ỨNG DỤNG KỸ THUẬT TÁCH THÔNG TIN TỪ VĂN BẢN TRONG QUÁ TRÌNH TẠO TRANG WEB CÓ NGỮ NGHĨA

Từ Minh Phương, Phạm Hoàng Duy, Trịnh Hữu Kiên

Học viện công nghệ bưu chính viễn thông

Email: phuongtm@fpt.com.vn

***Tóm tắt**: Web có ngữ nghĩa (Semantic Web là) mở rộng của Web truyền thống bằng cách bổ sung phần ý nghĩa (ngữ nghĩa) cho các trang Web sao cho các chương trình (các agent mềm) có thể "hiểu" được. Việc bổ sung ngữ nghĩa cho phép tăng cường khả năng tìm kiếm, chia sẻ, trao đổi thông tin tự động. Để tạo ra và sử dụng các trang Web có ngữ nghĩa cần có công cụ hỗ trợ thích hợp. Bài báo trình bầy về bộ công cụ như vậy do chúng tôi thiết kế và xây dựng. Đặc điểm quan trọng nhất của bộ công cụ là cho phép tự động tạo ra các chú giải của trang Web đã có nhờ sử dụng kỹ thuật tách thông tin tự động từ văn bản. Bộ công cụ được thử nghiệm với một ứng dụng cụ thể và cho kết quả khả quan.*

1. ĐẶT VẤN ĐỀ

Với nhiều tỷ trang web phân bố trên hầu hết các quốc gia, World Wide Web (WWW) là môi trường tốt cho việc biểu diễn và truy cập thông tin dạng số. Tuy nhiên, lượng thông tin khổng lồ đó cũng tạo ra những khó khăn lớn trong việc tìm kiếm, chia sẻ thông tin trên WWW. Hiện nay, thông tin trên WWW được biểu diễn chủ yếu dưới dạng ngôn ngữ tự nhiên (các trang web trên ngôn ngữ HTML). Cách biểu diễn đó phù hợp với con người nhưng lại gây ra nhiều khó khăn cho các chương trình làm nhiệm vụ hỗ trợ tìm kiếm, chia sẻ và trao đổi tin. Chương trình máy tính không "hiểu" được thông tin và dữ liệu biểu diễn dưới dạng thích hợp với con người.

Để giải quyết vấn đề này, nhiều tổ chức nghiên cứu và kinh doanh đã phối hợp nghiên cứu và phát triển *Web có ngữ nghĩa* (Semantic Web). Theo định nghĩa của giám đốc tổ chức World Wide Web Consortium (http: // www. w3c. org), đồng thời là cha đẻ của WWW, Web có ngữ nghĩa là sự mở rộng của WWW hiện tại bằng cách thêm vào các mô tả ý nghĩa (hay ngữ nghĩa) của thông tin dưới dạng mà chương trình máy tính có thể "hiểu" và do vậy cho phép xử lý thông tin hiệu quả hơn [1]. Như vậy, Web có ngữ nghĩa sẽ bao gồm các thông tin (trang web) được biểu diễn theo cách truyền thống cùng với ngữ nghĩa của các thông tin này được biểu diễn một cách tường minh. Việc thêm phần ngữ nghĩa cung cấp thêm tri thức cho các chương trình (các agent), giúp nâng cao chất lượng phân loại, tìm kiếm, trao đổi thông tin.

Muốn xây dựng Web có ngữ nghĩa cần có công cụ hỗ trợ. Trong bài báo này, chúng tôi mô tả bộ công cụ mà chúng tôi xây dựng phục vụ mục đích này cùng với các giải pháp kỹ thuật được lựa chọn và sử dụng. Phần quan trọng của bộ công cụ là phần tách thông tin tự động cho phép rút ngắn thời gian tạo phần ngữ nghĩa cho trang web. Để minh hoạ cho việc sử dụng và thử nghiệm bộ công cụ, bài báo cũng trình bầy một ứng dụng tìm kiếm thông tin với những trang web có ngữ nghĩa do bộ công cụ tạo ra.

2. THÀNH PHẦN CỦA WEB CÓ NGỮ NGHĨA

Để tiện cho việc mô tả chức năng của bộ công cụ, phần này sẽ trình bầy sơ lược về các thành phần của Web có ngữ nghĩa. Các thành phần của Web có ngữ nghĩa được chia thành ba nhóm chính như sau:

- Ontology và các ngôn ngữ dùng để biểu diễn ngữ nghĩa thông tin.
- Các công cụ tạo nên phần ngữ nghĩa cũng như cấu trúc hạ tầng của Web có ngữ nghĩa.
- Các ứng dụng sử dụng Web có ngữ nghĩa.

Chức năng từng nhóm được trình bầy dưới đây.

2.1. Ngôn ngữ cho Web có ngữ nghĩa

Cơ chế cho phép chia sẻ và trao đổi ngữ nghĩa của thông tin được biết đến và sử dụng lâu nhất là *ontology*. Ontology là bản mô tả một cách tường minh các khái niệm trong một miền ứng dụng nào đó cùng với quan hệ giữa những khái niệm này. Ontology cung cấp từ vựng chung cho việc trao đổi thông tin giữa các ứng dụng và dịch vụ Web. Bản thân phần ngữ nghĩa của Web có ngữ nghĩa bao gồm ontology và giá trị cụ thể của khái niệm định nghĩa trong ontology. Để biểu diễn ontology và dữ liệu cần có ngôn ngữ thích hợp. Trong quá trình hình thành Web có ngữ nghĩa, nhiều ngôn ngữ như vậy đã được đề xuất và phát triển, trong đó được biết đến nhiều nhất là RDF và RDFS [2], DAML+OIL [8,9].

RDF và RDF Schema. RDF (Resource Description Framework) là cơ chế cho phép mô tả dữ liệu về dữ liệu (meta data). RDF coi các đối tượng trên Web (trang web, đoạn văn, người, các đối tượng khác.v.v.) là các tài nguyên. Mỗi tài nguyên được mô tả bởi bộ ba *đối tượng - thuộc tính – giá trị*. Ví dụ, mệnh đề "Phương là tác giả bài báo tại trang web nào đó" sẽ được mô tả bởi bộ ba: http:// www..., tác giả, "Phương". RDF Schema (RDFS) là một biến thể đơn giản sử dụng cơ chế RDF. RDFS cho phép mô tả các thuộc tính đặc thù cho ứng dụng, đồng thời định nghĩa lớp các đối tượng có cùng thuộc tính đó. Việc định nghĩa lớp đối tượng với thuộc tính và quan hệ rất cần thiết cho việc xây dựng ontology.

DAML + OIL. RDF và RDF Schema chỉ cho phép biểu diễn ngữ nghĩa ở mức độ đơn giản. Để biểu diễn ngữ nghĩa bao gồm nhiều đối tượng có quan hệ lôgic phức tạp với nhau cần các phương tiện biểu diễn mạnh hơn. DAML (Darpa Agent Markup Language) và OIL (Ontology Interface Layer) là các phương tiện như vậy. DAML+OIL là một mở rộng của RDFS. Trong DAML+OIL, ngữ nghĩa được mô tả thông qua lôgic mô tả (descriptive logic) cho phép sử dụng lôgic bool khi mô tả quan hệ giữc các đối tượng và có nhiều kiểu quan hệ cơ sở hơn so với RDFS.

2.2. Công cụ cho Web có ngữ nghĩa

Để tạo và sử dụng Web có ngữ nghĩa cần có sự hỗ trợ của các loại công cụ sau.

Công cụ tạo và liên kết ontology. Các công cụ này cho phép tạo ra khái niệm, thuộc tính của khái niệm, quan hệ và phân cấp giữa các khái niệm. Công cụ loại này thường có giao diện đồ hoạ và tuân theo chuẩn của ứng dụng web. Ví dụ điển hình cho công cụ loại này là Protégé [11].

Công cụ chú giải (annotation tools). Công cụ chú giải cho phép tạo phần ngữ nghĩa, tức là giá trị cụ thể của khái niệm, thuộc tính và quan hệ từ dữ liệu thông thường (trang web) phù hợp với một ontology nào đó. Giá trị tạo ra có thể được biểu diễn bởi các ngôn ngữ được nhắc

tới ở phần trên. Hiện nay đa số công cụ chỉ cho phép chú giải bằng tay, do vậy quá trình chú giải thường đòi hỏi nhiều thời gian [6].

Các kho chứa. Sau khi tạo ra, các ontology và phần ngữ nghĩa phải được lưu vào kho chứa. Những kho này thực chất là cơ sở dữ liệu cho phép lưu các mô tả trên ngôn ngữ RDFS hay DAML+OIL và cho phép biến đổi câu truy vấn trên những ngôn ngữ này thành câu truy vấn SQL. Một trong những kho chứa điển hình là Sesame [7].

Dịch vụ suy diễn. Dịch vụ suy diễn cho phép tìm ra giá trị cụ thể của các khái niệm hoặc thuộc tính tương ứng với ontology có trong kho chứa. Một ví dụ hệ thống suy diễn kiểu này là Ontobroker [5].

2.3. Các ứng dụng

Web có ngữ nghĩa cho phép tăng cường chức năng, mức độ thông minh và tính tự động hoá của nhiều ứng dụng hiện có. Những lĩnh vực ứng dụng đặc biệt hứa hẹn cho Web có ngữ nghĩa là các dịch vụ Web, quản lý tri thức và thương mại điện tử [3].

Dịch vụ Web là các chương trình và thiết bị có thể truy cập thông qua hạ tầng WWW. Web có ngữ nghĩa cung cấp thông tin và tri thức cần thiết cho việc tìm kiếm, tương tác, chia sẻ và kết hợp các dịch vụ Web.

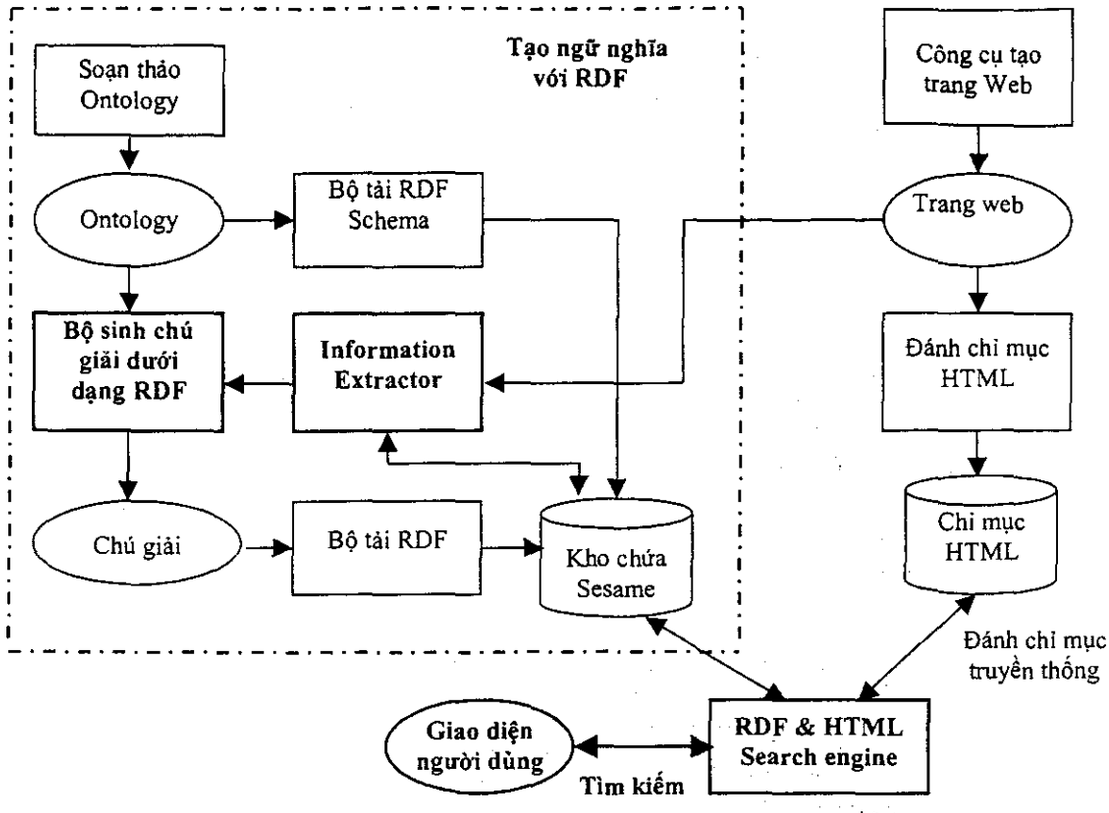
Quản lý tri thức liên quan đến việc thu thập, lưu trữ, tìm kiếm, truy cập và cung cấp thông tin, tri thức trong các tổ chức với mục đích tận dụng tài sản trí tuệ của chính tổ chức đó. Công việc này đòi hỏi một số chức năng hoàn chỉnh hơn các hệ thống quản lý văn bản hoặc dữ liệu thông thường như tìm kiếm thông minh, tự động tách thông tin từ văn bản, liên kết cơ sở dữ liệu, tự động tổng hợp văn bản. Những chức năng này có thể thực hiện được trên hạ tầng mà Web có ngữ nghĩa cung cấp.

Sự phát triển mạnh của *thương mại điện tử* hiện nay dẫn đến số lượng khổng lồ các giao dịch trên mạng. Để tự động hoá những giao dịch này, phần mềm hỗ trợ cần có khả năng: chuyển đổi giữa những dạng văn bản tồn tại trong giao dịch điện tử, hỗ trợ ontology mô tả hàng hoá và dịch vụ cho phép các agent tìm kiếm, phân loại và thương lượng về hàng hoá.

3. KHÁI QUÁT VỀ BỘ CÔNG CỤ

Mục tiêu của bộ công cụ là hỗ trợ toàn bộ quá trình tạo lập, lưu trữ và truy vấn phần ngữ nghĩa của trang web. Quá trình này đòi hỏi sự hỗ trợ của nhiều công cụ riêng biệt. Mặc dù nhiều công cụ như vậy là những công cụ có sẵn song chúng tôi cho rằng, việc kết hợp chúng trong một hệ thống thống nhất (với một số chỉnh sửa nhất định) là cần thiết để hỗ trợ quá trình tạo lập và truy vấn Web có ngữ nghĩa một cách hoàn chỉnh và đồng bộ.

Ngoài những công cụ có sẵn, hệ thống còn có một số thành phần do chúng tôi tự xây dựng. Quan trọng nhất trong số đó là mô đun chú giải trang web tự động sử dụng kỹ thuật tách thông tin từ văn bản. Chi tiết về việc tách thông tin về văn bản sẽ được trình bầy chi tiết trong phần sau của bài báo. Để minh hoạ cho hoạt động của bộ công cụ tạo ngữ nghĩa, một số môđun tìm kiếm thông tin dựa trên phần ngữ nghĩa vừa tạo ra cũng được bổ sung vào hệ thống. Các thành phần của toàn hệ thống được thể hiện trên hình 1.



Hình 1. Bộ công cụ tạo Web có ngữ nghĩa và ứng dụng đi kèm

Trong hình vẽ trên, hình chữ nhật là các khối chức năng, hình elip biểu diễn thông tin hoặc dữ liệu sinh ra từ những khối chức năng đó. Các hình chữ nhật có đường bao đậm và nền xám là những thành phần do chúng tôi tự xây dựng, hình với đường bao nhạt là những thành phần có sẵn được tích hợp vào hệ thống. Các thành phần có sẵn bao gồm bộ soạn thảo ontology Protégé [11], kho chứa mô tả RDF Sesame [7], bộ tải RDF và RDFS, một phần máy tìm kiếm sử dụng từ khoá truyền thống.

Phần tạo ngữ nghĩa được thực hiện bởi các mô đun nằm trong hình chữ nhật không liền nét ở góc trên bên trái. Đây cũng là phần chính của hệ thống. Phần ngữ nghĩa sinh ra sẽ được sử dụng cho ứng dụng tìm kiếm thông tin thông minh với máy tìm kiếm và giao diện thể hiện ở phía dưới hình vẽ. Để đảm bảo tính tương thích của phần tìm kiếm cho Web truyền thống (không có ngữ nghĩa), hệ thống còn bao gồm mô đun đánh chỉ mục HTML theo từ khoá (ở phía bên phải trên hình vẽ).

Hệ thống hoạt động như sau.

- Trước tiên, người sử dụng tạo ra ontology cho một miền ứng dụng cụ thể nhờ công cụ soạn thảo ontology. Sau đó ontology được chuyển thành mô tả trên RDFS và được chứa trong kho chứa Sesame.
- Sau khi đã tạo được ontology, bước tiếp theo là chú giải các trang web, tức là thêm phần ngữ nghĩa cho trang web bằng cách điền giá trị cho các khái niệm và thuộc tính trong ontology bằng thông tin lấy từ trang web. Thông thường, việc chú giải được thực hiện bằng tay. Với số lượng trang web lớn, công đoạn này đòi hỏi nhiều thời gian và dễ

sinh lỗi như để thiếu chú giải, chú giải không chính xác. Bộ công cụ của chúng tôi cho phép giải quyết vấn đề đó nhờ mô đun tách thông tin từ trang web và tạo chú giải tự động. Để chú giải cho một trang web, trang web được đưa và mô đun tách thông tin tự động. Dựa trên cấu trúc ontology, mô đun này tách từ trang web những thông tin về giá trị cụ thể của khái niệm và thuộc tính chứa trong ontology.

- Thông tin được tách ra ở bước trên được đưa sang bộ sinh chú giải. Mô đun này có nhiệm vụ tạo các bộ ba RDF mô tả những thông tin được tách ra và chuyển mô tả vừa được tạo ra sang kho chứa Sesame.
- Song song với quá trình trên, trang web cũng được đánh chỉ mục theo từ khoá như cách truyền thống.
- Cuối cùng, phần ngữ nghĩa được sử dụng trong máy tìm kiếm. Máy tìm kiếm sử dụng ngôn ngữ RQL để truy vấn kho chứa, đồng thời kết hợp với cơ chế suy diễn dựa trên ngữ nghĩa để đưa ra kết quả tìm kiếm thông minh. Câu truy vấn có thể được cho dưới dạng ngôn ngữ tự nhiên. Trong trường hợp đó, phần ngữ nghĩa của câu truy vấn được tách ra cùng bằng kỹ thuật tương tự như phần tách thông tin phục vụ chú giải.

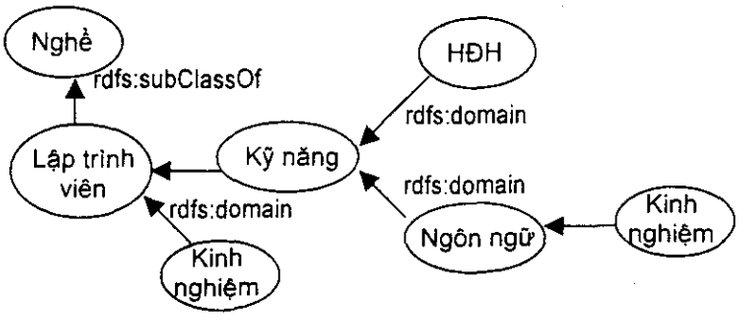
4. TÁCH THÔNG TIN TỪ VĂN BẢN VÀ CHÚ GIẢI TỰ ĐỘNG

Nhiệm vụ của khối tách thông tin từ văn bản là phát hiện những thông tin, dữ liệu tương ứng với các khái niệm trong ontology, tách những thông tin này và chuyển cho khối sinh chú giải. Ví dụ, xét đoạn văn bản sau lấy từ trang web đăng thông tin tuyển dụng lao động.

> Cần tuyển lập trình viên cho dự án thương mại điện tử. Ứng viên cần có ít nhất bốn năm kinh nghiệm, có khả năng làm việc với hệ điều hành Windows và Unix. Ứng viên phải sử dụng thành thạo các ngôn ngữ lập trình Java, Javascript, đặc biệt phải có kinh nghiệm lập trình Java không dưới ba năm. Ưu tiên những ứng viên có kỹ năng làm việc với cơ sở dữ liệu Oracle.

Giả sử ontology có các khái niệm, thuộc tính và quan hệ như mô tả trên hình 2. Quá trình tách thông tin phải cho kết quả sau:

```
nghề: lập trình viên
lập trình viên: kinh nghiệm :bốn năm
kỹ năng:
   hệ điều hành: Windows, Unix
   ngôn ngữ:  Javascript
              Java: kinh nghiệm :ba năm.
```



Hình 2: Một ví dụ ontology (không đầy đủ)

Có nhiều kỹ thuật tách thông tin được đề cập đến trong các nghiên cứu [4,10,12]. Do văn bản cần chú giải là văn bản có cấu trúc yếu (viết dưới dạng ngôn ngữ tự nhiên), đồng thời thông

tin tách ra phải có cấu trúc như ontology quy định nên chúng tôi đã lựa chọn kỹ thuật tách thông tin mô tả trong [4] - kỹ thuật cho phép thoả mãn tốt nhất hai yêu cầu này. Chúng tôi cũng thực hiện một số sửa đổi để quá trình tách thông tin phù hợp hơn với yêu cầu bộ công cụ [13].

Quá trình tách thông tin bao gồm những bước sau:

Bước1: Nhận biết *hằng* và *từ khoá*. Hằng là giá trị cụ thể của khái niệm hay thuộc tính chứa trong ontology. Từ khoá là từ hoặc cụm từ cho phép xác định hằng thuộc về khái niệm hay thuộc tính nào. Chẳng hạn, trong ví dụ trên "Java" là một hằng, còn "ngôn ngữ lập trình" là từ khoá cho biết hằng đó thuộc về thuộc tính "ngôn ngữ" của khái niệm "kỹ năng".

Hằng và từ khoá được xác định bằng cách sử dụng các quy tắc. Quy tắc ở đây là các mẫu được biểu diễn dưới dạng regular expression (như ở trong Perl) nhưng được mở rộng thêm bởi một số từ vựng. Ví dụ, mẫu nhận dạng thời gian kinh nghiệm được cho như sau

```
Lập trình viên: Kinh nghiệm case insensitive
    constant {extract Số, "[a-zA-Z\s]*\s+năm" };
    lexicon {Số case insensitive, filename "number.dat" };
    keyword {"\bkinh nghiệm\b"}
end;
```

Mẫu trên cho biết thuộc tính "kinh nghiệm" của "lập trình viên" được nhận dạng bởi biểu thức bắt đầu bởi một "Số", kết thúc bởi "năm"; "Số" là từ vựng chứa trong file có tên "number.dat" (từ vựng này liệt kê các xâu "một", "hai", "ba".v.v.); từ khoá đi kèm là "\bkinh nghiệm\b".

Các mẫu nhận dạng hằng và từ khoá được chứa trong ontology cùng với mô tả khái niệm và thuộc tính. Như vậy, chúng tôi đã mở rộng ontology bình thường để chứa thêm những thông tin này.

Khi bắt đầu quá trình tách thông tin, tất cả các mẫu được lần lượt sử dụng để tìm kiếm các hằng và từ khoá có trong văn bản. Kết quả nhận dạng hằng và từ khoá được chứa trong bảng như mô tả ở bước 2.

Bước 2: Tạo bảng Tên|Giá trị|Vị trí. Những hằng và từ khoá được nhận dạng ở bước trên được chứa trong một bảng. Mỗi dòng của bảng này chứa tên của khái niệm hoặc thuộc tính ứng với hằng hay từ khoá tìm được, giá trị tìm được, vị trí bắt đầu và kết thúc trong văn bản. Từ khoá được phân biệt với hằng bằng cách cho tiền tố KEYWORD ở trước. Ví dụ, từ đoạn văn bản trong ví dụ trên, ta xây dựng được bảng sau (chỉ thể hiện một phần của bảng)

```
...
lập trình viên:kinh nghiệm|bốn năm|80|86
ngôn ngữ:kinh nghiệm|bốn năm|80|86
KEYWORD lập trình viên:kinh nghiệm|kinh nghiệm|88|98
KEYWORD kỹ năng:hệ điều hành|hệ điều hành|126|137
kỹ năng:hệ điều hành|Windows|139|145
kỹ năng:hệ điều hành|Unix|151|154
KEYWORD kỹ năng:ngôn ngữ|ngôn ngữ lập trình|196|213
kỹ năng:ngôn ngữ|Java|212|215
kỹ năng:ngôn ngữ|Javascript|218|227
kỹ năng:ngôn ngữ|Java|270|273
...
```

Bước 3: Tạo thông tin ứng với ontology từ bảng trên. Ở bước này, thông tin từ bảng Tên|Giá trị|Vị trí được sử dụng để sinh ra giá trị cho khái niệm và thuộc tính trong bảng. Thực

chất của bước này là giải quyết mâu thuẫn hoặc không rõ ràng về thông tin trong bảng bằng cách sử dụng một số quy tắc heuristic. Ví dụ, trong bảng trên, ta thấy "bốn năm" được nhận dạng ở bước 1 vừa thuộc loại kinh nghiệm lập trình nói chung, vừa thuộc loại kinh nghiệm lập trình ngôn ngữ do phù hợp với mẫu của cả hai thuộc tính này. Hay "Java" cũng được nhận dạng hai lần, trong khi chỉ có thể cho một giá trị vào kho chứa. Ở đây, chúng tôi sử dụng một số heuristic sau:

- Nếu một khái niệm hoặc thuộc tính chỉ được phép có một giá trị nhưng bảng lại chứa nhiều giá trị thì chỉ giữ lại giá trị gần từ khoá tương ứng nhất. Ví dụ, trong bảng trên có hai hằng cho thuộc tính "lập trình viên:kinh nghiệm" là "ba năm" và "bốn năm". Heuristic này cho phép loại giá trị "ba năm" do nằm xa từ khoá "lập trình viên : kinh nghiệm".
- Nếu có nhiều hằng trùng nhau thì chỉ giữ lại hằng tương ứng với từ khoá gần nhất. Chẳng hạn, trong bảng trên có hai hằng "bốn năm" thì chỉ giữ lại hằng ứng với "lập trình viên:kinh nghiệm" vì nằm gần từ khoá này hơn.
- Nếu có nhiều giá trị hằng / từ khoá lồng nhau thì chỉ giữ lại hằng / từ khoá dài hơn. Chẳng hạn, từ khoá "kinh nghiệm lập trình" lồng từ khoá "kinh nghiệm" nhưng lại dài hơn, do đó chỉ giữ lại "kinh nghiệm lập trình" cho vị trí đó.
- Nếu một khái niệm chỉ có thể có một giá trị thì chọn hằng đầu tiên xuất hiện trong bảng.
- Các quan hệ một-nhiều thường được thể hiện bởi các hằng có vị trí lồng nhau trong văn bản.

Trong các quy tắc trên, khoảng cách dùng để so sánh được tính theo vị trí xuất hiện hằng và từ khoá trong văn bản. Sau khi áp dụng những heuristic trên, các hằng còn lại được chuyển sang bộ sinh chú giải để biến đổi về dạng RDF.

5. TRIỂN KHAI VÀ ỨNG DỤNG THỬ NGHIỆM

5.1. Triển khai hệ thống

Hệ thống được triển khai như một ứng dụng Web, mọi giao diện đều sử dụng web form và được hiển thị bằng trình duyệt. Lựa chọn này cho phép xây dựng và lưu trữ phần ngữ nghĩa tập trung trên máy chủ hoặc ngay trên máy cục bộ. Chúng tôi đã sử dụng những ngôn ngữ và công cụ sau để triển khai hệ thống.

Ngôn ngữ lập trình là ngôn ngữ Java. Java được lựa chọn do có nhiều ưu điểm: thích hợp với lập trình ứng dụng Web, cụ thể là hỗ trợ Servlet/JSP; là ngôn ngữ hoàn toàn hướng đối tượng; không phụ thuộc phần cứng và hệ điều hành, có thể kết nối với cơ sở dữ liệu thông qua JDBC. Ngoài ra, thư viện chuẩn của Java (từ phiên bản 1.4) hỗ trợ regular expression cần thiết cho phần tách thông tin.

Hệ thống bao gồm hai cơ sở dữ liệu, một dùng cho kho chứa Sesame và một chứa các thông tin quản lý của hệ thống. Cả hai đều được xây dựng sử dụng hệ quản trị CSDL MySQL. Đây là hệ quản trị cơ sở dữ liệu miễn phí với nhiều ưu điểm như nhanh, không đòi hỏi nhiều tài nguyên.

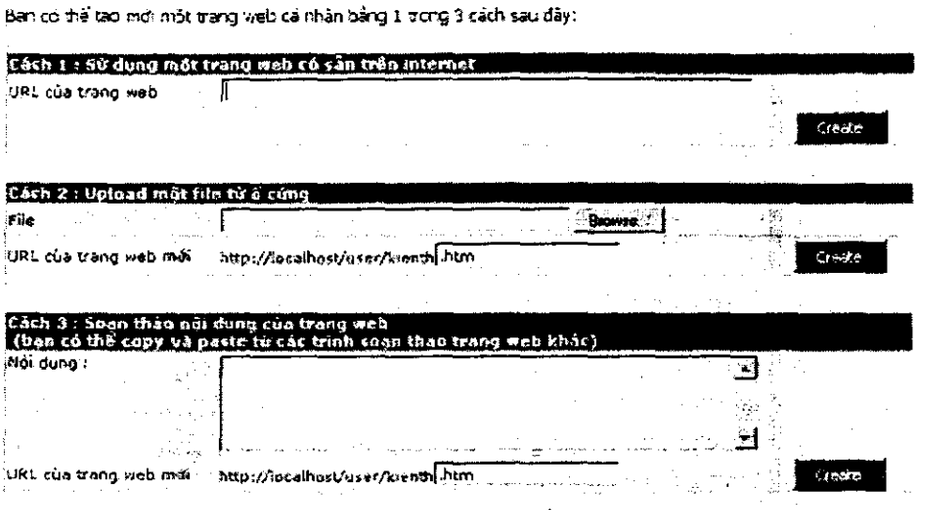
Phần mềm máy chủ Web là Tomcat 4.1 (http://jakarta.apache.org/tomcat). Đây là phần mềm miễn phí hỗ trợ Servlet / JSP.

Phần đánh chỉ mục trang web và tìm kiếm theo từ khoá được xây dựng dựa trên máy tìm kiếm Jakarta Lucene (http://jakarta.apache.org/lucene). Đây là máy tìm kiếm mã nguồn mở được viết trên Java và hỗ trợ nhiều tính năng tìm kiếm mở rộng với từ khoá.

5.2. Ứng dụng minh hoạ và thử nghiệm

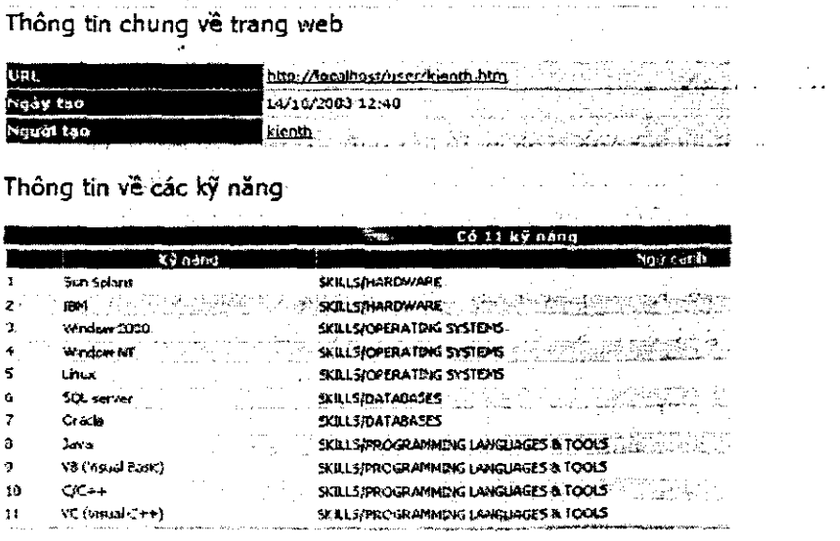
Với mục đích minh hoạ và thử nghiệm, hệ thống được sử dụng chú giải các trang web chứa thông tin cá nhân và kỹ năng của lập trình viên, sau đó phần tìm kiếm cho phép tìm kiếm thông tin về những người này dựa trên ngữ nghĩa hoặc từ khoá. Trước hết, một ontology về nghề lập trình và những kỹ năng, kinh nghiệm liên quan được tạo ra. Ontology này chỉ cần tạo một lần cho tất cả các trang web.

Sau khi có ontology, người dùng sử dụng giao diện của hệ thống để nhập trang web cần chú giải. Ở đây có thể tạo mới trang web và chú giải luôn hoặc tạo chú giải một trang có sẵn bằng cách tải trang đó lên. Giao diện nhập trang web cần chú giải được cho trên hình 3.



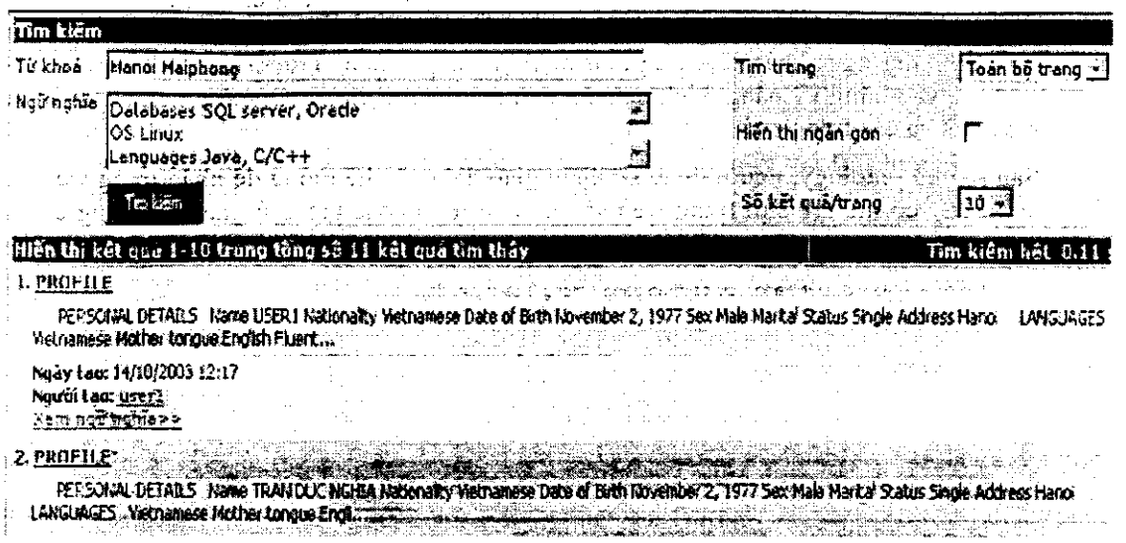
Hình 3. Nhập trang web cần chú giải

Sau khi xác định trang web cần tạo ngữ nghĩa và bấm nút "Create", bộ sinh tách thông tin sẽ sinh ra chú giải. Người dùng có thể xem những chú giải được tạo ra và có thể chỉnh sửa theo mong muốn. Hình 4 minh hoạ phần chú giải về kỹ năng của lập trình viên được tạo ra cho một trang web ví dụ



Hình 4: Chú giải về kỹ năng được tách từ trang web

Sau khi đã chú giải các trang web, người dùng có thể tìm kiếm thông tin theo từ khoá và/hoặc theo ngữ nghĩa như ví dụ trên hình 5.



Hình 5: Kết quả tìm kiếm kết hợp từ khoá và ngữ nghĩa

Để thử nghiệm hệ thống, chúng tôi sử dụng gần 200 trang thông tin cá nhân của lập trình viên đang làm việc tại trung tâm xuất khẩu phần mềm FPT Fsoft và một số trang lấy từ Internet. Những trang này được chú giải tự động bởi bộ công cụ, sau đó chú giải bằng tay và so sánh kết quả. Kết quả được đánh giá theo hai chỉ số *recall* (tỷ lệ thông tin tách được / thông tin có trong văn bản) và *precision* (tỷ lệ thông tin tách đúng / thông tin tách được). Thử nghiệm cho thấy, chất lượng xây dựng ontology ảnh hưởng nhiều nhất tới chất lượng chú giải. Sau khi hiệu chỉnh ontology, với gần 200 trang web cá nhân, giá trị của recall và precision tương ứng là 88% và 95%. Các chỉ số recall và precision như vậy là tương đối cao và phù hợp với đặc điểm của phương pháp tách thông tin đã lựa chọn. Kết quả chú giải tự động có thể hiệu chỉnh bằng tay sau đó để cho kết quả tốt nhất.

6. KẾT LUẬN

Bài báo trình bầy việc thiết kế và xây dựng bộ công cụ hỗ trợ tạo Web có ngữ nghĩa cùng với ứng dụng minh hoạ. Kết quả xây dựng công cụ cho thấy, việc sử dụng kỹ thuật tách thông tin từ văn bản cho phép giảm đáng kể thời gian chú giải thông tin trên trang Web, phần việc chiếm nhiều thời gian nhất khi tạo Web có ngữ nghĩa. Kết quả chú giải thông tin tự động có độ chính xác khá cao. Kinh nghiệm xây dựng bộ công cụ cũng cho thấy, việc hiệu chỉnh và tích hợp một số công cụ đã có sẵn cho phép giảm thời gian đồng thời tăng thêm tính năng bộ công cụ và tạo thuận lợi cho người sử dụng so với dùng từng công cụ riêng lẻ. Tuy nhiên, bộ công cụ còn thiếu một số chức năng tự động khác như tự động sinh ra ontology từ văn bản. Hướng phát triển tiếp theo là hoàn thiện và bổ sung các chức năng này.

7. LỜI CẢM ƠN

Nghiên cứu được thực hiện với sự hỗ trợ kinh phí của Hội đồng khoa học tự nhiên.

TÀI LIỆU THAM KHẢO

[1]. T. BERNERS-LEE, J. HENDLER, O. LASSILA, The Semantic Web, Scientific American, May 2001.

[2]. D. BRICKLEY, R.V. GUHA, Resource Description Framework (RDF) Schema Specification, World Wide Web Consortium, Proposed recommendation 2001.

[3]. Y. DING, D. FENSEL, M. KLEIN, B. OMELAYENKO, The semantic web: yet another hip? Data & Knowledge Engineering 41, Elsevier 2002, pp 205–227.

[4]. D.W. EMBLEY, D.M. CAMPBELL, R.D. SMITH, S.W. LIDDLE, Ontology-Based Extraction and Structuring of Information from Data-Rich Unstructured Documents, Proc. of 1998 ACM Inter. Conf. on Inform. and Knowledge Man., CIKM 1998, USA, pp 52-59

[5]. D. FENSEL, S. DECKER, M. ERDMANN, H.-P. SCHNURR, R. STUDER, A. WITT, Lessons learned from applying AI to the web, Journal of Cooperative Information Systems 9 (4) (2000).

[6]. S. HANDSCHUH, S. STAAB, CREAM – Creating metadata for the semantic web, Computer networks, vol. 42, Elsevier 2003, pp 557-571.

[7]. http://sesame.aidministrator.nl/.

[8]. http://www.ontoknowledge.org/oil.

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

[10]. N. KUSHMERIC, Wrapper induction: efficiency and expressiveness, Artificial intelligence, vol.118,2000.

[11]. N. F. NOY, M. SINTEK, S. DECKER, M. CRUBÉZY, R. W. FERGERSON, M. A. Musen, Creating semantic web contents with Protégé -2000, IEEE Intelligent systems, 3-4/2001, pp 60-71.

[12]. S. SODERLAND, Learning information extraction rules for semi-structured and free text. Machine learning, 34. Kluwer Academic Publishers.(1999)

[13]. TU MINH PHUONG, Information Extraction and Evaluation of Candidates with Fuzzy Set techniques, Proc. of Inter. Conf. on Fuzzy syst. and Knowl. discovery, FSKD 2002, Singapore, 2002, pp 481-485.+