LỜI CẢM ƠN

Đầu tiên, chúng em xin gởi lời cảm ơn đến Thầy, Cô khoa Công nghệ Thông tin trường Đại học Khoa học Tự nhiên đã tận tình dạy dỗ, dìu dắt chúng em suốt bốn năm đại học.

Chúng em cảm ơn Cô Phạm Thị Bạch Huệ, người tận tình hướng dẫn, giúp đỡ, động viên chúng em hoàn thành luận văn này.

Cuối cùng, chúng con cảm ơn Ba, Mẹ và những người thân đã khích lệ, hỗ trợ, động viên chúng con trong thời gian học tập, nghiên cứu để có được thành quả như ngày nay.

KHOA CNTT

Tháng 7 năm 2005

Sinh viên

Phạm Thị Mỹ Phượng – Từ Thị Ngọc Thanh

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

KHOA CNTT

Ngày…… tháng……năm 2005

Ký tên

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

Ngày…… tháng……năm 2005

Ký tên

KHOA CNTT

MỤC LỤC

KHOA CNTT

KHOA CNTT

DANH MỤC CÁC BẢNG

KHOA CNTT

DANH MỤC CÁC HÌNH

KHOA CNTT

DANH MỤC CÁC TỪ VIẾT TẮT

eDoc	Electronic document
eLib	Electronic library
eLearning	Electronic learning
www	World Wide Web
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
RDF	Resources Descriprion Framework
OIL	Ontology Inference Language
OWL	Ontology Web Language
XML	e**X**tensible **M**arkup **L**anguage

KHOA CNTT

DANH MỤC CÁC THUẬT NGỮ

Class	Lớp
Property	Thuộc tính
Metadata	Siêu dữ liệu
Subject	Chủ đề, chủ ngữ
Title	Tiêu đề
Namespace	Không gian tên
Predicate	Vị ngữ
Triple	Bộ ba (subject, predicate, object)

KHOA CNTT

MỞ ĐẦU

Hiện nay, hầu hết các hệ thống tìm kiếm trên Internet đều đi theo hướng truyền thống đó là tìm kiếm theo từ khoá (key word). Theo cách tìm kiếm này, khi ta gõ vào từ cần tìm, các hệ thống tìm kiếm sẽ hiển thị các tài liệu mà trong nó có chứa từ khoá cần tìm. Do đó, kết quả trả ra là một danh sách rất nhiều các tài liệu, mà có thể các tài liệu này không liên quan gì đến nội dung ta cần tìm. Và đôi khi các hệ thống này không đưa ra hết các tài liệu cần thiết, tức là thừa tài liệu không cần thiết nhưng lại thiếu hẳn những tài liệu quan trọng khác.

Vấn đề đặt ra là ta phải xây dựng một hệ thống tìm kiếm như thế nào để khắc phục hiện trạng nêu trên ?

Để giải quyết vấn đề này, ta cần xây dựng hệ thống tìm kiếm sao cho đáp ứng đầy đủ thông tin mà người dùng mong muốn, nghĩa là phải xây dựng hệ thống tìm kiếm theo ngữ nghĩa dựa trên thông tin người dùng đưa vào.

Từ nhận thức trên chúng em quyết định chọn đề tài: **Tìm kiếm ngữ nghĩa ứng dụng trên lĩnh vực eDoc** (những tài liệu điện tử tiếng Anh) với mục đích tìm hiểu và xây dựng một công cụ tìm kiếm theo ngữ nghĩa để có thể tìm kiếm thông tin chính xác và đầy đủ, để có thể hạn chế được phần nào vấn đề tìm kiếm theo từ khoá của các search engine hiện tại.

Các đối tượng nghiên cứu liên quan đến đề tài: eDoc, Semantic Web, RDF, OWL, Metadata,….

Trong phạm vi đề tài, vì thời gian thực hiện ngắn, nên chúng em chỉ thử nghiệm chương trình tìm kiếm trong một số lĩnh vực: Khoa học máy tính (Computer Science), Nghệ thuật (Art). Hai lĩnh vực này có vẻ như không liên hệ với nhau nhưng thực tế vẫn có những trường hợp cần phải phân biệt, ví dụ như tài liệu về "nghệ thuật lập trình" ("Art of programming") thì phải phân tài liệu về lĩnh vực khoa học máy tính chứ không phải nghệ thuật …. Tóm lại, ứng dụng mà chúng em xây dựng chỉ tìm kiếm thông tin trong các lĩnh vực nêu trên. Tuy nhiên, ứng dụng có thể dễ dàng mở rộng ra nhiều lĩnh vực còn lại.

KHOA CNTT

Chương 1 : TỔNG QUAN

1.1. Đặt vấn đề

Nhu cầu tìm kiếm, nắm bắt thông tin là một nhu cầu không thể thiếu trong đời sống của mỗi người. Khi việc sử dụng World Wide Web đã trở nên phổ biến rộng khắp, thì công việc của các search engine cũng trở thành một phần sống còn và có lợi ích cho Web. Các công cụ tìm kiếm trở thành những công cụ công cộng cho mọi người dùng của Internet; Google và Yahoo, cũng trở thành những cái tên quen thuộc.

Các công cụ tìm kiếm hiện nay dựa trên một trong hai dạng của công nghệ tìm kiếm Web: tìm kiếm do con người tự chỉ đường dẫn và tìm kiếm tự động.

Công cụ tìm kiếm do con người chỉ đường dẫn sử dụng một cơ sở dữ liệu của các từ khoá, các khái niệm, và các tham chiếu. Những công cụ tìm kiếm theo từ khoá trả về một dãy các trang, nhưng phương pháp đơn giản này thường dẫn đến hàng loạt các kết quả không liên quan và không xác thực. Hoạt động của một công cụ tìm kiếm dựa trên nội dung là: sẽ đếm số lượng các từ truy vấn (các từ khoá) so với các từ hiện diện trong mỗi trang được chứa trong chỉ mục của nó. Sau đó, công cụ tìm kiếm này sẽ sắp xếp các trang. Tiếp cận phức tạp hơn bằng cách đưa các vị trí của từ khoá vào một mức độ quan trọng cụ thể. Ví dụ, các từ khoá xuất hiện trong thẻ title của trang web thì quan trọng hơn trong phần body. Các kiểu khác của công cụ tìm kiếm do người dùng chỉ đường dẫn, như Yahoo, sử dụng các lược đồ chủ đề để giúp chỉ hướng tìm kiếm và trả về các kết quả có liên quan hơn. Những lược đồ chủ đề này do con người tạo ra. Bởi lí do này, chúng ta phải tốn chi phí tạo ra và duy trì trong các từ mang "ý nghĩa thời gian" (thay đổi theo thời gian), và rồi thì không được cập nhật thường xuyên như các hệ thống tự động.

Cách tiếp cận tìm theo từ khoá vẫn còn một số hạn chế, điều này đã làm giảm đi tính đúng đắn của các search engine. Ví dụ như các từ đồng âm khác nghĩa (chẳng hạn: bank (ngân hàng), bank (bờ sông), …) hoặc các từ có các biến thể khác nhau do có các tiền tố và hậu tố như student và students; small, smaller, smallest; …. Ngoài ra, các search engine không trả về các tài liệu có các từ đồng nghĩa với các từ trong câu

KHOA CNTT

truy vấn mà người dùng nhập vào. Key word không đủ để biểu diễn chính xác nhu cầu của người dùng cũng như nội dung các trang web, hạn chế này làm cho các search engine trả về những tài liệu không liên quan đến vấn đề mà người dùng quan tâm. Bởi vì t**ập hợp các từ khóa** là dạng biểu diễn sơ lược nhất của nội dung, và do đó, cách biểu diễn này là một dạng góc nhìn luận lý (logical view) của nội dung **mang mức độ thông tin thấp nhất,** đó chính là lý do cơ bản khiến cho các Search Engine hiện nay có **tỷ lệ số trang web hữu ích trên tổng số trang web trả về thấp.**

Google với 400 triệu tài liệu thu về mỗi ngày và trên 8 tỉ trang web được lập chỉ mục, và là công cụ tìm kiếm thông dụng nhất được sử dụng ngày nay, nhưng thậm chí với Google vẫn còn có nhiều vấn đề. Ví dụ, bằng cách nào bạn tìm kiếm chỉ với một lượng ít dữ liệu mà bạn cần trong một biển kết quả không liên quan được đưa ra?

Khi công nghệ trí tuệ nhân tạo (Artificial Intelligence_AI) phát triển mạnh, thì vấn đề đặt ra là làm thế nào để đưa ra những phương pháp tìm kiếm tốt hơn mà có thể thực sự tin cậy vào những kết quả tìm kiếm đó. Đó là xu hướng của những công cụ tìm kiếm dựa vào ngữ nghĩa và các agent tìm kiếm theo ngữ nghĩa. Một công cụ tìm kiếm ngữ nghĩa tìm kiếm các tài liệu có nghĩa tương tự nhau chứ không chỉ những từ ngữ tương tự nhau. Để Web trở thành một mạng ngữ nghĩa, phải cung cấp nhiều siêu dữ liệu về nội dung của nó, thông qua việc sử dụng các thẻ RDF (Resource Description Framework) và OWL (Ontology Web Language), các thẻ này sẽ giúp thực hiện đưa Web vào trong mạng ngữ nghĩa. Trong mạng ngữ nghĩa, ý nghĩa của nội dung được thể hiện tốt hơn, và những liên kết logic được thực hiện giữa những thông tin liên quan nhau.

Công cụ tìm kiếm ngữ nghĩa, chúng ta đề cập ở đây, có hai ưu điểm lớn so với các công cụ tìm kiếm truyền thống:

1. Nó chấp nhận các truy vấn được phát biểu ở ngôn ngữ tự nhiên.
2. Kết quả là tìm kiếm một mẫu thông tin; không phải là một danh sách các tài liệu có thể (hoặc không) chứa thông tin yêu cầu.

Thật vậy công cụ tìm kiếm ngữ nghĩa bắt đầu với lượng thông tin quá tải. Nó tiếp nhận một số các tác vụ không được ai ưa thích trong việc tìm kiếm thông tin hiện

KHOA CNTT

nay: mở ra mỗi tài liệu của danh sách kết quả và quét nó một cách thủ công để lấy thông tin. Theo cách đó, các công cụ tìm kiếm ngữ nghĩa có khả năng cách mạng hoá, hướng đến việc tìm kiếm thông tin điện tử một cách tự động: nó thay đổi mô hình tìm kiếm từ *việc thu hồi tài liệu đến việc trả lời câu hỏi*.

1.2. Bài toán giải quyết

Theo thống kê trong năm 2001: "Các nhân viên tốn trung bình 8 giờ một tuần, hay 16% giờ công hàng tuần của họ, để tìm kiếm và sử dụng nội dung thông tin bên ngoài. Chi phí lương chỉ riêng cho công ty của Mỹ là 107 tỉ đôla một năm. Việc tìm kiếm ngữ nghĩa là một cơ hội đầy ý nghĩa cho các công ty giúp cho nhân viên của họ có khả năng hơn và hiệu quả hơn trong việc đặt thông tin bên ngoài vào công việc của họ." Không cần nói nhiều thêm nữa. Sự quá tải thông tin là một vấn đề lớn trong xã hội thông tin.

Những khám phá tương tự cũng được tìm thấy trong nhiều nghiên cứu, làm nổi bật vấn đề: phải đưa ra giải pháp trong việc cải tiến xử lí tìm kiếm thông tin. Ngoại trừ những ích lợi to lớn mà các công cụ tìm kiếm mang lại cho chúng ta những năm gần đây bằng việc làm cho có thể truy cập đến hàng triệu các tài liệu, bất chấp vị trí vật lí và ngôn ngữ, thì chúng vẫn có một số hạn chế cơ bản. Ví dụ, chúng không "hiểu" các từ con người gõ vào và do đó đạt tới một số lượng khổng lồ của các kết quả sai. Hơn nữa, chúng hoạt động hiệu quả khi hỏi về những sự kiện, chẳng hạn như "Kerry" và "vua của Tây Ban Nha". Tuy nhiên, chúng thực hiện nhiều kết quả không tốt nếu câu truy vấn nói về *sự liên hệ* giữa các khái niệm chẳng hạn như "Những quốc gia nào đã tham gia trong chiến tranh Iraq?" và "tổng thống nước Pháp theo chính Đảng nào?"

Có ba vấn đề cần được cải tiến để cải thiện các kết quả của công cụ tìm kiếm là:

(i) Công cụ tìm kiếm cần cho phép những truy vấn phức tạp hơn (ví dụ trong ngôn ngữ tự nhiên),

(ii) Công cụ tìm kiếm cần "hiểu" những gì con người hỏi, và

(iii) Công cụ tìm kiếm phải cung cấp câu trả lời cho truy vấn (có thể sao lưu lại những liên kết đến các tài liệu mà cho ra câu trả lời).

KHOA CNTT

1.3. Hướng tiếp cận

Có hai tiếp cận để cải thiện các kết quả tìm kiếm thông qua phương pháp ngữ nghĩa:

1. Kiến trúc của Semantic Web.
2. Lập chỉ mục cho ngữ nghĩa tiềm tàng (Latent Semantic Indexing).

Tuy nhiên, hầu hết các công cụ tìm kiếm dựa trên ngữ nghĩa phải chịu những vấn đề thực thi bởi qui mô của mạng ngữ nghĩa rất lớn. Nhằm mục đích làm cho tìm kiếm ngữ nghĩa trở nên hiệu quả trong việc tìm kiếm các kết quả mong muốn, mạng này phải chứa một lượng lớn các thông tin liên quan. Cùng lúc đó, một mạng rộng lớn tạo ra những khó khăn trong việc xử lí nhiều đường dẫn có thể có cho một giải pháp liên quan.

Chúng ta sử dụng khía cạnh sắc bén của công nghệ Web ngữ nghĩa – kết hợp chặt chẽ sự phối hợp của các công nghệ tiên tiến – làm cho mô hình có thể chuyển nhanh trong việc tìm kiếm thông tin.

- **Công nghệ xử lí ngôn ngữ tự nhiên** cho phép người dùng hỏi những câu hỏi mà họ muốn, hơn là phải nêu lên những từ khoá có liên quan trong câu hỏi của họ.
- **Các Ontology định nghĩa lĩnh vực quan tâm**. Chúng được xem như là "bộ não" của công cụ tìm kiếm, bởi vì nó cố gắng hiểu những câu truy vấn của người dùng trong các từ của ontology này. Theo cách này chú ý rằng công cụ tìm kiếm ngữ nghĩa của chúng ta không phải là có mục đích thông thường như Google, mà nó có ý định áp dụng đối với một lĩnh vực hay khu vực cụ thể (ví dụ về lĩnh vực pháp lí, văn hoá, thể thao v.v…).
- **Phân tích tri thức**. Công nghệ này chuyển dữ liệu không có cấu trúc sang thông tin có cấu trúc. Nó rút trích thông tin từ các văn bản tự do,

KHOA CNTT

các văn bản bán cấu trúc và cấu trúc để phát sinh ra ontology với tri thức thật sự.

- **Truy cập tri thức thông minh**. Các câu trả lời cho các truy vấn đạt được do việc truy vấn ontology được đưa ra tự động, và được biểu diễn trong những dạng khác nhau:
  - ***"Dữ liệu"*** của thực thể chính được hỏi đến (ví dụ trong lĩnh vực xã hội, dữ liệu của một nghệ sĩ).
  - ***Định hướng ngữ nghĩa***. Những từ của các câu trả lời được tự động siêu liên kết đến các khái niệm ontology con, cho phép định hướng bằng "ý nghĩa".
  - ***Các thể thông minh và liên kết thông minh***. Các câu trả lời luôn được sao lưu bởi các nguồn và các tài liệu chúng dựa vào. Khi những tài liệu đó được tra cứu, thì phần mềm gán thẻ và liên kết sẽ tự động nhận ra các từ chứa ý nghĩa lĩnh vực và liên kết chúng đến ontology, hay thêm vào các thẻ thông minh với những hoạt động được định nghĩa trong ontology.
  - ***Sự "tưởng tượng" thông minh***. Thông thường, các câu trả lời phát sinh ra nhiều các khái niệm liên quan và các mối quan hệ. Phầm mềm "tưởng tượng" thông minh cho phép một khái niệm đi xuyên qua tri thức này.

Có một vấn đề mà công cụ tìm kiếm ngữ nghĩa được định nghĩa ở đây vẫn chưa thể hoàn tất so với những công cụ tìm kiếm với mục đích thông thường (không có ngữ nghĩa) như Google đó là: phạm vi. Trong Google bạn có thể tìm kiếm với bất kỳ từ khoá nào trong bất kỳ lĩnh vực nào. Nếu các từ khoá xuất hiện trong một số tài liệu trên Web, Google sẽ tìm thấy nó. Một công cụ tìm kiếm ngữ nghĩa cần một số tri thức nâng cao: nó cần biết ý nghĩa, được biểu diễn trong một ontology. Thực tế là các ontology – trong trạng thái thi hành hiện tại – vẫn còn làm bằng thủ công, hạn chế chúng trong những mục đích thông thường. Do đó, các công cụ tìm kiếm ngữ nghĩa là những công cụ quan trọng cho những lĩnh vực cụ thể. Trong trường hợp này, mục đích

KHOA CNTT

của các công cụ tìm kiếm ngữ nghĩa là bổ sung cho các công cụ tìm kiếm thông thường, hơn là cạnh tranh như những đối thủ .

KHOA CNTT

Chương 2 : CƠ SỞ LÝ THUYẾT

2.1. Chiến lược tìm kiếm thông tin của các bộ tìm kiếm (Search Engine)

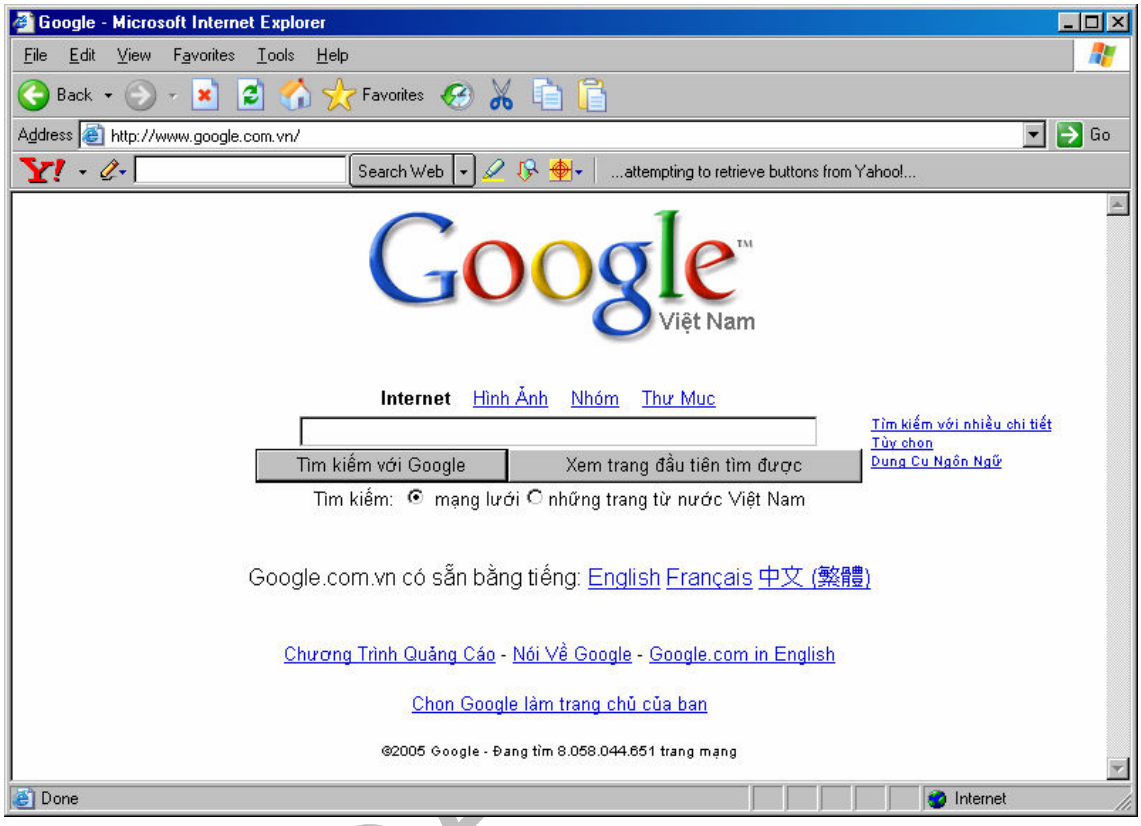
2.1.1. Một số search engine thông dụng:

Sau đây là danh sách một số search engine. Tại sao chúng được xem là những search engine "lớn"? Đó là bởi vì chúng được biết đến nhiều và sử dụng tốt. Đối với các chuyên gia web, các công cụ tìm kiếm lớn là danh sách những nơi quan trọng nhất bởi chúng phát sinh ra một lượng rất lớn các trang web tiềm tàng. Đối với những người tìm kiếm, các công cụ tìm kiếm phổ biến thường trả ra các kết quả đáng tin cậy hơn. Những search engine này rất có thể được duy trì tốt và nâng cấp khi cần thiết, để giữ thế cân bằng với tốc độ phát triển của web.

Những search engine sau là tất cả những lựa chọn tốt nhất để bắt đầu khi tìm kiếm thông tin:

KHOA CNTT

2.1.1.1. Google: http://www.google.com/



Hình 1: Giao diện của Google

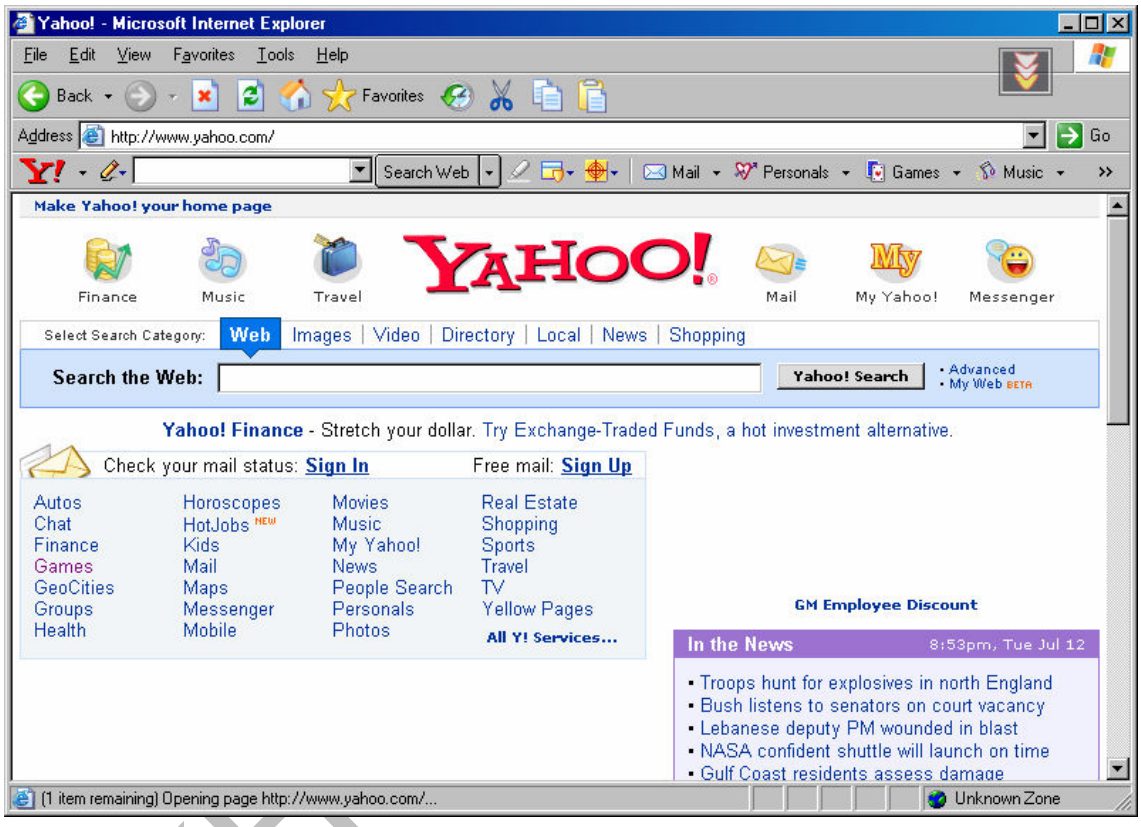
Nguyên thuỷ, Google là một đề án của trường đại học Stanford được thực hiện bởi hai sinh viên Larry Page và Sergey Brin gọi là BackRub. Đến năm 1998, thì đổi tên thành Google, và đồ án này đã trở thành công ty riêng Google đặt tại khuôn viên trường đại học. Nó vẫn còn được lưu giữ cho đến ngày nay.

Google là công cụ tìm kiếm nổi tiếng, tốt nhất trong các lựa chọn để tìm kiếm thông tin trên web. Dịch vụ dựa vào crawler, spider cung cấp trang web với thông tin đưa ra toàn diện cùng với mức độ liên quan tốt. Đây là công cụ tốt nhất hiện nay trong việc tìm kiếm bất cứ thứ gì bạn muốn.

Tuy nhiên, Google cung cấp chọn lựa để tìm kiếm chủ yếu về các trang web. Sử dụng hộp tìm kiếm trên trang chủ Google, bạn có thể dễ dàng định vị các ảnh qua

web, những đề nghị được đặt trong các nhóm thảo luận Usenet, định vị thông tin tin tức hay thực hiện tìm kiếm sản phẩm.

2.1.1.2. Yahoo: http://www.yahoo.com/



Hình 2: Giao diện của Yahoo

Đưa ra năm 1994, Yahoo là "thư mục" cũ nhất của web, một nơi mà các nhà biên tập tổ chức các trang web trong các danh mục. Tuy nhiên, vào tháng 10 năm 2002, Yahoo chuyển sang lập danh sách dựa vào crawler cho những kết quả chính của nó. Công cụ này sử dụng công nghệ từ Google cho đến tháng 2 năm 2004. Hiện nay, Yahoo sử dụng công nghệ tìm kiếm riêng của mình.

Yahoo Directory vẫn tồn tại. Bạn sẽ chỉ ra các liên kết "danh mục" phía dưới một số các trang web liệt kê trong kết quả trả về của một tìm kiếm từ khoá. Khi được

đề xuất, những trang web này dẫn bạn đến một danh sách các trang web đã được xem xét và phê chuẩn bởi một nhà biên tập.

Công nghệ AltaVista và AllTheWeb được phối hợp với kỹ thuật Inktomi, một công cụ tìm kiếm dựa trên crawler, để tạo nên một Yahoo crawler hiện nay.

2.1.1.3. Ask Jeeves: http://www.askjeeves.com/

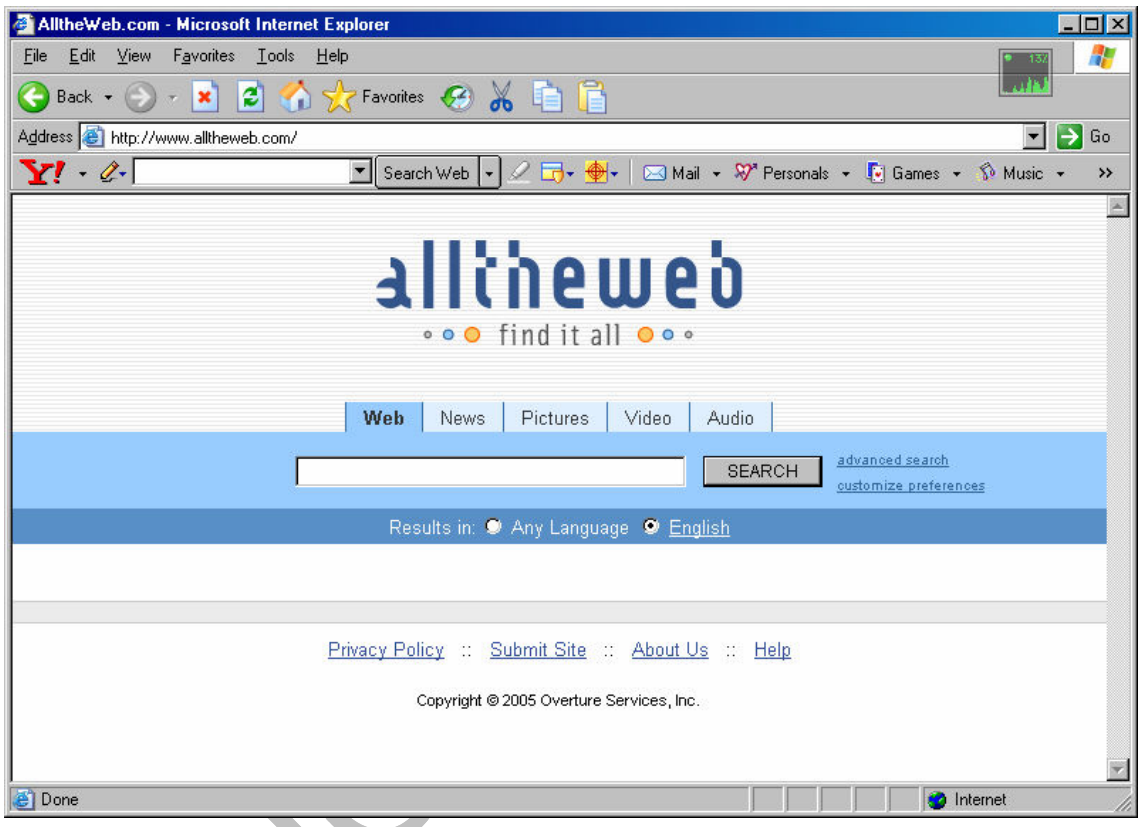
Hình 3: Giao diện của Ask Jeeves

Ask Jeeves bắt đầu nổi tiếng từ năm 1998 và 1999, được biết như là một công cụ tìm kiếm "ngôn ngữ tự nhiên" cho phép ta tìm kiếm bằng cách hỏi những câu hỏi và trả về kết quả với những gì *có vẻ là* trả lời đúng về mọi thứ.

Thực sự, công nghệ không phải là những gì làm cho Ask Jeeves thực thi tốt. Bên cạnh các bối cảnh, công cụ này tại một thời điểm có khoảng 100 trình soạn thảo

giám sát các log tìm kiếm. Sau đó chúng vào trong web và định vị những site mà chúng cho là tốt nhất tương xứng với các truy vấn phổ biến nhất.

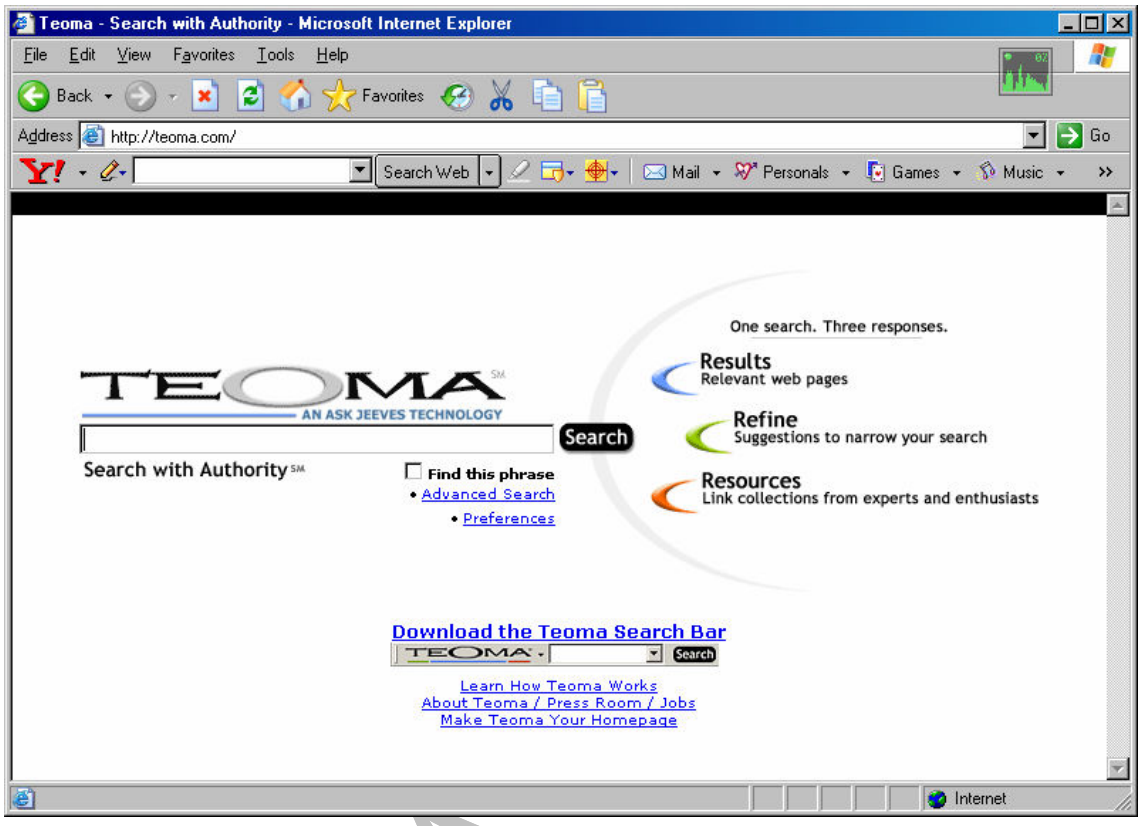
2.1.1.4. AllTheWeb: http://www.alltheweb.com/



Hình 4: Giao diện của AllTheWeb

Được Yahoo cung cấp nguồn, có thể thấy AllTheWeb là một "tìm kiếm thuần tuý" ("pure search") nhẹ nhàng hơn, tuỳ biến hơn và dễ chịu hơn là khi thực hiện ở Yahoo. Tiêu điểm là trong tìm kiếm web, ngoại trừ tin tức, tìm kiếm hình ảnh, video, MP3 và FPT cũng được đưa ra.

2.1.1.5. Teoma: http://www.teoma.com/



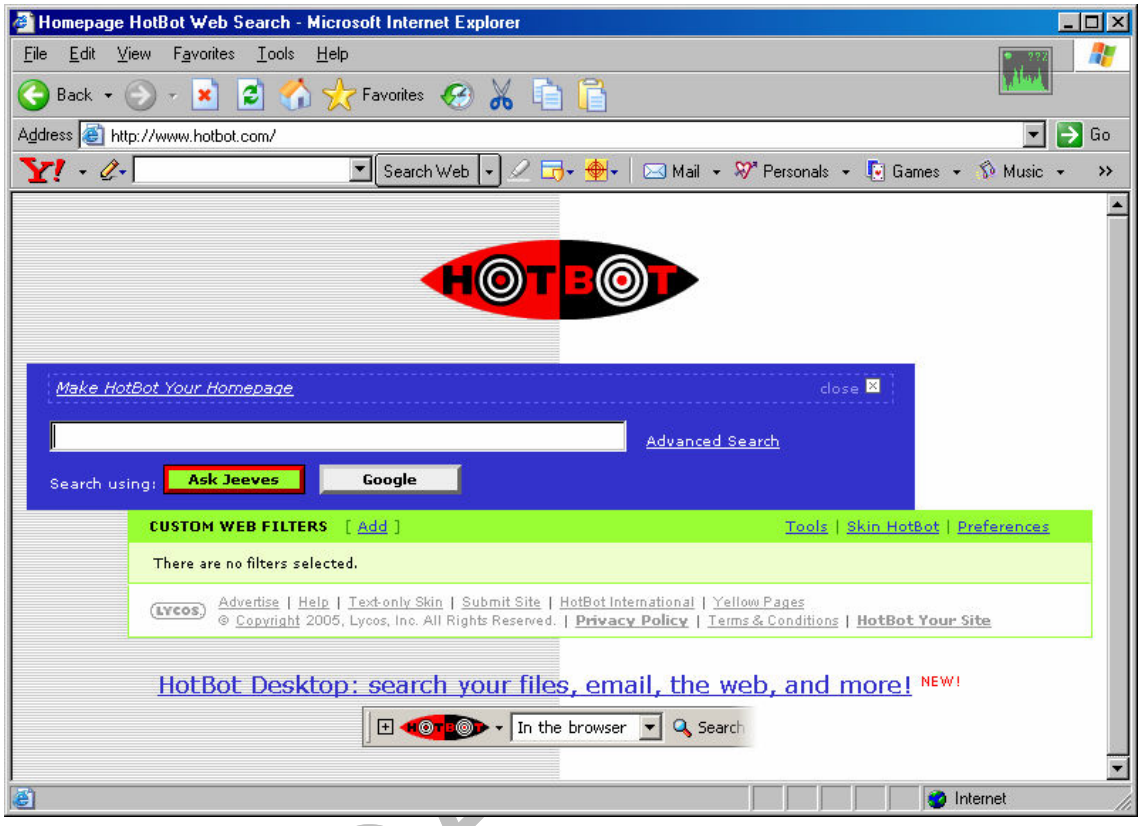
Hình 5: Giao diện của Teoma

Teoma là một công cụ tìm kiếm dựa trên crawler được sở hữu bởi Ask Jeeves. Nó có số lượng trang web được chỉ mục nhỏ hơn Google và Yahoo. Năm 2000, Teoma ra đời cùng với thành công của mình: đưa ra được những thứ liên quan. Tính năng "Refine" của công cụ này đề xuất ra những chủ đề để khảo sát sau khi bạn thực hiện một tìm kiếm.

Teoma được Ask Jeeves mua vào tháng 9 năm 2001 và cũng cung cấp một số kết quả cho web site này.

KHOA

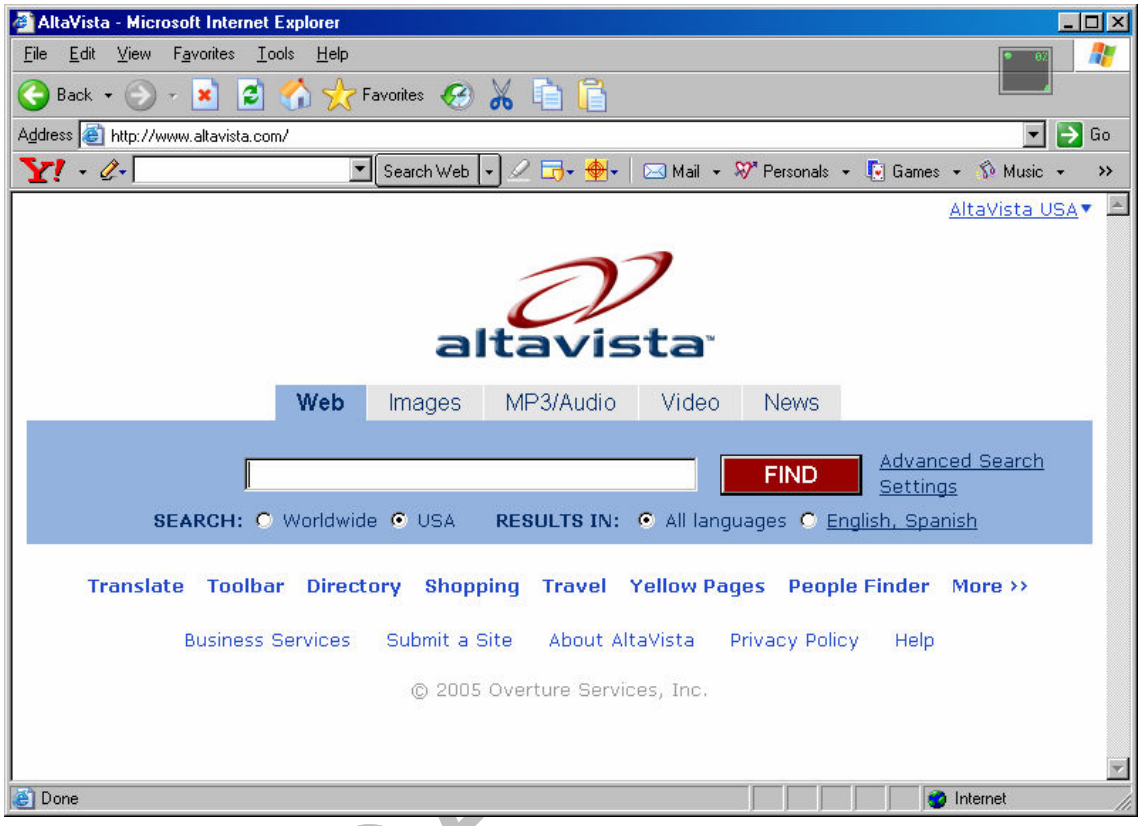
2.1.1.6. HotBot: http://www.hotbot.com/



Hình 6: Giao diện HotBot

HotBot hỗ trợ truy cập dễ dàng đến 3 trang web search engine dựa vào crawler lớn: Yahoo, Google, và Teoma. Không như một meta search engine, nó không thể pha trộn các kết quả từ tất cả các crawler này với nhau. Do đó, nó là một cách nhanh, dễ dàng để lấy các "ý kiến" tìm kiếm web khác nhau trong một nơi.

2.1.1.7. AltaVista: http://www.altavista.com/

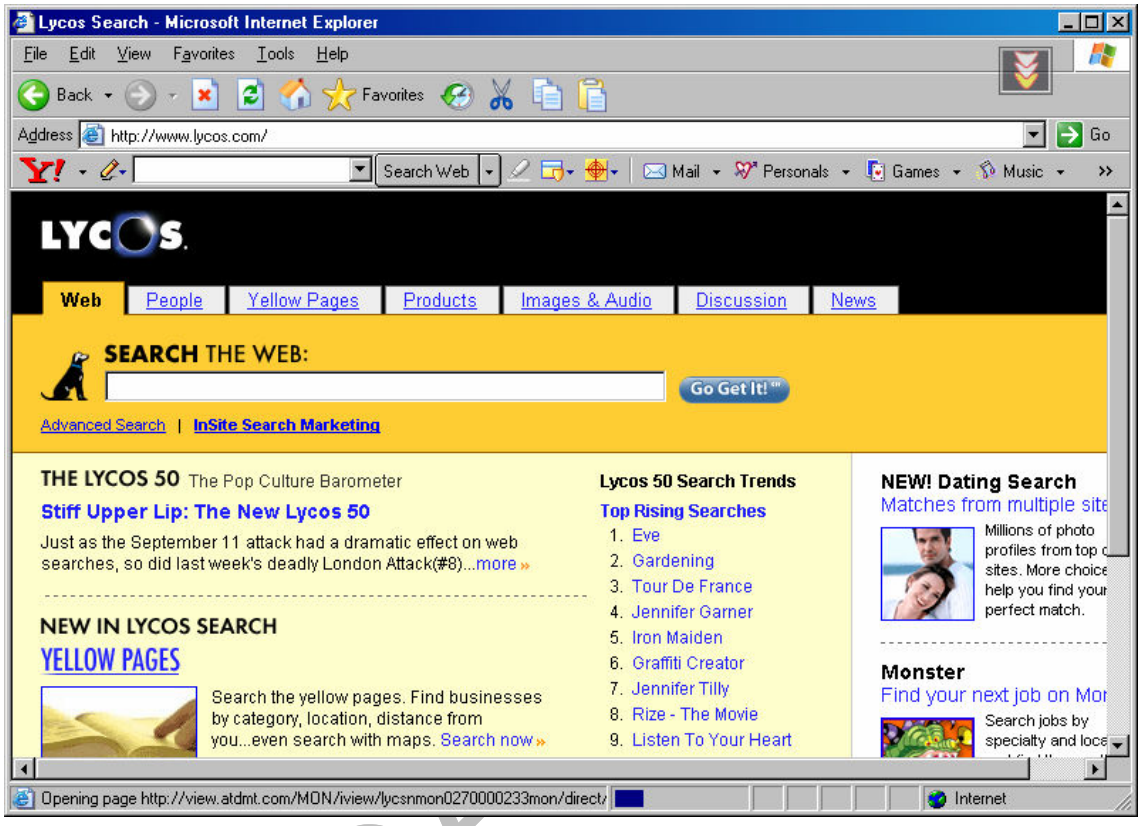


Hình 7: Giao diện của Altavista

AltaVista được đưa ra vào tháng 9 năm 1995 và được xem như là "Google" trong một vài năm, nó cung cấp những kết quả liên quan và đã có một nhóm người dùng yêu thích công cụ tìm kiếm này. Nhưng từ sau năm 1998, người ta không còn ưa chuộng AltaVista nữa, bởi vì sự mới mẻ của các danh sách AltaVista và tin tức được đưa ra của crawler trong trang web này không được cập nhật thường xuyên.

Ngày nay, AltaVista một lần nữa tập trung vào tìm kiếm. Các kết quả đến từ Yahoo, và cho phép đến các trang web để tìm hình ảnh, MP3/Audio, Video, các danh sách danh mục con người và các kết quả tin tức. Nếu muốn một cảm giác nhẹ nhàng hơn Yahoo nhưng vẫn có các kết quả của Yahoo, AltaVista là một chọn lựa tốt.

2.1.1.8. Lycos: http://www.lycos.com/



Hình 8: Giao diện của Lycos

Lycos là một trong những công cụ tìm kiếm cũ nhất trên web, được đưa ra năm 1994. Được mô tả như là những cổng truy cập web (web portal) hay những trung tâm truy cập, là nơi mà người dùng đi vào để lấy thông tin cho mọi lĩnh vực, kể cả tán gẫu, gởi thư điện tử,…

KHOA

Search Engine	**Google**	**AlltheWeb**	**AltaVista**	**Teoma**
Database	google.com	alltheweb.com	altavista.com	teoma.com
Kích thước(# trang)	Khoảng 8 tỉ (1 tỉ không đánh chỉ mục trên toàn văn bản)	Khoảng 3 tỉ, chỉ mục trên toàn văn bản.	Khoảng 1 tỉ	Khoảng 1 tỉ
Đa phương tiện (multimedia)	Hỗ trợ	Hỗ trợ	Hỗ trợ	Không hỗ trợ
Toán tử				
Mặc định	AND	AND	AND	AND
Loại trừ	-	-	-	-
Cụm từ	Dùng dấu " "	Dùng dấu " "	Dùng dấu " "	Dùng dấu " "
Rút gọn	Không hỗ trợ Dùng ký tự * để thay thế cho các ký tự trong dấu " "	Không hỗ trợ	Dùng ký tự *	Không hỗ trợ
Boolean	OR (chỉ dùng cho danh từ riêng)	AND, OR, ANDNOT, RANK, ()	AND, OR, ANDNOT, NEAR, ()	OR (chỉ dùng cho tên riêng)

KHOA CNTT

Stop words	Thông thường bỏ qua các từ thông dụng + nếu muốn tìm và phải đặt trong cặp dấu " "		Dùng dấu " " trong search cơ bản Bỏ qua trong search nâng cao	Thông thường bỏ qua các từ thông dụng + nếu muốn tìm
Danh từ riêng	Không hỗ trợ	Không hỗ trợ	Hỗ trợ	Không hỗ trợ
Giới hạn field cần tìm	intitle: inurl: allintitle: Allinurl: filetype: Link:site: <u>Trong search nâng cao :</u> cache:info:	Normal.title: url.all: Link.all: Link.extension: :	Title: domain: Link: image: Text: url: host: Anchor: applet:	intitle: inurl: site: geoloc: lang: last: afterfate:
Các đặc tính đặc biệt	~ tìm từ đồng nghĩa Giới hạn bởi ngôn ngữ Nhiều kiểu file : pdf, doc,… Caches : trang web khi đánh chỉ mục	Duyệt qua các URL Trong tìm nâng cao : giới hạn bởi ngày, domain, địa chỉ iP	Giới hạn bởi ngày, vị trí, ngôn ngữ Trong tìm nâng cao : sử dụng *sortby* để lọc và sắp xếp kết quả.	Dùng *refine* để tối ưu kết quả. *Resource* để có được các trang và liên kết tập trung trên chủ đề cần tìm.

KHOA CNTT

Ưu điểm				
Ưu điểm chính	Rất tốt với những trang có độ phổ biến cao. Các trang tin tức gần đây	Tốt như Google. Không có stopword.	Dùng nhiều toán tử Boolean trong tìm kiếm. Trong tìm nâng cao hỗ trợ hiển thị kết quả theo độ phổ biến của từ.	Tính độ phổ biến tốt, dựa vào số lượng trang web cùng chủ đề với các trang đang xét. Thường đạt kết quả đáng khích lệ.
Search Engine	Google	AlltheWeb	AltaVista	Teoma

Bảng 1 : Bảng hướng dẫn nhanh về cách sử dụng một số search engine phổ biến

KHOA CNTT

Search engine	**Cơ sở dữ liệu**	**Toán tử**	**Lựa chọn tìm kiếm**	**Linh tinh**
Google http://www.google.com Hỗ trợ tìm kiếm nâng cao Hệ thống thư mục chủ đề (Subject	Toàn văn bản của các trang web, .pdf, .doc, .xls, .ps, .wpd (4.3B, + 1B một phần của chỉ mục URLs)	AND (mặc định) OR (danh từ riêng) + cho các stop word thông dụng, cho các URL hoặc các trang cụ thể (ví	Dùng * để rút gọn. Dùng “” tìm cụm từ. Fields : intitle:, inurl:, link:, site: Tìm trên hệ thống danh mục các chủ đề trong	Kiểm lỗi chính tả. Lưu trữ các trang đã lập chỉ mục. Tốt cho tìm các trang hay bị lỗi 404. Phiên dịch đến 5 ngôn ngữ.

Directory) Hệ thống thư mục mở (Open Directory)	Tin tức : cập nhật thường xuyên (4500 nguồn). Các dạng file ảnh Nhóm : Usenet từ 1981 đến nay	dụ +edu) - loại trừ	thư mục web. Tìm các trang web tương tự.	~ tìm từ đồng nghĩa.
AlltheWeb http://allthewеb.com Hỗ trợ tìm kiếm nâng cao	Toàn bộ văn bản các trang web, .pdf, Flash, (3.1B toàn bộ chỉ mục URLs) Tin tức : cập nhật thường xuyên (3000 nguồn) Tranh ảnh Video Audio FPT	AND (mặc định) OR, phải đặt các từ trong dấu " ". ANDNOT, RANK - để loại bỏ	Không rút gọn. Dùng dấu " " cho cụm từ. Field intitle:inurl: link:site: Trong tìm nâng cao : giới hạn theo ngày, ngôn ngữ, domain, file format, địa chỉ iP.	Kiểm lỗi chính tả. Tìm nâng cao : tranh ảnh, video. Hỗ trợ sử dụng kỹ thuật "clusters" để tối ưu câu truy vấn.
AltaVista http://altavista.com	Toàn bộ văn bản các trang web (khoảng	AND (mặc định) Trong tìm nâng	Dấu * để rút gọn. Dấu "" cho cụm từ.	Kiểm lỗi chính tả. Phiên dịch : 8

KHOA CNTT

Hỗ trợ tìm kiếm nâng cao Hệ thống thư mục chủ đề (Subject Directory) Hệ thống thư mục mở (Open Directory)	1B) và file .pdf. Tin tức (3000 nguồn), ảnh, MP3/Audio, Video.	cao hoặc danh từ riêng trong tìm cơ bản : AND, OR, ANDNOT, NEAR, dấu () lồng nhau. - cho loại trừ.	Tìm nâng cao : giới hạn ngày, ngôn ngữ.	ngôn ngữ của Châu Âu & các ngôn ngữ của Châu Á. AltaVistaPrima : tối ưu câu hỏi.
Teoma http://teoma.com Hỗ trợ tìm kiếm nâng cao	Toàn bộ văn bản trang web (khoảng 1B)	AND (mặc định) OR (danh từ riêng) + hoặc “” cho stopword - để loại bỏ	Không rút gọn. Dùng dấu “ “ cho cụm từ. Field intitle:inurl: site:geoloc:lang:last: afterdate:beforedate: betweendate: Trong tìm nâng cao : giới hạn theo ngày, ngôn ngữ, domain, file format, địa chỉ iP.	Kiểm lỗi chính tả. Gom nhóm kết quả *Refine* để tối ưu câu hỏi. *Resource* để có các trang hoặc liên kết tập trung vào chủ đề.

KHOA CNTT

AskJeeves www.ask.com	Nhận kết quả từ CSDL của Teoma. Tìm sản phẩm : PriceGrabber.com, Tìm tranh ảnh : Picsearch.com Tìm tin tức : Moreover.com.	Giống Teoma. Đối với những câu hỏi đơn giản, xuất hiện cửa sổ đối thoại.	Giống Teoma. Click vào *Remove Frame* để thấy URLs của các trang.	Kiểm lỗi chính tả.
AskJeeves for Kids www.ajkids.com	Trả lời tốt các câu hỏi đơn giản. Games cho trẻ em, Tin tức theo từng nhóm tuổi.	Hỏi bằng ngôn ngữ tự nhiên. Không sử dụng các toán tử Boolean.	Click vào *No frames* để thấy URL của trang kết quả.	Dẫn đến các trang phục vụ học tập : tự điển, vật lý, khoa học, bản đồ, lịch sử,…
Yahoo http://dir.yahoo.com	Xem xét các trang web (khoảng 13K)	AND (mặc định) OR	Cụm từ : “” Rút gọn : * Fields t: title, u:URL	Nhiều dịch vụ trong Yahoo: Tin tức : từng giờ. Thể thao :tỉ số,.. Bản đồ, thời tiết,

KHOA CNTT

				mua sắm.

Bảng 2: Sơ lược về các đặc trưng của một số search engine thông dụng trên Internet

2.1.2. Chiến lược tìm kiếm

Từ "search engine" thường được sử dụng rộng rãi để mô tả các công cụ tìm kiếm dựa trên crawler và các thư mục do con người cung cấp. Đây là hai loại của các search engine tập hợp các danh sách của chúng trong những cách khác nhau hoàn toàn.

Search engine dựa vào crawler gồm 3 phần:

- **Bộ thu thập thông tin – Robot**

Robot là một chương trình tự động duyệt qua các cấu trúc siêu liên kết để thu thập tài liệu và một cách đệ quy nó nhận về tất cả các tài liệu có liên kết với tài liệu này.

Robot được biết đến dưới nhiều tên gọi khác nhau : spider, web wanderer hoặc web worm, crawler… Những tên gọi này đôi khi gây nhầm lẫn, như từ ' spider ', ' wanderer ' làm người ta nghĩ rằng robot tự nó di chuyển và từ ' worm ' làm người ta liên tưởng đến virus. Về bản chất robot chỉ là một chương trình duyệt và thu thập thông tin từ các site theo đúng giao thức web. Những trình duyệt thông thường không được xem là robot do thiếu tính chủ động, chúng chỉ duyệt web khi có sự tác động của con người.

- **Bộ lập chỉ mục – Index**

Hệ thống lập chỉ mục hay còn gọi là hệ thống phân tích và xử lý dữ liệu, thực hiện việc phân tích, trích chọn những thông tin cần thiết (thường là các từ đơn , từ ghép , cụm từ quan trọng) từ những dữ liệu mà robot thu thập được và tổ chức thành

KHOA CNTT

cơ sở dữ liệu riêng để có thể tìm kiếm trên đó một cách nhanh chóng, hiệu quả. Hệ thống chỉ mục là danh sách các từ khoá, chỉ rõ các từ khoá nào xuất hiện ở trang nào, địa chỉ nào.

❖ Bộ tìm kiếm thông tin – Search Engine

Search engine là cụm từ dùng để chỉ toàn bộ hệ thống bao gồm bộ thu thập thông tin, bộ lập chỉ mục và bộ tìm kiếm thông tin. Các bộ này hoạt động liên tục từ lúc khởi động hệ thống, chúng phụ thuộc lẫn nhau về mặt dữ liệu nhưng độc lập với nhau về mặt hoạt động.

Search engine tương tác với user thông qua giao diện web, có nhiệm vụ tiếp nhận và trả về những tài liệu thoả yêu cầu của user.

Nói nôm na, tìm kiếm từ là tìm kiếm các trang mà những từ trong câu truy vấn (query) xuất hiện nhiều nhất, ngoại trừ stopword (các từ quá thông dụng như mạo từ a, an, the,…). Một từ trong câu truy vấn càng xuất hiện nhiều trong một trang thì trang đó càng được chọn để trả về cho người dùng. Và một trang chứa tất cả các từ trong câu truy vấn thì tốt hơn là một trang không chứa một hoặc một số từ. Ngày nay, hầu hết các search engine đều hỗ trợ chức năng tìm cơ bản và nâng cao, tìm từ đơn, từ ghép, cụm từ, danh từ riêng, hay giới hạn phạm vi tìm kiếm như trên đề mục, tiêu đề, đoạn văn bản giới thiệu về trang web,…..

Ngoài chiến lược tìm chính xác theo từ khoá, các search engine còn cố gắng ‘hiểu’ ý nghĩa thực sự của câu hỏi thông qua những câu chữ do người dùng cung cấp. Điều này được thể hiện qua chức năng sửa lỗi chính tả, tìm cả những hình thức biến đổi khác nhau của một từ. Ví dụ : search engine sẽ tìm những từ như speaker, speaking, spoke khi người dùng nhập vào từ speak.

KHOA CNTT

Nguyên lý hoạt động

Search engine điều khiển robot đi thu thập thông tin trên mạng thông qua các siêu liên kết (hyperlink). Khi robot phát hiện ra một site mới, nó gởi tài liệu (web page) về cho server chính để tạo cơ sở dữ liệu chỉ mục phục vụ cho nhu cầu tìm kiếm thông tin.

Bởi vì thông tin trên mạng luôn thay đổi nên robot phải liên tục cập nhật các site cũ. Mật độ cập nhật phụ thuộc vào từng hệ thống search engine. Khi search engine nhận câu truy vấn từ user, nó sẽ tiến hành phân tích, tìm trong cơ sở dữ liệu chỉ mục và trả về những tài liệu thoả yêu cầu.

2.2. Semantic Web

2.2.1. Khái niệm

"Web ngữ nghĩa" là một dạng mở rộng của web hiện nay, mà cho phép ta truy tìm, chia sẻ, phối hợp, sử dụng lại và rút trích thông tin một cách chính xác, dễ dàng."(Tim – Berners Lee, XML – 2000).

Web ngữ nghĩa là một mạng lưới thông tin được liên kết theo cách mà máy tính có thể dễ dàng xử lý được trên quy mô toàn cầu. Chúng ta có thể xem web ngữ nghĩa như là một cơ sở dữ liệu toàn cầu được liên kết với nhau.

Web ngữ nghĩa được phát triển bởi Tim – Berners Lee, nhà phát minh của WWW, URIs, HTTP, và HTML. Hiện nay có một nhóm nghiên cứu tại tập đoàn WWW đang cải tiến, mở rộng và tiêu chuẩn hoá hệ thống ngữ nghĩa.

Dữ liệu trong tập tin HTML thường hữu ích trong một số trường hợp. Phần lớn dữ liệu trên web là dạng HTML nên khó sử dụng trên quy mô lớn, bởi vì nó không có một hệ thống toàn cầu để xuất bản dữ liệu.

Do đó, Web ngữ nghĩa được xem như là một giải pháp kỹ thuật.

Web ngữ nghĩa được xây dựng chủ yếu trên cú pháp sử dụng URIs để biểu diễn dữ liệu, thường thấy là cấu trúc dựa trên bộ ba (subject, predicate, object), ví dụ: nhiều bộ ba của dữ liệu URI có thể được cất giữ trong cơ sở dữ liệu, hoặc thay thế lẫn nhau

KHOA CNTT

trên word wide web bằng cách sử dụng một tập các cú pháp đặc biệt được pháp triển chuyên biệt phục vụ cho nhiệm vụ đó. Cú pháp này được gọi là cú pháp RDF.

Web ngữ nghĩa yêu cầu dữ liệu không những máy có thể đọc được mà còn mong muốn máy có thể hiểu được. Trích dẫn câu nói của Tim – Berners Lee:

"*The semantic web goal is to be a unifying system which will (like the web for human communication) be as un-restraining as possible so that the complexity of reality can be described*".

Tạm dịch là: "Mục đích của web ngữ nghĩa là để một hệ thống hợp nhất (giống như web dành cho sự giao tiếp của người) càng không bị cản trở càng tốt để mà độ phức tạp của thực tế có thể được mô tả".

Với web ngữ nghĩa, nó sẽ dễ dàng nhận biết toàn bộ phạm vi của các công cụ và ứng dụng khó giải quyết trong khuôn khổ của web hiện tại.

Hai công nghệ quan trọng cho việc phát triển semantic web là: e**X**tensible **M**arkup **L**anguage (**XML**) và **R**esource **D**escription **F**rameword (**RDF**). XML cho phép mọi người có thể tạo ra các tag (thẻ) của riêng mình. Còn RDF thì trình bày ngữ nghĩa, RDF sử dụng tập các ***triple*** để mô tả các khái niệm cơ sở.

URI (Uniform Resource Identifier):

Một URI đơn giản dùng để nhận biết một trang web: giống như các chuỗi bắt đầu với "http" hay "ftp" mà bạn thường thấy trên word wide web. Bất kỳ ai cũng có thể tạo ra một URI và quyền sở hữu chúng được uỷ quyền một cách rõ ràng, chính vì vậy chúng tạo nên cơ sở quan niệm để xây dựng web toàn cầu. Thực ra, word wide web có thể xem như là: bất kỳ thứ gì mà có URI được coi như là "on the web".

Các URIs là các chuỗi ký tự có thể nhận biết các tài nguyên trên web. Thông qua việc sử dụng URIs, chúng ta có thể sử dụng cùng cách đặt tên đơn giản để tham chiếu đến các tài nguyên dưới các nghi thức (protocol) khác nhau như là: HTTP, FTP, GOPHER, EMAIL, ….

URLs (Uniform Resource Locator): là một dạng được sử dụng rộng rãi của URIs, được sử dụng rất phổ biến trên web, là các địa chỉ của các tài nguyên. Mặc dù thường được biết đến như là các URLs, nhưng URIs cũng có thể được tham chiếu đến

KHOA CNTT

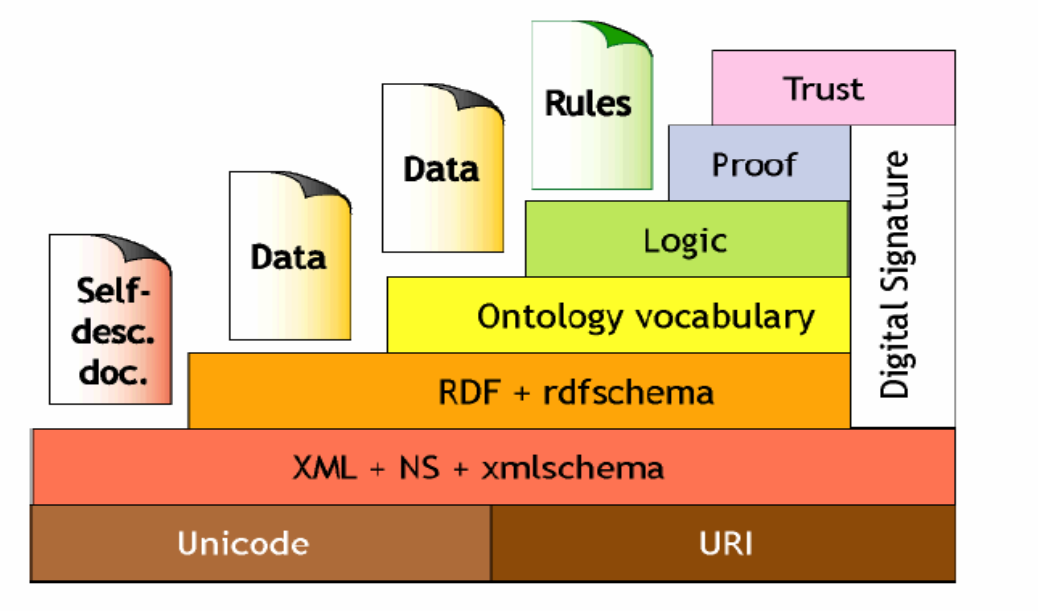
các khái niệm trong semantic web. Ví dụ, giả sử bạn có một quyển sách có tên là "Machine Learning", thì URI của nó sẽ như sau:

http://www.cs.bris.ac.uk/home/pw2538/book/title#machinelearning

Lưu ý là mọi thứ trên web đều có một URI duy nhất.

2.2.2. Kiến trúc

Web ngữ nghĩa được xây dựng theo mô hình kiến trúc phân tầng gồm có 7 tầng, các tầng như sau:



Hình 9: Kiến trúc tầng của Semantic web.

Tầng Unicode + URI:

Nhằm bảo đảm việc sử dụng tập ký tự quốc tế và cung cấp phương tiện để định danh các đối tượng trong Web ngữ nghĩa.

Tầng XML + NS + Lược đồ XML:

Cùng với các định nghĩa về namespace và schema bảo đảm rằng ta có thể tích hợp các định nghĩa web ngữ nghĩa với các chuẩn dựa trên XML khác.

Tầng RDF + Lược đồ RDF:

Dùng siêu dữ liệu mô tả tài liệu trên Web để máy có thể hiểu được chúng.

Tầng Ontology:

Lược đồ RDF cung cấp các công cụ để định nghĩa những từ vựng, cấu trúc và các ràng buộc trong việc mô tả cho siêu dữ liệu về các tài nguyên Web. Nhưng lược đồ RDF chưa thật sự đầy đủ cho việc mô hình hoá và hỗ trợ suy luận trên Semantic Web. Ngôn ngữ Ontology OIL được đề ra là một dạng mở rộng của lược đồ RDF. Nó cho phép thể hiện ngữ nghĩa hình thức, giúp hỗ trợ suy diễn tự động.

Tầng Logic:

Tầng logic được xem như là một cơ sở luật trên Semantic Web. Bản chất của cơ sở luật này có dạng như một hệ chuyên gia. Tầng này sẽ hỗ trợ các dịch vụ như : phân loại văn bản, rút trích dữ liệu.

Tầng Proof:

Trong khi tầng logic giúp hỗ trợ suy luận dựa vào cơ sở luật thì tầng Proof được dùng để chứng minh các suy diễn của hệ thống bằng cách liên kết các dữ kiện.

Tầng Trust:

Trong Web ngữ nghĩa các thông tin được sử dụng chung như một cơ sở dữ liệu toàn cầu, nên cần phải có một cái gì đó để bảo mật. Đó là nguyên nhân của sự ra đời của chữ ký điện tử, nó giúp cho thông tin trên Web đáng tin cậy hơn. Trust engine là một hệ thống đang được xây dựng dựa trên nền tảng của chữ ký điện tử. Các kỹ thuật để xây dựng chúng còn đang trong giai đoạn nghiên cứu và thử nghiệm.

KHOA CNTT

2.2.3. Các thách thức đặt ra cho Semantic web

2.2.3.1. Thách thức 1: Tính sẵn có của nội dung (The availability of content)

Nội dung của web ngữ nghĩa là nội dung web được chú thích theo các ontology đặc biệt, các ontology này định nghĩa ngữ nghĩa của các từ hoặc các khái niệm xuất hiện trong cùng một nội dung. Một sự mở rộng đơn giản đối với HTML là được dùng để chú thích các trang web với thông tin về ontology. Việc tạo nội dung semantic web là một thách thức lớn, bởi vì "cơ sở hạ tầng" của semantic web vẫn còn đang được xây

dựng (chưa hoàn chỉnh – RDF, OIL, DAML+OIL,…), hiện tại có rất ít nội dung web ngữ nghĩa có sẵn.

2.2.3.2. Thách thức 2: Các ontology sẵn có, phát triển và tiến hoá

Các ontology là chìa khóa đối với semantic web bởi vì chúng là những bộ chuyên chở ngữ nghĩa được chứa trong semantic web, có nghĩa là chúng cung cấp một tập từ vựng và ngữ nghĩa chú thích. Có 3 vấn đề chính cần được giải quyết đối với thách thức này, hai vấn đề đầu có liên quan đến các vấn đề về việc phát triển các ontology truyền thống mà cho đến tận bây giờ các vấn đề này vẫn chưa được giải quyết, và vấn đề thứ ba còn lại có liên quan nhiều đến khung cảnh mới của semantic web:

Vấn đề thứ nhất là việc xây dựng các ontology hạt nhân (*kernel)* để được sử dụng bởi tất cả các domain. Những khởi đầu tồn tại đối với việc xây dựng một số kernel ontology này là chúng phải được ứng dụng trong những domain khác nhau.

Vấn đề thứ hai là cung cấp sự hỗ trợ mang tính chất giải pháp và công nghệ đối với hầu hết các hoạt động của *tiến trình phát triển ontology*, bao gồm:

a. Sự thu thập tri thức, mô hình khái niệm và mã hoá ontology trong các ngôn ngữ semantic web (RDFS, OIL, DAML+OIL), và các ngôn ngữ mới – các ngôn ngữ mới này có thể sẽ được đưa ra trong những năm sắp tới [Maedche, Staab – 2001] .
b. Sự sắp xếp và ánh xạ ontology, sự tích hợp ontology, các công cụ chuyển đổi ontology, và các công cụ xây dựng ontology, nếu các ontology tồn tại sắp được sử dụng lại [Fensel et al, 2001], [Noy, Musen 2000].
c. Các công cụ kiểm tra tính bền vững cho các ontology được sử dụng lại [Gomez-Perez 1996].

Vấn đề thứ ba là sự tiến hoá của các ontology và mối quan hệ của chúng đối với các dữ liệu đã được chú thích. Các công cụ quản lý cấu hình là cần thiết cho sự điều khiển các phiên bản của mỗi ontology cũng như sự phụ thuộc lẫn nhau giữa chúng và

KHOA CNTT

các chú thích. Tất cả các vấn đề này có thể là không quan trọng lắm, nhưng cần thiết phải giải quyết trước khi một semantic web thực sự ra đời.

2.2.3.3. Thách thức 3: Scalability of semantic web content

Một khi chúng ta đã có nội dung của semantic web, chúng ta sẽ phải quan tâm đến việc phải quản lý nó như thế nào, có nghĩa là cách tổ chức nó như thế nào, nơi lưu trữ nó và cách để tìm được nội dung đúng đắn. Có 2 vấn đề chính trong thách thức này:

a. Vấn đề thứ nhất có liên quan đến việc lưu trữ và tổ chức của các trang web ngữ nghĩa (semantic web pages). Semantic web "cơ sở" bao gồm các trang được chú thích dựa trên ontology, cấu trúc liên kết của các trang này phản ánh cấu trúc của WWW, có nghĩa là các trang liên kết với những trang khác thông qua các **hyperlink.** Theo cách liên kết này (hyperlink) thì không khai thác được đầy đủ ngữ nghĩa của các trang web ngữ nghĩa. Chiến lược **semantic indexes** được đề xuất để gom nhóm nội dung của semantic web dựa trên các chủ đề cụ thể. Semantic indexes sẽ được phát sinh tự động bằng cách sử dụng thông tin của ontology và các tài liệu đã được chú thích.
b. Vấn đề thứ hai có liên quan đến việc dễ dàng tìm kiếm thông tin trên semantic web, nói cách khác là có liên quan đến việc phối hợp giữa các semantic indexes.

2.2.3.4. Thách thức 4: Đa ngôn ngữ

Việc học dựa trên sự phân tán của ngôn ngữ thông qua nội dung của WWW chỉ ra rằng thậm chí nếu tiếng Anh là ngôn ngữ ưu thế hơn đối với các tài liệu, một số tài nguyên được viết bằng ngôn ngữ khác cũng rất quan trọng: Tiếng Anh 68,4%; Tiếng Nhật 5,9%; Tiếng Đức 5,8%; Tiếng Trung Quốc 3,9%; Tiếng Pháp 3,0%; Tiếng Tây Ban Nha 2,4%; Tiếng Nga 1,9%; Tiếng Italia 1,6%; Tiếng Bồ Đào Nha 1,4%; Tiếng Hàn 1,3%; Các ngôn ngữ khác 4,6% [www.vilaweb.com]. Tính đa dạng của ngôn ngữ còn quan trọng hơn nhiều đối với các tài nguyên WWW. Đa ngôn ngữ đóng vai trò

KHOA CNTT

ngày càng lớn đối với các cấp độ sau: ở cấp độ ontology, ở cấp độ chú thích, và ở cấp độ giao diện người dùng.

Ở cấp độ ontology, những người thiết kế ontology có thể muốn sử dụng ngôn ngữ địa phương của mình cho việc phát triển ontology mà trong đó các chú thích sẽ được gắn vào. Bởi vì không phải tất cả người sử dụng đều là những người xây dựng ontology, nên cấp độ này có độ ưu tiên thấp nhất. Sự tồn tại cuả đa ngôn ngữ và các tài nguyên ngôn ngữ học, như là WordNet [wordnet], EuroWordnet [eurowordnet],…có thể được xem xét tỉ mỉ để hỗ trợ vấn đề đa ngôn ngữ ở cấp độ này.

Ở cấp độ chú thích (**annotation**), chú thích của nội dung có thể được thực hiện trong nhiều ngôn ngữ khác nhau. Bởi vì nhiều người dùng (đặc biệt là các nhà cung cấp nội dung) sẽ thích chú thích nội dung hơn là phát triển các ontology, sự hỗ trợ phù hợp là cần thiết phải để cho các nhà cung cấp (nội dung) chú thích nội dung bằng ngôn ngữ địa phương của họ. Để có thể phát sinh nội dung web ngữ nghĩa bằng tất cả khả năng, chúng ta không thể yêu cầu chú thích nội dung từ tiếng Pháp sang tiếng Đức được và ngược lại.

Cuối cùng ở cấp độ giao diện người dùng, hàng tỉ người muốn truy xuất vào nội dung thích hợp bằng ngôn ngữ địa phương của họ bất chấp ngôn ngữ nguồn – ngôn ngữ mà trong đó các chú thích được trình bày. Mặc dù hiện tại, đa số nội dung đều được viết bằng tiếng Anh, chúng ta hy vọng rằng sẽ có nhiều nội dung hơn được viết bằng nhiều ngôn ngữ khác. Bất kỳ hướng tiếp cận nào của semantic web cũng nên bao gồm các tiện ích truy xuất thông tin trong nhiều ngôn ngữ. Các công nghệ quốc tế hoá và địa phương hoá nên được xem xét cẩn thận đối với việc truy xuất thông tin cá nhân dựa trên ngôn ngữ địa phương của người dùng.

2.2.3.5. Thách thức 5: Visualization – sự mường tượng

Với sự gia tăng thông tin vượt bậc, sự mường tượng (hình dung) của trực giác về thông tin sẽ trở nên rất quan trọng, bởi vì người dùng sẽ yêu cầu sự dễ dàng để nhận biết sự phù hợp của nội dung cho mục đích của họ ngày càng gia tăng. Thêm vào đó việc sử dụng *semantic indexes* và các routers cho việc lưu trữ, tổ chức và tìm kiếm

KHOA CNTT

thông tin, về sau này sẽ yêu cầu một bước quan trọng trong sự mường tượng. Các công nghệ nên cho phép đối với các công nghệ 3 chiều và sự mường tượng mới để mường tượng ra nội dung của semantic web trong bất kỳ một ngôn ngữ web hiện tại nào (RDFS, OIL, DAML + OIL). Thông qua công nghệ hiển thị đồ hoạ thời gian thực 3D thoả đáng và việc khai thác các mối quan hệ ngữ nghĩa, một giao diện ba chiều mới có thể được phát sinh một cách tự động. Theo cách này, nhiều thông tin hơn có thể được trình bày trong một không gian nhỏ hơn, và người dùng có thể tương tác với các site một cách thực tế và tiện lợi [Van Harmelen et al 2001].

2.2.3.6. Thách thức 6: Sự chuẩn hoá các ngôn ngữ semantic web

Semantic web là một lĩnh vực đang nổi bật và WWW Consortium sẽ đưa ra các giới thiệu về các ngôn ngữ và công nghệ sẽ được sử dụng. Để vươn lên đến mức nghệ thuật trong semantic web, và các công cụ phần lớn phụ thuộc vào ngôn ngữ semantic web mà chúng được hỗ trợ, thì nhu cầu chuẩn hoá ngôn ngữ semantic web là một đòi hỏi cần thiết.

2.2.4. So sánh web và web ngữ nghĩa

Điểm giống nhau giữa Web và Web ngữ nghĩa: cả 2 đều dùng những liên kết (link) URI, nhưng Web ngữ nghĩa sử dụng các link này rất nhiều, việc sử dụng link làm gia tăng tính chính xác của thông tin.

Sự khác nhau cơ bản giữa Web và Web ngữ nghĩa:

Web ngữ nghĩa	**Web**
Web ngữ nghĩa là một không gian thông tin trong đó thông tin được biểu diễn thông qua một ngôn ngữ mà máy và người đều có thể hiểu được.	Web là một không gian thông tin chứa đựng thông tin chỉ hướng vào việc biểu diễn trong một ngôn ngữ tự nhiên mà chỉ có người mới hiểu được.
Web ngữ nghĩa là một dữ liệu liên kết với nhau một cách ngữ nghĩa và hình thức.	Web là một tập hợp thông tin liên kết với nhau một cách không hình thức.

KHOA CNTT

2.2.5. Các khái niệm liên quan

2.2.5.1. Metadata

Metadata là thông tin có cấu trúc mô tả, giải thích, định vị hoặc mặt khác làm cho dễ dàng truy vấn, sử dụng, quản lý một tài nguyên thông tin. Metadata thường được gọi là dữ liệu về dữ liệu (từ điển dữ liệu), hoặc là thông tin về thông tin.

Metadata là thông tin về thông tin, metadata được sử dụng rộng rãi trong thế giới thực cho mục đích tìm kiếm. Ví dụ, bạn muốn mượn một vài quyển sách ở một thư viện nào đó thông qua máy tính. Thường thì thư viện sẽ cung cấp một hệ thống tra cứu, hệ thống này cho phép bạn liệt kê sách theo tên tác giả (author), theo tựa sách (title), theo chủ đề (subject), v.v…. Danh sách liệt kê này chứa nhiều thông tin quan trọng như: tên tác giả, tựa sách, ISBN, và thông tin quan trọng nhất là nơi cất giữ sách. Bạn cần vài thông tin (trong trường hợp này là nơi cất giữ sách) mà bạn muốn biết và bạn sử dụng metadata (trong trường hợp này là: tên tác giả, tựa sách, và chủ đề) để lấy được sách.

Có 3 kiểu metadata:

a. Descriptive metadata: mô tả một tài nguyên cho những mục đích như là khám phá hoặc là nhận diện. Nó có thể bao gồm các phần tử như là: titles, astract, author, và keywords.
b. Structural metadata: ví dụ: cho biết các đối tượng phức hợp liên kết với nhau như thế nào, các trang (pages) được sắp xếp thành các chương như thế nào.
c. Administrative metadata: cung cấp thông tin giúp cho việc quản lý một tài nguyên, như là nó được tạo ra khi nào và như thế nào, kiểu file, và các thông tin kỹ thuật khác, và những ai có thể truy cập đến nó.

2.2.5.2. Namespace

Chúng ta có thể mở rộng tập từ vựng của chúng ta thông qua các namespace – là các nhóm của tên các phần tử và tên các thuộc tính. Giả sử, nếu bạn muốn gộp (include) một ký hiệu (symbol) được mã hoá trong một ngôn ngữ đánh dấu

KHOA CNTT

nào đó trong một tài liệu XML, thì bạn có thể khai báo một namespace (không gian tên) mà symbol đó thuộc về. Thêm vào đó, chúng ta có thể tránh được tình huống hai đối tượng XML trong các không gian tên khác nhau với cùng một tên mà có ý nghĩa khác nhau thông qua các đặc trưng của các namespace. Giải pháp là gán một tiền tố nhận biết namespace mà mỗi phần tử hoặc các thuộc tính thuộc về. Cú pháp của namespace như sau:

ns-prefix:local-name

Trong đó **ns-prefix** là tên của namespace, và **local-name** là tên của phần tử hoặc thuộc tính.

Ví dụ về namespace:

Tài liệu XML dưới đây là một thư viện sách. Chúng ta bắt đầu bằng phần tử gốc có tên thẻ là <libarary>, bên trong thẻ gốc chứa các phần tử sách <book> và tựa sách <title> như sau:

```
<library>
    <book>
        <title>
            Earthquakes for lunch
        </title>
    </book>
</library>
```

KHOA CNTT

Không gian tên cục bộ (local namespace):

Chúng ta có thể đặt thuộc tính xmlns ở phần tử gốc hay ở bất kỳ thẻ nào khác. Khi thuộc tính này không nằm trong thẻ gốc thì ta gọi đó là không gian tên cục bộ.

Ví dụ: Xem đoạn xml dưới đây:

```
<minhkhai: library
    xmlns: minhkhai= http://www.minhkhai.com.vn/spec>
```

```
<minhkhai:book>
        <minhkhai:title>
                Earthquakes for lunch.
        </minhkhai:title>
 </minhkhai:book>
<amazon:book
        xmlns:amazon=http://www.amazon.com.lib>
        <amazon:title>
                Earthquakes for lunch.
        </amazon:title>
 </amazon:book>
```

Trong ví dụ này thì namespace: xmlns:amazon=http://www.amazon.com.lib được gọi là không gian tên cục bộ.

2.2.6. Ontology

Thuật ngữ "ontology" được vay mượn từ triết học. Ý nghĩa đầu tiên của nó là "the branch of metaphysics that deals with the nature of being" [The American Heritage® Dictionary of the English Language: Fourth Edition (2000)].

Ontology là một công nghệ quan trọng mang tính chất xương sống, vì nó cung cấp một đặc tính quan trọng: ontology giao tiếp được giữa ngữ nghĩa hình thức mà máy tính có thể hiểu được với ngữ nghĩa của thế giới thực mà con người có thể hiểu được.

Những Ontology được phát triển trong trí tuệ nhân tạo để tri thức dễ dàng chia sẻ và sử dụng lại. Kể từ đầu thập niên 90 của thế kỷ XX, Ontology đã trở thành một đề tài nghiên cứu phổ biến đối với các tổ chức nghiên cứu trí tuệ nhân tạo, bao gồm những kỹ sư về tri thức (Knowledge), xử lý ngôn ngữ tự nhiên và trình bày tri thức.

Ontology không chỉ làm cho tri thức có thể sử dụng lại dễ dàng hơn, nó còn là nền tảng của việc tạo ra các chuẩn bởi vì nó làm rõ các khái niệm bên cạnh một thuật ngữ hoặc một mô hình. Yêu cầu trên thực tế không phải chỉ dành cho một khái niệm

KHOA CNTT

duy nhất, mà là đối với một sự tương tác mơ hồ giữa các khái niệm phức tạp và chi tiết (có thể được trình bày trong nhiều ngôn ngữ khác nhau).

Gần đây, khái niệm Ontology đã trở nên phổ biến hơn nhiều trong các lĩnh vực như sự tích hợp thông minh, những hệ thống thông tin hợp tác, phục hồi thông tin, giao dịch thương mại điện tử, và quản lý tri thức. Mục đích của Ontology là hướng đến tri thức miền, nên sự phát triển của nó thường là một quá trình xử lý kéo theo nhiều yếu tố khác.

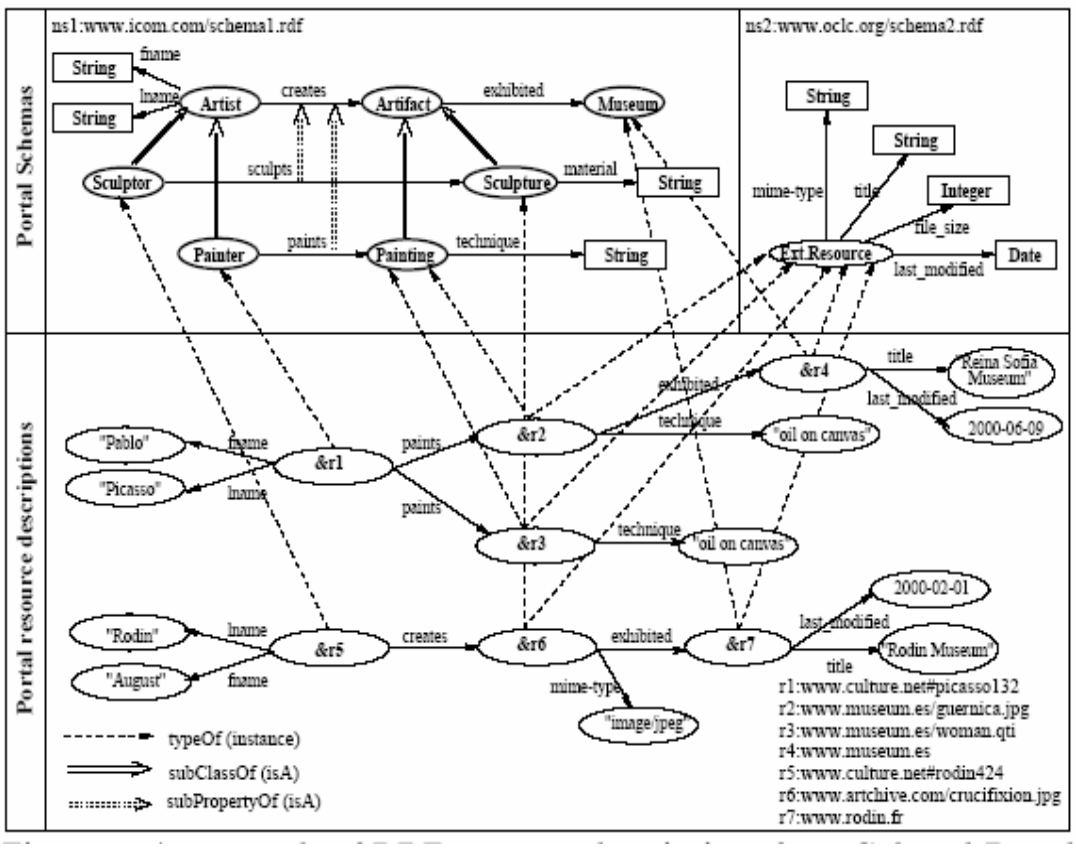
Từ lúc ra đời đến nay, Ontology đã có rất nhiều định nghĩa. Tuy nhiên, đặc điểm cốt lõi của Ontology vẫn là: "Một ontology là một sự chỉ định *tường minh*, *hình thức* và *chia sẻ* về một *khái niệm* dùng chung". Trong đó:

- Một *khái niệm* tham chiếu đến một mô hình trừu tượng của một vài hiện tượng nào đó trong thế giới thực mà xác định những khái niệm có liên quan về hiện tượng đó.
- *Tường minh* là những khái niệm và những ràng buộc trên nó được sử dụng một cách rõ ràng.
- *Hình thức* tham chiếu đến công việc mà ontology phải thực hiện để máy tính có thể hiểu được.
- *Chia sẻ* phản ánh rằng một ontology giữ tri thức đồng nhất, nghĩa là nó không bị hạn chế bởi một cá nhân hay một nhóm riêng lẻ nào.

Hiện nay có nhiều ontology lớn như: CYC, WordNet, ….

Ví dụ về ontology:

KHOA CNTT



Hình 10: Một Ontology đơn giản

2.2.7. Rdf

2.2.7.1 Khái niệm :

RDF là từ viết tắt của **R**esource **D**escription **F**ramework. RDF được đề cử bởi W3C cho một mô hình và ngôn ngữ siêu dữ liệu (metadata) chuẩn. RDF là một bộ khung cho việc mô tả các tài nguyên trên web.

RDF cung cấp mô hình dữ liệu và cú pháp để các phần độc lập nhau có thể chuyển đổi cho nhau và sử dụng được RDF.

2.2.7.2 Cấu trúc :

RDF là khung sườn (framework) cho việc xử lý metadata, và nó mô tả các mối quan hệ giữa các tài nguyên thông qua các thuộc tính và các giá trị. RDF được xây dựng dựa trên các luật như sau:

KHOA

Resource: Mọi thứ được mô tả bằng biểu thức RDF được gọi là một resource (tài nguyên). Mỗi tài nguyên có một URI và nó có thể là toàn bộ trang web hoặc là một phần của trang web.

Property: "Property là một khía cạnh, đặc trưng, thuộc tính hoặc quan hệ riêng biệt được dùng để mô tả một tài nguyên" – trích trong *W3C, Resource Description Framework (RDF) Model and Syntax Specification*. Chú ý là một **property** cũng có thể là một **resource** bởi vì nó có những tính chất riêng của nó.

Statements: Một statements được dùng để kết hợp một *resource*, một *property* và một *value* của nó. Ba phần riêng biệt này được biết như là "**subject**", "**predicate**", và "**object**". Ví dụ, "The Author of http://www.cs.bris.ac.uk/home/pw2538/index.html is Peng Wang" là một statement. Chú ý rằng *value* của câu này có thể là một chuỗi ký tự mà cũng có thể là một resource.

Ví dụ về RDF:

Một statement (phát biểu) có thể được xem như là một đồ thị trong RDF.

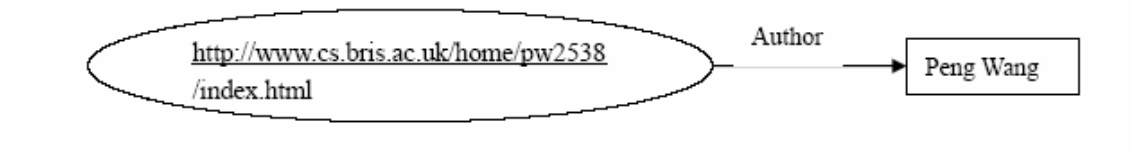
Phát biểu như sau:

"The Author of http://www.cs.bris.ac.uk/home/pw2538/index.html is Peng Wang"

Câu trên được phân tích thành 3 phần:

Subject (Resource)	http://www.cs.bris.ac.uk/home/pw2538/index.html
Predicate (Property)	Author
Object (Literal)	Peng Wang

Được biểu diễn dưới dạng đồ thị như sau:



KHOA CNTT

Chiều của mũi tên luôn hướng từ **subject** đến **object** của phát biểu (statement). Và đồ thị có thể đọc theo cách sau: "<subject> **HAS** <predicate> <object>", ví dụ: "http://www.cs.bris.ac.uk/home/pw2538/index.html has author Peng Wang".

Nếu chúng ta gán một URI cho thuộc tính *author,* thì sẽ có :

http://www.cs.bris.ac.uk/home/pw2538/terms/author

Để trình bày ngắn gọn, chúng ta đưa ra một số tiền tố (prefix) để tránh phải viết lại toàn bộ địa chỉ URI tham chiếu đến. Có một số tiền tố gắn liền với các URI được sử dụng rộng rãi sau:

Tiền tố **rdf:** là không gian tên cho URI:

http://www.w3.org/1999/02/22-rdf-syntax-ns#

Tiền tố **rdfs:** là không gian tên cho URI:

http://www.w3.org/2000/01/rdf-schema#

Tiền tố **daml:** là không gian tên cho URI:

http://www.daml.org/2001/03/daml+oil#

Tiền tố **xsd:** là không gian tên cho URI:

http://www.w3.org/2001/XMLSchema#

Trong ví dụ này, chúng ta dùng không gian tên là **pwterms** để đại diện cho địa chỉ URI mà ta tham chiếu đến: http://www.cs.bris.ac.uk/home/pw2538/terms

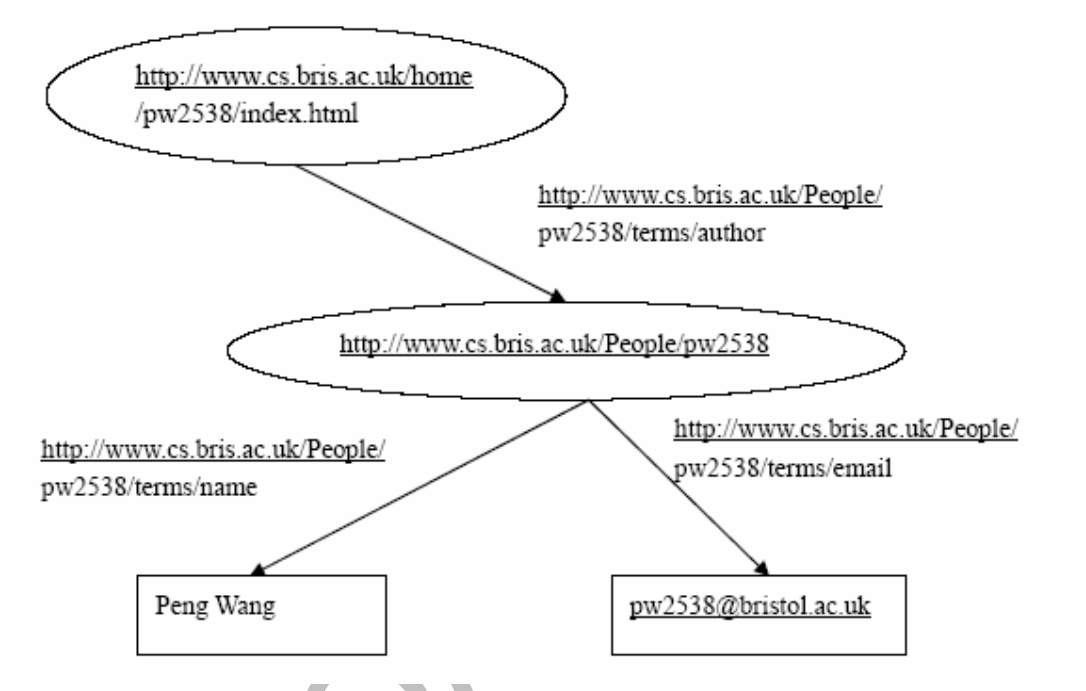
Khi đó cú pháp RDF cho câu phát biểu: "The Author of http://www.cs.bris.ac.uk/home/pw2538/index.html is Peng Wang" là:

```
1  <?xml version="1.0"?>
2  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3           xmlns:pwterms=" http://www.cs.bris.ac.uk/home/pw2538/terms">
4            <rdf:Description
5                    rdf:about="http://www.cs.bris.ac.uk/home/pw2538/index.html">
                     <pwterms:author>Peng Wang</pwterms:author>
6            </rdf:Description>
7    </rdf:RDF>
```

KHOA CNTT

Một câu phát biểu khác: "Một người có mã số sinh viên là pw2538 có tên là Peng Wang và có địa chỉ email là pw2538@bristol.ac.uk . Người này là tác giả của tài nguyên http://www.cs.bris.ac.uk/home/pw2538/index.html"

Có đồ thị như sau:



Có cú pháp RDF:

KHO

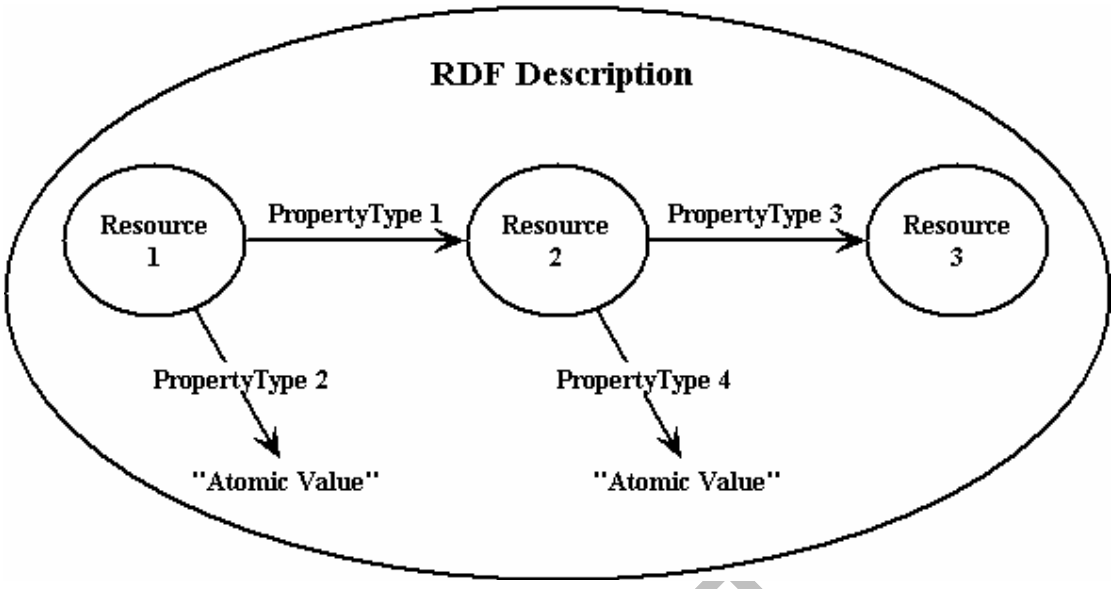
```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.                  xmlns:pwterms=" http://www.cs.bris.ac.uk/home/pw2538/terms">

4.    <rdf:Description about="http://www.cs.bris.ac.uk/home/pw2538/index.html">
5.          < pwterms:author rdf:resource="http://www.cs.bris.ac.uk/People/pw2538"/>
6.    </rdf:Description>

7.    <rdf:Description about="http://www.cs.bris.ac.uk/People/pw2538">
8.          <pwterms:Name>Peng Wang</pwterms:Name>
9.          <pwterms:Email>pw2538@bristol.ac.uk</pwterms:Email>
10.   </rdf:Description>

11. </rdf:RDF>
```

Mô hình dữ liệu RDF (RDF Data Model):

RDF cung cấp một mô hình cho việc mô tả các tài nguyên. Tài nguyên có các tính chất (property) – thuộc tính hoặc là đặc trưng. RDF định nghĩa *tài nguyên* như là một đối tượng bất kỳ có thể nhận biết duy nhất bằng một URI. Các property được kết hợp với các tài nguyên được nhận biết bởi các *property – types*, và các property – types này có các *values* tương ứng. Property – types mô tả mối quan hệ của các *values* được kết hợp với các tài nguyên. Trong RDF, các *values* có thể được xem như là nguyên tử trong tự nhiên (chuỗi text, số, v.v…) hoặc là các loại tài nguyên khác.

Bản chất cốt lõi của RDF là một mô hình độc lập cú pháp cho việc trình bày các tài nguyên và sự mô tả tương ứng của chúng.

KHOA CN



Hình 11: Mô hình dữ liệu RDF

Mô hình dữ liệu RDF là một đồ thị có gán nhãn định hướng, trong đó các nút là các tài nguyên (những thực thể với URI) hoặc những ký tự, và các cạnh là những thuộc tính. Như đã giới thiệu, một phát biểu RDF là một bộ ba (Chủ ngữ, Vị ngữ, Bổ ngữ). Trong đó, tài nguyên là Chủ ngữ của một phát biểu có thuộc tính mà giá trị của nó là Bổ ngữ của một phát biểu. Một Bổ ngữ có thể là tài nguyên hoặc có thể là một giá trị ký tự. Một phát biểu có thể được đại diện như một đồ thị, bằng cách vẽ một cung từ một nút (Chủ ngữ) đến nút khác (Bổ ngữ).

RDF là một cách thành lập cho việc xử lý siêu dữ liệu, nó cung cấp *interoperability (thao tác giữa các phần)* giữa các ứng dụng mà chuyển đổi thông tin máy có thể hiểu được trên web. RDF nhấn mạnh các tiện ích để có thể xử lý tự động các tài nguyên web.

KHOA CNTT

2.2.7.3 RDF Schema – một ngôn ngữ mô tả từ vựng

Ngôn ngữ được định nghĩa trong đặc tả này (specification) gồm một tập hợp các tài nguyên mà có thể được sử dụng để mô tả các thuộc tính của các tài nguyên RDF khác (bao gồm cả các thuộc tính) – định nghĩa tập từ vựng RDF của ứng dụng xác định. Tập từ vựng này chủ yếu được định nghĩa trong một không gian tên được gọi là “rdfs”, và được nhận biết bởi tham chiếu URI: http://www.w3.org/2000/01/rdf-

schema#. Đặc tả này cũng sử dụng tiền tố "rdf" để tham chiếu đến không gian tên RDF chính: http://www.w3.org/1999/02/22-rdf-syntax-ns#.

Hệ thống class và property trong RDF Schema cũng tương tự như các hệ thống kiểu của các ngôn ngữ hướng đối tượng như Java. Tuy nhiên, RDF khác với các hệ thống khác ở chỗ thay vì định nghĩa một class trong quan hệ của các thuộc tính mà thể hiện của nó có thể có, RDF Schema sẽ định nghĩa các thuộc tính trong quan hệ của các lớp của tài nguyên mà chúng ứng dụng. Đây là nhiệm vụ của *rdfs:domain* và *rdfs:range* được mô tả trong đặc tả này. Ví dụ, chúng ta có thể định nghĩa thuộc tính **eg:author**, có miền là **eg:Document** và giới hạn là **eg:Person,** nhưng trái lại một hệ thống hướng đối tượng kinh điển có thể định nghĩa một cách đặc trưng một class **eg:Book** với một thuộc tính được gọi là **eg:author** của kiểu **eg:Person**.

Từ vựng Domain and Range

Đặc tả này giới thiệu tập từ vựng RDF cho việc mô tả cách sử dụng đầy đủ ngữ nghĩa của các property và các class trong dữ liệu RDF. Ví dụ, một lược đồ RDF có thể mô tả giới hạn trên các kiểu của các value thích hợp với một số thuộc tính.

RDF Schema cung cấp cơ chế (kỹ thuật) cho việc mô tả thông tin này, nhưng không thể nói trong trường hợp nào thì ứng dụng nên sử dụng nó và sử dụng như thế nào. Các ứng dụng khác nhau sẽ sử dụng thông tin này theo nhiều cách khác nhau. Ví dụ, các công cụ kiểm tra dữ liệu có thể sử dụng thông tin này để tìm ra các lỗi trong dataset, một trình soạn thảo giao tiếp giữa người và máy có thể đề nghị những giá trị thích hợp, và một ứng dụng suy luận có thể sử dụng nó suy luận rồi đưa ra thông tin mới từ dữ liệu ban đầu.

Lược đồ RDF (RDF Schema) có thể mô tả các mối quan hệ giữa các từ vựng từ nhiều lược đồ được phát triển độc lập nhau. Bởi vì tham chiếu URI được sử dụng để nhận biết các class và property trên web, nên nó có thể tạo ra các thuộc tính (property) mới có *domain* và *range* mà giá trị của nó được định nghĩa trong một namespace khác.

Đặc tả này không cố gắng để liệt kê tất cả các hình thức có thể có của việc mô tả từ vựng mà nó được sử dụng để trình bày ngữ nghĩa của các class và property của

KHOA CNTT

RDF. Thay vào đó, chiến lược mô tả từ vựng RDF thừa nhận rằng có nhiều kỹ thuật mà thông qua đó ngữ nghĩa của các class và property được cho biết, và để xuất bản một số quy ước cho việc sử dụng RDF/XML để mô tả các đặc trưng của các class và property cuả RDF.

Lược đồ tốt hơn hoặc là các ngôn ngữ "ontology" như là DAML+OIL, W3C, các ngôn ngữ suy luận dựa trên luật, và các chủ nghĩa hình thức khác, mỗi loại sẽ góp phần cho khả năng của chúng ta nắm bắt được sự tổng hợp đầy đủ ngữ nghĩa về dữ liệu trên web. Các nhà thiết kế từ vựng RDF có thể tạo và phát triển các ứng dụng web ngữ nghĩa bằng cách sử dụng tiện ích **The basic RDF Schema 1.0**, trong khi trình bày các ngôn ngữ mô tả từ vựng tốt hơn – cách này cũng sử dụng hướng tiếp cận này.

Sơ lược về RDF Schema

Bảng này trình bày một cách tổng quát về tập từ vựng cơ sở của RDF

Tên lớp	Ghi chú
rdfs:Resource	The class resource, everything.
rdfs:Literal	This represents the set of atomic values, eg. textual strings.
rdfs:XMLLiteral	The class of XML literals.
rdfs:Class	The concept of Class
rdf:Property	The concept of a property.
rdfs:Datatype	The class of datatypes.
rdf:Statement	The class of RDF statements.
rdf:Bag	An unordered collection.
rdf:Seq	An ordered collection.
rdf:Alt	A collection of alternatives.

KHOA CNTT

rdfs:Container	This represents the set Containers.
rdfs:ContainerMembershipProperty	The container membership properties, rdf:1, rdf:2, ..., all of which are sub-properties of 'member'.
rdf:List	The class of RDF Lists

Bảng 3 : Các lớp trong RDF

Property name	**comment**	**domain**	**range**
rdf:type	Indicates membership of a class	rdfs:Resource	rdfs:Class
rdfs:subClassOf	Indicates membership of a class	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	Indicates specialization of properties	rdf:Property	rdf:Property
rdfs:domain	A domain class for a property type	rdf:Property	rdfs:Class
rdfs:range	A range class for a property type	rdf:Property	rdfs:Class
rdfs:label	Provides a human-readable version of a resource name.	rdfs:Resource	rdfs:Literal
rdfs:comment	Use this for descriptions	rdfs:Resource	rdfs:Literal
rdfs:member	a member of a container	rdfs:Container	*not specified*
rdf:first	The first item in an RDF list. Also often called the head.	rdf:List	*not specified*
rdf:rest	The rest of an RDF list after the first item. Also often called the tail.	rdf:List	rdf:List

KHOA CNTT

rdfs:seeAlso	A resource that provides information about the subject resource	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	Indicates the namespace of a resource	rdfs:Resource	rdfs:Resource
rdf:value	Identifies the principal value (usually a string) of a property when the property value is a structured resource	rdfs:Resource	*not specified*
rdf:subject	The subject of an RDF statement.	rdf:Statement	rdfs:Resource
rdf:predicate	the predicate of an RDF statement.	rdf:Statement	rdf:Property
rdf:object	The object of an RDF statement.	rdf:Statement	*not specified*

Bảng 4:Các thuộc tính của RDF

(Mô tả các từ vựng của RDF được trình bày trong phần Phụ lục [1].)

KHOA CNTT

2.3. eDoc

2.3.1. Tìm hiểu eLearning

2.3.1.1. Khái niệm

eLearning hay còn gọi là Online Learning, chuẩn cho tất cả các hình thức của việc học.

Online learning liên quan đến việc sử dụng các công nghệ mạng (như là: Internet hay là mạng thương mại – bussiness network) cho việc phân phát, hỗ trợ, đánh giá việc dạy học chính qui và không chính qui.

"Học" xảy ra ở đâu và như thế nào? Ở: các tài nguyên và các tài liệu trực tuyến, các thư viện điện tử, các tài liệu; và các khoá học, các buổi thảo luận, chats, email, hội nghị, và các ứng dụng chia sẻ tri thức. Một chú ý quan trọng là online learning không nhất thiết phải diễn ra trực tuyến (online). Sử dụng công nghệ cho việc học thường là một yếu tố phụ đối với lớp học và các cơ hội học trực tiếp (face – to – face).

Một số nguyên nhân để sử dụng online learning:

a. Việc truy cập được cải thiện và tính linh động: Mọi người có thể đăng nhập vào bất kỳ một máy tính nào, ở tại nhà hoặc ở nơi làm việc, vào bất kỳ lúc nào kể cả ngày lẫn đêm, để lấy bài học hoặc tham khảo đến các tài liệu học.
b. Phân phối nhanh hơn và tiết kiệm chi phí: Đối với các tổ chức cần truyền đạt thông tin quan trọng mà thông tin này nhanh chóng trở nên lỗi thời (ví dụ, phiên bản mới nhất của một sản phẩm), thì hình thức online hầu như là rẻ hơn và nhanh hơn nhiều so với việc người truyền đạt phải bay qua nhiều quốc gia để gặp gỡ những học viên ở lớp học với hàng tiếng đồng hồ.
c. Cải tiến việc điều hành và chuẩn hoá: Trong môi trường thương mại quốc tế ngày nay, nhiều tổ chức mở rộng trên phạm vi toàn cầu. Sự khác nhau về kiến thức và kỹ năng của các cá nhân dạy có thể sẽ làm cho chất lượng học của các học viên ở những nơi khác nhau sẽ khác nhau: ví dụ những người học ở New Delphi sẽ có chất lượng huấn luyện khác với những người ở New York. Online learning cung cấp thông tin nhất quán, phổ biến đối với các đối tượng ở khắp nơi.

Làm nổi bật thông tin truyền đạt và sự cộng tác: Thông qua những phần mềm nào đó sẽ cho phép những người học được giao tiếp với nhau, cộng tác với nhau qua các dự án, và chia sẻ tài liệu mà không cần phải gặp mặt trực tiếp.

KHOA CNTT

2.3.1.2. Các chuẩn của eLearning

Ngành công nghiệp eLearning tiếp tục được mở rộng mỗi ngày, và các chuẩn cần thiết để tạo nội dung bài học ngày càng trở nên phức tạp.

Trước khi một "qui ước" của eLearning trở thành "standards" (chuẩn), nó được gọi là "specification" (đặc tả). Specification được duyệt bởi một tổ chức – tổ chức này được mọi người công nhận, như là IEEE chẳng hạn.

Một số chuẩn của eLearning:

a. Tập phần tử siêu dữ liệu Dublin Core

Tập phần tử siêu dữ liệu Dublin Core (The Dublin Core metada element set) là chuẩn cho sự mô tả tài nguyên thông tin xuyên domain (băng qua nhiều domain). Ở đây, tài nguyên thông tin được định nghĩa là bất kỳ thứ gì mà có thể nhận biết được. Đối với các ứng dụng Dublin Core, một tài nguyên sẽ là một tài liệu điện tử (electronic document).

Siêu dữ liệu Dublin Core được dùng cho việc tìm kiếm và chỉ mục cho các siêu dữ liệu dựa trên Web. Tập siêu dữ liệu này cung cấp từ vựng ngữ nghĩa như: "Description", "Creator" và "Date" cho việc mô tả những đặc trưng thông tin quan trọng của các tài nguyên Internet.

Tập siêu dữ liệu Dublin Core cung cấp 15 từ vựng:

- Title: Tên được gán cho tài nguyên.
- Creator: Thực thể có trách nhiệm tạo ra tài nguyên. Ví dụ như: cá nhân, tổ chức hay một dịch vụ nào đó.
- Subject: Chủ đề nội dung của tài nguyên.
- Description: Mô tả nội dung của tài nguyên.
- Publisher: Thực thể có nhiệm vụ tạo ra tài nguyên.
- Contributor: Thực thể có đóng góp vào nội dung của tài nguyên.
- Date: Ngày tài nguyên được tạo.
- Type: Thể loại nội dung của tài nguyên.
- Format: Dạng lưu trữ vật lý của tài nguyên.

KHOA CNTT

- Identifier: Một tham chiếu cụ thể đến tài nguyên trong một ngữ cảnh cho phép.
- Source: Tham chiếu đến một tài nguyên mà tài nguyên được dẫn xuất.
- Language: Ngôn ngữ sử dụng bởi nội dung của tài nguyên.
- Relation: Tham chiếu đến một tài nguyên liên quan
- Coverage: Mở rộng nội dung của tài nguyên
- Right: Thông tin về quyền sở hữu tài nguyên.

b. LOM (Learning Object Metadata)

LOM là một chuẩn về eLearning hiện tại được phát triển bởi tổ chức IEEE. Tổ chức chuẩn hoá công nghệ học (**L**earning **T**echnology **S**tandards **C**ommittee) của IEEE đã phát triển chuẩn LOM nhằm giúp cho việc sử dụng và sử dụng lại của các tài nguyên học được hỗ trợ công nghệ như là việc huấn luyện dựa trên máy tính, và việc học từ xa.

Trong một hệ thống eLearning, đối tượng học là những gì có thể được sử dụng, kế thừa hay tham khảo trong việc hỗ trợ công nghệ học. Hiện tại một số đối tượng đang được tiếp tục phát triển nhằm đáp ứng nhu cầu học thay đổi nhanh chóng. Việc thiếu thông tin hay siêu dữ liệu về đối tượng học tạo ra nhiều cản trở, hạn chế cho khả năng quản lý, khám phá và sử dụng đối tượng học.

LOM giải quyết vấn đề trên bằng cách định nghĩa một cấu trúc cho việc mô tả một đối tượng học. LOM chỉ ra cú pháp và ngữ nghĩa của các siêu dữ liệu đối tượng học, định nghĩa các thuộc tính nhằm mô tả đầy đủ và thoả đáng các đối tượng học.

Mục đích của LOM:

- Cho phép người học hay người hướng dẫn tìm kiếm, đánh giá đối tượng học.

KHOA CNTT

- Cho phép chia sẻ và trao đổi các đối tượng học qua bất kỳ công nghệ có hỗ trợ hệ thống học.
- Cho phép phát triển các đối tượng học theo các đơn vị có khả năng kết hợp hay phân rã theo một phương pháp phù hợp.
- Cho phép các agent máy tính linh động là tự động trong việc tổ chức các bài học cung cấp đến người học.
- Nó hoàn toàn dựa trên chuẩn và quan tâm đến các đối tượng học trong môi trường mở và phân tán.
- Cho phép các công nghệ mới kết hợp với các đối tượng học.
- Cung cấp cho các nhà nghiên cứu chuẩn hỗ trợ và sưu tập dữ liệu liên quan đến hiệu quả của các đối tượng học.

LOM định nghĩa một tập tối thiểu các thuộc tính (attributes) để quản lý, định vị, và đánh giá các đối tượng học. Các thuộc tính được gom nhóm thành 8 phạm trù:

- General: chứa đựng thông tin về toàn bộ đối tượng.
- Lifecycle: chứa đựng siêu dữ liệu về sự tiến hoá của các đối tượng.
- Technical: với sự mô tả của các đặc trưng và yêu cầu kỹ thuật.
- Educational: chứa đựng các thuộc tính về giáo dục hoặc sư phạm.
- Rights: mô tả quyền sở hữu và các điều kiện sử dụng
- Relation: nhận biết các đối tượng có liên quan với nhau.
- Annotation: chứa đựng các chú thích và ngày, tác giả của các chú thích này.
- Classification: nhận biết các bộ nhận diện hệ thống phân loại khác cho đối tượng.

Bên trong mỗi phạm trù là một tập các phần tử dữ liệu có thứ tự, mà giá trị của chúng là các metadata. Ví dụ: Các phần tử siêu dữ liệu liên quan đến việc học được tìm thấy trong phạm trù *Education* là Typical Age Range, Difficulty, Typical Learning Time, và Interactivity Level.

KHOA CNTT

c. vCard

vCard là chuẩn được giới thiệu và phát triển bởi IMC (Internet Mail Consortium). Các thông tin cá nhân thông thường rất phức tạp và có nhiều loại khác nhau. Hiện tại có một số chuẩn đề xuất các cấu trúc cho việc trao đổi thông tin cá nhân PDI (Personal Data Interchange). Mục đích của chuẩn này là nhằm giải quyết nhu cầu sưu tập và trao đổi thông tin cá nhân qua nhiều kênh thông tin khác nhau như điện thoại, thư điện tử hay đối thoại trực tiếp.

Chuẩn vCard phù hợp cho việc trao đổi dữ liệu cá nhân giữa các ứng dụng và hệ thống. Định dạng của vCard hoàn toàn độc lập với phương pháp dùng để truyền tải nó. Việc truyền tải này có thể là trao đổi một hệ thống tập tin, mạng chuyển mạch công cộng, mạng dây dẫn hay mạng không dây. vCard nhắm đến việc trao đổi thông tin cá nhân. Trong môi trường thương mại ngày nay, thông tin này thường được trao đổi trên các thẻ thương mại và vCard định nghĩa những thông tin này dựa trên các đối tượng thẻ thương mại điện tử.

d. SCORM (Shareable Content Object Reference Model)

SCORM định nghĩa mô hình kết hợp giữa nội dung và môi trường thực thi cho các đối tượng học. Đây là một mô hình tham chiếu đến một tập các kỹ thuật liên quan việc thiết kế nhằm đáp ứng yêu cầu nội dung học dựa trên Web, những yêu cầu này bao gồm khả năng tái sử dụng, truy xuất, khả năng tương tác của các đối tượng học.

e. IMS (Instructional Management Systems)

IMS đang được phát triển và xúc tiến trở thành chuẩn mở cho các hoạt động eLearning như sử dụng, sắp xếp các nội dung giáo dục và mở rộng các khái niệm tổng quát như: thiết kế người học, theo dõi và báo cáo quá trình người học nhằm thực hiện việc trao đổi thông tin giữa các hệ thống học khác nhau.

KHOA CNTT

Mục đích của IMS:

- Định nghĩa các chuẩn kỹ thuật nhằm nâng cao khả năng tương tác giữa ứng dụng và dịch vụ trong môi trường học phân tán hiện nay.
- Hỗ trợ việc sát nhập đặc tả của IMS vào trong các sản phẩm và dịch vụ trên toàn thế giới. Sự chấp nhận đặc tả rộng rãi sẽ cho phép phân phối môi trường và nội dung học từ nhiều tác giả lại với nhau.

2.3.2. Tìm hiểu eLib

Elib (electronic library hay cò gọi là digital library) là một thư viện ẩn. Từ 'electronic library' ngụ ý là một sưu tập của các tài nguyên thông tin điện tử được nối mạng cùng kỹ thuật liên kết và cơ sở hạ tầng quản trị. Bạn có thể truy cập nó từ bất cứ máy PC hay laptop có nối mạng nào từ bất cứ nơi nào trên thế giới ở bất cứ thời điểm nào.

Elib lưu trữ và chỉ mục hàng vạn sách, báo, tạp chí về đủ các chủ đề trên thế giới, chẳng hạn như vật lí, thiên văn, sinh hoá, công nghệ sinh học, hoá học và công trình xây dựng hoá chất, các thiết bị xây dựng, công trình xây dựng môi trường, khoa học thực phẩm, và an toàn sức khoẻ và vệ sinh .v.v… cũng như các tài liệu về thông tin tiểu sử, lí lịch cá nhân, nghề nghiệp, các tổ chức, hội liên hiệp, và du lịch v.v…. Thư viện điện tử này được sử dụng phổ biến nhất trong các trường đại học và những trung tâm nghiên cứu khoa học. Tất nhiên, đối tượng sử dụng nó chính là những sinh viên, nghiên cứu sinh và các nhà khoa học.

Những chương trình Electronic library được xây dựng dựa trên những chuẩn thống nhất do các hội đồng, tổ chức lớn trên thế giới lập ra. Một số tổ chức định chuẩn lớn trên giới như **W3C** (World Wide Web Consortium), **ISO** (International Organization for Standardization), **NISO** (National Information Standards Organization),… . Có nhiều chuẩn cho nhiều khía cạnh khác nhau của việc lưu trữ và truy cập thông tin điện tử, bao gồm các chuẩn về thu hồi thông tin (Information Retrieval Standard), thao tác giữa các phần (Interoperability), định dạng tài nguyên,

KHOA CNTT

nhận dạng tài nguyên, mô tả tài nguyên,… Sau đây là một số chuẩn sử dụng trong eLib liên quan đến vấn đề truy cập thông tin điện tử:

- **Chuẩn về thu hồi thông tin:**

Kiểu chuẩn này cho phép thông tin giữa các hệ thống khác nhau, làm cho thuận tiện trong việc khám phá và truy cập thông tin điện tử. Ví dụ như chuẩn thu hồi thông tin ISO 23950 (tương đương với ANSI Z39.50) định nghĩa một hướng chuẩn cho hai máy tính liên lạc và chia sẻ thông tin với nhau. Nó đã được thiết kế để hỗ trợ khám phá tài nguyên và thu hồi tài nguyên của những tài liệu "full-text", dữ liệu mục lục, các hình ảnh và multimedia. Chuẩn này dựa trên kiến trúc client-server và độc lập với các hệ thống cụ thể, hoàn toàn điều hành trên Internet.

Z39.50:

Z39.50 là một trong một nhóm các chuẩn được sản xuất để làm cho dễ dàng kết nối các hệ thống máy tính. Chuẩn này chỉ ra các định dạng và thủ tục chi phối việc trao đổi các thông điệp giữa client và server, cho phép người dùng có thể tìm kiếm các cơ sở dữ liệu từ xa, nhận diện các dòng dữ liệu có định rõ các chuẩn, và thu hồi một vài hay tất cả các dòng được nhận diện và có liên quan, cụ thể với việc tìm kiếm và thu hồi thông tin trong cơ sở dữ liệu. Một trong những thuận lợi lớn trong việc sử dụng Z39.50 là nó cho phép truy cập như nhau đến một số lượng lớn nguồn thông tin thay đổi khác nhau.

Z39.50 thừa nhận rằng việc thu hồi thông tin gồm hai thành phần chính – chọn thông tin dựa trên những tiêu chuẩn và thu hồi thông tin đó, và nó cung cấp một ngôn ngữ chung cho cả hai hành động đó. Z39.50 chuẩn hoá cách xử sự mà trong đó client và server thông tin với nhau và hoạt động ngay khi có những khác biệt giữa các hệ thống máy tính, các công cụ tìm kiếm và các cơ sở dữ liệu.

EDI (Electronic Data Interchange)

EDI được biết đến như một chuẩn công nghệ thông tin quốc gia. Ở EDI, dữ liệu mà theo truyền thống được chuyển vào trong các tài liệu giấy thì được truyền hay được thông tin một cách điện tử tùy vào các luật và các định dạng được thiết lập. Dữ

KHOA CNTT

liệu liên đới với mỗi kiểu của tài liệu chức năng, ví dụ như bảng mua bán hay hoá đơn, được vận chuyển lẫn nhau như là một thông điệp điện tử. Dữ liệu đã định dạng có thể được vận chuyển từ người tạo ra đến người nhận thông qua thông tin liên lạc bằng cáp hay vận chuyển vật lí vào trong thiết bị lưu trữ điện tử.

EDI đưa đến một chuỗi các thông điệp giữa hai nơi, ví dụ người mua và người bán, mỗi người có thể xem như là người tạo ra hay người nhận. Các thông điệp từ người mua đến người bán sẽ bao gồm, ví dụ như dữ liệu cần thiết cho yêu cầu đối với sự trích dẫn (request for quotation_ RFQ), các biên lai mua bán, các thông báo việc vận chuyển tàu thuyền, và các hoá đơn. Việc thực thi của EDI yêu cầu viện sử dụng của một họ các chuẩn liên kết với nhau. Họ chuẩn này phải bao gồm các chuẩn cho các kiểu thông điệp (cũng được gọi là các "nhóm giao dịch" _ "transaction set"), và cho việc vận chuyển thư, các yếu tố dữ liệu, và các chuỗi của các yếu tố dữ liệu được sắp xếp gọi là các segment dữ liệu. Một chuẩn thông điệp hay chuẩn transaction set định nghĩa chuỗi các segment dữ liệu mà tạo thành thông điệp và transaction set đó. Thư mục segment dữ liệu liệt kê tất cả các segment dữ liệu, và định nghĩa định danh và chuỗi của các yếu tố dữ liệu tạo nên nó. Tự điển yếu tố dữ liệu cung cấp các chuẩn của tất cả các yếu tố dữ liệu. Việc vận chuyển thư cung cấp thông tin điều khiển về các thông điệp thêm vào cho các hệ thống vận chuyển và tiếp nhận. Việc chuẩn hoá của các định dạng thông điệp, và của các segment dữ liệu và yếu tố dữ liệu trong các thông điệp đó, làm cho có thể thu thập, tháo rời và xử lí các thông điệp bằng máy tính với các kết quả có thể có thể đoán trước.

KHOA CNTT

ILL (Internet Loan Library)

Nghi thức ILL (ISO 10160/1) được phát triển để giữ nhiều giao dịch được liên kết bao gồm các hoạt động yêu cầu tài liệu gồm nhiều người tham gia. Về khái niệm thì nó tương đương với EDI và bao gồm việc cung cấp cho định nghĩa các data element được yêu cầu, định nghĩa một nhóm các thông điệp và các mối quan hệ của nó, và một cú pháp cho việc lập cấu trúc thông điệp.

Nghi thức ILL có vẻ như có nhiều để cung cấp các dịch vụ yêu cầu, đặc biệt khi chúng trở nên phân tán nhiều hơn. Sự truyền thông từ hệ thống này sang hệ thống khác của các thông điệp có cấu trúc cho phép một phạm vi rộng lớn các thi hành được tự động, và các thủ tục bằng tay hay phối hợp cho việc theo vết, gọi về,… được tự động. Công dụng của nó trong các dịch vụ tương tác đối với yêu cầu các tài liệu cần nghiên cứu xa hơn nữa.

- **Chuẩn mã hoá tài nguyên:**

Những chuẩn này định nghĩa các kiểu hiển thị khác nhau của thông tin điện tử. Bao gồm các chuẩn:

- Định dạng mô tả trang (ví dụ postscript, PDF)
- Định dạng đồ họa (ví dụ TIFF, GIF, JPEG)
- Thông tin cấu trúc (SGML, HTML, XML)
- Định dạng hình ảnh động và audio.
- Nén (ví dụ: gzip, jar, tar, zip).

- **Chuẩn nhận dạng tài nguyên:**

Gồm một số chuẩn sau:

- **DOI (Digital Object Identifier)**

Digital Object Identifier là một hệ thống được phát triển bởi Bowker và CNRI (Corporation for National Research Initiative) ở US, theo một yêu cầu về các đề xuất cho công nghệ nhận dạng nội dung kỹ thuật số được đưa ra bởi Association of American Publishers. Hệ thống DOI có ba thành phần: phần định danh, thư mục và cơ sở dữ liệu. Hệ thống này cho phép các bộ định dạng qui định những mức khác nhau, và cho các hệ thống khác (ví dụ SICI, ISSN) được thêm vào.

Hệ thống DOI có thể được định nghĩa như là "một bộ nhận dạng duy nhất có thể giải quyết được và nhiều mảng của dữ liệu trạng thái kiểu kết hợp trong một cơ sở quản lí thông tin". Diễn tả những phần của định nghĩa như sau:

KHOA CNTT

a. Một "bộ nhận dạng duy nhất": nhiệm vụ của DOI là duy nhất đối với một mảng của đặc tính tri thức. Định nghĩa của mảng này được chỉ rõ bởi một số mảng chính của thông tin về nó (siêu dữ liệu) mà thuộc vào thể loại cụ thể: dù thực thể là một bài báo hay một video clip, ví dụ như vậy. Định danh này là một chuỗi không rõ ràng; nó không chứa bất cứ tri thức cú pháp về thực thể này.
b. "có thể giải quyết được"; với "dữ liệu trạng thái kết hợp": đi sâu vào thông qua hệ thống Internet từ bộ nhận dạng đó đến một hay nhiều mảng của dữ liệu kết hợp. Những mảng này biểu diễu trạng thái hiện tại (giá trị) của một số kiểu dữ liệu (ví dụ như một URL). Những mảng này của dữ liệu có thể hiển thị, hay dẫn đến, các dịch vụ sử dụng DOI như là một điểm thực thể.
c. "một cơ sở quản lí thông tin": một khi một mảng dữ liệu thu được do sự phân tích, thì siêu dữ liệu về thực thể được định danh có thể thi hành với siêu dữ liệu từ những nguồn khác (ví dụ về ngữ cảnh) để xây dựng các dịch vụ và các giao dịch tự động. Khả năng thi hành này được hoàn tất thông qua việc quản lí siêu dữ liệu trong một hướng được điều khiển, phù hợp với một kiến trúc thi hành mà làm cho DOI có thể đưa ra những ứng dụng ở một bộ nhận dạng liên tục đơn giản.

- **SICI**

Chuẩn SICI là chuẩn ANSI/NISO Z39.56-1996 định nghĩa những luật lệ về mã dùng nhận dạng duy nhất chuỗi các item (ví dụ như các số báo) và mỗi thành phần (ví dụ như bài báo) chứa trong một chuỗi. SICI là từ viết tắt của Serial Item and Contribution Identifier và được sử dụng trong chuẩn này để chỉ mã của chính nó.

Chuẩn này được định nghĩa cho việc sử dụng với chuỗi các xuất bản trong tất cả các định dạng. Đối với mục đích của chuẩn này, một chuỗi được định nghĩa như là một xuất bản phát hành trong những phần liên tục ở những khoảng trống đều đặn hay

KHOA CNTT

không đều đặn, mang bậc số và/hoặc thứ tự thời gian (numerical and/or chronological designation), và có xu hướng được tiếp tục vô hạn.

SICI có xu hướng được tạo ra và sử dụng bởi các thành viên của cộng đồng thư mục tham gia vào những chức năng kết hợp với việc quản lí của các chuỗi và các phần mà chúng chứa đựng, các chức năng như sắp thứ tự, bổ sung vào thư viện, yêu cầu, thu tiền nhuận bút, quản lí quyền, thu hồi trực tuyến, liên kết cơ sở dữ liệu, và phân phát tài liệu.

Bộ nhận dạng được xây dựng theo chuẩn này được sử dụng trong những ứng dụng: Electronic Data Interchange (EDI), mã vạch Serial Industry Systems Advisory Committee (SISAC), truy vấn Z39.50, Uniform Resource Names (URNs), thư điện tử, và bản ghi của con người trong in ấn. Chuẩn này không định nghĩa bất cứ hệ thống vận chuyển nào rõ rệt hay ý nghĩa của việc thực thi.

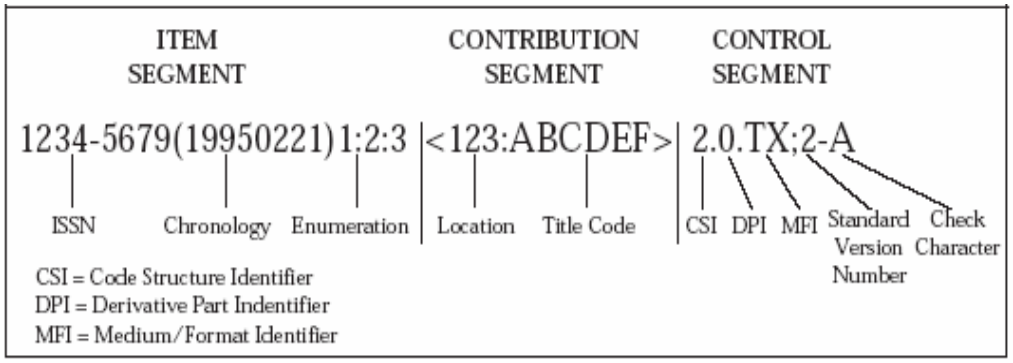
SICI sử dụng chuỗi số chuẩn quốc tế (International Standard Serial Number _ ISSN) để định nhận diện chuỗi tiêu đề. Do đó, để sử dụng chuẩn này trong việc xây dựng một item hay góp phần nhận diện vật chất được phát sinh trong chuỗi này, thì chuỗi này phải được gán vào trong một ISSN.

Chuẩn SICI là một sự kết hợp của các segment được định nghĩa, tất các chúng đều được yêu cầu. Những segment này là:

a. Item Segment, các data element cần mô tả chuỗi item (ISSN, bảng niên đại, bảng liệt kê)
b. Contribution Segment, các data element cần nhận diện các phần trong một item (vị trí, mã tiêu đề, và những sắp xếp thứ tự theo số trong một trường hợp cụ thể của SICI).
c. Control Segment, các data element cần ghi lại những element quản trị đó mà định nghĩa sự đánh giá, phiên bản, và định dạng của biểu diễn mã. Đây là segment quan trọng nhất của SICI. Sự phiên dịch và xử lí được định nghĩa bởi segment điều khiển này.

Ví dụ:

KHOA CNTT



- **Chuẩn mô tả tài nguyên:**

  Chuẩn này có thể làm cho dễ dàng khám phá tài nguyên hiệu quả. Bao gồm:

  - **AACR2**_ một tập các mã được sử dụng cho việc mô tả các tài liệu thư viện
  - **Dublin Core**_ một chuẩn siêu dữ liệu mô tả được phát triển cho việc mô tả tài nguyên trên Internet. (Được mô tả bên trên).
  - **MARC** (Machine-Readable Cataloguing)_ một chuẩn siêu dữ liệu mô tả phát triển cho mục đích mục lục.

Chuẩn MARC đang được giám sát bởi hội đồng thông tin thư mục có thể đọc bằng máy (Machine-Readable Bibliographic Information Committee) kết hợp với văn phòng phát triển mạng và các chuẩn MARC ở thư viện của cơ quan lập pháp Hoa Kì.

Các định dạng MARC là các chuẩn cho việc biểu diễn và truyền thông của thông tin thư mục và quan hệ trong việc thi hành có thể đọc bằng máy "Dòng MARC chứa một chỉ dẫn đến dữ liệu của nó, hay một ít các "biển chỉ đường"("signposts"), trước mỗi mảng thư mục của thông tin. Có ba loại nội dung MARC chỉ rõ: các thẻ, các bộ mã lãnh vực con, và các chỉ thị.

Thuận lợi trong việc sử dụng siêu dữ liệu MARC là chúng không phải phát triển phương pháp chỉ rõ lĩnh vực của việc tổ chức thông tin thư mục, thông tin này lưu công việc và cho phép dữ liệu danh mục có thể cộng tác và trao đổi với các thư viện khác. "Sử dụng chuẩn MARC ngăn chặn việc lặp lại công việc và cho phép các thư viện chia sẻ tốt hơn các tài nguyên thư mục". MARC là một chuẩn công nghiệp

KHOA CNTT

diện rộng mà mục đích chính của nó là đưa việc truyền đạt của thông tin trong một hướng chuẩn, bằng cách đó làm cho dễ dàng truy cập thường xuyên đến các dòng dữ liệu.

- **EDA** (Encoded Archival Description)_ được sử dụng bởi các chuyên viên lưu trữ văn thư cho việc mã hoá những giúp đỡ tìm kiếm.

EAD là một chuẩn được sử dụng để mã hoà những giúp đỡ trong việc tìm kiếm sử dụng SGML và/hay XML. Mục đích của việc sử dụng EAD là thực hiện lưu trữ tài nguyên từ nhiều cơ sở có khả năng truy cập nhiều hơn đến người dùng. EAD cũng khuyến khích cộng đồng lưu trữ văn thư tán thành các chuẩn cấu trúc dữ liệu và làm việc với nhau trong sự hình thành của các hội đồng và các cơ sở dữ liệu thống nhất. Hiện tại, thư viện của văn phòng chuẩn MARC và phát triển mạng của cơ quan lập pháp Hoa Kỳ hoạt động như là cơ quan bảo dưỡng cho EAD và cung cấp tài liệu chính thức cho trang web của nó. Cộng đồng chuyên viên lưu trữ văn thư của Mỹ hoạt động như người chủ của EAD, và bàn tròn SAA EAD có trách nhiệm tiếp tục giám sát và phát triển.

Giúp đỡ tìm kiếm là gì? Những giúp đỡ tìm kiếm là những hướng dẫn chi tiết, nó mô tả và sáng tác những sưu tập của các tài liệu giấy cá nhân chưa xuất bản, các hồ sơ tổ chức, và hình ảnh. Chúng giúp người nghiên cứu nhận dạng và định vị các hộp hay các thư mục quan tâm được yêu cầu cho công việc nghiên cứu. Chúng cũng cung cấp thông tin cơ bản về tổ chức, người, hay gia đình đã tạo ra các tài liệu hay hình ảnh, một tổng quan của những sưu tập và việc sắp xếp của chúng, và một danh sách lưu trữ chi tiết. Giúp đỡ tìm kiếm là những công cụ của việc mô tả lưu trữ.

2.3.3. Tìm hiểu eDoc

2.3.3.1. Khái niệm

Edoc là từ viết tắt của “electronic document” hay còn gọi là digital document. Đây là một khái niệm mang tính tổng quát, chỉ tất cả những tài liệu trên web, chẳng hạn như các trang tin tức, tạp chí điện tử, các tài liệu chuyên ngành hay

KHOA CNTT

các sách điện tử. Edoc được xem là nguồn tài nguyên chính cho các đề án eLib, eLearning. Những đề án này tập hợp, tổ chức lại một cách logic các eDoc xoay quanh một chủ đề cụ thể nào đó nhằm mục đích giúp cho người dùng có thể dễ dàng tìm thấy các tài liệu điện tử trong hàng vạn tài liệu, phục vụ cho nhu cầu nghiên cứu của người dùng.

2.3.3.2. Phạm vi sử dụng của eDoc

eDoc được sử dụng/ áp dụng trong tất cả các hoạt động, nơi nào có phần mềm và các thiết bị công nghệ được ứng dụng để tạo, lưu trữ, chuyển đổi và nhận thông tin thì ở đó cần có eDoc.

2.3.3.3. Các yêu cầu đối với eDocs

- eDoc được tạo, sử dụng, chuyển đổi và lưu trữ với sự hỗ trợ của các thiết bị công nghệ và sự hỗ trợ của các phần mềm.
- eDoc phải được biểu diễn trong hình thức đầy đủ nghĩa nhất
- eDoc phải có cấu trúc phù hợp, phổ dụng được nhiều người sử dụng, và có các thuộc tính cho phép xác nhận tính xác thực của nó.

2.3.3.4. Cấu trúc của eDoc

- Electronic document bao gồm 2 phần không thể tách rời được : general part và especial part.
- **General part** bao gồm thông tin thể hiện nội dung của tài liệu. Nếu một tài liệu được gởi đến một người xác định, thông tin về người này được thể hiện trong phần general part.
- **Especial part** gồm một hoặc nhiều chữ ký điện tử.

2.3.3.5. Bảo mật trong eDoc

Khi một tổ chức muốn thực hiện các giao dịch thương mại trực tuyến, việc bảo đảm an toàn và bí mật của thông tin được sử dụng trong suốt các giao dịch, cũng như việc cung cấp xác thật và toàn vẹn thông tin là rất cần thiết. Bởi vì nhiều

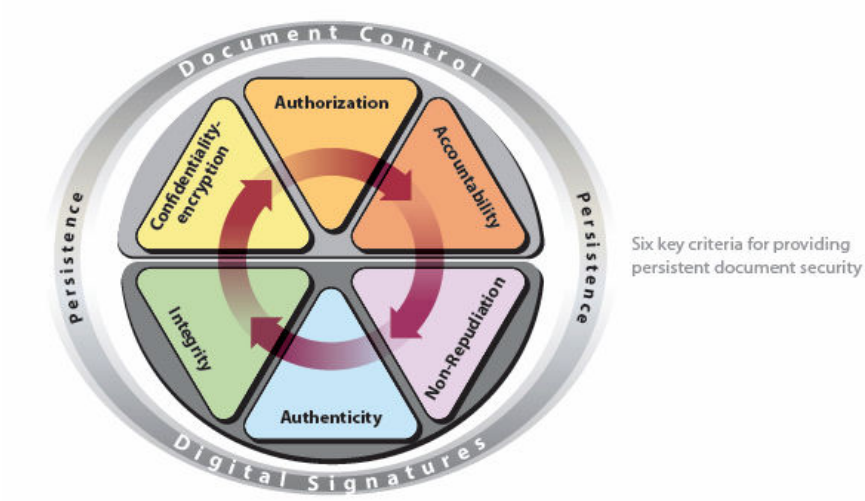
KHOA CNTT

giao dịch tự động dựa trên tài liệu điện tử, loại tài liệu này chứa đựng thông tin rất nhạy cảm, các tổ chức phải bảo đảm hoàn toàn những tài liệu này. Nhiều giải pháp bảo mật thông tin cố gắng bảo vệ các tài liệu điện tử chỉ bảo đảm được ở mức lưu trữ cục bộ hoặc trong suốt quá trình chuyên đổi. Tuy nhiên các giải pháp bảo mật này không cung cấp chế độ bảo vệ cho toàn bộ chu trình sống của một tài liệu điện tử. Khi một tài liệu được chuyên đến cho người nhận thì chế độ bảo vệ cho nó cũng mất đi, và tài liệu này có thể được chuyên đến hoặc được xem một cách cố ý hay vô tình bởi người nhận mà không thể nào chứng thực được ngươi này có được quyền chuyển tiếp hoặc xem hay không?

Một giải pháp hiệu quả hơn nhiều là bảo vệ tài liệu bằng cách gán các thông số bảo mật mà được gởi kèm với nó. Sáu tiêu chuẩn cần phải có để cung cấp chế độ bảo vệ hiệu quả hơn cho một tài liệu điện tử trong suốt chu trình sống của nó:

1. Confidentiality
2. Authorization
3. Accountability
4. Integrity
5. Authenticity
6. Non-repudiation

KHOA CNTT



Hình 12 : Tiêu chuẩn đánh giá tính bảo mật của eDoc

2.3.3.6. Đánh giá

Cho đến hiện tại eDoc vẫn chưa thật sự có một chuẩn nào . Tài liệu eDoc trên Internet vô cùng phong phú, đa dạng, chứa đựng một lượng thông tin khổng lồ trên web. Tuy nhiên, cũng vì nó quá phong phú, đa dạng nên thật sự khó khăn cho việc đề xuất ra một chuẩn để tất cả các tài liệu eDoc tuân theo.

Trong khi đó, eLearning, eLib với số lượng tài liệu khiêm tốn hơn nhưng thực sự đã tuân theo các chuẩn riêng của mình và được mọi người chấp nhận. Với những chuẩn riêng của mình, tài liệu eLearning, eLib dễ dàng tiến đến với web ngữ nghĩa.

2.4. Một số vấn đề trong xử lí ngôn ngữ tự nhiên:

Xử lí ngôn ngữ tự nhiên (Natural Language Processing) là bài toán lí thú nhất và cũng khó khăn nhất của ngành máy tính từ hơn 50 năm qua. Ước mơ dùng máy tính để xử lí ngôn ngữ, muốn máy tính hiểu được ngôn ngữ tự nhiên như con người, đã gặp phải trở ngại lớn nhất từ phía ngôn ngữ, đó là tính nhập nhằng (ambiquity) vốn có của ngôn ngữ tự nhiên. Tuy nhiên, từ hơn nửa thế kỉ qua, các nhà ngôn ngữ học và các nhà

KHOA CNTT

tin học đã cùng nhau từng bước khắc phục được đáng kể các trở ngại này và đã đạt nhiều kết quả tương đối khả quan.

2.4.1. Vấn đề trong việc xử lí văn bản:

Văn bản đầu vào ở dạng text, chẳng hạn như các trang HTML, chưa được xử lí. Cần phải có thêm tầng tiền xử lí để xử lí sơ bộ văn bản đầu vào, rồi phân tách nó thành các đơn vị rõ ràng (như đoạn, câu, từ, …) để cho hệ thống dễ xử lí. Bài toán tiền xử lí văn bản bao gồm các công việc sau:

- Xử lí sơ bộ văn bản đầu vào (làm sạch văn bản) bằng cách xoá bỏ những kí tự, những mã điều khiển, những phần không cần thiết trong bài toán.
- Trong mỗi văn bản, khối tiền xử lí sẽ nhận ra các tiêu đề, các chú thích, các thông tin thêm vào (tác giả, ngày…)(nếu có), và nội dung chính của văn bản.
- Trong mỗi đoạn văn, khối tiền xử lí sẽ phân rã nó thành các câu. Đây là giai đoạn khó nhất. Cao hơn nữa, khối này có thể phân tích câu thành những mệnh đề (phase) để giảm bớt gánh nặng cho hệ đồng thời tăng chất lượng cũng như tốc độ xử lí của hệ.

2.4.2. Vấn đề xử lí ngữ nghĩa:

Trong xử lý ngôn ngữ tự nhiên, bài toán gán nhãn ngữ nghĩa (sense tagger), hay còn gọi là "khử nhập nhằng ngữ nghĩa của từ" (Word Sense Disambiguation, viết tắt là WSD) là bài toán khó khăn nhất và cũng là bài toán trọng tâm mà đến nay thế giới vẫn chưa thể giải quyết ổn thoả được. Để giải quyết bài toán này, đến nay trên thế giới đã có rất nhiều mô hình với nhiều hướng tiếp cận khác nhau, chủ yếu gồm các hướng:

- ❖ Dựa trên trí tuệ nhân tạo (AI – based): đây là cách tiếp cận sớm nhất (1960) với những lý thuyết rất hay về mạng ngữ nghĩa, khung ngữ nghĩa và các ý niệm nguyên thuỷ (như: THING, DO, CAUSE,…) và các quan hệ như IS – A, PART – OF, …. Tuy nhiên, do hầu hết các tri thức về ngữ nghĩa trong cách tiếp cận này đều được xây dựng bằng tay (không thể xây dựng được nhiều tri thức về thế giới thực), vì vậy các mô hình này đều dừng lại ở mức độ biểu diễn trên một

KHOA CNTT

vài câu (demonstration on "toy" program). Vấn đề khó khăn của cách tiếp cận này là tình trạng thiếu tri thức.

❖ Dựa trên cơ sở tri thức (Knowledge – Based):

Vào đầu thập niên 80, người ta đã chuyển sang hướng khai thác tri thức tự động từ các từ điển điện tử (MRD: Machine – Readable Dictionaries) như các từ điển đồng nghĩa (thesaurus), LDOCE, LLOCE,… để có thể phần nào khắc phục hạn chế của hướng tiếp cận dựa trên trí tuệ nhân tạo (tình trạng thiếu tri thức). Kết quả của hướng tiếp cận này là sự ra đời của: mạng WordNet – một cơ sở tri thức khổng lồ về ngữ nghĩa của từ vựng theo hướng liệt kê nét nghĩa; hệ CORELEX theo hướng hệ thống nét nghĩa; và FrameNet về vai trò (case – roles) của động từ. Tuy nhiên, các cơ sở tri thức nói trên cũng chỉ là những nguồn thông tin để hệ thống chọn nghĩa tham khảo, còn chọn thông tin nào trong số những thông tin có liên quan đó thì ta phải tự xác định trong từng trường hợp cụ thể.

❖ Dựa trên ngữ liệu (Corpus – Based):

Hướng tiếp cận này sẽ rút ra các qui luật xử lý ngữ nghĩa (bằng thống kê, bằng máy học,…) từ những kho ngữ liệu lớn đã có sẵn và áp dụng các luật này cho các trường hợp mới. Thực ra cách tiếp cận này đã được nêu ra rất sớm (1940), nhưng do nguồn ngữ liệu hạn chế, thiết bị xử lý chưa hiện đại, nên không có điều kiện để phát triển. Mãi đến thập niên 1990, khi mà công nghệ phát triển mạnh, đã có thể vượt qua được những khó khăn của mình, cách tiếp cận này được hồi sinh và phát triển ngày càng mạnh mẽ cho đến ngày hôm nay.

Hiện nay, cách tiếp cận dựa trên ngữ liệu kết hợp với tri thức có sẵn là hướng tiếp cận đang được nhiều nhà ngôn ngữ học – máy tính quan tâm.

2.4.2.1. Khái niệm về nhãn ngữ nghĩa từ:

Từ khảo sát ý nghĩa từ vựng của mỗi từ, ta thấy mỗi từ có thể mang nhiều nghĩa khác nhau, nhưng trong một ngữ cảnh cụ thể, thì nó chỉ mang một nghĩa nhất định trong số những nghĩa đó. Để dễ phân biệt các nghĩa từ vựng khác nhau, các nhà ngữ

KHOA CNTT

nghĩa học, từ vựng học và tâm lý học – ngôn ngữ đã phân chia toàn bộ các ý nghĩa từ vựng có thể có thành hệ thống các ý niệm (cây ý niệm) và mỗi ý niệm như vậy được coi như là một *nhãn ngữ nghĩa của từ.*

2.4.2.2. Một số hệ thống nhãn ngữ nghĩa:

Cho đến nay, việc xây dựng một hệ thống nhãn ngữ nghĩa thống nhất vẫn chưa hoàn tất và vẫn đang tồn tại nhiều hệ thống nhãn khác nhau (mặc dù hệ thống nhãn ở mức từ pháp đã được thống nhất và xác định rõ ràng từ lâu). Vấn đề khó khăn là có những từ ta không biết nên phân vào ý niệm nào (lấy ý nghĩa nào) vì cách phân loại còn tuỳ thuộc vào mục đích và lĩnh vực sử dụng.

Ngoài ra, nếu hệ thống nhãn ngữ nghĩa được phân quá mịn thì số nhãn sẽ rất lớn (hàng chục/ trăm ngàn nhãn) và không thể gán nhãn tự động được (vì khi đó, ta cần ngữ liệu huấn luyện lớn hàng tỉ từ). Còn nếu hệ thống nhãn phân quá thô (quá ít nhãn), thì nó sẽ không đáp ứng được một số nhu cầu phân biệt nghĩa trong thực tế (chẳng hạn nhu cầu khử mơ hồ những trường hợp cùng nhãn ngữ nghĩa nhưng có ý nghĩa từ vựng khác nhau).

Một số hệ thống nhãn ngữ nghĩa thông dụng hiện nay bao gồm LLOCE (Longman Lexicon Of Contemporary English), LDOCE (Longman Dictionary Of Contemporary English), CORELEX, WordNet…. Đề tài chọn và sử dụng kho ngữ liệu WordNet là chủ yếu trong giai đoạn xử lí ngôn ngữ tự nhiên.

Hệ thống nhãn ngữ nghĩa WordNet

WordNet là một hệ cơ sở tri thức khổng lồ về ngữ nghĩa của từ vựng tiếng Anh với hơn 100.000 ý niệm khác nhau, được xây dựng bởi các nhà ngôn ngữ học – máy tính, ngôn ngữ học – tâm lý và ngôn ngữ học – tri nhận ở Đại học Princeton (Mỹ) từ đầu thập niên 1980. WordNet là một hệ trực tuyến (on – line) cho phép mọi người ở khắp mọi nơi được tự do (miễn phí) khai thác hay sử dụng cho các mục đích nghiên cứu, học tập.

WordNet là một kho tàng tri thức ngữ nghĩa từ vựng khổng lồ được nhiều nhà ngôn ngữ học và ngôn ngữ học – máy tính khai thác, ứng dụng thành công trong nhiều bài toán xử lý ngữ nghĩa. Hiện nay, WordNet đang được các nhà khoa học về ngôn

KHOA CNTT

ngữ, tâm lý, máy tính trên toàn thế giới tiếp tục khai thác, đóng góp để cải tiến ngày càng hoàn thiện hơn. WordNet có nhiều ưu điểm không thể chối cãi, đó là: tính khoa học, tính hệ thống, tính mở (open), tính dễ sử dụng, tính phổ thông, tính phát triển,… Chính vì vậy, đến nay, đã có một số công trình bản địa hoá (localization) WordNet theo ngôn ngữ của một số nước, như: Pháp, Nhật, Tây Ban Nha, Hàn, Nhật,….và gần đây là Việt Nam.

WordNet không chỉ đơn thuần là nhóm các từ đồng nghĩa hay các từ có quan hệ ngữ nghĩa với nhau thành từng lớp như một số từ điển LDOCE, LLOCE,… mà WordNet còn là một hệ thống các ý niệm có quan hệ nhiều mặt với nhau, tạo thành một mạng lưới phức tạp. Mục tiêu cơ bản của WordNet là chứa các thông tin về *ngữ nghĩa của từ*. Chính vì vậy, ngay từ đầu, ta phải xác định cách hiểu về đơn vị *từ* trong WordNet là như thế nào, sau đó ta tìm hiểu về tập đồng nghĩa (synset) – thành phần cơ bản của WordNet để áp dụng vào việc bản địa hoá WordNet thành ngôn ngữ của chúng ta.

2.4.2.3. Các nguồn tri thức để xử lý ngữ nghĩa:

Để xử lý ngữ nghĩa, người ta phải kết hợp nhiều nguồn tri thức: từ các tri thức về ngôn ngữ (như: hình thái, ngữ pháp, ngữ nghĩa) cho đến các tri thức ngoài ngôn ngữ (tri thức về thế giới thực). Các nguồn tri thức đó thường bao gồm:

2.4.2.3.1. Tri thức về từ loại

Trong trường hợp các từ đồng tự (homograph) và có nghĩa khác nhau với các từ loại khác nhau và ứng với một từ loại chỉ có một nghĩa duy nhất, thì nhờ thông tin từ loại, chúng ta sẽ xác định được chính xác nghĩa của chúng. Ví dụ, từ "can" có nghĩa là "có thể" (trợ động từ), "cái hợp" (danh từ), "đóng hợp" (động từ). Vì vậy, với các trường hợp này, nếu biết được chính xác từ loại, chúng ta hoàn toàn khử được nhập nhằng nghĩa của chúng. Ví dụ: " I_{PRO} can_{AUX} can_{V} a_{DET} can_{NN}" (Tôi *có thể đóng hộp* một *cái hộp)*.

Theo thống kê trong từ điển LLOCE , có tới 88% mục từ thuộc dạng nói trên, ngoài ra có 7% trường hợp mà mục từ (tập các từ đồng tự) có nhiều từ loại, mỗi từ loại

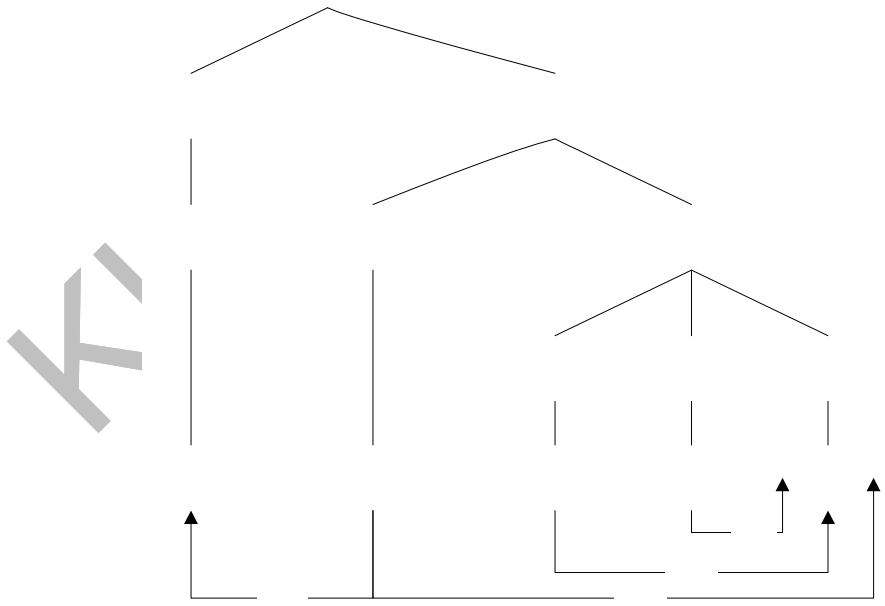
KHOA CNTT

có thể có nhiều nghĩa khác nhau, nhưng trong đó có ít nhất một từ loại có duy nhất một nghĩa. Đối với trường hợp này, ta có thể khử nhập nhằng nghĩa nếu từ loại của nó (trong ngữ cảnh) chính là từ loại mà chỉ có một nghĩa.

2.4.2.3.2. Tri thức về quan hệ cú pháp và ràng buộc ngữ nghĩa:

Trường hợp một từ trong một từ loại có nhiều hơn một nghĩa, thì thông tin từ loại không đủ để khử nhập nhằng nghĩa. Ví dụ: từ "bank" (có 2 từ loại là động từ và danh từ), với từ loại danh từ có các nghĩa: "ngân hàng", "bờ sông", "dãy",…. Trong trường hợp này, ta cần sử dụng thêm tri thức về thế giới thực thông qua các ràng buộc ngữ nghĩa (selectional restriction) giữa các thành phần cú pháp (S – V – O – M) trong câu. Ví dụ, trong câu "I enter an old bank", sau khi qua phần gán nhãn ngữ pháp, ta được:

$[I_{PRO}]_{NP}$ [$enter_{V}$ [an_{DET} old_{ADJ} $bank_{N}]_{NP}]_{VP}$ và cây cú pháp như hình dưới đây:



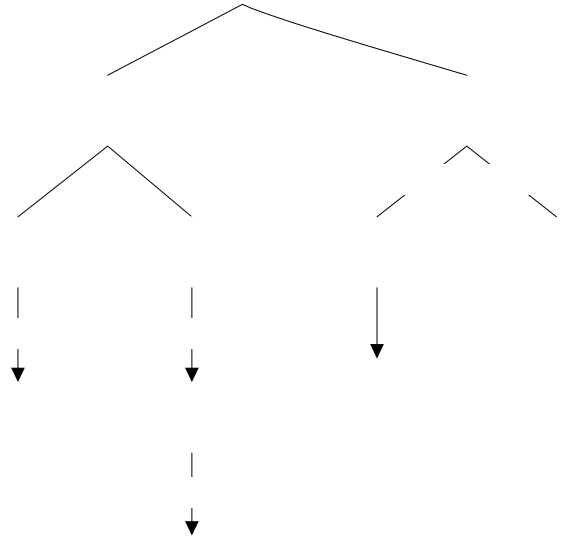
Hình 13 Các quan hệ cú pháp và ràng buộc ngữ nghĩa

Trên cây cú pháp này, ta xác định được các quan hệ cú pháp như: S – V (chủ ngữ – động từ), V – O (động từ – đối từ), A – N (tính từ – danh từ), D – N (định từ – danh từ). Mỗi từ thực (content words) trong câu trên, cho dù đã xác định được từ loại chính xác, nhưng đều vẫn gây nhập nhằng về ngữ nghĩa. Ví dụ, động từ "enter" (đi vào / nhập), danh từ "bank" (ngân hàng/ bờ sông/ dãy), tính từ "old" (già/ cũ). Vì vậy, chúng ta phải sử dụng đến những ràng buộc ngữ nghĩa như sau:

Từ	Ràng buộc / nhãn ngữ nghĩa	Ràng buộc
I (tôi)	Type: Person (Người)	
Enter1 (đi vào)	S:Human (người)	O:Closed – SPA (không kín)
Enter2 (nhập)	S:Human (người)	O: Data (dữ liệu)
Bank1 (ngân hàng)	Type: Hou (nhà cửa, không gian kín)	
Bank2 (bờ sông)	Type: Nat (công trình thiên nhiên, không gian hở)	
Old1 (già)	N: Ani (có sự sống)	
Old2 (cũ)		

Bảng 5: Danh sách các nghĩa và ràng buộc của các từ thực trong câu.

KHOA CNTT



Hình 14 Cây quyết định trong việc chọn nghĩa phù hợp.

Qua việc duyệt cây từ trên xuống với gốc là động từ (Enter), cuối cùng ta chọn được các nghĩa phù hợp: enter1 (đi vào), bank1 (ngân hàng), và old2 (cũ). Trong việc xét điều kiện ràng buộc về ngữ nghĩa, chúng ta phải xét đến tính cấp bậc (hierachical) trong hệ thống nhãn ngữ nghĩa (ontology) mà trong đó khái niệm con sẽ kế thừa các nét nghĩa của khái niệm cha và có thêm nét nghĩa mới riêng của chúng. Thông tin về đặc điểm ngữ nghĩa (type) của từng mục từ thực cũng như các ràng buộc đã được xác định trong từ điển LDOCE và FrameNet.

2.4.2.3.3. Tri thức về ngôn từ (Collocation)

Ràng buộc về ngữ nghĩa giữa các thành phần cú pháp không phải lúc nào cũng giải quyết được mọi nhập nhằng, vì có những quan hệ tiềm ẩn về logic, về ngữ nghĩa hoặc thậm chí do thói quen mà việc nhận biết phải đòi hỏi những tri thức thế giới thực mà đến nay người ta cũng chưa thể tích hợp hết vào từ điển hay các cơ sở tri thức khác trong máy tính.

KHOA CN

Ví dụ, danh từ "bank" trong câu "I go to the bank…" có nghĩa gì? Ta sẽ chọn nghĩa nào trong số các nghĩa: "ngân hàng/ bờ (sông) / dãy"; danh từ "way" là "đường (đi) / cách (thức)"?; danh từ "letter" là "bức thư / chữ cái"?;…. Nếu ta chỉ xét các ràng buộc về ngữ nghĩa (không phải lúc nào các ràng buộc này cũng có mặt đầy đủ) thì ta khó mà có thể xác định được chính xác nghĩa của các từ nhập nhằng đó.

Vì vậy, để khử nhập nhằng trong những trường hợp này, người ta thường xét đến hình thái và ngữ nghĩa của các từ lân cận hay còn gọi là ngôn từ (collocation). Chẳng hạn khi thấy "bank … river" → "bờ sông", "bank … account/money" → "ngân hàng"; "way to" → "đường (đi)", "way of" → "cách thức"; "write … letter … to" → "bức thư", "… letter A" → "chữ cái", "… letters, digits, symbols …" → "chữ cái", "write … papers, letters, messages,…" → "bức thư";….

Phạm vi lân cận của từ cần khử ngữ nghĩa có thể là bên trái 1, 2 hay n từ và bên phải 1, 2 hay n từ. Việc chọn lựa lân cận này phụ thuộc vào từng trường hợp và cá nhân cụ thể.

2.4.2.3.4. Tri thức về chủ đề (subject)

Trong một số trường hợp nhập nhằng, chúng ta có thể xác định được nghĩa đúng của từ nếu ta biết được chủ đề của văn bản. Chẳng hạn từ "bank", nếu đang nói về vấn đề "tài chính" thì nó thường có nghĩa là "ngân hàng"; từ "driver" → "trình điều khiển" (nếu chủ đề là lĩnh vực "tin học"); "sentence" → "câu" (nếu chủ đề là "ngôn ngữ / văn phạm") hoặc "bản án" (nếu đang nói về "pháp luật"); "element" → "nguyên tố" (trong "hoá") / "phần tử" (trong "toán / tin học");….

Để xác định được chủ đề của văn bản đang cần dịch, **ta cần xem xét sự xuất hiện của một số từ chuyên môn trong lĩnh vực đó**. Chẳng hạn, nếu trong văn bản ta thấy xuất hiện các từ như: "ellipsis" (tỉnh lược), "bilingual" (song ngữ), "anaphora" (thế đại từ), "pharse" (ngữ), … thì ta có thể đoán nhận văn bản này đang nói về chủ đề "ngôn ngữ học"; tương tự cho các từ "computer", "memory", "peripherals", "CPU",… → đang nói về "tin học", ….

KHOA CNTT

Chính vì vậy, trong từ điển LDOCE/ LLOCE đều có mã số chủ đề cho các từ chuyên môn này. Chúng ta có thể xác định được chủ đề một cách tự động bằng cách xem xét các từ chuyên môn lân cận từ đang cần khử nhập nhằng.

2.4.2.3.5. Tri thức về tần suất nghĩa của từ

Một từ không phải lúc nào cũng thuộc về một chủ đề nhất định (trong từ điển LDOCE, hơn 56% từ thuộc dạng này), vì vậy tính thông dụng của một nghĩa nào đó còn được dựa trên độ đo về tần suất (frequency) xuất hiện của từ đó đối với nghĩa cụ thể đó. Ví dụ, danh từ "pen"có nghĩa thông dụng nhất là "bút/ viết" (bên cạnh các nghĩa ít thông dụng hơn, như: "chuồng", "lồng chim"); "ball" thường có nghĩa là "quả banh/ hòn bi" hơn là "buổi khiêu vũ",…

Độ đo tần suất xuất hiện của mỗi nghĩa của mỗi từ được thống kê trên những ngữ liệu rất lớn thuộc nhiều loại văn bản khác nhau. Chính vì vậy, trong WordNet và trong LDOCE, các nghĩa được sắp xếp theo thứ tự giảm dần (nghĩa thông dụng nhất sẽ được liệt kê đầu tiên).

2.4.2.3.6. Tri thức trong định nghĩa của nghĩa từ (definition):

Trong các từ điển LDOCE/ WordNet, mỗi nghĩa sẽ được định nghĩa và có ví dụ kèm theo. Ví dụ, từ "bank" trong LDOCE sẽ có các nghĩa kèm định nghĩa của nó như:

- "land along the side of a river, lake, etc." (đất dọc bên sông / hồ)
- "a place where money is kept and paid …." (nơi giữ tiền và trả tiền …)
- "a row, a line of …" (một hàng, một dãy …)

Dựa trên thông tin trong các định nghĩa này, và so sánh với thông tin của ngữ cảnh, ta có thể xác định được nghĩa phù hợp của từ trong ngữ cảnh đó. Để thực hiện điều này, Wilks et.al. đã tính toán phần giao (overlap) của tất cả các tổ hợp nghĩa của các từ thực trong câu tiếng Anh dùng để định nghĩa mỗi nghĩa của từ.

KHOA CNTT

2.4.2.4. Gán nhãn ngữ nghĩa

Khử nhập nhằng nghĩa của từ là bài toán đặc trưng trong gán nhãn ngữ nghĩa. Tức là nghĩa của từ đa nghĩa sẽ được xác định ngay nếu biết nhãn ngữ nghĩa của nó, ví dụ: danh từ "bank" sẽ có nghĩa là "ngân hàng" nếu được gán nhãn là "HOU", và có nghĩa "bờ (sông)" nếu gán nhãn "NAT", ….

Trong các mô hình gán nhãn ngữ nghĩa theo cách tiếp cận dựa trên các nguồn tri thức nói trên, người ta thường sử dụng bộ nhãn có độ mịn (granularity) khác nhau. Bộ nhãn càng mịn (chi tiết hàng trăm ngàn nhãn như WordNet) thì độ chính xác của việc gán nhãn sẽ thấp hơn nhưng khả năng khử nhập nhằng nghĩa của nó sẽ cao hơn (vì không có trường hợp nào cùng nhãn mà khác nghĩa). Ngược lại, nếu chọn bộ nhãn càng thô (chỉ có 36 nhãn như LLOCE), thì độ chính xác trong gán nhãn sẽ cao hơn và tất nhiên khả năng khử nhập nhằng nghĩa sẽ thấp hơn (sẽ có nhiều trường hợp cùng nhãn nhưng khác nghĩa).

Ngoài ra, việc gán nhãn ngữ nghĩa còn được phân biệt theo quy mô gán nhãn: hoặc là gán cho một số ít các từ điển hình (như Hwee Ng và Hian Lee cho một từ *interest,* David Yarowsky cho 12 từ,…) hoặc là gán cho hầu hết các từ thực (như Mark Stevenson và Yorick Wilks, Mona Diab và Philip Resnik).

Việc chọn nguồn tri thức nào cho mỗi tình huống được hệ thống quyết định bằng phương pháp học giám sát trên ngữ liệu đã được gán nhãn ngữ nghĩa chính xác (đây chính là ngữ liệu huấn luyện hay còn gọi là ngữ liệu vàng). Giải thuật học có thể là mạng Neural, cây quyết định, MBL, TBL,… mà trong đó các giải thuật học dựa trên ký hiệu (symbolic) tỏ ra chính xác hơn.

2.4.2.5. Các mức độ nhập nhằng trong xử lý ngữ nghĩa:

2.4.2.5.1. Nhập nhằng mức từ vựng:

Như câu ví dụ "I enter the bank" ở trên, sau khi phân tích cú pháp, máy tính đã xác định được mối quan hệ giữa động từ "enter" (đi vào) và đối từ của nó là "bank" (là *ngân hàng* hay *bờ sông*?) thì phải cần phân tích ngữ nghĩa của động từ "enter" và danh từ "bank". Trong trường hợp này máy sẽ vận dụng các ý niệm của ngôn ngữ học tri

KHOA CNTT

nhận để biết rằng "enter" là hành động "*đi vào không gian kín* (close space)" và danh từ "bank" với nghĩa là "bờ sông" có thuộc tính là "không gian hở" thì sẽ không thoả thuộc tính này, chỉ có "bank" với nghĩa "ngân hàng" là sẽ thoả điều kiện "không gian kín" này, nên cuối cùng máy tính sẽ chọn nghĩa "ngân hàng".

2.4.2.5.2. Nhập nhằng mức cấu trúc:

Ví dụ xét ngữ "*Old man and woman"*, ta có 2 phân tích: "[Old man] and [woman]" và "Old [man and woman]" và máy tính sẽ chọn cách phân tích thứ nhì (do tính cân bằng vốn có trong cấu trúc song song của liên từ "and"). Tuy nhiên, nếu xét "Old man and child", ta cũng sẽ có 2 phân tích: "[Old man] and [child]" và "Old [man and child]" và máy tính sẽ chọn cách phân tích thứ nhất, vì máy thấy cấu trúc thứ nhì là vô lý (do có sự đối lập giữa thuộc tính "trẻ" trong "child" và già trong "man").

2.4.2.5.3. Nhập nhằng mức liên câu:

Ví dụ xét câu "*The monkey ate the banana because <u>it</u> was hungry"* (con khỉ ăn chuối vì <u>nó</u> đói). Trong một số trường hợp, máy tính hiện nay có thể xác định được đại từ "it" (nó) thay thế cho từ nào: "monkey" (khỉ) hay "banana" (chuối). Để giải quyết được nhập nhằng này, máy tính phải xem lại mệnh đề trước và vận dụng tri thức về thế giới thực có trong WordNet để biết rằng "chỉ có khỉ mới có khả năng đói" nên sẽ chọn "*it* thay thế cho *monkey*". Còn trong câu: "*The monkey ate the banana because <u>it</u> was ripe" (con khỉ ăn chuối vì <u>nó</u> chín)*, thì máy tính sẽ biết rằng "chỉ có *chuối* mới có khả năng *chín)*, nên sẽ chọn "*it* thay thế cho *banana"*.

2.4.3. Phân loại văn bản (Text Classification)

Trong thời đại ngày này, thời đại của thông tin, lượng văn bản ngày càng lớn và ta cần phân loại các văn bản thành các nhóm chủ đề khác nhau, như: theo chuyên ngành (Toán, Lý, Hoá, Văn, Sử, …), theo lĩnh vực (Khoa học, Văn hoá, Xã hội, Chính trị, …), …. Do khối lượng quá lớn, ta không thể phân loại thủ công bằng tay được. Vì vậy, một chương trình máy tính phân loại tự động được yêu cầu. Để xây dựng chương trình này, người ta đã dùng nhiều cách tiếp cận khác nhau, như: dựa trên từ khoá, dựa

KHOA CNTT

trên trường ngữ nghĩa của các từ có tần số xuất hiện cao, mô hình Maximum Entropy, dựa trên lý thuyết tập thô, …

Đối với tiếng Anh, các kết quả trong lĩnh vực này rất khả quan. Còn đối với tiếng Việt, gần đây đã có một số công trình nghiên cứu về vấn đề này và đã có một số kết quả ban đầu nhưng còn hạn chế do phần phân tích hình thái (tách từ) và từ điển ý niệm (phân loại ngữ nghĩa) cho tiếng Việt chưa hoàn thiện. Bên cạnh việc phân loại văn bản, người ta cũng quan tâm đến các ứng dụng gom cụm văn bản nhằm nhóm các văn bản có nội dung tương tự nhau (theo các thông số của văn bản) lại với nhau.

KHOA CNTT

Chương 3 : MÔ HÌNH VÀ GIẢI THUẬT

3.1. Công nghệ tìm kiếm ngữ nghĩa trên thế giới hiện nay:

Hầu hết các hiệu quả gần đây của các công cụ tìm kiếm dựa vào ngữ nghĩa là phụ thuộc cao vào công nghệ xử lí ngôn ngữ tự nhiên để phân tích và hiểu câu truy vấn. Một trong những công cụ tìm kiếm đầu tiên và thông dụng nhất này là Ask Jeeves (http://www.askjeeves.com/). Nó liên kết những điểm mạnh của phần mềm phân tích ngôn ngữ tự nhiên, xử lí khai khoáng dữ liệu, và tạo cơ sở tri thức với những phân tích theo kinh nghiệm. Người dùng có thể gõ các truy vấn bằng ngôn ngữ tự nhiên và nhận được những trả lời thoả đáng.

Một ví dụ dựa trên ngữ nghĩa khác là Albert (http://www.albert.com/). Ưu điểm lớn nhất của nó là cung cấp nhiều ngôn ngữ thêm vào cho tiếng Anh, ví dụ như tiếng Pháp, Tây Ban Nha, Đức. Loại này của search engine cần một số đông người để xây dựng nên một mạng ngữ nghĩa rất lớn nhằm mục đích hướng tới việc thực thi hợp lí.

Một kiểu nâng cao khác của công cụ tìm kiếm Internet là Cycorp (http://www.cyc.com/). Cyc liên kết cơ sở tri thức lớn nhất trên thế giới với Internet. Cyc (en-cyc-lopedia) là một cơ sở tri thức bao la và đa ngữ cảnh. Với Cyc Knowledge Server, nó cho phép các site Internet thêm vào tri thức ngữ nghĩa thông dụng và phân biệt những nghĩa khác nhau của các khái niệm nhập nhằng.

3.1.1. Các hiệu quả tìm kiếm ngữ nghĩa hiện nay

Khi công nghệ Web trí tuệ nhân tạo trở nên nâng cao hơn, sử dụng các thẻ RDF và OWL sẽ đưa ra những cơ hội ngữ nghĩa cho tìm kiếm. Tuy nhiên, kích thước của mạng đang được tìm kiếm sẽ phải thiết lập một khoảng trống cho giải pháp phức tạp và do đó ảnh hưởng mạnh đến khả năng xuất hiện của các kết quả thành công.

Nhiều công ty lớn đang thật sự hướng đến vấn đề của tìm kiếm ngữ nghĩa. Sự phát triển của Microsoft về Web có lẽ phụ thuộc vào khả năng của nó để hoàn thiện công cụ tìm kiếm mà dẫn đầu là Google. Kết quả là Microsoft đã đưa ra một chương

KHOA CNTT

trình tìm kiếm mới gọi là MSNBot, nó lướt qua Web để xây dựng một chỉ mục của các liên kết HTML và các tài liệu. MSNBot được dự định như là một công nghệ mà kết hợp các ứng dụng cho hệ điều hành Windows. Sau đó Microsoft sẽ kết nối công cụ tìm kiếm của nó với cổng MSN trong phiên bản Windows kế tiếp của nó nhằm làm cho dễ dàng tìm kiếm e-mail, spreadsheets và các tài liệu trên các PC (Personal Computer), các mạng hợp nhất, cũng như Web.

3.1.2. Công nghệ tìm kiếm

Tìm kiếm ngữ nghĩa giải quyết với các khái niệm và các mối quan hệ logic. Nếu xem xét các vấn đề thực tế của tìm kiếm ngữ nghĩa, chúng ta sẽ thấy rằng cây tìm kiếm đứng trước tình trạng thiếu logic đưa đến vấn đề chưa hoàn tất (Incompleteness Problem) hay vấn đề "ngắc ngứ" (Halting Problem).

Đầu tiên hãy xem xét **vấn đề chưa hoàn tất**. Kết luận có thể được xem như là một sự suy diễn của một dãy logic gắn lại với nhau. Ở mỗi điểm, có thể có nhiều hướng khác nhau để tới một suy diễn mới. Vì vậy, nhằm đạt hiệu quả, có một nhóm các khả năng phân nhánh để bằng cách nào đó hướng đến một giải pháp đúng. Và nhóm các phân nhánh đó có thể trải ra trong các hướng mới lạ.

Ví dụ, bạn có thể muốn cố gắng định nghĩa "ai là người mà Kevin Bacon biết" dựa trên thông tin về mối quan hệ gia đình của anh ta, những phim của anh ta, hay những tiếp xúc công việc của anh ta. Do đó, có nhiều hơn một hướng để đưa đến một số các kết quả. Các kết quả này nằm trong một nhóm phân nhánh các khả năng có thể có. Do vậy, kết luận trong hệ thống của chúng ta là một loại của vấn đề tìm kiếm, được biểu thị như là một cây tìm kiếm.

Có thể bắt đầu ở đỉnh của cây, ở gốc, hay từ các nhánh. Đỉnh của cây có thể là câu truy vấn được hỏi. Mỗi bước lần xuống các nút con trong cây này có thể được xem như một suy diễn logic tiềm tàng di chuyển hướng đến việc cố gắng xác nhận câu truy vấn nguyên thủy mà sử dụng bước suy diễn logic này. Hướng rẽ quạt của các khả năng có thể được xem như cây phân nhánh này, trở nên rậm rạp hơn và sâu hơn. Mỗi tiếp cận này kết thúc bằng việc trở thành một trong các bước con, đến một nút con.

KHOA CNTT

Tưởng tượng rằng mỗi nút trong cây này biểu thị một vài hướng để xác nhận. Mỗi liên kết từ một nút cha cao hơn đến một nút con biểu thị một câu lệnh logic. Bây giờ vấn đề này là chúng ta có một cây lớn của các khả năng.

Trong một hệ thống logic phức tạp, có một số lượng lớn các chứng cớ tiềm tàng. Một số chúng dài và không rõ ràng nếu chỉ có một chứng cớ. Được chứng minh vào những năm 1930, một số hệ thống logic đủ phức tạp vốn đã là không đầy đủ (không thể quyết định). Nói cách khác, có các câu lệnh mà không thể được chứng minh một cách logic. Luận cứ của nó cho điều đó liên quan đến một vấn đề khác, vấn đề "ngắc ngứ" (Halting Problem).

Vấn đề halting suy ra rằng các thuật giải hiện nay sẽ không bao giờ kết thúc trong một câu trả lời. Khi nói về Web, chúng ta nói về hàng triệu các sự kiện và hàng chục ngàn luật mà có thể nối kết đan lại với nhau trong những hướng phức tạp, vì thế không gian của các chứng cứ tiềm tàng là vô tận và cây này theo logic sẽ trở nên vô tận. Theo đó, chúng ta sẽ đi vào các vấn đề không hoàn tất vốn có; ví dụ như chúng ta không thể thấy mỗi chứng cứ có thể có và thu tất cả các câu trả lời.

Chúng ta sẽ đi vào tình trạng không hoàn tất bởi vì cây tìm kiếm quá lớn. Vì thế hướng tiếp cận của chúng tôi là chỉ phải tìm kiếm trên các phần của cây. Có một chiến lược nổi tiếng cho việc bằng cách nào để chỉ ra các vấn đề tìm kiếm như vậy. Một chiến lược là tìm kiếm cây theo "chiều sâu" (depth-first).

Tìm kiếm chiều sâu sẽ bắt đầu ở đỉnh cây và đi xuống sâu đến mức có thể một số đường dẫn nào đó, mở rộng các nút khi chúng ta đi, cho đến khi tìm thấy một kết thúc chết (dead end). Một kết thúc có thể là một đích (thành công) hay một nút mà chúng ta không thể tạo ra các con mới. Vì vậy hệ thống không thể chứng minh bất cứ thứ gì ngoài điểm này.

Hãy xem qua tìm kiếm theo chiều sâu và xoay theo trục của cây. Chúng ta bắt đầu ở nút đỉnh và đi sâu nhất có thể:

1) Bắt đầu ở nút cao nhất.
2) Đi xuống sâu nhất có thể theo một hướng.

KHOA CNTT

3) Khi chúng ta đi vào một kết thúc, sao lưu nút cuối cùng mà từ đó chúng ta rời khỏi. Nếu có một đường dẫn mà chúng ta chưa đi, thì hãy lần theo nó. Cứ theo chọn lựa này cho đến khi chúng ta thấy một kết thúc hay một đích đến.
4) Đường dẫn này dẫn đến một kết thúc khác, vì thế đi trở lại một nút và cố gắng ở nhánh khác.
5) Đường dẫn đưa đến một điểm đích. Nói cách khác, nút cuối cùng này là một kết quả khả quan cho truy vấn. Vì thế chúng ta có một câu trả lời. Hãy tìm kiếm những đáp án khác bằng cách đi lên một vài node và sau đó đi xuống một đường dẫn mà chúng ta chưa đi thử.
6) Tiếp tục cho đến khi thấy nhiều hơn những điểm kết thúc và sử dụng hết những khả năng tìm kiếm.

Ưu điểm của tìm kiếm theo chiều sâu là: đây là một cách hiệu quả theo thuật toán để tìm kiếm các cây trong một định dạng. Nó giới hạn số lượng không gian mà ta có để duy trì việc nhớ những thứ mà ta chưa nhìn thấy. Tất cả những thứ mà chúng ta phải nhớ là lưu lại đường dẫn.

Khuyết điểm của tìm kiếm này là một khi chúng ta bắt đầu đi xuống một hướng, chúng ta sẽ đi đến tất các các con đường cho đến cuối cùng.

Một chiến lược khác cho tìm kiếm là tìm kiếm theo chiều ngang trước. Ở đây chúng ta tìm kiếm từ lớp này sang lớp khác. Đầu tiên chúng ta cố gắng thực hiện tất cả các kiểm chứng ở bước 0 và sau đó chúng ta cố gắng thực hiện tất cả các kiểm chứng ở bước 1, v.v… Ưu điểm của tìm kiếm theo chiều ngang là chúng ta được bảo đảm nhận các kiểm chứng đơn giản nhất trước khi chúng ta đến những cái phức tạp hơn. Điều này được đưa ra do những lợi ích của Ockham’s Razor. Nếu có một kiểm chứng ở bước thứ n, chúng ta sẽ tìm thấy nó trước khi chúng ta xem xét đến bước thứ n+1. Khuyết điểm của tìm kiếm theo chiều ngang là chúng ta có những cây rất sâu, chúng ra cũng có những cây rất rậm rạp mà chúng ta có hàng ngàn hay hàng chục ngàn các nút con. Khuyết điểm khác của tìm kiếm này là số lượng không gian chúng ta phải sử

KHOA CNTT

dụng để lưu tất cả các kết quả mức thứ 3 trước khi chúng ta khảo sát nó. Với tìm kiếm theo chiều rộng, chúng ta càng đi vào cây càng sâu thì không gian yêu cầu càng lớn.

Vì thế chúng ta nhận ra rằng hai trong các thuật giải cổ điển cho tìm kiếm, theo chiều dọc và chiều ngang, sẽ dẫn đến những vấn đề về các hệ thống lớn.

Có hai lớp cơ bản của các giải thuật tìm kiếm được sử dụng để cố gắng giải quyết các giới hạn về vấn đề không hoàn tất và tình trạng ngắc ngứ là: không có đủ thông tin và có đủ thông tin. **Các tìm kiếm không đầy đủ thông tin**, hay không nhìn thấy, thì không có thông tin về số lượng các bước hay chi phí đường dẫn từ trạng thái hiện tại đến đích. Những tìm kiếm kiểu này bao gồm: tìm theo chiều sâu (depth-first), theo chiều rộng (breadth-first), chi phí không đổi (uniform-cost), giới hạn chiều sâu (depth-limiting) và tìm kiếm sâu thêm lặp đi lặp lại (iterative deepening). **Các tìm kiếm đầy đủ thông tin**, hay heuristic, có đầy đủ thông tin về đích đến; thông tin này thường là chi phí đường dẫn ước lượng cho nó hay là ước đoán số lượng các bước xuất phát từ nó. Thông tin này được biết như là heuristic search agent. Nó cho phép các tìm kiếm có đầy đủ thông tin thực hiện tốt hơn những tìm kiếm không đủ thông tin và làm cho chúng hành xử trong một dáng vẻ hoàn toàn "lí trí". Những tìm kiếm này bao gồm: các tìm kiếm best-first, hill-climbing, beam, A*, và IDA* (iterative deepening A*).

3.1.3. Các Web search agent

Trong khi các công cụ tìm kiếm là mạnh và quan trọng cho tương lai của Web, thì có một hình thức hoạt động khác của tìm kiếm cũng đóng vai trò quyết định: các trạm tìm kiếm Web (Web search agent). Một Web search agent sẽ không thực hiện như một công cụ tìm kiếm thương mại. Các công cụ tìm kiếm này sử dụng cơ sở dữ liệu tra cứu từ một cơ sở tri thức (Knowledge Base).

Trong trường hợp của Web search agent, tự các trang Web được tìm kiếm và máy tính cung cấp một giao diện cho người dùng. Các kết quả tri giác của agent là các tài liệu được kết nối thông qua Internet sử dụng HTTP. Các hoạt động của agent được định nghĩa nếu tìm thấy đích đến của việc tìm một trang Web chứa một điểm đích

KHOA CNTT

được chỉ rõ (ví dụ như từ khoá hay cụm từ) và nếu không , thì tìm một vị trí khác để viếng thăm. Nó hoạt động trong môi trường sử dụng các phương pháp đầu ra để cập nhật người dùng ở trạng thái của tìm kiếm hay các kết quả kết thúc.

Cái gì làm cho "trí tuệ" của agent có khả năng ra quyết định có lí trí khi đưa ra một chọn lựa. Nói cách khác, đưa ra một đích đến, chúng sẽ ra quyết định đi theo những hành động mà dẫn đến đích trong một cách đúng lúc.

Một agent thường có thể phát sinh ra tất cả các kết quả có thể có của một sự kiện, nhưng sau đó nó sẽ cần tìm kiếm thông qua những kết quả đó để tìm kiếm một đích đến mong muốn và thực thi đường dẫn (chuỗi các bước) bắt đầu ở trạng thái ban đầu hay trạng thái hiện tại, để đến trạng thái của đích đến mong muốn. Trong trường hợp của Web search agent thông minh, nó sẽ cần sử dụng một tìm kiếm để định hướng thông qua Web để tới đích của nó.

Việc xây dựng một Web search agent thông minh cần những kỹ thuật cho tìm kiếm nhiều và kết hợp từ khoá, ngăn chận "handling" và khả năng tự nảy mầm khi nó sử dụng hết hoàn toàn một không gian tìm kiếm. Đưa ra một điểm đích, Web search agent xử lí để tìm kiếm thông qua một số đường dẫn cần thiết. Agent này sẽ dựa vào từ khoá. Phương pháp được ủng hộ này là để bắt đầu từ một vị trí "hạt giống" (do người dùng cung cấp) và tìm tất cả những vị trí khác được liên kết trong một dạng cây đến gốc (vị trí hạt giống) chứa điểm đích.

Search agent cần biết điểm đích (ví dụ từ khoá hay cụm từ), nơi mà bắt đầu, lặp lại bao nhiêu lần điểm đích để nhận thấy sẽ xem bao lâu (ràng buộc thời gian), và phương pháp gì nên được định nghĩa tiêu chuẩn cho việc chọn đường dẫn (các phương pháp tìm kiếm). Những vấn đề này được đưa ra trong phần mềm.

Việc thực thi cần một số tri thức của lập trình, làm việc với sockets, HTTP, HTML, sắp xếp, và tìm kiếm.

Có nhiều ngôn ngữ trong những thi hành trên Web, những giao diện lập trình ứng dụng (APIs) nâng cao, và khả năng phân tách văn bản tốt hơn mà có thể sử dụng để viết một Web agent.

KHOA CNTT

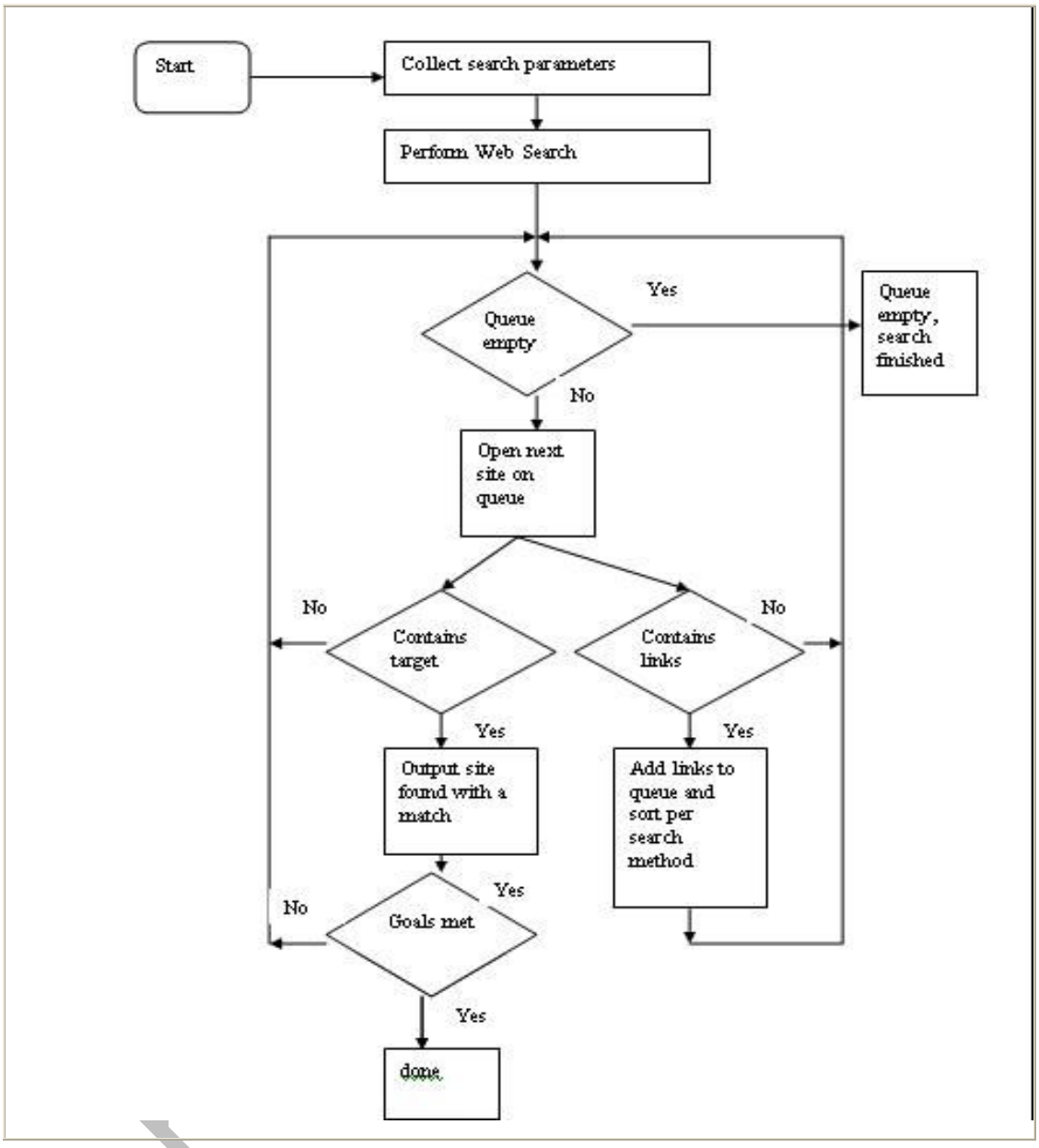
Sử dụng thuật giải sắp xếp nâng cao và hiệu quả sẽ giúp cải thiện thực thi của Web search agent.

Thiết kế Web search agent gồm bốn giai đoạn: khởi tạo, nhận thức, hành động và hiệu quả. Trong **giai đoạn khởi tạo**, Web search agent nên tạo lập tất cả các biến, cấu trúc và mảng. Cũng nên lấy thông tin cơ sở cần cho việc chỉ đạo săn tìm điểm đích, đích đến, một vị trí bắt đầu và phương pháp tìm kiếm. **Giai đoạn nhận thức**, được tập trung sử dụng tri thức được cung cấp để tiếp xúc với một trang và thu hồi thông tin từ vị trí đó. Nó nên được nhận diện nếu hiện diện điểm đích và nên nhận ra các đường dẫn đến những vị trí URL khác. **Giai đoạn hành động** lấy tất cả những thông tin mà hệ thống biết và định nghĩa nếu đích đến được tìm thấy (điểm đích được tìm thấy và việc săn tìm kết thúc).

Nếu việc săn tìm vẫn còn hoạt động nó phải ra quyết định đi đến nơi nào tiếp theo. Đây là sự thông minh của agent, và phương pháp của tìm kiếm cho biết Web agent sẽ "thông minh" bao nhiêu. Nếu một liên kết không tìm thấy, việc săn tìm kết thúc, và nó cung cấp đầu ra cho user.

Web search agent di chuyển từ giai đoạn khởi tạo đến một vòng lặp bao gồm các giai đoạn nhận thức, hoạt động và hiệu quả cho đến khi đạt được đích đến hay không.

KHOA CNTT



Hình 15: **Dòng cơ sở tìm kiếm Web**

3.2. Các bước xây dựng một ứng dụng semantic search engine:

Một ví dụ của công nghệ tìm kiếm ngữ nghĩa là TAP. TAP là một đề án phân tán gồm những nhà nghiên cứu từ Standford, IBM, và W3C. TAP tạo đòn bẩy cho công nghệ tự động và bán tự động rút ra những cơ sở tri thức từ phần thân không có

cấu trúc hay bán cấu trúc của văn bản. Hệ thống này có thể sử dụng thông tin vừa học để học thêm thông tin mới, và có thể sử dụng để thu hồi thông tin.

Trong TAP, các tài liệu sẵn có được phân tích sử dụng công nghệ ngữ nghĩa và chuyển sang thành các tài liệu Web ngữ nghĩa sử dụng công nghệ tự động hay thủ công với các gói tri thức có cấu trúc ngày càng sâu hơn. Công nghệ thu hồi thông tin truyền thống được nâng cao với tri thức có cấu trúc sâu để cung cấp các kết quả chính xác hơn. Cả hai phép phân tích tự động và được hướng dẫn sử dụng các hệ thống và các agent lập luận thông minh.

Các giải pháp xây dựng nên một công nghệ trung tâm được gọi là các Semantic Web Template. Thực hiện biểu diễn tri thức, sự sáng tạo, sự tiêu thụ và duy trì của tri thức trở nên trong suốt đối với người dùng. Mô hình dữ liệu RDF là cơ sở của công nghệ biểu diễn tri thức Web ngữ nghĩa và TAP sử dụng RDF Schema và OWL.

Khó khăn của việc tự tạo ra tri thức yêu cầu một máy tri thức có thể dùng để dịch các tài liệu sang những ngôn ngữ tượng trưng và logic được yêu cầu. Các ontology sử dụng vốn từ vựng chính của tri thức được yêu cầu để định nghĩa các khái niệm và mối quan hệ mà các trường hợp của khái niệm đó nắm giữ.

3.3.1. Xây dựng kiến trúc Web ngữ nghĩa:

Kiến trúc Web ngữ nghĩa được phát triển dựa trên ý tưởng của việc chú thích các trang Web bằng các thẻ RDF và OWL để biểu diễn chi tiết các ontology ngữ nghĩa. Tuy nhiên, giới hạn của các hệ thống này là chúng chỉ xử lí các trang Web đã được chú thích bằng những thẻ ngữ nghĩa cụ thể.

Ontology mô tả các khái niệm và mối quan hệ với một tập từ vựng tiêu biểu. Mục đích của việc xây dựng ontology là chia sẻ và sử dụng lại tri thức. Từ khi Web ngữ nghĩa là một mạng phân tán, có những ontology khác nhau mô tả những điều tương đương một cách ngữ nghĩa. Kết quả là, cần thiết để lập sơ đồ các yếu tố của những ontology này nếu chúng ta muốn xử lí thông tin trên qui mô của Web. Một tiếp cận cho tìm kiếm ngữ nghĩa có thể dựa trên việc phân loại văn bản cho những ánh xạ ontology so sánh mỗi yếu tố của một ontology này với mỗi yếu tố của ontology khác,

KHOA CNTT

và sau đó định nghĩa quan hệ tương đương trên mỗi một cặp cơ sở. Những item được liên kết có giá trị tương đương của nó lớn hơn một ngưỡng nào đó.

3.3.2. Lập chỉ mục ngữ nghĩa tiềm tàng:

Bây giờ chúng ta đề cập đến việc thực thi Latent Semantic Indexing (LSI – lập chỉ mục ngữ nghĩa tiềm tàng) có thể cải tiến những khả năng tìm kiếm ngày nay mà không có những giới hạn nghiêm trọng của mạng Web ngữ nghĩa rộng lớn.

Việc dựa vào tiêu chuẩn của độ chính xác, phẩm chất và sự thu hồi đòi hỏi nhiều hơn "sức mạnh cơ bắp". Gán các công cụ mô tả và phân loại cho văn bản cung cấp một thuận lợi quan trọng, bằng cách trả về các tài liệu không cần chứa liên kết theo từng chữ một cho truy vấn tìm kiếm của chúng ta. Các bộ dữ liệu được mô tả đầy đủ có thể cung cấp một bức tranh về phạm vi và sự phân tán của bộ sưu tập tài liệu nói chung. Điều này có thể được thực hiện bởi việc nghiên cứu cấu trúc của các danh mục và các danh mục con (được gọi là sự phân loại_ taxonomy).

Một trở ngại nghiêm trọng cho sự tiếp cận đến việc phân loại dữ liệu này là vấn đề vốn có trong bất cứ kiểu của taxonomy – trên thế giới đôi khi chống lại sự phân loại. Ví dụ, cà chua là trái cây hay rau quả?

Và điều gì xảy ra khi chúng ta kết nối hai tập tài liệu được chỉ mục trong những hướng khác nhau? Các giải pháp được gọi là các "ontology taxonomy" (phân loại ontology).

Các tìm kiếm từ khoá thông thường tiếp cận một tập tài liệu mà một tài liệu chứa hay không chứa một từ đưa ra.

Chỉ mục ngữ nghĩa tiềm tàng (LSI) thêm một bước quan trọng cho việc xử lí chỉ mục tài liệu. Thêm vào việc ghi những từ khoá mà một tài liệu chứa, phương pháp này khảo sát toàn bộ tập dữ liệu, để thấy những tài liệu khác chứa một số từ tương đương với các từ đó. LSI được phát triển đầu tiên ở Bellcore trong cuối những năm 80. LSI xem các tài liệu có nhiều từ thông dụng là có nghĩa, và xem những tài liệu ít từ thông dụng là có ít ngữ nghĩa. Mặc dù thuật giải LSI không hiểu tí gì về nghĩa của các từ, nó nhận ra các khuôn mẫu.

KHOA CNTT

Khi bạn tìm kiếm một cơ sở dữ liệu chỉ mục LSI, công cụ tìm kiếm này xem xét những giá trị tương tự mà nó tính toán cho mỗi từ của nội dung, và trả về các tài liệu mà nó nghĩ là thích hợp nhất với câu truy vấn. Bởi vì hai tài liệu có thể rất gần nghĩa với nhau thậm chí nếu chúng không cùng chung một từ khoá đặc biệt, LSI không yêu cầu một sự phân tích lấy tương xứng để trả về các kết quả hữu dụng. Ở những vị trí mà một tìm kiếm theo từ khoá đơn giản sẽ không thực hiện được nếu không có phân tích lấy tương xứng, thì LSI sẽ thường trả về những tài liệu liên quan mà không chứa tất cả những từ khoá đó.

3.3.2.1. Tìm kiếm lấy nội dung

Việc lập chỉ mục ngữ nghĩa tiềm tàng xem xét các mẫu từ trong một tập tài liệu. Ngôn ngữ tự nhiên có nhiều những từ không cần thiết, và không phải mỗi từ xuất hiện trong tài liệu đều chứa ngữ nghĩa. Các từ được sử dụng thường xuyên trong tiến Anh thường không chứa nội dung, ví dụ như các từ chức năng, liên từ, giới từ, và các động từ thường. Bước đầu tiên trong việc thực thi LSI là chọn lọc những từ xa lạ từ một tài liệu. Để thu được nội dung ngữ nghĩa từ một tài liệu:

1. Tạo một danh sách hoàn chỉnh tất cả các từ xuất hiện trong bộ sưu tập.
2. Lược bỏ các mạo từ, các giới từ, và các liên từ
3. Lược bỏ các động từ thông dụng (know, see, do, be…)
4. Lược bỏ các đại từ
5. Lược bỏ các tính từ thông dụng (big, late, high…)
6. Lược bỏ các từ "frilly" (therefore, thus, however, albeit,…)
7. Lược bỏ một số từ xuất hiện trong mọi tài liệu.
8. Lược bỏ các từ xuất hiện chỉ trong một tài liệu.

3.3.2.2. Stemming (lemmatize)

Công cụ tìm kiếm ngữ nghĩa là một giải pháp hiệu quả đáng chú ý. Nó có thể phát hiện được 2 tài liệu tương tự nhau thậm chí nếu chúng không có bất kỳ một từ

KHOA CNTT

nào chung và công cụ tìm kiếm ngữ nghĩa này có thể loại bỏ những tài liệu chỉ dùng chung những từ quan tâm một cách phổ biến.

Một số công việc khởi đầu cần thiết để thu thập tài liệu sẵn sàng cho việc lập chỉ mục thì rất đặc trưng ngôn ngữ, chẳng hạn như **stemming (lemmatize).** Đối với các tài liệu tiếng Anh, chúng ta sử dụng thuật toán được gọi là **The Porter Stemmer** để khử các phần đuôi thông thường của từ, để trả về dạng gốc của nó. (Ví dụ: writing → write, writes → write, …).

Việc đầu tiên là áp dụng đối với các tài liệu riêng biệt, và chúng ta gán cho nó một trọng số cục bộ. Các từ xuất hiện nhiều lần trong một tài liệu thì có trọng số lơn hơn những từ chỉ xuất hiện 1 lần.

Chúng ta đưa ra một giải thuật tạo ra trang web của các tài liệu và các từ – liên kết tất cả các tài liệu với các từ. Cho một mô hình các từ và các tài liệu, một người có thể thiết lập các giá trị dựa trên sự khác biệt của tài liệu so với các tài liệu khác. 'Giá trị' của một tài liệu bất kỳ so với các tài liệu khác có thể được thiết kế như là một hàm của số lượng các kết nối mà phải được thông qua để thiết lập một kết nối giữa các tài liệu. Nếu 2 tài liệu được liên kết với nhau bởi nhiều đường đi (đường kết nối) thì hai tài liệu này có thể có cùng một mức độ tương quan.

Trọng số của từ là **sự chuẩn hoá của 2 từ có nghĩa thông thường**:

- Các từ xuất hiện nhiều lần trong một tài liệu thì có nhiều ngữ nghĩa hơn từ chỉ xuất hiện một lần.
- Những từ được sử dụng thường xuyên thì có thể đáng quan tâm hơn những từ bình thường.

KHOA CNTT

<u>Mô tả giải thuật:</u>

Với mỗi tài liệu:

1. "Stem" (lược bỏ tiền tố và hậu tố) tất cá các từ và bỏ đi những từ có nghĩa thường xuyên xuất hiện.
2. Đối với mỗi từ:
   a. Đánh dấu lại mỗi tài liệu mà có mối quan hệ trực tiếp đến từ này.

b. Tính điểm cho mỗi tài liệu dựa trên hàm tính khoảng cách từ tài liệu xuất phát đến các mối quan hệ của từ.

3. Với mỗi tài liệu có mối quan hệ mới chưa được đánh dấu thì tiến hành lưu vết.

Lặp lại các thao tác như trên một cách đệ qui.

Giải thuật tính trọng số chi tiết được sử dụng như sau:

1. Đối với mỗi lần tăng khoảng cách, chia điểm số cho 2.

2. Điểm số cho mỗi tài liệu bằng với giá trị giới hạn chia cho căn bậc hai tính phổ biến của từ.

Toàn bộ thuật giải này đưa ra một cái nhìn ngữ nghĩa thấp dựa vào đường đi từ một tài liệu đến sơ đồ từ.

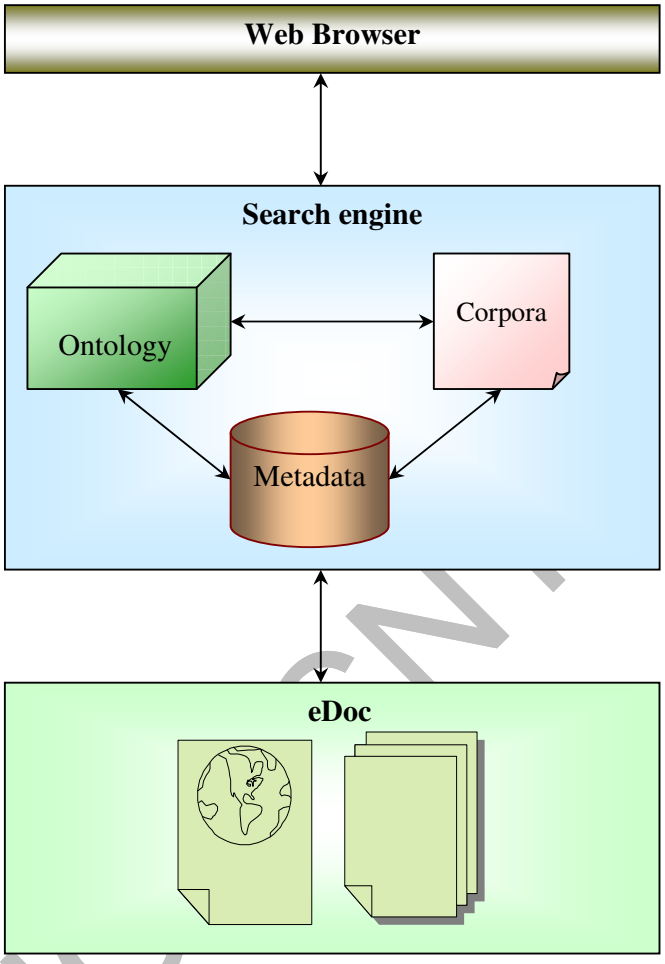
Chuẩn được trình bày ở đây là trường hợp đơn giản nhất và nó có thể được cải tiến theo nhiều cách khác nhau. Có nhiều giải thuật tính điểm khác có thể được sử dụng. Thêm vào đó, một từ điển đồng nghĩa có thể được áp dụng để giúp khắc phục các vấn đề ngữ nghĩa.

Một thử thách đáng quan tâm là làm cho giải thuật làm việc để mà khi các tài liệu mới được thêm vào chúng sẽ lập tức tự tính điểm. Một thách thức khác là tìm ra một cách mà có thể đưa giải thuật đến nhiều máy.

3.3. Mô hình đề nghị cho ứng dụng tìm kiếm ngữ nghĩa trên lĩnh vực eDoc

Từ những cơ sở lí thuyết đã nghiên cứu trên, chúng em tổng hợp lại và đề nghị mô hình cho ứng dụng tìm kiếm ngữ nghĩa trong lĩnh vực eDoc.

KHOA CNTT



Hình 16: Mô hình đề nghị cho ứng dụng tìm kiếm ngữ nghĩa trên lĩnh vực eDoc

- **Web Browser:**

Đóng vai trò giao diện giao tiếp với người dùng. Nó thực hiện vai trò tiếp nhận câu truy vấn của người dùng và hiển thị kết quả câu truy vấn.

- **Search engine:**

Đây là phần chính của chương trình. Search engine thực hiện tất cả các thao tác xử lí cần có của hệ thống:

- Đóng vai trò như web robot, thu thập tài liệu điện tử trên mạng.

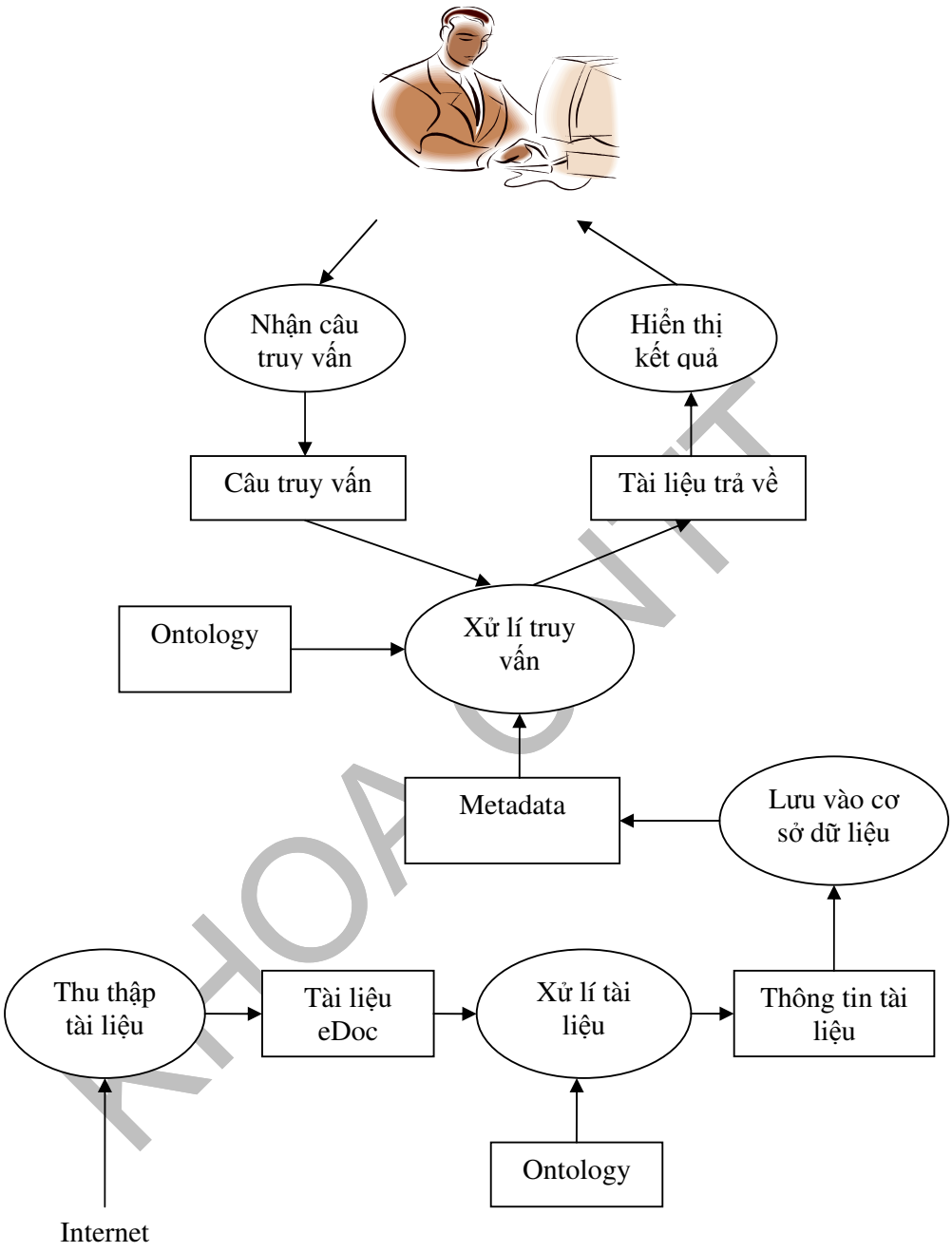
- Thực hiện như bộ lọc, search engine tiến hành thu thập, xử lí, rút trích siêu dữ liệu cho các tài liệu bằng cách phân tách từ, lược bỏ những từ không cần thiết chỉ giữ lại danh sách các danh từ và động từ, sau đó tiến hành thống kê tần số xuất hiện của các lĩnh vực trong tài liệu và cuối cùng lưu trữ siêu dữ liệu cho nội dung của tài liệu đó, sử dụng chuẩn siêu dữ liệu Dublin Core.
- Tổ chức và lưu trữ các Ontology cho mối quan hệ ngữ nghĩa giữa các đối tượng trong thực tế. Hình thức tổ chức, lưu trữ dạng tập tin RDF.
- Tổ chức và lưu trữ các kho ngữ liệu (corpora). Đây cũng được xem là một Ontology, biểu diễn mối quan hệ thành phần_bộ phận của đối tượng, đồng thời kho ngữ liệu cũng cho phép xác định các từ đồng nghĩa với nhau dựa vào khái niệm synset. (Chi tiết về các kho ngữ liệu được mô tả bên dưới). Sử dụng hình thức lưu trữ bảng trong SQL Server vì dữ liệu này có nhu cầu truy vấn cao.
- Thiết kế siêu dữ liệu để mô tả mối quan hệ giữa các tài nguyên (các tài liệu eDoc) với các đối tượng trong Ontology. Cũng sử dụng hình thức lưu trữ dạng cơ sở dữ liệu quan hệ.
- Thực hiện phân tích câu truy vấn của người dùng, lấy những từ quan trọng, từ đó phân tích ngữ nghĩa của câu truy vấn dựa vào Word Net và các Ontology đồng thời truy vấn các siêu dữ liệu để trả về cho Web Browser các tài liệu đúng với ngữ nghĩa câu truy vấn của người dùng.

- **eDoc**

Chỉ tất cả các tài liệu điện tử trên mạng, cụ thể là các file dạng HTML, PDF, CHM, ASP, PHP…

KHOA CNTT

Qui trình xử lí của tầng search engine:



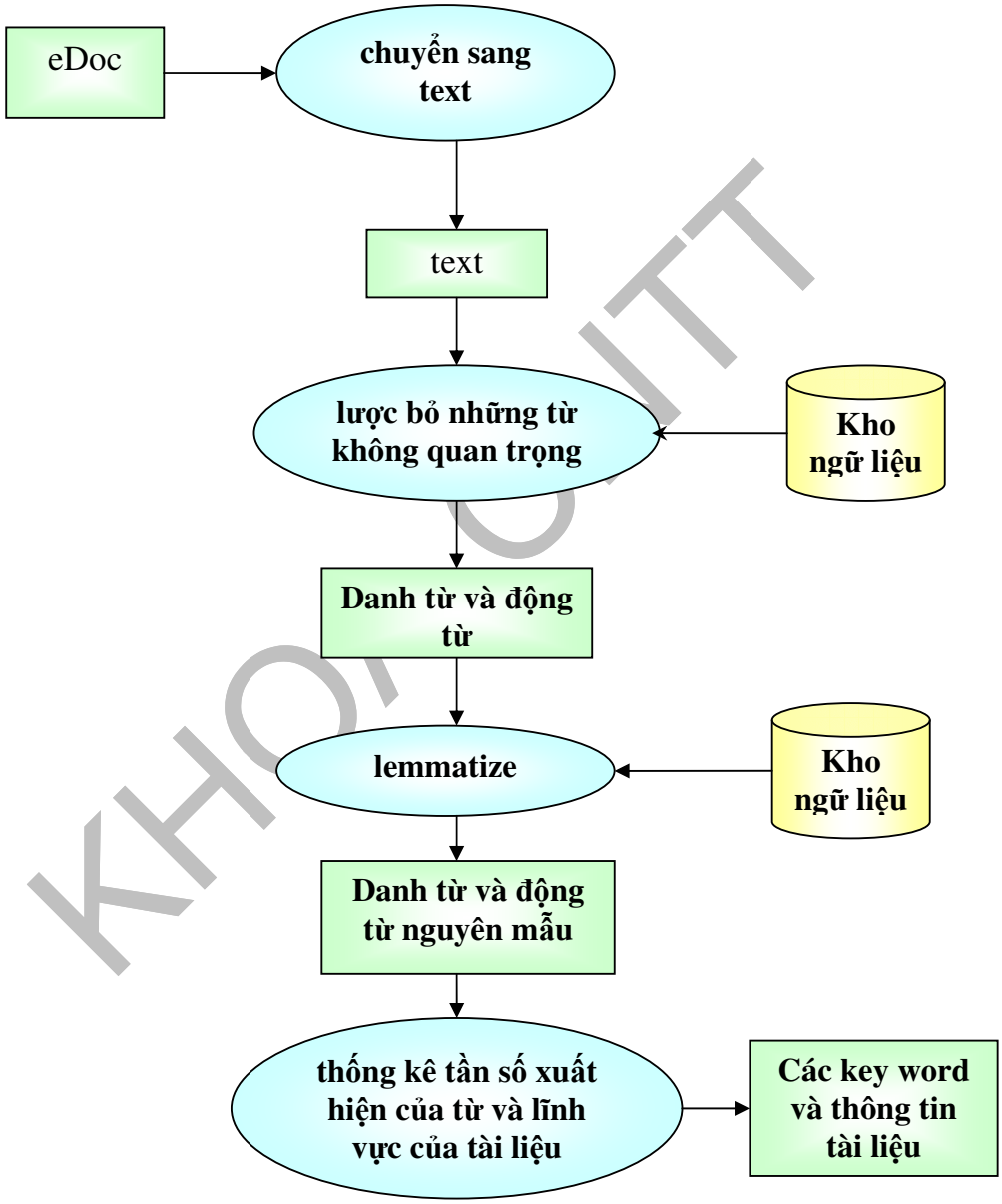
Hình 17: Qui trình xử lý của tầng search engine

KHOA CNTT

3.4. Các giải thuật sử dụng

3.4.1. Giải thuật xử lý tài liệu:

Tài liệu sau khi được thu thập về sẽ được xử lí thông qua bộ lọc. Sơ đồ giải thuật:



KHOA CNTT

Hình 18: Giải thuật xử lý tài liệu:

Giải thuật cho bước lemmatize:

Kho ngữ liệu sử dụng cho việc stemming là WORDNET vì số lượng từ trong kho ngữ liệu là khá lớn (với trên 100 000 danh từ và 11 000 động từ), các từ sử dụng ở dạng nguyên mẫu. Ngoài ra trong tự điển của WORDNET có file "noun.exc" và "verb.exc", đây là hai file để chuyển các danh từ dạng số nhiều bất qui tắc sang số ít và chuyển các động từ quá khứ và tiếp diễn dạng bất qui tắc về nguyên mẫu.

Các bước stemming đơn giản:

B1: Kiểm tra từng từ, nếu từ này có trong "noun.exc" hay "verb.exc" thì lấy dạng nguyên mẫu của nó.

B2: Nếu không có thì:

- Nếu từ này kết thúc bằng "**s**" thì: tiến hành bỏ "s" theo luật.
  - Nếu từ kết thúc bằng "ss", "chs", "shs", "xs", "is", "zs" thì đây không phải là số nhiều.
  - Nếu từ kết thúc là " 's " thì đây là dạng sở hữu cách nên bỏ hai kí tự này.
  - Bỏ kí tự 's' ở cuối từ.
  - Kiểm tra trong kho ngữ liệu danh từ và động từ, nếu có từ này thì đây là từ nguyên mẫu.
  - Nếu không có (nghĩa là từ này chưa ở dạng nguyên mẫu) thì:
    - Nếu từ kết thúc bằng "se", "che", "she", "xe", "ze" thì bỏ kí tự 'e' sau cùng.
    - nếu từ kết thúc bằng "ie" thì bỏ "ie" thêm "y".

- Nếu từ này kết thức bằng "**ed**" thì:
  - Bỏ "ed".
  - Kiểm tra trong kho ngữ liệu động từ, nếu có thì đây là dạng nguyên mẫu.

KHOA CNTT

- Nếu không có thì:
  - Nếu từ có hai kí tự cuối giống nhau thì bỏ một kí tự cuối.
  - Nếu từ kết thúc bằng "i" thì thay bằng "y".
  - Còn các trường hợp còn lại thì thêm vào cuối kí tự 'e'.

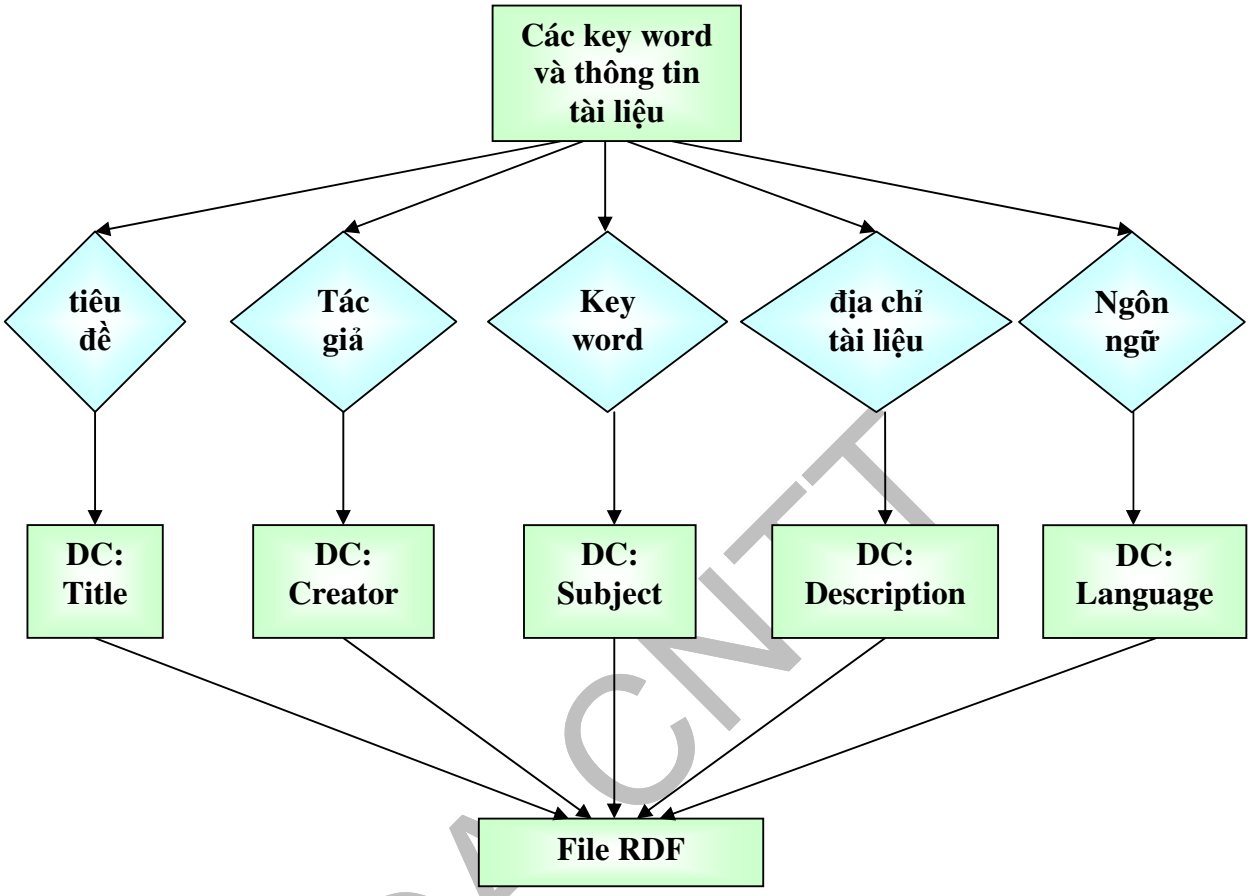
➢ Nếu từ này kết thức bằng "**ing**" thì:

- Bỏ "ing".
- Kiểm tra trong kho ngữ liệu động từ, nếu có thì đây là dạng nguyên mẫu.
- Nếu không có thì:
  - Nếu từ có hai kí tự cuối giống nhau thì bỏ một kí tự cuối.
  - Nếu từ kết thúc bằng "y" thì thay "y" bằng "ie".
  - Còn các trường hợp còn lại thì thêm vào cuối kí tự 'e'.

3.4.2. Giải thuật rút trích siêu dữ liệu:

Sau khi đã xử lí tài liệu để lấy các thông tin về tài liệu, chương trình xây dựng metadata để mô tả tài liệu đó. Metadata sử dụng chuẩn Dublin Core để mô tả và đưa về lưu trữ dạng RDF.

KHOA CNTT



Hình 19: Giải thuật rút trích siêu dữ liệu

Sử dụng các tag chính:

- title: mô tả tên tài liệu
- identifier: mô tả URI của tài liệu
- language: ngôn ngữ tài liệu
- description: mô tả thông tin tài liệu
- subject: các từ khoá cho tài liệu (một số trang HTML có thẻ meta này, kết hợp với một số từ thống kê được trong nội dung tài liệu).

KHOACNTT

Nội dung của các tag này chủ yếu được lấy trong phần HEAD của file HTML. Trừ tag identifier và subject được thêm vào từ thông tin nhận diện tài nguyên của robot và thông tin thống kê key word.

3.4.3. Giải thuật phân loại lĩnh vực cho tài liệu:

Một tài liệu, sau khi được rút trích thông tin ở phần header, sẽ được xử lí nội dung để phân loại lĩnh vực cho nó. Các lĩnh vực được đưa ra để phân loại chính là những lớp con (subclass) trong ontology. Và hình thức phân loại là sử dụng một tập các từ ứng với mỗi lớp con bao gồm các từ đồng nghĩa và các từ chi tiết hơn của lớp con đó, gọi là các từ chuyên ngành. Việc xây dựng tự điển các từ này dựa vào kho ngữ liệu WordNet và Tropes (công cụ phân loại văn bản).

Ví dụ, trong lĩnh vực "khoa học máy tính" thì có những lớp con như "máy tính" (computer), "lập trình" (programming)…. Và lớp con "máy tính" (computer) lại chứa các từ riêng của nó như: computing machine, hardware, CPU….

Các bước phân loại lĩnh vực:

B1: Dựa vào danh sách các từ chuyên ngành, tìm trong tài liệu và đếm số lần xuất hiện của nó, con số này được xem như là trọng số của từ trong tài liệu.

B2: Cộng các trọng số của từ trong từng lớp con để tính trọng số cho mỗi lớp con.

B3: Lớp con nào có trọng số cao nhất thì được xem là lớp tối ưu và tài liệu sẽ được xếp vào lớp con đó.

Và mối quan hệ giữa tài liệu với các lớp con sẽ được lưu trữ theo dạng chỉ mục Doc_Onto.

3.4.4. Giải thuật xử lí câu truy vấn:

Các bước phân tích lĩnh vực của câu truy vấn cũng được thực hiện tương tự như giải thuật phân loại lĩnh vực cho tài liệu. Từ việc phân tích đó, những tài liệu thuộc lĩnh vực tối ưu của câu truy vấn sẽ được đưa ra và xem như đó là kết quả trả về cho người dùng.

KHOA CNTT

Chương 4 : CHƯƠNG TRÌNH ỨNG DỤNG

4.1. Giới thiệu chương trình ứng dụng:

Trong chương này, chúng em xây dựng một công cụ tìm kiếm để minh hoạ cho việc tìm kiếm Web trên Internet có kết hợp với ngữ nghĩa. Mô hình xây dựng được hiện thực dựa trên cơ sở áp dụng và phát triển các mô hình Web ngữ nghĩa mà chúng em đã trình bày trong các chương trước.

Chương trình ứng dụng sẽ thực hiện việc tìm kiếm ngữ nghĩa thông qua các công nghệ Web ngữ nghĩa hiện có và các giải pháp mà chúng em đã đề xuất:

- Chương trình có sử dụng công cụ RDF Gateway.
- Thi hành trên I.E5.
- Chương trình có sử dụng công cụ RDF editor.

4.2. Kiến trúc của ứng dụng:

Để thiết kế công cụ tìm kiếm ngữ nghĩa ứng dụng trên eDoc, chúng em đề xuất một **kiến trúc mô hình** hỗ trợ việc tìm kiếm trên Internet và Intranet gồm các công đoạn sau:

❖ **Công đoạn 1: Thiết kế ontology.**

Các Ontology thường lưu dưới dạng tập tin có đuôi: .rdf, .rdfs, .owl, .daml, .xml, ….

Ontology mô tả mối quan hệ giữa các đối tượng trong thực tế. Ontology do các chuyên gia về các lĩnh vực đã được tạo sẵn, để sẵn trên Internet. Đặc tính của các Ontology này là cho phép mọi người có thể chia sẻ, tạo, đọc và ghi trên nó. Do đó, chúng ta có thể phát triển Ontology theo ý muốn.

Các Ontology cũng được tạo từ những tập tin cấu trúc dạng: HTML, RDF, Image, Excel, WinWord, SQL Server, Oracle, …. Các Ontology này sẽ được tạo ra thông qua một công cụ soạn thảo, sau đó chúng sẽ được lưu dưới dạng tập tin có đuôi: .rdf, .rdfs, .owl, .daml, ….

KHOA CNTT

Các công cụ có thể dùng để soạn thảo Ontology là:

- Sử dụng HTML Parser.
- Protégé
- RDF Editor
- ….

❖ **Công đoạn 2: Xây dựng ứng dụng.**

Các bước chính trong quá trình xây dựng ứng dụng:

- Bước 1: Dùng các phần mềm như Crawlers, Spiders, … đóng vai trò là các robot thu thập thông tin trên internet, cũng như là để thu thập các Ontology từ trên internet.
- Bước 2: Dùng tiện ích RDF Query Analyzer trong phần mềm RDF Gateway để đưa các file Ontology(thu được ở Bước 1) vào cơ sở dữ liệu của RDF Gateway.
- Bước 3: Xây dựng ứng dụng:
  - Tiến hành phân loại Ontology (đã thu được) theo những lĩnh vực cần tìm.
  - Tài liệu sau khi đã thu thập (ở Bước 1), tiến hành rút trích siêu dữ liệu với các thành phần quan tâm: title, author, keyword, subject, description, …. Rồi phân loại tài liệu theo lĩnh vực.
  - Siêu dữ liệu rút trích được sẽ được đưa xuống cơ sở dữ liệu SQL Server. Đồng thời cũng xây dựng mối quan hệ giữa các đối tượng trong Ontology với siêu dữ liệu rút trích.
  - Với truy vấn người dùng nhập vào, vào cơ sở dữ liệu tiến hành truy vấn và trả ra kết quả cho người dùng.

KHOA CNTT

4.3. Mô tả phạm vi ứng dụng

4.3.1. Mô tả bài toán:

Trong ứng dụng này, chúng em tích hợp các Ontology (lấy từ internet) vào một thư mục ở máy cục bộ để tiện cho việc minh hoạ ứng dụng. Tuy nhiên, ta có cũng có thể lấy các ontology này trực tiếp từ internet. Các ontology được lưu vào localhost:

http://localhost/eDocSearch/Library/RDF/

Ở đây chỉ sử dụng những ontology cho từng lĩnh vực nhất định, nếu một lĩnh vực có nhiều ontology hoặc một ontology ứng dụng cho nhiều lĩnh vực thì ta phải tiến hành phân loại ontology theo lĩnh vực (đây là hướng mở rộng của luận văn).

Ứng dụng được xây dựng nhằm minh họa cho việc tìm kiếm ngữ nghĩa trên lĩnh vực edoc, phạm vi ứng dụng giới hạn trong lĩnh vực như sau:

- Khoa học máy tính (computer scient).
- Nghệ thuật (art) .

4.3.2. Xác định yêu cầu:

- **Yêu cầu lưu trữ:**

  Lưu thông tin ngữ nghĩa cần tìm (các đối tượng) từ các ontology vào trong CSDL, thông tin mô tả các thuật ngữ tương đương hỗ trợ cho việc tìm kiếm.

- **Yêu cầu tra cứu:**

  Tìm kiếm các tài liệu liên quan đến thuật ngữ mà người dùng gõ vào.

- **Tính hiệu quả:**

  Kết quả tìm kiếm phải phù hợp, chính xác, nhanh chóng theo công nghệ Semantic Web.

- **Tính tiến hoá:**

  Các tài liệu hỗ trợ nhiều tài liệu hơn, nhiều lĩnh vực hơn, ….

- **Tính tương thích:**

  Người dùng chỉ cần một trình duyệt web và kết nối được đến server.

KHOA CNTT

- **Tính tiện dụng:**

  Giao diện thân thiện, dễ sử dụng, người dùng chỉ cần gõ vào một thuật ngữ cần tìm kiếm rồi nhấn vào nút Search.

- **Tính bảo mật:**

  Người dùng chỉ xem được kết quả tra cứu dưới dạng tĩnh (htm/html).

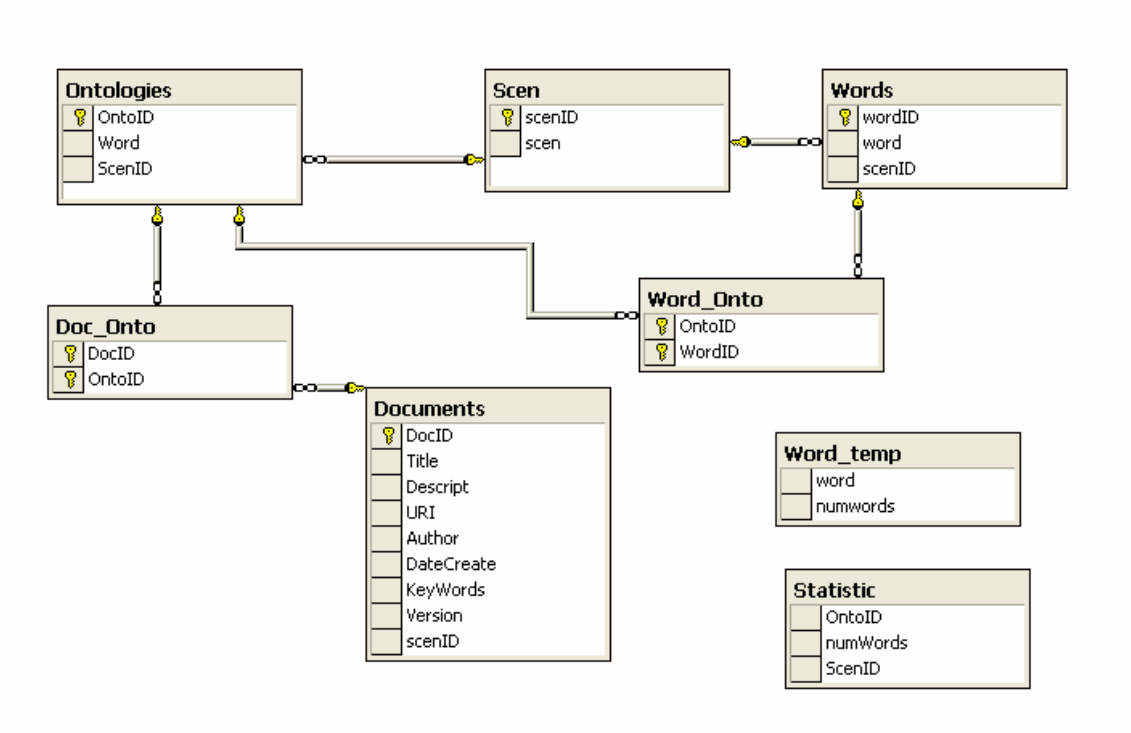
- **Tính dễ bảo trì:**

  Dễ dàng phát triển hay thêm các ontology thuận lợi.

4.4. Xây dựng ứng dụng:

4.4.1. Thiết kế dữ liệu:

Dữ liệu được lưu trữ trong SQL Server 2000. Bao gồm các bảng:



Hình 20: Sơ đồ dữ liệu quan hệ của ứng dụng

Tên bảng	Các trường	Mô tả
DOCUMENTS	DocID varchar(12) Title text Descript text URI varchar(200) Author varchar(200) Datacreate varchar(12) Keywords text Version varchar(50) ScenID char(3)	Bảng lưu trữ thông tin của các tài liệu cùng với lĩnh vực mà tài liệu đó thuộc về.
ONTOLOGIES	OntoID varchar(12) Word varchar(50) ScenID char(3)	Bảng lưu trữ thông tin các đốn tượng của ontology.
DOC_ONTO	DocID varchar(12) OntoID varchar(12)	Mối quan hệ giữa tài liệu và các đối tượng của ontology
WORDS	WordID varchar(10) Word varchar(50) ScenID char(3)	Có thể xem đây là danh sách các từ có thể có trong một lĩnh vực.
WORD_ONTO	WordID varchar(10) OntoID varchar(12)	Các từ tham chiếu đến một đối tượng của Ontology
STATISTIC	OntoID varchar(12) NumWords int ScenID char(3)	Đây là bảng tạm dùng để lưu trữ số từ tìm thấy trong tài liệu ứng với một đối tượng trong Ontology. Bảng này sử dụng để phân loại tài

KHOA CNTT

		liệu theo một lĩnh vực.
WORD_TEMP	Word varchar(50) Numwords int	Đây cũng là một bảng tạm nhằm lưu các từ có trong tài liệu ứng để sau này lấy các key word cho tài liệu.

Bảng 6 Mô tả cơ sở dữ liệu cho ứng dụng

Đặc biệt bảng Ontology được xây dựng từ những tài liệu RDF. Sử dụng RDF gateway để truy vấn và cache dữ liệu vào bảng này giúp tìm kiếm nhanh chóng và dễ dàng hơn.

4.4.2. Thiết kế xử lý:

Chương trình sử dụng ngôn ngữ lập trình C# kết hợp với ASP.NET.
Sử dụng SQL Server 2000 để lưu trữ dữ liệu.

Chương trình có 2 module:

STT	Module	Ý nghĩa
1	eDocSearch	Thực hiện giao tiếp với người dùng, tiếp nhận câu truy vấn, xử lí câu truy vấn, và hiển thị kết quả cho người dùng.
2	eDocSearchAdministrator	Quản lí cơ sở dữ liệu các từ, các ontology, các tài liệu. Thu thập tài liệu từ Internet, và xử lí tài liệu.

Bảng 7 Các module của chương trình

KHOA CNTT

Các lớp đối tượng cho từng module:

- **Module eDocSearch:**

STT	Lớp đối tượng	Ý nghĩa
1	UserQuery.cs	Có trách nhiệm xử lí câu truy vấn của người dùng, và trả ra kết quả cho câu truy vấn.

Bảng 8 Module eDocSearch

- **Module eDocSearchAdministrator:**

STT	Lớp đối tượng	Ý nghĩa
1	Database.cs	Thực hiện kết nối cơ sở dữ liệu SQL server và RDF gateway.
2	Spider.cs	Thu thập tài liệu từ Internet
3	DocumentProcess.cs	Quản lí cơ sở dữ liệu tài liệu (rút trích metadata cho tài liệu, phân loại lĩnh vực cho tài liệu).
4	TextProcess.cs	Có trách nhiệm xử lí văn bản (lược bỏ các từ không quan trọng, thực hiện "lemmatize")
5	Word_database.cs	Quản lí cơ sở dữ liệu các từ chuyên ngành cho từng lĩnh vực.
5	ManageOntology.cs	Quản lí cơ sở dữ liệu Ontology
6	DatabaseProcess.cs	Xử lí Ontology, chuyển từ dạng lưu trữ RDF sang cơ sở dữ liệu quan hệ SQL server.

Bảng 9 Module eDocSearch

KHOA CNTT

4.5. Kết quả chương trình

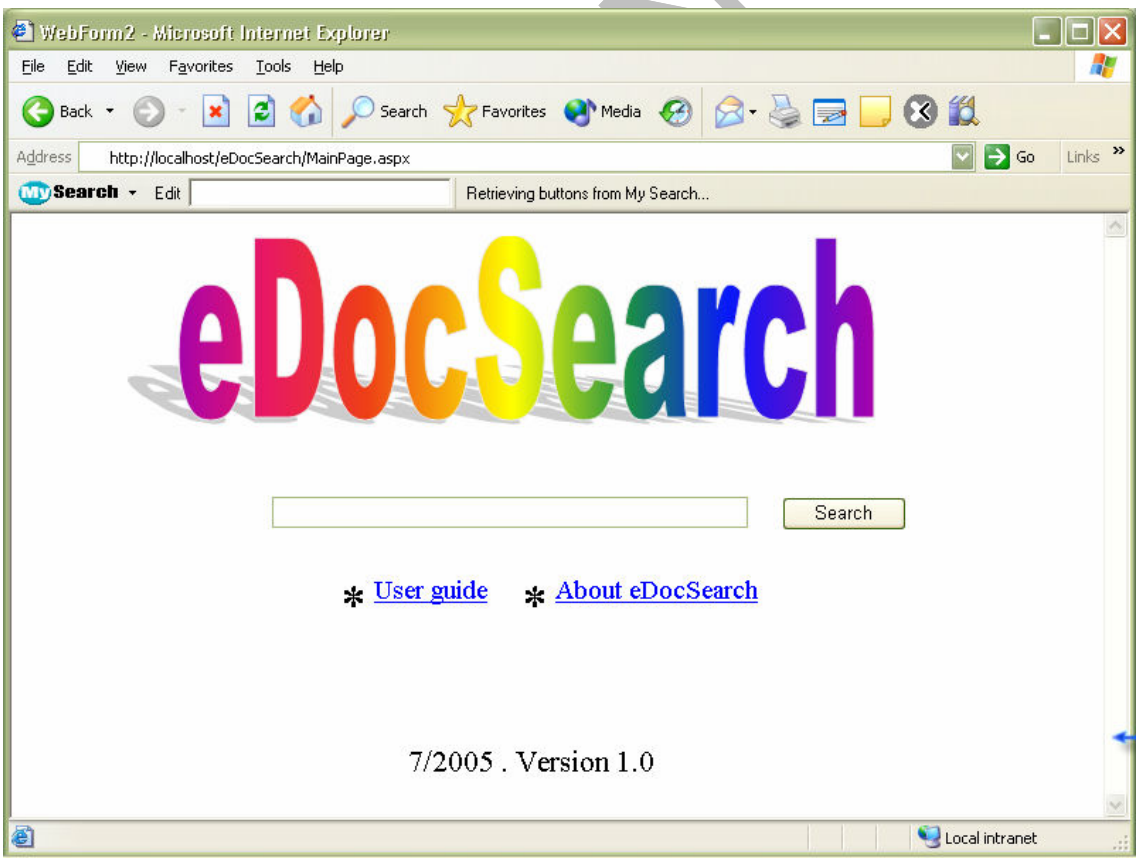
Tài liệu cho việc tìm kiếm thử nghiệm được download về và lưu trong máy chủ ở thư mục http://localhost/eDocSearch/DataTest/. Số lượng tài liệu khoảng 500 tài liệu cho cả hai lĩnh vực.

Môi trường ứng dụng: Máy Celeron, 256 MB RAM, 1.2 GB, hdh Windows XP.

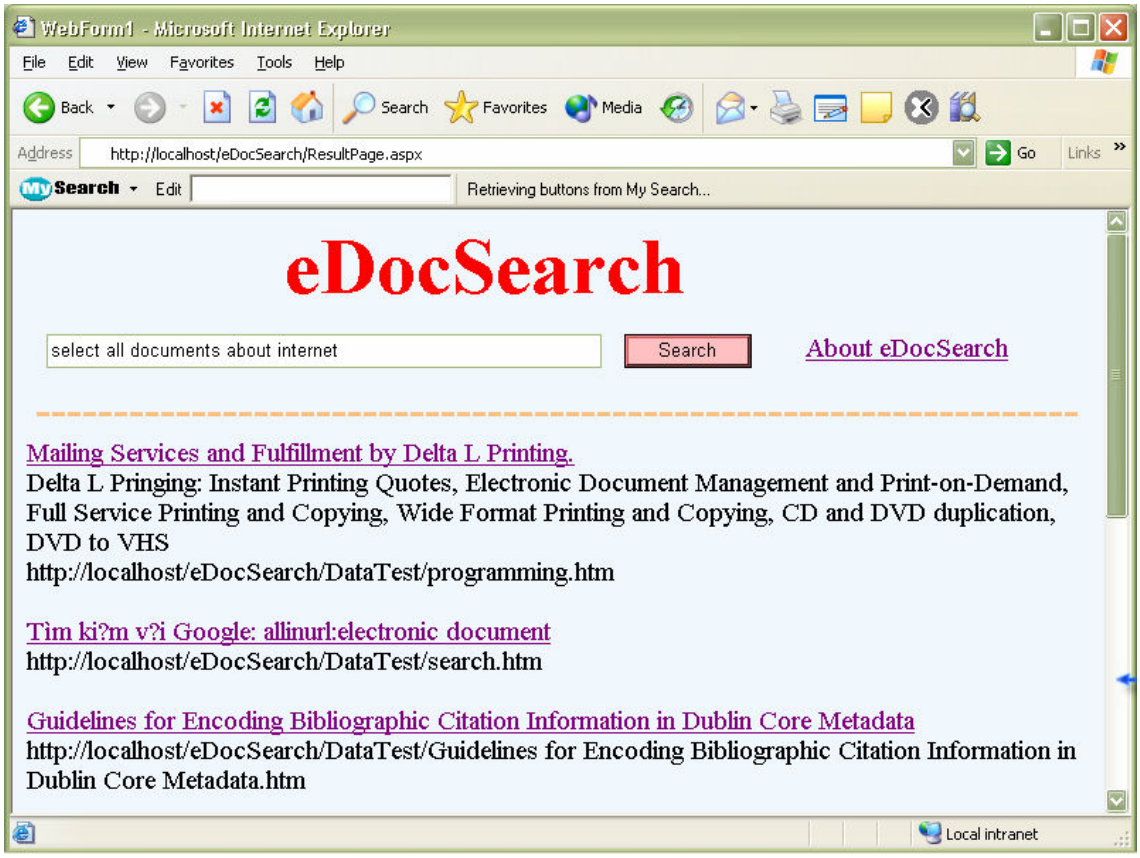
- Thời gian xử lý văn bản ~ 2s/tài liệu
- Thời gian xử lý truy vấn nhanh.
- Phân loại văn bản theo lĩnh vực: 91%

Chương trình cho phép người dùng truy vấn những vấn đề quan tâm bằng ngôn ngữ tự nhiên.

Giao diện chính của chương trình:

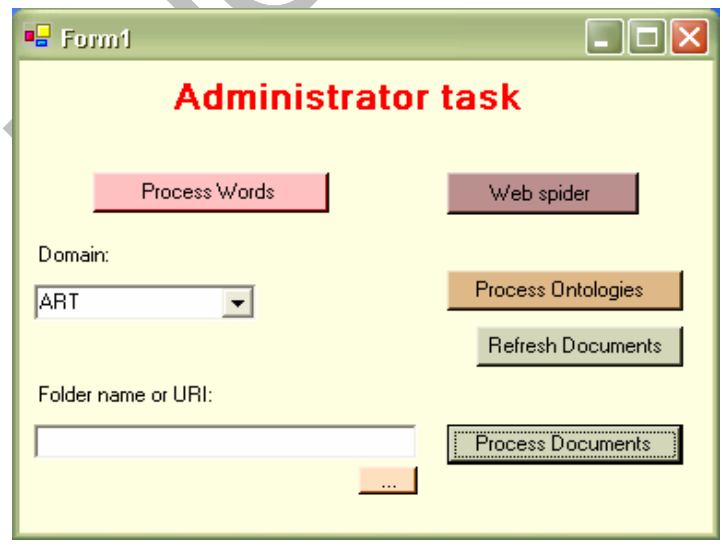


Hình 21: Giao diện chính của ứng dụng



Hình 22: Giao diện kết quả tìm kiếm của ứng dụng

Giao diện quản lí tài nguyên:



Hình 23: Giao diện quản lí tài nguyên

4.6. Thực nghiệm chương trình

- **Danh sách các câu truy vấn thử nghiệm chương trình:**

STT	Từ truy vấn	Số tài liệu trả về	Số tài liệu không đúng nội dung
1	Programming	14	3
2	Oop	10	1
3	Asp	10	1
4	Assembly	9	2
5	Java	12	3
6	Visual basic	3	0
7	C#	10	1
8	Data	7	3
9	Database	76	33
10	Metadata	32	14
11	Register	0	0
12	Security	5	1
13	Computer science	63	25
14	Computing	47	17
15	Algorithm	45	9
16	Machine translation	52	17
17	Computer vision	62	27
18	Internet	46	6
19	www	43	18
20	Site	43	18
21	Server	57	22
22	Computer	29	24
23	Hardware	11	7

KHOA CNTT

24	Information processing	9	7
25	Natural language processing	10	8
26	Sofrware	12	6
27	Freeware	7	2
28	Shareware	7	2
29	Virus	6	0
30	Norton antivirus	5	0
31	Graphic	5	3
32	Picture	9	7
33	Artwork	15	7
34	Art school	100	90
35	Artist	12	3
36	Gallery	19	17
37	Museum	19	8
38	Clip art	100	90
39	Painting	36	27
40	Landscape	11	6
41	Portrait	10	7

Bảng 10 Các câu truy vấn thử nghiệm

KHOA CNTT

- **Kết quả thống kê truy vấn theo từng lĩnh vực**:

Công thức thống kê:

Độ chính xác của lĩnh vực = trung bình cộng(phần trăm chính xác của từng từ trong lĩnh vực đó).

❖ Computer & information science:

STT	Tên lĩnh vực	Độ chính xác
1	Programming	87%
2	Data	57%
3	Security	93%
4	Computer science	65%
5	Internet	67%
6	Computer	26%
7	Information science	21%
8	Software	64%
9	Virus	100%

Bảng 11 Thống kê lĩnh vực khoa học máy tính

❖ Art:

STT	Tên lĩnh vực	Độ chính xác
1	Art and artwork	10%
2	Artist	75%
3	Gallery	11%
4	Museum	58%
5	Art school	10%
6	Painting	25%
8	Music	70%
9	Music style	65%

Bảng 12 Thống kê lĩnh vực nghệ thuật.

KHOA CNTT

➢ **Nhận xét:**

- Ứng dụng chỉ xây dựng trên hai lĩnh vực là nghệ thuật và khoa học máy tính nên mọi tài liệu đưa vào đều được phân vào một trong hai lĩnh vực này do đó làm giảm đi độ chính xác.
- Số tài liệu trả về cho mỗi từ trong cùng một lớp con trong ontology là không bằng nhau do phương pháp xử lí câu truy vấn là: lấy những tài liệu trong cùng lớp con của ontology và đồng thời lấy những tài liệu có từ khoá có trong với từ khoá của câu truy vấn.
- Độ chính xác trong việc phân loại tài liệu theo từng lớp con chưa cao do các lớp con trong ontology thiết kế chưa đầy đủ, chưa bao hàm hết các khái niệm trong một lĩnh vực và số từ trong một lĩnh vực chưa nhiều và đầy đủ.
- Mặt khác, độ chính xác trong việc phân loại của tài liệu còn bị ảnh hưởng do số lượng từ của nội dung trong tài liệu ít (tài liệu chỉ chứa đa số là các hyperlink và các hình ảnh).
- Lĩnh vực nghệ thuật có độ chính xác thấp do các từ trong mỗi lớp con của ontology không được phân biệt rõ ràng, một từ có thể nằm ở nhiều lớp và số lượng từ ít.

Tóm lại, chương trình ứng dụng đạt hiệu quả tốt trong việc phân loại tài liệu theo lĩnh vực lớn, còn đối với từng lớp con trong mỗi lĩnh vực thì hiệu quả chưa cao. Người quản trị có thể nâng cao hiệu quả của chương trình bằng cách xây dựng tất cả các lĩnh vực trong thực tế, bổ sung các từ trong từng lớp con của mỗi lĩnh vực theo xu hướng càng nhiều từ đặc trưng cho lớp càng tốt (mức cô lập giữa các lớp càng cao).

KHOA CNTT

Chương 5 : KẾT LUẬN

5.1. Đánh giá kết quả nghiên cứu

5.1.1. Ưu điểm

Về cơ bản luận văn đã thực hiện tốt các nội dung đề ra và đạt được một số kết quả nhất định :

- Luận văn đã trình bày cơ sở lý thuyết về nguyên lý vận hành cũng như ưu và khuyết điểm của một hệ thống search engine.
- Luận văn trình bày rõ mô hình Web ngữ nghĩa cùng với các đối tượng của nó như RDF, OWL, …
- Trình bày các vấn đề về ngữ nghĩa cũng như các hướng giải quyết trong việc xử lí ngôn ngữ tự nhiên nhằm giúp máy tính "hiểu" được câu hỏi của người dùng.
- Từ những cơ sở nghiên cứu lí thuyết, luận văn đã đề ra mô hình cho việc xây dựng công cụ tìm kiếm ngữ nghĩa, và thực hiện cài đặt một công cụ tìm kiếm các tài liệu điện tử phù hợp với ngữ nghĩa của câu truy vấn của người dùng.
- Luận văn có thể xác định tương đối chính xác lĩnh vực mà tài liệu thuộc về. Và phần nào xác định được lĩnh vực của câu truy vấn của người dùng.

❖ Ý nghĩa thực tiễn:

➢ Tìm hiểu mô hình, nắm vững công nghệ tìm kiếm ngữ nghĩa để áp dụng chi tiếng Việt.

❖ Ý nghĩa khoa học:

➢ Đây là công cụ phục vụ cho nhu cầu phân loại văn bản, phân loại tài liệu học tập.

KHOA CNTT

5.1.2. Khuyết điểm:

Tuy nhiên, do vấn đề về ngữ nghĩa là một vấn đề phức tạp và rộng lớn nên luận văn chỉ đề ra một số hướng nghiên cứu hiện nay ở một số lĩnh vực hữu hạn, không thể bao hàm hết được các khái niệm cũng như ngôn ngữ của con người.

Những vấn đề được đề xuất trong luận văn nhằm mục đích đưa ra một hướng giải quyết mang tính chất tham khảo nên có thể sẽ có nhiều điểm chưa tối ưu, cần được hoàn thiện hơn.

Trong chương trình ứng dụng, luận văn sử dụng cơ sở dữ liệu các từ đặc trưng cho từ lĩnh vực, cơ sở dữ liệu này được xây dựng chủ yếu dựa vào WordNet, song vẫn còn hạn chế về số lượng các từ riêng cho từng chuyên ngành. Nếu câu truy vấn của người dùng hỏi về những từ không nằm trong cơ sở dữ liệu thì có thể sẽ không tìm thấy kết quả. Và việc phân loại các từ lĩnh vực mang tính chủ quan nên có thể chưa tối ưu.

Việc phân loại tài liệu theo lĩnh vực tương đối tốt do có số lượng từ khá nhiều nhưng việc phân loại câu truy vấn của người dùng, sử dụng một lượng từ rất ít nên có một số câu truy vấn không có kết quả trả về.

Ngoài ra, luận văn chỉ sử dụng cơ sở dữ liệu các tài liệu lưu sẵn về trên máy chủ nên số lượng các tài liệu chưa lớn.

5.2. Hướng phát triển

Chương trình ứng dụng của luận văn được xây dựng dựa trên những vấn đề cơ bản, song nó có thể phát triển để ngày càng hoàn thiện và tối ưu hơn. Những hướng phát triển của luận văn:

- Mở rộng tìm kiếm trong tất cả các lĩnh vực.
- Tìm kiếm trên nhiều ontology, phân loại ontology.
- Thực sự tìm kiếm online.
- Ứng dụng cho Tiếng Việt.

KHOA CNTT

TÀI LIỆU THAM KHẢO

I. Luận văn, luận án:

[I.1] Đặng Thị Quỳnh Chi. Luận văn thạc sĩ tin học. **Nghiên cứu về mô hình, khám phá và khai thác các mối quan hệ trên web ngữ nghĩa, xây dựng ứng dụng.** Người hướng dẫn khoa học: Nguyễn Tiến Dũng.

[I.2] Lê Thuý Ngọc, Đỗ Mỹ Nhung. Luận văn cử nhân tin học. **Tìm hiểu về Search Engine và xây dựng ứng dụng minh hoạ cho Search Engine tiếng Việt.** GVHD: Nguyễn Thị Diễm Tiên.

II. Sách, eBooks:

[II.1] Ying Ding, Dieter Fensel, Michel Klein, and Borys Omelayenko. The Semantic Web: Yet another Hip?**.** Data and knowedgle engineering, 2002.

[II.2] Eero Hyvonen. Semantic web Kick – off in Finland vision, Technologies, Research, and Applications; May 19, 2002 .

[II.3] Đinh Điền, Giáo trình Xử Lý Ngôn Ngữ Tự Nhiên, tháng 12/2004.

[II.4] Dr. V. Richard Benjamins, Jesús Contreras; Six challenges for the semantic web; April 2002.

[II.5] Nicola Guarino; Some Ontological Principles for Designing Upper Level Lexical Resources; 28 – 30 May 1998.

[II.6] Urvi Shah, Tim Finin, Anupam Joshi, R. Scott Cost, James Mayfield; Information Retrieval on the Semantic Web*.

[II.7] Luke K. McDowell; Meaning for the Masses: Theory and Applications for Semantic Web and Semantic Email Systems; 2004.

[II.8] Gareth Osler;The Semantic Web Through Semantic Data – A Four Tier Architecture Model ; 4 Mar 2005.

KHOA CNTT

[II.9] Julius Stuller; Network of Excellence Semantic Web; 7 June 2002.

[II.10] Peter Dolog and Wolfgang Nejdl; Challenges and Benefits of the Semantic Web for User Modelling.

[II.11] Pang Wang; A Search Engine Based on the Semantic Web; May, 2003.

[II.12] Karen Sparck Jones; What's new about the Semantic Web? Some questions; December 2004, 18 – 23.

[II.13] Mark Klein, Abraham Bernstein; Searching for Services on the Semantic Web Using Process Ontology; July 30 – August 1, 2001.

[II.14] Michael Sintek, Stefan Decker; TRIPLE – A Query Language for the Semantic Web; November 2 2001.

[II.15] Stefan Decker, Vipul Kashyap; The Semantic Web: Semantics for Data on the Web; September 10 2003.

[II.16] Catherine C. Marshall; Taking a Stand on the Semantic Web; 2003.

[II.17] Eric Miller, Ralph Swick; Semantic Web Activity: Adcanced Development; 07/09/2003.

[II.18] Tim Berners – Lee; Semantic Web Road map; 10/14/1998.

[II.19] Raul Corazzon; Ontology. A resource guide for philosophers; 06/01/2005.

[II.20] John F.Sowa; Guided Tour of Ontology; June 03 2005.

[II.21] John F. Sowa; Building, Sharing, and Merging Ontologies; June 03 2005.

[II.22] ISO; Information and documentation – The Dublin Core metadata element set; 02/26/2003.

[II.23] IEEE; Draft Standard for Learning Object Metadata; 15 July 2002.

[II.24] Shigeo SUGIMOTO, Jun ADACHI, Stuart WEIBEL; 68th IFLA Council and General Conference; August 24 2002.

[II.25] Stiching SURF; DARE use of Dublin Core, version 2.0; December 2004.

KHOA CNTT

[II.26] CEN/ISSS MII – DC (WI3) Report; Guidance for the Deployment of Dublin Core Metadata in Corporate Environments; 8/20/2004 DRAFT.

[II.27] Kazuhiko Asou, Takako Nakahara, Takao Namiki; A report on Dublin Core based research information service on mathematics; 10/26/2001.

[II.28] Western States Digital Standards Group, Metadata Working Group; Western States Dublin Core Metadata Best Practices, Version 2.0; 01/12/2005.

[II.29] Jay Cross, CEO, Internet Time Group; eLearning; mid – 1999.

[II.30] ADOBE; A primer on electronic document security; 11/2004.

[II.31] Gerhard U. Bartsch; Introduction to Electronic Document Management Whitepaper ; March 16 2003.

[II.32] Andreas Hotho; Using Ontologies to Improve the Text Custering and Classification Task; January 14 2005.

[II.33] Norman Paskin; DOI: implementing a standard digital identifier as the key to effective digital rights management; March 9 2000.

III. Website:

[III.1] W3C SemanticWeb Activity http://www.w3.org/2001/sw

[III.2] Semantic web server http://www.semanticwebserver.com

[III.3] RDF http://www.w3.org/RDF

[III.4] Tim Berners – Lee *Notation3* http://www.w3.org/DesignIssues/Notation3.html

[III.5] http://www.cimtech.co.uk

[III.6] http://www.adobe.com/security

[III.7] RDQL: RDF Data Query Language http://www.htl.hp.com/semweb/rdql.html

[III.8] RDF/XML Syntax Specification http://www.w3.org/TR/rdf-syntax-grammar/

[III.9] DAML http://www.daml.org

[III.10] RDF Data http://www.rdfdata.org

KHOA CNTT

[III.11] National Information Standards Organization http://www.niso.org

[III.12] Intellidimension: Delivering a Platform for the Semantic Web http://www.intellidimension.com/

[III.13] eLib http://purl.org/metadata/dublin_core.

KHOA CNTT

PHỤ LỤC

1. Cú pháp RDF:

rdfs:Resource

Tất cả mọi thứ được mô tả bởi RDF được gọi là resources và là thành viên của class rdfs:Resource

rdfs:Literal

Lớp rdfs:Listeral đại diện cho một lớp các giá trị ký tự như là strings và intergers. Ví dụ: thuộc tính giá trị: chuỗi text

rdfs:XMLLiteral

Lớp rdfs:XMLLiteral đại diện cho lớp giá trị chuỗi của XML.

rdfs:Class

Lớp này tương ứng với khái niệm chung type hoặc là catalog của tài nguyên.

RDF class membership (quan hệ thành viên lớp RDF) được sử dụng để đại diện cho types và catalog của tài nguyên. Hai lớp có thể có cùng thành viên.

rdf:Property

rdf:Property đại diện cho những tài nguyên có thuộc tính RDF.

rdfs:Datatype

rdfs:Datatype đại diện cho những tài nguyên có các kiểu dữ liệu RDF.

rdf:type

Thuộc tính rdf:type cho biết một tài nguyên là thành viên của class nào.

Khi một tài nguyên có một thuộc tính rdf:type mà giá trị của thuộc tính này là một số class xác định, thì chúng ta nói rằng tài nguyên là một *instance of* của class xác định này.

KHOA CNTT

Giá trị của thuộc tính rdf:type sẽ luôn là một tài nguyên – tài nguyên này là một thể hiện (instance) của rdfs:Class. Tài nguyên này được biết như là rdfs:Class bản thân nó là một tài nguyên của một rdf:type rdfs:Class. (Bản thân nó cũng là một kiểu – type của một lớp).

rdfs:subClassOf

Thuộc tính rdfs:subClassOf đại diện cho mối quan hệ chuẩn hoá giữa các class của một tài nguyên. Thuộc tính rdfs:subClassOf là một transitive.

rdfs:subPropertyOf

Thuộc tính rdfs:subPropertyOf là một thể hiện (instance) của rdf:Property, được sử dụng để xác định một thuộc tính là một chuẩn của một cái khác.

Hệ thống cấp bậc thuộc tính con có thể được sử dụng để trình bày hệ thống cấp bậc của các ràng buộc về *range* và *domain*.

Chú ý: Thuật ngữ "super – property" đôi khi được sử dụng để cho biết mối quan hệ giữa một số thuộc tính với nhiều thuộc tính phổ biến khác, ví dụ là mối quan hệ rdfs:subPropertyOf.

rdfs:range

Một thể hiện của rdf:Property được sử dụng để cho biết các class nào mà giá trị của một thuộc tính sẽ là thành viên của nó.

Giá trị của một thuộc tính rdfs:range luôn luôn là một Class. Thuộc tính rdfs:range bản thân nó có thể được sử dụng để biểu diễn điều này: The *rdfs:range* of *rdfs:range* is the class *rdfs:Class.* Điều này cho thấy rằng bất kỳ một tài nguyên nào là giá trị của thuộc tính range sẽ là một class.

Thuộc tính rdfs:range chỉ được áp dụng đối với các thuộc tính. Điều này cũng được miêu tả trong RDF thông qua việc sử dụng thuộc tính *rdfs:domain*. The *rdfs:Domain* of *rdfs:range* is the class *rdf:Property.* Điều này cho thấy rằng thuộc tính *range* áp dụng đối với các tài nguyên mà bản thân nó cũng là các thuộc tính (property).

rdfs:domain

Một thể hiện của *rdf:Property* được sử dụng để cho biết class nào sẽ có thành viên là bất kỳ một tài nguyên nào sao cho thuộc tính của nó được chỉ định.

KHOA CNTT

The *rdfs:domain* of *rdfs:domain* is the class *rdf:Property*. Điều này cho thấy rằng thuộc tính domain được sử dụng trên các tài nguyên là các thuộc tính.

The *rdfs:range* of *rdfs:domain* is the class *rdfs:Class.* Điều này cho thấy rằng bất kỳ một tài nguyên nào mà là giá trị của một thuộc tính domain sẽ là một class.

rdfs:label

Thuộc tính rdfs:label được sử dụng để cung cấp phiên bản tên của tài nguyên mà con người có thể đọc được.

rdfs:comment

Thuộc tính rdfs:comment được sử dụng để cung cấp sự mô tả tài nguyên mà con người có thể đọc được.

Một dòng chú thích bằng text (textual comment) giúp làm rõ ngữ nghĩa của các class và các property của RDF.

Các lớp và các thuộc tính RDF Utility và Container

RDF định nghĩa thêm một số class và property, bao gồm xây dựng cách biểu diễn các container và các phát biểu RDF, và cách mở rộng mô tả từ vựng RDF trên world wide web.

Các lớp và các thuộc tính RDF Container

rdfs:Container

Lớp rdfs:Container là một super – class của các lớp Container của RDF, ví dụ: rdf:Bag, rdf:Seq, rdf:Alt.

rdf:Bag

Lớp rdf:Bag đại diện cho cấu trúc container ‘Bag’ của RDF, và là một lớp con của lớp rdfs:Container.

KHOA CNTT

rdf:Seq

Lớp rdf:Seq đại diện cho cấu trúc container 'Sequence' của RDF, và là một lớp con của lớp rdfs:Container.

rdf:Alt

Lớp rdf:Alt đại diện cho cấu trúc container 'Alt' của RDF, và là lớp con của lớp rdfs:Container.

rdfs:ContainerMembershipProperty

Lớp rdfs:ContainerMembershipProperty với tư cách là thành viên của thuộc tính rdfs:member và các thuộc tính _1, _2, _3, … có thể được sử dụng để cho biết quan hệ thành viên của các container **Baq, Seq,** và **Alt.** rdfs:ContainerMembershipProperty là một lớp con (subclass) của rdf:Property. Mỗi thuộc tính trong quan hệ thành viên của container là một **rdfs:subPropertyOf** của thuộc tính **rdfs:member.**

rdfs:member

Thuộc tính rdfs:member là một siêu thuộc tính (super – property) của các thuộc tính trong quan hệ thành viên của container.

rdf:List

Lớp rdf:List đại diện cho lớp các danh sách liệt kê (Lists) của RDF. Nó được sử dụng với các construct như 'first', 'rest', và 'nil', và nó được hỗ trợ trong cú pháp RDF/XML.

rdf:first

Thuộc tính rdf:first đại diện cho mối quan hệ giữa rdf:List và phần tử (item) đầu tiên của nó.

rdf:rest

KHOA CNTT

Thuộc tính rdf:rest đại diện cho mối quan hệ giữa phần tử (item) rdf:List với các phần tử còn lại trong danh sách (list), hoặc với phần tử cuối của nó (ví dụ, rdf:nil).

rdf:nil

Tài nguyên rdf:nil đại diện cho một rdf:List rỗng (empty).

Các lớp và các thuộc tính RDF Utility

rdfs:seeAlso

Thuộc tính rdfs:seeAlso được sử dụng để cho biết một tài nguyên có thể cung cấp thông tin RDF thêm vào về tài nguyên chủ đề (subject resource).

rdfs:isDefinedBy

Thuộc tính rdfs:isDefinedBy là một thuộc tính con của rdfs:seeAlso, và cho biết tài nguyên nào đang định nghĩa tài nguyên chủ đề.

rdf:value

Thuộc tính rdf:value nhận biết giá trị chủ yếu (thường là chuỗi) của một thuộc tính khi giá trị thuộc tính là một tài nguyên có cấu trúc (structured resource).

rdf:Statement

Lớp rdf:Statement đại diện cho các phát biểu về các thuộc tính của các tài nguyên. rdf:Statement là domain (lĩnh vực) của các thuộc tính: rdf:predicate, rdf:subject và rdf:object.

Các thể hiện (instance) rdf:Statement độc lập khác có thể có cùng giá trị cho các thuộc tính *predicate, subject* và *object* của chúng.

rdf:subject

Chủ đề của một phát biểu (statement) RDF.

Thuộc tính rdf:subject cho biết một tài nguyên là chủ đề của một số phát biểu RDF.

KHOA CNTT

The rdfs:domain **of** rdf:subject **is** rdf:Statement **and the** rdfs:range **is** rdfs:Resource. Thuộc tính này có thể được sử dụng để xác định tài nguyên nào được mô tả bởi một phát biểu RDF.

rdf:predicate

Vị ngữ (predicate) của một phát biểu RDF.

The rdfs:domain **of** rdf:predicate **is** rdf:Statement **and the** rdfs:range **is** rdfs:Resource. Thuộc tính này được sử dụng để xác định vị ngữ nào được sử dụng trong một phát biểu RDF.

rdf:object

Túc từ (tân ngữ) của một phát biểu RDF.

The rdfs:domain **of** rdf:object **is** rdf:Statement. Thuộc tính **range** không được định nghĩa cho thuộc tính này bởi vì các giá trị của rdf:object có thể bao gồm cả **Literals** và **Resources.** Thuộc tính này có thể được sử dụng để xác định túc từ của một phát biểu RDF.

KHOA CNTT

2. RDF Gateway:

Công ty Intellidimension, nằm tại Windsor, Vermont (USA) đã tạo ra một nền RDF thương mại được gọi là RDF Gateway. Điểm mạnh của công cụ này là tính dễ sử dụng và mang chuyển. RDF Gateway chỉ giới hạn trên nền Microsoft Windows, hiện nay vẫn chưa có một kế hoạch nào cho sự ra đời của một phiên bản cho Linux hay một hệ điều hành khác.

Sản phẩm RDF Gateway ra đời cùng lúc với sự ra đời của công ty Intellidimension vào tháng 6 năm 2000. Phiên bản kiểm nghiệm beta của nó được ra mắt vào năm 2001. Những nhà lập trình đã đề xuất và thảo luận các tính năng của hệ thống trong diễn đàn thảo luận chung của W3C. Cuối cùng thì phiên bản thương mại 1.0 ra đời vào ngày 3 tháng 3 năm 2003.

Bởi vì đây là một phần mềm thương mại, nên nó cũng cần có bản quyền. Tuy nhiên vẫn là miễn phí đối với các mục đích học tập phát triển.

2.1. Kiến trúc của RDF Gateway:

RDF Gateway là một server nhẹ và nhanh, nó có thể liên kết các tính năng của một hệ quản trị cơ sở dữ liệu và web server. Nó được thiết kế như là một khung nền cho việc tập hợp, truy vấn, chuyển đổi và phân phối dữ liệu RDF.

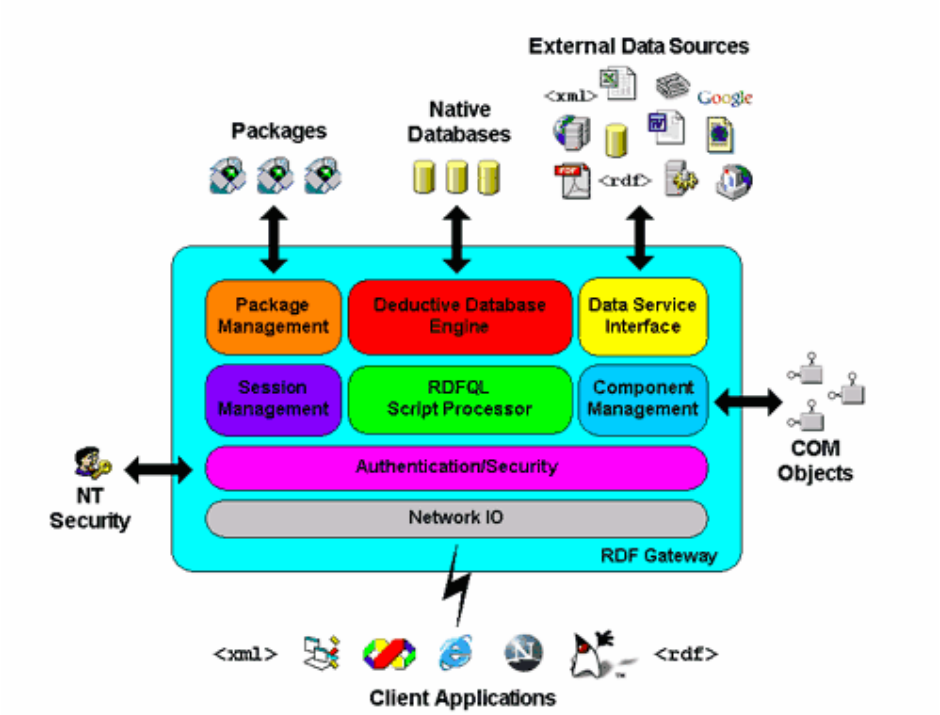


Figure 4 Architecture of RDF Gateway

Hình 24: Kiến trúc của RDF Gateway

- **Bộ xử lý bản mã RDFQL (RDFQL Script Processor)**

RDFQL Script Processor là một máy ảo ưu tiên (preemptive virtual machine) có thể biên dịch, lưu trữ, và thực thi các đoạn script RDFQL. RDFQL là một ngôn ngữ scripting phía server dựa trên ECMA Script (Java Script). RDFQL tích hợp các mở rộng truy vấn tựa SQL để dễ dàng truy cập đến bộ máy cơ sở dữ liệu suy diễn của RDF Gateway. RDFQL script processor cho phép các trang (pages) – có sự kết

hợp của script và nội dung tĩnh tương tự như Microsoft Active Server Pages (ASP). Server được được kết nối đến RDFQL thông qua một thư viện của các đối tượng bên trong (Server, Session, Request, Response, …).

- **Database Engine**

RDF Gateway có một bộ máy cơ sở dữ liệu suy diễn – được thiết kế từ nền không hỗ trợ hoặc có hỗ trợ RDF. Nó thực hiện đánh giá truy vấn theo chiến lược bottom – up, được tổ chức liên đoạn theo tất cả các tài nguyên dữ liệu xác định. Khả năng suy luận logic của bộ máy cung cấp sự hỗ trợ cho cú pháp các luật khai báo của RDFQL. Bộ máy cơ sở dữ liệu không truy cập đến một hệ thống quản lý dữ liệu bên ngoài.

- **Data Service Interface: (Giao diện dịch vụ dữ liệu)**

Giao diện dịch vụ dữ liệu cho phép các tài nguyên dữ liệu từ bên ngoài được tích hợp với RDF Gateway. Một nhà cung cấp dịch vụ dữ liệu là một mô – đun thực thi giao diện này và biểu diễn các nội dung của một kiểu xác định của tài nguyên dữ liệu như là dữ liệu RDF. RDFQL cho phép tổ chức liên đoạn các câu vấn tin được thi hành thông qua nhiều dịch vụ dữ liệu. Giao diện mở này làm cho nó có thể sử dụng bất kỳ một nhà cung cấp dịch vụ dữ liệu sẵn có hiện tại nào hoặc phát triển một nhà cung cấp theo ý mình cho một nguồn dữ liệu.

- **Authentication/Security:**

RDF Gateway có một mô hình bảo mật dựa trên quyền và sự cho phép điều khiển truy xuất đến server và các tài nguyên cơ sở dữ liệu. RDF Gateway hỗ trợ cho những user của nó và các role cũng như user và group của NT. Một NT user luôn đượ chứng thực bằng cách sử dụng một sự uỷ nhiệm của NT cho tài khoản. Sự hỗ trợ của RDF Gateway cho đối với các user và group của NT làm cho có thể quản trị bảo mật từ bên ngoài.

- **Network IO**

Giao diện mạng hỗ trợ cả HTTP và TCP/IP dựa trên giao thức. Tầng nhập xuất mạng (network IO layer) hỗ trợ lược đồ chứng thực mạng bảo mật như là

KHOA CNTT

NT Challenge/ Response (NTLM). Một client kết nối đến server thông qua một interface (giao diện).

- **Package Management**

RDF Gateway cho phép thực thi các ứng dụng để được phát triển và triển khai như là các package. Một package bao gồm các trang server RDF, các trang HTML, các hình ảnh hoặc bất kỳ một kiểu file nào khác.

- **Component Management**

RDFQL hỗ trợ COM trong script phía server của nó. Điều này cho phép tính năng của RDF Gateway có thể được mở rộng hoặc đối với các ứng dụng được tích hợp với RDF Gateway.

- **Session Management**

Bộ quản lý phiên làm việc cho phép lưu lại trạng thái của người dùng trên server.

2.2. Tính năng (Features)

- **Biểu diễn các bộ ba RDF vào trong các bảng dữ liệu:**

Hệ biến hoá RDBMS (RDBMS paradigm) của việc lưu trữ dữ liệu trong các bảng được lắp vào để lưu trữ các bộ ba RDF (triples). Mô hình dữ liệu của các bảng là một bộ ba bao gồm: predicate, subject, và object. Các cột của bảng không có tên nhưng luôn chứa 3 thành phần của bộ ba này theo thứ tự. Lưu ý là predicate là thành phần đầu tiên. Có một cột tuỳ chọn thứ tư cho lưu trữ siêu dữ liệu về triple (bộ ba), siêu dữ liệu này được gọi là "context" của bộ ba. Trường context có thể lưu trữ một định danh tài nguyên mà định danh này có thể được sử dụng để giải quyết các vấn đề bảo mật hoặc nhận diện tài nguyên của bộ ba hoặc bất kỳ một tính năng quen thuộc nào.

- **Other data sources: (Các nguồn dữ liệu khác)**

Các nguồn dữ liệu bên ngoài và các cơ sở dữ liệu đang hoạt động được truy xuất từ server được bao quanh các đối tượng của nguồn dữ liệu. Một đối tượng

KHOA CNTT

nguồn dữ liệu (datasource object) có cấu trúc giống như một table, chứa đựng các bộ ba trong các dòng. Có sự hỗ trợ cho các bảng trong bộ nhớ và nó có thể tạo các trình bao bọc cho dữ liệu bên ngoài.

- **Databases**

Việc lưu trữ các bảng được phân thành các phần trong cơ sở dữ liệu. Một server có thể chứa nhiều cơ sở dữ liệu khác nhau, một bảng có thể được tạo trong một cơ sở dữ liệu. Format của cơ sở dữ liệu là một định dạng file sở hữu, mỗi cơ sở dữ liệu được lưu trong một file.

- **RDFQL script language:**

Ngôn ngữ scripting dựa trên ECMA script, thường được biết đến như là Javascript. Các khái niệm sau được cung cấp:

- Functions (các hàm)
- Variables và Arrays (các biến và các mảng)
- Câu lệnh loops và If
- Exception handling (bắt lỗi)
- Import các file script khác.
- Comments (các chú thích)
- Các câu lệnh (phát biểu) trong RDF Gateway.

Các câu lệnh cho RDF Gateway bao gồm mỗi khía cạnh của server và giúp người lập trình truy cập đến tất cả các tính năng của nó. Một ví dụ là công cụ cấu hình server, công cụ này là một trang web được viết bằng RDFQL được thông dịch bởi một web server được tích hợp, và cho phép truy xuất đến tất cả các đối tượng của server như là: các table, các database, user và package.

Để tìm ra được các dataset của bộ ba RDF, một đối tượng RDF node được cung cấp, nó thu thập tất cả các predicate và subject của một đối tượng đã cho và làm cho nó có thể thay đổi giá trị của các subject.

Để chạy các câu truy vấn trên server, một tập các câu lệnh cơ sở dữ liệu cần phải sẵn sàng. Các câu lệnh cơ sở dữ liệu đóng gói trong RDFQL script, câu lệnh

KHOA CNTT

này thường được biết từ các câu lệnh SQL trong các file source C được tích hợp bởi một trình biên dịch trước.

Truy cập đến các đối tượng ActiveX và COM được hỗ trợ thông qua phương thức khởi gán (construct) của ngôn ngữ ActiveXObject.

Nếu đoạn script RDFQL được đánh giá trong ngữ cảnh của web server, thì các đối tượng chứa dữ liệu session, request và response được cung cấp.

- **Adding and retrieving data (thêm và truy vấn dữ liệu)**

Các lệnh thao tác dữ liệu thì tương tự với cú pháp lệnh trong SQL. Tính năng được mở rộng đối với các nhu cầu xác định của RDF. Có các câu lệnh như: INSERT, SELECT và DELETE. Các câu lệnh này sử dụng các biến (variable) để ràng buộc dữ liệu, tương tự như ngôn ngữ RQL được sử dụng bởi RDFSuite.

```
INSERT {
        [http://www.artchive.com/]
        [http://www.icom.com/schema.rdf#technique]
        [http://www.artchive.com/rembrandt/abraham.jpg]
        'Oil on canvas'
} INTO museum;
```

Ví dụ này chỉ ra cách nào để insert một bộ ba (triple) vào table "museum". Bộ ba được viết giữa 2 dấu ngoặc nhọn ('{' và '}') và chứa 4 giá trị:

- Context
- Predicate
- Subject
- Object hoặc Literal

Thông tin ngữ nghĩa của bộ ba này có nghĩa là: bức ảnh "abraham.jpg" thuộc về lĩnh vực "Oil on Canvas" và thông tin này được lấy từ "www.artchive.com".

```
SELECT ?a, ?b, ?c USING museum
WHERE {?a ?b ?c} AND ?c LIKE "Oil";
```

KHOA CNTT

Để truy vấn các triple từ một table, thì câu lệnh SELECT được sử dụng. Ví dụ này truy xuất tất cả các triple mà có chứa từ "oil" trong giá trị đối tượng literal. Chú ý là triple ở giữa 2 dấu ngoặc nhọn chỉ chứa 3 giá trị, context được bỏ đi.
Dữ liệu có thể được lấy từ các nguồn dữ liệu bên ngoài hoặc chuyển đổi (tranfer) từ một bảng này đến một bảng khác.

```
var doc = new DataSource(
"inet?url=file://c:/Museum.xml&parsetype=rdf");
SELECT ?a, ?b, ?c USING #doc WHERE {?a ?b ?c};
INSERT {?p ?s ?o} INTO museum USING #doc
WHERE {?p ?s ?o};
```

Trong ví dụ này, một dữ liệu RDF được lấy từ một file text và được insert vào bảng museum. Lưu ý là trong RDFQL Javascript, code được trộn với một đoạn code giống như SQL – biến javascript "doc" được sử dụng trong lệnh cơ sở dữ liệu như là "#doc".

- **Built – in Webserver (Webserver gắn liền)**

RDF Gateway có một Webserver gắn liền.Giao diện cấu hình và quảnlý được xuất bản dưới dạng web. Các nhà phát triển ứng dụng có thể tạo các trang web với web server này, bằng cách sử dụng ngôn ngữ RDFQL script. Tính năng này có thể được sử dụng trong việc debug và phát triển, nhưng cũng có thể sử dụng để xây dựng toàn bộ các ứng dụng web bằng cách sử dụng RDF Gateway. Đối với vấn đề sử dụng các đối tượng ActiveX thông qua RDFQL, web server được xem là rất mạnh.

- **RDF Query Analyzer**

Các câu lệnh và các câu truy vấn RDFQL có thể được tạo ra bằng cách sử dụng ứng dụng ảo này (RQF Query Analyzer).

KHOA CNTT

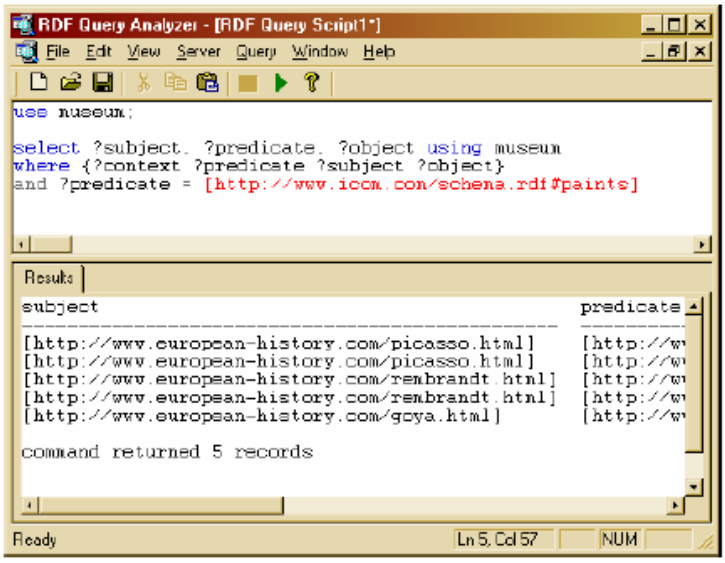


Figure 5: RDF Query Analyzer

Hình 25: Giao diện của RQF Query Analyzer.

Query analyzer thì tương tự như các sản phẩm lượng gía truy vấn của các SQL Server phổ biến. Các script phức tạp có thể được tạo ra ở đây và được sử dụng trong các trang web hoặc các ứng dụng khác. Các câu truy vấn có thể được đánh giá lại một lần nữa ở RDF Gateway cục bộ hoặc ở xa, trình soạn thảo văn vản (text editor) có cú pháp ở dạng được highlight và có thể lưu và mở các câu vấn tin.

- **Inference Engine (máy suy diễn)**

Bộ máy cơ sở dữ liệu RDF Gateway gồm một máy suy diễn. Các câu lệnh bộ ba RDF mới, có thể được phát sinh một cách tự động dựa trên các luật suy diễn và các bộ ba đã có sẵn. Các hàm (function) có thể được định nghĩa, các hàm này rút trích dữ liệu từ cơ sở dữ liệu dựa trên các luật. Các luật này được định nghĩa trong ngôn ngữ RDFQL script và có thể được sử dụng trong các dòng lệnh thao tác cơ sở dữ liệu.

```
RULEBASE schema
{
    INFER {[rdf:type] ?s ?class} FROM
    {[rdf:type] ?s ?subclass} AND
```

KHOA CN

```
        {[rdfs:subClassOf] ?subclass ?class};
};

SELECT ?p ?s ?o USING #ds RULEBASE schema WHERE
{[rdf:type] ?s ?o} AND {?p ?s ?o};
```

Ví dụ này định nghĩa một luật (rule) cho lược đồ RDF (RDF Schema) về các subclass (lớp con): Nếu môt subject là một kiểu (type) của X và X được định nghĩa là subject của Y, thì subject cũng là một kiểu của Y. Rồi sau đó luật này được sử dụng cho câu lệnh SELECT để truy vấn tất cả các class và các class xuất phát của tất cả các subject.

Vốn RDF Schema không được hỗ trợ bởi RDF Gateway, nó phải được mô tả theo các luật suy diễn.

- **Client Libraries**

RDF Gateway có các thẻ cài client (client drivers) cho Microsoft ADO và Sun Microsystems JDBC. Điều này cho phép RDF Gateway hỗ trợ rộng khắp nhiều client như là các web browser, các ứng dụng Windows, các ứng dụng Java, XML hoặc RDF dựa trên clients.

- **Security**

Khi truy cập vào RDF Gateway thông qua http, ADO hoặc các protocols khác, người dùng phải được nhận biết bằng cách sử dụng username và password. Một tài khoản người dùng (user account) là "anonymous" được cung cấp cho việc truy xuất chung (mọi người đều có thể truy cập vào với user account này).

Hệ thống bảo mật sử dụng cả hai loại: cơ sở dữ liệu bảo mật của windows để xác nhận các người dùng windows và một cơ sở dữ liệu người dùng bên trong. Cũng như Internet Explore, NT Authentication (sự chứng thực ở mức NT) có thể được sử dụng với http.

Mỗi thành phần (item) được quản lý bởi RDF Gateway có thể bị giới hạn đối với các user được định nghĩa, các thành phần này bao gồm: các package, các

KHOA CNTT

table, data source và các component. Ở cấp độ table, dễ dàng để sửa đổi việc đọc, viết, và xoá các quyền cho các user riêng biệt.

Một khái niệm bảo mật dựa trên dòng đối với các lệnh RDF trong các table được dựa trên cột "context", trường thêm vào thứ tư này được add vào subject, predicate, và object. Một user có thể được cho phép để đọc, viết, và xoá các quyền (rights) đối với một context riêng biệt.

Không hề có sự hỗ trợ đối với một nhóm người sử dụgn (user group – trong RDF Gateway không có khái niệm này).

- **Configuration and Management (cấu hình và quản lý)**

Các sự sắp đặt (setting) cấu hình chi tiết được truy cập thông qua giao diện web, giao diện này được dẫn vào nhờ web server gắn liền. Người dùng phải đăng nhập vào bằng cách sử dụng một account có vai trò là administrator của windows. Ứng dụng web này được đặt tên là "RDF Gateway Management Utility" và cung cấp truy cập đến các databases, tables, users, contexts, ActiveX Components, Data Services, Roles, Packages, MimeTypes và Timers. Đối với hầu hết các thành phần này, thì các chọn lựa bảo mật về sự cho phép có thể được đặt.

Tiện ích quản lý được thực thi như là RDF Gateway web package.

Trên đây là những giới thiệu bao quát về RDF Gateway. Ngoài ra ta cũng có thể xem thêm về cú pháp cuả RDF Gateway khá chi tiết trong phần help của tiện ích RDF Query Analyzer.

KHOA CNTT

3. Hệ thống nhãn ngữ nghĩa:

Hệ thống nhãn ngữ nghĩa được giới thiệu ở đây gồm 3 tiểu hệ thống nhỏ ứng với 3 từ loại: danh từ, động từ và tính từ. Trong mỗi tiểu hệ thống, lại được phân thành 2 cấp: cấp cơ bản chứa một số ít các nhãn chung nhất, sử dụng nhiều nhất và là những nhãn viết tắt (gợi nhớ) dễ nhớ; và cấp chuyên sâu là những nhãn theo hệ thống

LLOCE. Ngoài ra, trong phần này, cũng còn liệt kê một số hệ thống nhãn ngữ nghĩa khác như WordNet, CoreLex.

3.1. Nhãn ngữ nghĩa cơ bản cho danh từ:

STT	Nhãn	Mô tả	Ý nghĩa
1	ABS	Abstraction	Những gì trừu tượng
2	ACT	Act	Hành động
3	AGT	Agent	Tác nhân
4	ANM	Animal	Sinh vật
5	ART	Artifact	Nhân tạo
6	ATR	Attribute	Thuộc tính
7	BDY	Body	Cơ thể con người
8	CEL	Cell	Tế bào
9	CHM	Chemical	Hoá chất
10	COM	Communication	Truyền tin
11	CON	Consequence	Hậu quả
12	ENT	Entity	Thực thể
13	EVT	Event	Biến cố
14	FEL	Feel	Sự cảm nhận
15	FEM	Female	Giống cái/ nữ
16	FOD	Food	Thức ăn
17	FRM	Shape, form	Hình dạng
18	GAS	Gas	Thể khí
19	GRB	Group biology	Nhóm sinh học
20	GRP	Group	Nhóm nói chung
21	GRS	Group social	Nhóm xã hội
22	HOU	House	Công trình xây dựng
23	HUM	Human	Con người

KHOA CNTT

24	LFR	Life form	Sự sống
25	LIN	Line	Đường, nét, dấu vết
26	LIQ	Liquid	Thể lỏng
27	LME	Linear measure	Đo lường
28	LOC	Location	Vị trí
29	LOG	Location geography	Vùng địa lý
30	MAL	Male	Giống đực/ nam
31	MEA	Measure	Đại lượng
32	MIC	Microoragnism	Vi sinh vật
33	MOT	Motion	Sự chuyển động
34	NAT	Natural object	Vật thể thiên nhiên
35	PHM	Phenomenon	Hiện tượng
36	PHO	Physical object	Vật thể vật lý
37	PLT	Plant	Thực vật
38	POS	Possession	Sự sở hữu
39	PRO	Process	Quá trình
40	PRT	Part, piece	Bộ phận
41	PSY	Psychological	Thuộc tính tâm lý
42	QUD	Definite quantity	Đại lượng hữu hạn
43	QUI	Indefinite quantity	Đại lượng vô hạn
44	REL	Relation	Quan hệ
45	SOL	Solid	Thể rắn
46	SPC	Space	Không gian
47	STA	State	Trạng thái
48	SUB	Substance	Chất liệu
49	TME	Time	Thời gian
50	UNT	Unit	Thuộc đơn vị

Bảng 13: Nhãn ngữ nghĩa cơ bản cho danh từ

KHOA CNTT

3.2. Nhãn ngữ nghĩa cơ bản cho động từ:

STT	Nhãn	Mô tả	Ý nghĩa
1	VBDY	Body	Các động từ của cơ thể: ăn, mặc, …
2	VCHG	Change	Các động từ thuộc về sự thay đổi: tăng, đổi, …
3	VCOG	Human	Các động từ tri nhận: suy nghĩ, xét đoán,…
4	VCOM	Communication	Các động từ truyền thông: kể, hỏi, ra lệnh, …
5	VCMP	Competition	Các động từ về cạnh tranh: chiến đấu, thi đấu, …
6	VCSM	Consumption	Các động từ về tiêu thụ: ăn, uống, …
7	VCON	Contact	Các động từ về tiếp xúc: đánh, đào, …
8	VCRE	Creation	Các động từ về sự tạo lập: sơn, khâu, thi hành, …
9	VEMO	Emotion	Các động từ về cảm giác: yêu, ghét,…
10	VMOT	Motion	Các động từ về chuyển động: đi, bay, bơi, …
11	VPER	Perception	Các động từ về giác quan: nghe, thấy, cảm thấy, …
12	VPOS	Possession	Các động từ về sở hữu: mua, bán, sở hữu, …
13	VSOC	Social	Các động từ về hoạt động xã hội: bầu cử, tại vì, …

KHOA CNTT

14	VSTA	Stative	Các động từ về trạng thái, quan hệ không gian.
15	VWEA	Weather	Các động từ về thời tiết: mưa, tuyết, sấm, …

Bảng 14: Nhãn ngữ nghĩa cơ bản cho động từ

3.3. Nhãn ngữ nghĩa cơ bản cho tính từ:

STT	Nhãn	Mô tả	Ý nghĩa
1	ACOL	Color	Các tính từ về màu sắc: đỏ, xanh, …
2	ASIZ	Size	Các tính từ về kích thước: tròn, dẹt, …
3	ATME	Time	Các tính từ thuộc về thời gian: lâu, mau, …
4	ASPC	Space	Các tính từ thuộc về không gian: lớn, nhỏ, dài, …
5	ASTR	Strength	Các tính từ về sức mạnh: mạnh, yếu, …
6	ADEG	Degree	Các tính từ về mức độ: nhiều, ít, …
7	AFEA	Feature	Các tính từ về đặc điểm, nội dung: khó, hay, …
8	AREF	Reference	Các tính từ bổ nghĩa sở chỉ: former (president)
9	AREL	Relation	Các tính từ quan hệ: Vietnamese (war)

KHOA CNTT

Bảng 15 : Nhãn ngữ nghĩa cơ bản cho tính từ

3.4. Hệ thống nhãn ngữ nghĩa LDOCE

STT	Mã ngữ nghĩa cơ bản		Mã ngữ nghĩa phát sinh	
1	A	Con vật (animal)	E	Chất rắn/ lỏng (S + L)
2	B	Con vật cái	K	Người/con vật đực

		(female animal)		(D +M)
3	C	Vật cụ thể (concrete)	O	Người/ con vật (A + H)
4	D	Con vật đực (male animal)	R	Người/con vật cái (B + F)
5	F	Người nữ (female human)	U	Tập hợp người/con vật (Col. + O)
6	G	Khí (gas)	V	Thực vật/ con vật (P + A)
7	H	Người (human)	W	Vật trừu tượng/cụ thể (T + I)
8	I	Vật cụ thể không có sự sống	X	Vật trừu tượng/ người (T + H)
9	J	Vật rắn di chuyên được	Y	Vật trừu tượng/ có sự sống (T + Q)
10	L	Chất lỏng (liquid)	1	Người /chất rắn (H + S)
11	M	Người nam (male human)	2	Trừu tượng/ chất rắn (T + S)
12	N	Vật rắn không di chuyển được	6	Chất lỏng/ trừu tượng (L + T)
13	P	Thực vật (plant)	7	Chất khí/ chất lỏng (G + L)
14	Q	Có sự sống (animate)		
15	S	Chất rắn (solid)		
16	T	Trừu tượng (abstract)		

KHOA CNTT

17	Z	Không đánh dấu (unmarked)		
18	4	Vật thể trừu tượng (abs physic)		
19	5	Chất hữu cơ (organic material)		

Bảng 16: Hệ thống nhãn ngữ nghĩa LDOCE

4. Hệ cơ sở tri thức ngữ nghĩa từ vựng WordNet

4.1. Hệ thống nhãn ngữ nghĩa của danh từ:

Trước hết, ta sẽ tìm hiểu những hạn chế trong cách lưu trữ thông tin về ngữ nghĩa của danh từ ở từ điển thông thường, từ đó, chúng ta mới thấy những ưu thế của WordNet trong cách lưu trữ, truy xuất, cập nhật các thông tin đó.

4.1.1. Tổ chức của danh từ trong từ điển thông thường:

Khi ta tra một danh từ nào đó trong các từ điển thông thường, ta sẽ nhận được những lời giải thích có vẻ khá đầy đủ. Ví dụ, tra từ "tree" (cây), ta sẽ nhận được định nghĩa "*tree* is a *plant* that is large, woody, perennial and has a distinct trunk" (*cây* là một *thực vật* mà có thân, sống lâu năm, có gỗ, kích thước lớn). Đối với những người có kiến thức phổ thông, có thể chấp nhận định nghĩa này. Nhưng nếu chúng ta muốn biết sâu hơn như "cây có rễ, có tế bào xen – lu – lô, là tổ chức có sự sống, …" thì ta cần phải tra ngữ nghĩa của từ "plant", tuy nhiên khi tra từ "plant", ta sẽ nhận được hai lời giải thích hoàn toàn khác nhau: một dành cho nghĩa "nhà máy" và một dành cho nghĩa "thực vật". Câu hỏi đặt ra là, khi muốn truy xuất tự động, thì máy tính sẽ chọn nghĩa nào? Đây là hạn chế của các từ điển thông thường.

Các từ điển thông thường chủ yếu thiếu thông tin mang tính cấu trúc (structure), vì định nghĩa của nó chỉ mang thông tin có tính dữ kiện (fact), và do cách tổ chức theo vần abc, nên không thể chứa ở mỗi từ mọi thông tin có liên quan trong

KHOA CNTT

định nghĩa của nó được, vì làm như vậy sẽ trùng lắp thông tin, kích thước của từ điển sẽ vô cùng lớn và không kinh tế.

Cuối cùng, một khuyết điểm lớn nhất mà hầu hết các từ điển thông thường đều gặp phải, đó là việc định nghĩa vòng tròn. Nghĩa là: dùng từ W_a để định nghĩa từ W_b, rồi lại có chỗ lại dùng từ W_b để định nghĩa lại từ W_a.

4.1.2. Tổ chức dữ liệu danh từ trong WordNet

Thấy được các khuyết điểm của từ điển thông thường, WordNet lưu trữ danh từ thành một hệ thống phân cấp hình cây dựa theo quan hệ hạ danh (hyponymy) và thượng danh (hypernymy). Xuất phát từ gốc là một ý niệm cha rất tổng quát, dựa theo quan hệ thượng danh (hypernymy), tả giả phân (nhánh) thành các ý niệm con cụ thể hơn, rồi cũng từ chính các ý niệm con này, lại tiếp tục phân nhỏ nữa thành các ý niệm chi tiết hơn, và cứ như thế đến khi không còn cần thiết phân chia nữa (trung bình cỡ chục cấp) và nút tận cùng đó (nút lá) chính là các danh từ.

Ví dụ, "cây sồi" (oak) là một loài "cây" (tree), "cây" là một loài "thực vật" (plant), "thực vật" là một loài "hữu cơ" (organism). Trong WordNet sẽ diễn tả như sau: oak @ $\rightarrow$ tree @ $\rightarrow$ plant @ $\rightarrow$ organism, với ký hiệu "@ $\rightarrow$" để trỏ đến nút cha, thể hiện quan hệ hạ danh (hyponymy), hay còn gọi là quan hệ ISA. Đối lập với quan hệ hạ danh là quan hệ thượng danh (hypernymy) và trong WordNet, quan hệ này được ký hiệu là "~ $\rightarrow$" để trỏ đến nút con, ví dụ: organism ~ $\rightarrow$ plant ~ $\rightarrow$ tree ~ $\rightarrow$ oak (vì WordNet được lưu trữ dưới dạng điện tử, nên WordNet chỉ cần lưu quan hệ hyponymy một cách tường minh, còn quan hệ hypernymy sẽ được tự động suy ra từ quan hệ hyponymy).

Với cách tổ chức phân cấp như trên, WordNet không cần lưu mọi tính chất của mỗi ý niệm (nút), mà chỉ cần lưu đặc điểm riêng của ý niệm đó mà thôi, còn các tính chất khác được tự động suy diễn ra từ đặc tính chung được kế thừa từ ý niệm cha cùng với các đặc tính khác của các ý niệm con. Điều này giúp cho WordNet khắc phục được các khuyết điểm của từ điển thông thường (không lưu trùng lắp thông tin mà vẫn chứa đầy đủ thông tin, tiết kiệm không gian lưu trữ).

KHOA CNTT

Ngoài ra, với các tổ chức phân cấp có kế thừa như trên, WordNet khắc phục được hiện tượng định nghĩa vòng quanh, không bao giờ có hiện tượng từ W_a định nghĩa từ W_b, rồi chính W_b lại định nghĩa W_a. Vì theo tổ chức hình cây, mỗi loại quan hệ chỉ có một chiều nhất định, ví dụ quan hệ thượng danh, chỉ có chiều từ trên xuống dưới, đi từ tổng thể đến chi tiết (chuyên biệt hoá), còn quan hệ hạ danh thì ngược lại: đi từ dưới lên trên, đi từ chi tiết đến tổng thể (tổng quát hoá).

Tuy nhiên, không phải mọi thông tin về thế giới thực đều được lưu trong các ý niệm của WordNet, nên trên thực tế, ta cũng không thể có được đầy đủ hoàn toàn các tri thức về thế giới thực của "cây" như tri thức của người được. Ví dụ: WordNet không lưu những thông tin, như: "cây" cho bóng mát, cây khô có thể làm củi đun, …. Hiện nay, WordNet chưa liên kết "bác sĩ" với "bệnh viện", chưa thể liên kết "vợt", "banh", "lưới", … với "sân chơi tennis".

4.1.3. Các ý niệm nguyên thuỷ (primitive semantic)

Trong WordNet, ta có "gia phả" của từ "oak" như sau: {oak} @→ {tree} @→ {plant, flora} @→ {organism, living thing} @→ {thing, entity}. Như vậy, ý niệm {thing, entity} là một ý niệm gốc, ý niệm cao nhất, tổng quát nhất, chính vì vậy nó chẳng mang một ý nghĩa gì (vì nó là cái gì đó rất chung chung) và mọi ý niệm trong WordNet đều dẫn tới ý niệm gốc đó (đều là con cháu của nó). Tuy nhiên, nếu ta tổ chức cây ý niệm danh từ với một gốc ý niệm duy nhất trên cây thì sẽ khiến cho cây có kích thước rất lớn, việc tổ chức các nhãn cho các ý niệm phải chi tiết hơn để tránh trùng nhau. Ví dụ: giữa "plant" của ý niệm "thực vật" và "plant" của ý niệm "nhà máy", WordNet phải dùng 2 nhãn (dạng từ) khác nhau để phân biệt, hơn nữa, sự gom về chung một gốc lớn như vậy thì cũng chẳng có kế thừa được thông tin gì (vì các ý niệm gốc là rất chung chung, ít thông tin).

Chính vì vậy mà WordNet đã phân thành 25 gốc chính như bảng dưới đây mô tả. Các gốc này được gọi là các ý niệm nguyên thuỷ. Mỗi cây như vậy được lưu thành một tập tin riêng rẽ. Chính vì vậy, mà khi gặp nhãn "plant" (thực vật) như trên, thì máy tính không nhầm lẫn với "plant" có nghĩa "nhà máy", vì cây ý niệm mà chứa "tree" là cây mà có ý niệm nguyên thuỷ là {plant} (thực vật) được lưu riêng biệt với

KHOA CNTT

cây ý niệm mà có chứa "plant" với nghĩa là "nhà máy" (ý niệm này được lưu trong cây khác, cây mà có ý niệm nguyên thuỷ là {artifact}).

Quan sát 25 ý niệm nguyên thuỷ đó, ta thấy có một số ý niệm có những nét nghĩa chung nhau (ví dụ: {animal}, {person}, {plant} đều là những vật có sự sống), chính vì vậy mà trong WordNet, những ý niệm có chung nét nghĩa như vậy sẽ được nhóm với nhau để tạo thành con của một ý niệm cao hơn. Sau khi nhóm rút gọn lại, trong WordNet chỉ còn 11 ý niệm nguyên thuỷ (những ý niệm được in nghiêng trong bảng dưới đây).

Entity (thực thể tiếp xúc được)	Organism (vật có sự sống)	Animal (súc vật)	
		Person (người)	
		Plant (thực vật)	
	Object (vật thể không có sự sống)	Artifact (đồ nhân tạo)	
		Natural object (vật thể tự nhiên)	Body (cơ thể)
		Substance (chất)	Food (thức ăn)
	Abstraction (trừu tượng)	Attribute (thuộc tính)	
		Quantity (số lượng)	
		Relation (quan hệ)	
		Time (thời gian)	
	Psychology feature (về tâm lý)	Cognition (tri nhận)	
		Feeling (cảm giác)	
		Motivation (tình cảm)	
		Natural phenomenon (hiện tượng tự nhiên)	Process (quá trình)
		Activity (hoạt động)	
		Event (biến cố)	
		Group (nhóm người)	
		Location (vị trí)	
		Possession (sở hữu)	

KHOA CNTT

		Shape (hình dạng)	
		State (trạng thái)	

Bảng 17:Sự phân lớp danh từ trong WordNet

Các ý niệm trong bảng trên đây được gọi là những ý niệm nguyên thuỷ (primitive semantic component). Từ những ý niệm nguyên thuỷ này, WordNet đã xây dựng nên hệ thống cây phân lớp cho danh từ theo quan hệ hạ danh (hyponymy) và thượng danh (hypermyny).

Với cách sắp xếp như trên, trong thực tế sử dụng WordNet, tác giả thấy độ sâu của cây WordNet rất cạn (cỡ 10 – 12 cấp) và gần một nửa trong số các ý niệm phải đi qua đó, mang ý nghĩa kỹ thuật nhiều hơn.

4.1.4. Đặc điểm riêng của mỗi ý niệm trong hệ phân cấp:

Theo cách tổ chức của WordNet, các ý niệm con cùng kế thừa một ý niệm cha, cần phải có một số đặc tính riêng nhằm phân biệt với ý niệm cha và các ý niệm anh em với nó. Các đặc tính phân biệt này gồm 3 loại, ví dụ với ý niệm {robin} (chim cổ đỏ), nó có 3 loại đặc tính sau:

- ☞ Thuộc tính (attributes), (nối với tính từ) [màu = đỏ, kích thước = nhỏ]
- ☞ Bộ phận (parts) (nối với dnh từ) [mỏ, lông,cánh]
- ☞ Chức năng (functions) (nối với động từ) = [hót, bay]

Tương tự, ý niệm {canary} (chim vàng anh) cũng là con của ý niệm {bird} (chim), có thuộc tính [màu = vàng, kích thước = nhỏ], có bộ phận [mỏ, lông, cánh], có khả năng [hót, bay, đẻ trứng]. Vậy ta thấy giữa {robin} và {canary} (đều cùng là loài chim), có điểm khác biệt về màu sắc. Như vậy, thông tin của một ý niệm chính là thông tin kế thừa từ ý niệm cha cộn thêm các đặc tính riêng của nó. Vậy ta có thể nói synset {A} là con của synset {B} nếu tất cả các đặc tính của synset {B} đều có trong synset {A}. Vì vậy một từ thuộc synset con, có thể làm tiền trí tự (antecendent) thay cho một từ thuộc synset cha, hay có thể thay cho một đối từ của một động từ với điều kiện đối từ đó thuộc synset cha. Ví dụ:

KHOA CNTT

- Trong câu "Tôi đưa anh ấy một *cuốn tiểu thuyết* hay, nhưng cuốn sách đó làm anh ta buồn". Ta có *cuốn tiểu thuyết* là ý niệm con của ý niệm *cuốn sách*, nên có thể làm tiền trí tự cho từ *cuốn sách.*
- Trong câu "Tôi uống nước", có thể thay thế đối từ "nước" của động từ "uống" bằng bất kỳ đối từ nào mà thuộc ý niệm con của nó, như: *nước ngot, nước trà, nước suối, …*

4.2. Hệ thống nhãn ngữ nghĩa của động từ:

Động từ là từ loại quan trọng nhất và là từ bắt buộc phải có đối với mọi câu tiếng Anh. Dựa trên đặc điểm của động từ, ta có thể xác định cấu trúc của câu (A.S. Hornby). Dựa trên động từ, ta có thể xác định các vai trong câu (Fillmore). Số lượng động từ trong tiếng Anh chỉ bằng 1/3 số lượng danh từ, còn mức độ mơ hồ nghĩa của động từ thì lại cao hơn (trung bình một động từ có 2.11 nghĩa, còn danh từ có 1.74 nghĩa). Nghĩa của động từ rất uyển chuyển, linh động theo các danh từ có liên quan đến nó. WordNet chia các động từ thành 15 nhóm (ở trên) để chỉ các *biến cố* (event), *hành động* (action) hay *trạng thái* (state) khác nhau dựa theo sự phân chia về mặt ngữ nghĩa, như: nhóm *động từ chỉ chức năng và việc chăm sóc cơ thể, sự nhận thức, quan hệ xã hội, ….*

Việc xây dựng tập đồng nghĩa (synset) cho động từ cũng gặp nhiều khó khăn hơn so với danh từ vì khó xác định từ đồng nghĩa. Ta thấy trong tiếng Anh có một số động từ đồng nghĩa, như: begin – commence (bắt đầu), end – terminate (kết thúc), buy – purchase (mua), hide – conceal (giấu), … nhưng thực chất việc dùng lẫn lộn các động từ đồng nghĩa này không phải lúc nào cũng đúng. Ví dụ: người ta thường nói "Where have you hidden Dad's slippers?" (Anh giấu dép của Dad ở đâu?) chứ không nói là "Where have you concealed Dad's slippers?".

Việc biểu diễn ngữ nghĩa và tổ chức động từ là điều khó khăn nhất so với các từ loại khác. Có rất nhiều cách tiếp cận khác nhau để biểu diễn ngữ nghĩa của động từ, chủ yếu là phân rã ngữ nghĩa động từ thành dạng này hay dạng khác. Sau đây là một số cách phân giải ngữ nghĩa động từ.

KHOA CNTT

4.2.1. Sự phân giải ngữ nghĩa của động từ:

Hầu hết các cách tiếp cận đối với ngữ nghĩa động từ là cố gắng phân giải ngữ nghĩa động từ thành một số hữu hạn các thành phần ý niệm – ngữ nghĩa phổ quát (universal semantic – conceptial components), hay còn gọi là ý niệm nguyên thuỷ, nguyên tố, sơ khởi, vị từ nguyên tử, danh từ đánh dấu (noun marker), ví dụ: động từ "kill" (giết) = {CAUSE TO BECOME NOT ALIVE} (gây ra sự dẫn đến không sống). Cách tiếp cận này đã nhận được nhiều ý kiến khác nhau, có người đồng tình (Katz, Lakoff, Jackendoff, Schank, Miller) nhưng cũng có người phản đối cho là không thích hợp (Chomsky và một số người khác).

Sự phân tích ngữ nghĩa quan hệ của động từ khác với sự phân giải ngữ nghĩa của động từ. Sự phân giải ngữ nghĩa chủ yếu dựa trên các ý niệm cơ sở (đơn vị ngữ nghĩa nhỏ nhất), còn sự phân tích ngữ nghĩa quan hệ lại dựa vào các ý niệm căn bản đã hình thành trong đầu óc của con người. Ví dụ: như quan hệ CAUSE (nguyên nhân) liên kết các cặp động từ *teach* (dạy) – *learn* (học), *show* (chỉ) – *see* (thấy), dựa trên quan hệ này cũng giúp ta phân biệt một cách có hệ thống đâu là tha động từ (transitive verb) và đâu là tự động từ (intransitive verb).

4.2.2. Quan hệ kéo theo của động từ:

Trong WordNet, mỗi từ loại được tổ chức dựa theo một quan hệ chính nào đó, ví dụ: danh từ thì dựa theo quan hệ hạ danh (hyponymy), tính từ thì dựa theo quan hệ phản nghĩa (antonymy), còn động từ thì dựa vào quan hệ kéo theo (entialment).

Giữa quan hệ kéo theo có phần nào đó giống quan hệ bộ phận (meronymy), nhưng không thích hợp cho ý nghĩa V1 là bộ phận của V2 giống như bên danh từ. Ví dụ: ta thử xét có phải "thinking" (sự suy nghĩ) là một bộ phận của "planning" (việc hoạch định) hay không? Nhưng nhiều người cho rằng động từ không thể phân chia bộ phận giống như danh từ vì: các danh từ và các bộ phận của danh từ đều có sở chỉ vật (referent) cụ thể, phân biệt trong khi đó bên động từ thì không được rõ ràng như vậy. Ngoài ra, quan hệ giữa 2 động từ còn phụ thuộc vào thời gian thực hiện, xảy ra hành động, biến cố (bên danh từ: quan hệ bộ phận không phụ thuộc vào thời gian). Một

KHOA CNTT

hành động hay biến cố được gọi là một bộ phận của một hành động hay biến cố khác chỉ khi nó là một phần, một giai đoạn trong quá trình thực hiện của hành động kia.

Tóm lại, qua quan sát các trường trên, ta rút ra nhận định sau: *nếu V1 kéo theo V2 và nếu thời gian diễn ra V1 nằm trong hay bao hàm thời gian diễn ra V2 thì giữa V1 và V2 có quan hệ bộ phận – toàn thể (part – whole).*

4.2.3. Quan hệ cách thức đặc biệt của động từ:

Trong WordNet, quan hệ hạ danh (hyponymy) đóng vai trò chính trong việc tổ chức danh từ, ví dụ: "canary" (chim vàng anh) là một loại (hạ danh của) "bird" (chim), nhưng đối với động từ, ta thấy không thích hợp nếu nói "limp" (đi khập khiễng) là một loại của "walk" (đi bộ). Điều này là do: sự khác biệt ngữ nghĩa giữa 2 động từ thì khác với những đặc trưng phân biệt giữa 2 danh từ trong quan hệ hạ danh.

Trong việc xem xét quan hệ "hạ danh" của động từ, người ta nhận thấy nó không đơn giản như danh từ, mà nó liên quan đến sự cân nhắc tỉ mỉ về ngữ nghĩa trên các trường nghĩa (semantic field) khác nhau. Ví dụ: khi phân tích các động từ chuyển động: "slide" (trượt) và "pull" (kéo), người ta nhận thấy rằng chúng là một sự kết hợp khác nhau giữa nét nghĩa MOVE (chuyển động) với nét nghĩa MANNER (cách thức) . Chính vì vậy, mà trong WordNet, đã sử dụng một quan hệ mới, được gọi là quan hệ cách thức (troponymy) để diễn tả "V1 là V2 với cách thức đặc biệt", ví dụ: "limp" (đi khập khiễng) có quan hệ cách thức với đặc biệt với "walk" (đi bộ) vì "đi khập khiễng là một cách thức đi bộ đặc biệt". Cách thức đặc biệt phải được hiểu rộng không chỉ là cách thức để hành động, mà còn có thể là ý định, động cơ, môi trường, … để hành động, để xảy ra biến cố, để hình thành trạng thái.

Trong mọi quan hệ cách thức đặc biệt, giữa động từ V1 của một động từ V2 tổng quát hơn, bao giờ cũng có quan hệ V1 cũng kéo theo V2. Ví dụ như: khi diễn ra hành động "đi khập khiễng" thì hiển nhiên lúc đó cũng phải diễn đang diễn ra hành động "đi bộ".

Vì vậy, ta có thể nói: *quan hệ cách thức đặc biệt (troponymy) là một trường hợp đặc biệt của quan hệ kéo theo (entailment).* Một quan hệ kéo theo mà trong đó thời gian diễn ra 2 hành động của 2 động từ là trùng nhau. Còn giữa hai động từ "buy/

KHOA CNTT

pay" hay "snore/ sleep" thì chỉ là quan hệ kéo theo mà thôi chứ không có quan hệ cách thức đặc biệt (vì thời gian diễn ra của 2 hành động không trùng nhau).

KHOA CNTT