điểm:

* Mỗi đỉnh trên đồ thị đại diện cho một *khái niệm* (cụm các trang web).
* Đồ thị có hướng và các cạnh được đánh trọng số.
* Hai đỉnh được nối với nhau khi tồn tại một đường dẫn từ trang web thuộc cụm này dẫn đến một trang web thuộc cụm kia. Trọng số của cạnh chính là số lượng đường dẫn nối hai cụm.

Với giải pháp này, cách hiện thực hóa của chúng em mang tính cải tiến so với phương pháp của các tác giả như đã được nhắc đến ở mục 4.8. Trong đó :

* Do nhận nhận xét các *mối quan hệ* có tính *duy nhất* và *đối xứng* của các tác giả, các *mối quan hệ* (đại diện bằng các cạnh trên đồ thị) sẽ không được gán trọng số và vô hướng. Ngược lại, chúng em lưu trữ trọng số và hướng của các mối quan hệ sẽ giúp cung cấp thêm thông tin cho các chuyên gia ontology để thuận lợi cho việc tinh chỉnh sau này.
* Nhận xét các *mối quan hệ* có tính *bắc cầu* khiến các cạnh nối giữa 2 đỉnh sẽ bị xóa đi nếu tồn tại một đường đi vô hướng khác dài hơn từ đỉnh này đến đỉnh kia. Chúng em vẫn lưu giữ lại những cạnh nối trực tiếp này, vì như vậy, quan hệ giữa 2 đỉnh có cạnh nối trực tiếp vẫn có ý nghĩa hơn là mối quan hệ được suy ra từ tính chất bắc cầu.

## Phân hệ Tinh chỉnh (Revision)



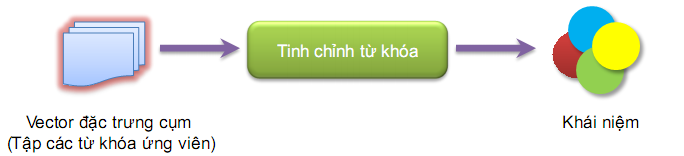
Hình ‑ Giao diện phân hệ Tinh chỉnh

### Kiến trúc phân hệ



Hình 5‑24 User Control phân hệ Tinh chỉnh

Kiến trúc phân hệ chỉ gồm một User Control là RevisionUC đóng vai trò hỗ trợ giao diện trong việc lựa chọn các từ khóa đại diện cho *khái niệm*.



Hình 5‑25 Mô hình xử lý tổng quát Phân hệ Tinh chỉnh

Người chuyên gia ontology đóng vai trò quan trọng trong việc vận hành toàn bộ hệ thống, điều chỉnh các thông số và tinh chỉnh các *khái niệm*, *mối quan hệ* cuối cùng nhằm đảm bảo kết quả ontology của hệ thống sao cho phù hợp nhất với mục đích sử dụng trong tương lai của ontology. Đối với hệ thống được triển khai, chuyên gia ontology là người lựa chọn các plug-in của hệ thống và điều chỉnh các tham số cần thiết cho các plug-in cần tham số. Cụ thể như sau:

* Điều chỉnh tham số ngưỡng tinh chỉnh số lượng trang web trong nhóm (mục 5.3.3)
* Điều chỉnh ngưỡng để lựa chọn từ khoá ứng viên triển vọng cho *vector đặc trưng* (mục 5.4.2).
* Điều chỉnh các tham số giá trị của *vector trọng số* trong việc xây dựng *vector đặc trưng* của trang web (mục 5.4.3).

# Kết luận

* Nội dung của Chương 6 trình bày các kết quả đạt được và hướng phát triển của đề tài.

## Các kết quả đạt được

Chúng em đã tiến hành tìm hiểu và nghiên cứu về ontology và các vấn đề liên quan đến ontology. Đồng thời, chúng em đã tìm hiểu các phương pháp khác nhau đê xây dựng ontology, và chú trọng vào các phương pháp rút trích ontology với nguồn dữ liệu lấy từ web.

Qua quá trình nghiên cứu và tìm hiểu đó, chúng em tiến hành phân tích cụ thể quy trình rút trích ontology từ web [[2](#Tim09)] do nhóm tác giả Du C. Timon, Li Feng, và King Irwin đề xuất. Những phân tích cụ thể về quy trình này cho thấy mức độ tổng quát cao của quy trình giúp cho nó có thể được áp dụng vào nhiều ngữ cảnh khác nhau và có độ tùy biến cao tùy thuộc vào người sử dụng.

Do đó, chúng em đã tiến hành xây dựng Ontology Extractor Framework để hiện thực hóa quy trình này. Trong quá trình xây dựng framework này, đề tài đã nêu ra và phân tích các vấn đề phát sinh khác nhau cũng như đưa ra hướng giải quyết cho các vấn đề này. Đề tài cũng đã xây dựng kiến trúc của hệ thống này phục vụ việc tùy biến theo nhu cầu cụ thể của người dùng.

Ngoài framework được xây dựng, chúng em còn tiến hành hiện thực hóa các plug-in cụ thể để sử dụng trên framework này. Trong quá trình hiện thực hóa này, chúng em đã phân tích các vấn đề nảy sinh và đề ra hướng giải quyết cho những vấn đề này mà nhóm tác giả của quy trình này không đề cập đến.

## Hướng phát triển của đề tài

Những dự định trong tương lai để phát triển framework này là:

* Hoàn thiện các chức năng hiện có của framework.
* Tăng cường tính tiện dụng cho framework.
* Xây dựng các bộ plug-in khác nhau để hoạt động trên framework này.
* Danh mục tài liệu tham khảo

x

|  |  |
| --- | --- |
| [1] | Tom Gruber. (2009) Ontology (Computer Science) - definition in Encyclopedia of Database Systems. [Online]. <http://tomgruber.org/writing/ontology-definition-2007.htm> |
| [2] | Du C. Timon, Li Feng, and King Irwin, "Managing knowledge on the Web - Extracting ontology from HTML Web," *Decision Support Systems*, vol. 47, no. 4, pp. 319-331, November 2009. |
| [3] | Neches Robert et al., "Enabling technology for knowledge sharing," , 1991, pp. 36-56. |
| [4] | Ehrig Marc, Haase Peter, Hefke Mark, and Stojanovic Nenad, "Similarity for Ontologies - a Comprehensive Framework," in *Proceedings of* 13*th European Conference on Information Systems*, 2005. |
| [5] | Sure York, Bloehdorn Stephan, Haase Peter, Hartmann Jens, and Oberle Daniel, "The SWRC Ontology - Semantic Web for Research Communities," in *Progress in Artificial Intelligence,* 12*th Portuguese Conference on Artificial Intelligence*, 2005, pp. 218-231. |
| [6] | N. Guarino and P. Giaretta, "Ontologies and Knowledge Bases: Towards a Terminological Clarification," in *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, Amsterdam, 1995, pp. 25-32. |
| [7] | Bernaras Amaia, Laresgoiti Iñaki, and Corera Manuel Jose, "Building and Reusing Ontologies for Electrical Network Applications," in 12*th European Conference on Artificial Intelligence*, Budapest, 1996, pp. 298-302. |
| [8] | Thomas R. Gruber, "A translation approach to portable ontology specifications," pp. 199-220, 1993. |
| [9] | Studer Rudi, Benjamins Richard V., and Fensel Dieter, "Knowledge Engineering: Principles and Methods," *IEEE Transactions on Data and Knowledge Engineering*, pp. 161-197, 1998. |
| [10] | Fensel Dieter, *Ontologies: a silver bullet for knowledge management and electronic commerce*.: Springer-Verlag, 2001. |
| [11] | MeSH: Medical Subject Headings. [Online]. <http://www.nlm.nih.gov/mesh/meshhome.html> |
| [12] | Rector L. A. et al., "A Terminology Server For Medical Language and Medical Information Systems," in *Methods of Information in Medicine*, 1995, pp. 147–157. [Online]. <http://www.opengalen.org/> |
| [13] | Gene Ontology. [Online]. <http://www.geneontology.org/> |
| [14] | OBO: Open Biological Ontologies. [Online]. <http://www.obofoundry.org/> |
| [15] | Weibel S., Gridby J., and Miller E., "OCLC/NCSA Metadata Workshop Report," Dublin, 1995. |
| [16] | Christiane Fellbaum, Ed., *WordNet An Electronic Lexical Database*.: MIT Press, 1998. |
| [17] | Lenat B. Douglas, "CYC: A Large-Scale Investment in Knowledge Infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33-38, Nov. 1995. |
| [18] | Fensel Dieter, Erdmann Michael, and Studer Rudi, "Ontology Groups: Semantically Enriched Subnets of the WWW," in *Proceedings of the International Workshop Intelligent Information Integration during the* 21*st German Annual Conference on Artificial Intelligence*, Freiburg, 1997. |
| [19] | Studer Rudi et al., "Ontologies and the Configuration of Problem-Solving Methods," in *Proceedings of the* 10*th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, p. 1996. |
| [20] | Lenat B. Douglas and Guha V. R., *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, 1st ed. Boston, USA: Addison-Wesley Longman Publishing , 1989. |
| [21] | Gomez-Perez Asuncion, Corcho Oscar, and Fernandez-Lopez Mariano, *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, 1st ed.: Springer, 2004. |
| [22] | Luke Sean and Heflin Jeff. (2000) SHOE 1.01. Proposed Specification. [Online]. <http://www.cs.umd.edu/projects/plus/SHOE/spec.html> |
| [23] | Lassila Ora and Swick R. Ralph. (1999) Resource Description Framework (RDF) Model and Syntax Specification. [Online]. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/> |
| [24] | Brickley Dan and Guha V. R. (2003) RDF Vocabulary Description Language 1.0: RDF Schema. [Online]. <http://www.w3.org/TR/rdf-schema/> |
| [25] | Horrocks Ian et al., "OIL in a Nutshell," in *Knowledge Acquisition, Modeling and Management,* 12*th International Conference*, Juan-les-Pins, 2000, pp. 1-16. |
| [26] | Horrocks Ian, van Harmelen Frank, and Patel-Schneide F. Peter. (2001) Reference description of the DAML+OIL (March 2001) ontology markup language. [Online]. <http://www.daml.org/2001/03/reference.html> |
| [27] | Dean Mike and Schreiber Guus. (2003) OWL Web Ontology Language Reference. [Online]. <http://www.w3.org/TR/owl-ref/> |
| [28] | Decker Stefan, Erdmann Michael, Fensel Dieter, and Studer Rud, "Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information," in *IFIP TC*2*/WG*2*.*6 *Eighth Working Conference on Database Semantics*, Rotorua, 1999, pp. 351-369. |
| [29] | Fensel Dieter, Decker Stefan, Erdmann Michael, and Studer Rudi, "Ontobroker: The Very High Idea," in *Proceedings of the Eleventh International Florida Artificial Intelligence Research Society Conference*, Sanibel Island, 1998, pp. 131-135. |
| [30] | Fensel Dieter et al., "On2broker: Semantic-Based Access to Information Sources at the WWW," in *Proceedings of the IJCAI-*99 *Workshop on Intelligent Information*, Stockholm, 1999. |
| [31] | Benjamins Richard V., Wielinga J. Bob, Wielemaker Jan, and Fensel Dieter, "An Intelligent Agent for Brokering Problem-Solving Knowledge," in *Engineering Applications of Bio-Inspired Artificial Neural Networks, International Work-Conference on Artificial and Natural Neural Networks*, Alicante, 1999, pp. 693-705. |
| [32] | Berners-Lee Tim, Hendler James, and Lassila Ora, "The Semantic Web," *Scientific American*, 2001. |
| [33] | Domingue John and Dzbor Martin, "Magpie: Supporting Browsing and Navigation on the Semantic Web," in *Proceedings of the* 2004 *International Conference on Intelligent User Interfaces*, Madeira, 2004, pp. 191-197. |
| [34] | Cimiano Philipp, Handschuh Siegfried, and Staab Steffen, "Towards the self-annotating web," in *Proceedings of the* 13*th international conference on World Wide Web*, New York, 2004, pp. 462-471. |
| [35] | Dill Stephen et al., "SemTag and seeker: bootstrapping the semantic web via automated semantic annotation," in *Proceedings of WWW’* 03, 2003, pp. 178-186. |
| [36] | Guha R. and Mccool R., "Tap: Towards a Web of Data". |
| [37] | Ngan Duy Le and Goh Angela, "Matching Semantic Web Services Using Different Ontologies," in *Web Engineering,* 5*th International Conference, ICWE* 2005, Sydney, 2005, pp. 302-307. |
| [38] | Ehrig Marc and Sure York, "Ontology Mapping - An Integrated Approach," in *The Semantic Web: Research and Applications, First European Semantic Web Symposium*, Heraklion, 2004, pp. 76-91. |
| [39] | Shin Dong-Hoon, Lee Kyong-Ho, and Suda Tatsuya, "Automated generation of composite web services based on functional semantics," *Journal of Web Semantics*, vol. 7, no. 4, pp. 332-343, 2009. |
| [40] | Mehrnoush Shamsfard and Ahmad Abdollahzadeh Barforoush, "The state of the art in ontology learning: a framework for comparison," *The Knowledge Engineering Review*, vol. 18, no. 4, pp. 293 - 316, 2003. |
| [41] | Vipul Kashyap, "Design and Creation of Ontologies for Environmental Information Retrieval," in *Proceedings of the* 12*th Workshop on Knowledge Acquisition, Modeling and Management*, 1999. |
| [42] | Andrew Brent Williams and Costas Tsatsoulis, "An Instance-based Approach for Identifying Candidate Ontology Relations within a Multi-Agent System," in *Proceedings of the First Workshop on Ontology Learning OL'*2000, Berlin, 2000. |
| [43] | "Learning Classification taxonomies from a classification knowledge based system," in *Proceedings of the First Workshop on Ontology Learning OL'*2000, Berlin, 2000. |
| [44] | Faure David and Poibeau Thierry, "First experiences of using semantic knowledge learned by ASIUM for information extraction task using INTEX," in *ECAI'*2000 *Workshop on Ontology Learning, Proceedings of the First Workshop on Ontology Learning OL'*2000, Berlin, 2000. |
| [45] | Shamsfard Mehrnoush and Barforoush Abdollahzadeh Ahmad, "Learning ontologies from natural language texts," *Int. J. Hum.-Comput. Stud.*, vol. 60, pp. 17-63, 2004. |
| [46] | Agirre Eneko, Ansa Olatz, Hovy H. Eduard, and Martínez David, "Enriching very large ontologies using the WWW," in *ECAI'*2000 *Workshop on Ontology Learning, Proceedings of the First Workshop on Ontology Learning OL'*2000, Berlin, 2000. |
| [47] | Faatz Andreas and Steinmetz Ralf, "Ontology Enrichment with Texts from the WWW," in *Semantic Web Mining*, Helsinski, 2002. |
| [48] | Heyer Gerhard, Läuter Martin, Quasthoff Uwe, Wittig Thomas, and Wolff Christian, "Learning Relations Using Collocations," in |
| [49] | Jiang Xing and Tan Ah-Hwee, "Mining Ontological Knowledge from Domain-Specific Text Documents," in *ICDM Proceedings of the Fifth IEEE International Conference on Data Mining*, Washington, DC, 2005, pp. 665 - 668. |
| [50] | Maddi Reddy Govind, Velvadapu Chakravarthi, Srivastava Sadanand, and Lamadrid James Gil de, "Ontology Extraction from text documents by Singular Value Decomposition," in *Proceedings of the ADMI*, 2001. |
| [51] | Buttler David, Liu Ling, and Pu Calton, "A Fully Automated Object Extraction System for the World Wide Web," in *Proceedings of the International Conference on Distributed Computing Systems*, 2001, pp. 361-370. |
| [52] | Crescenzi Valter, Mecca Giansalvatore, and Merialdo Paolo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," in *Proceedings of the Twenty-seventh VLDB Conference*, 2001, pp. 109-118. |
| [53] | Davulcu Hasan, Vadrevu Srinivas, and Nagarajan Saravanakumar, "OntoMiner: bootstrapping ontologies from overlapping domain specific web sites," in *Proceedings of the* 13*th international conference on World Wide Web*, New York, 2004, pp. 500-501. |
| [54] | Han Hyoil and Elmasri Ramez, "Learning Rules for Conceptual Structure on the Web," *Journal of Intelligent Information Systems*, vol. 22, pp. 237-256, 2004. |
| [55] | Kietz Jörg-Uwe, Volz Raphael, and Maedche D. Alexander, "Extracting a domain-specific ontology from a corporate intranet," in *Proceedings of the* 2*nd workshop on Learning language in logic and the* 4*th conference on Computational natural language learning*, Lisbon, 2000, pp. 167-175. |
| [56] | Hamp Birgit and Feldweg Helmut, "GermaNet - a Lexical-Semantic Net for German," in *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid, 1997, pp. 9-15. [Online]. <http://www.sfs.uni-tuebingen.de/lsd/> |
| [57] | Bergman K. Michael. (2001, August) The Deep Web: Surfacing Hidden Value. [Online]. <http://quod.lib.umich.edu/cgi/t/text/text-idx?c=jep;view=text;rgn=main;idno=3336451.0007.104> |

x