

# XÂY DỰNG ỨNG DỤNG TÌM KIẾM SÁCH ĐIỆN TỬ BẰNG PHƯƠNG PHÁP TÌM KIẾM NGỮ NGHĨA TRÊN ONTOLOGY

NGUYỄN PHÚC HÙNG<sup>1,\*</sup>, ĐẶNG THỊ PHÚC<sup>2</sup>, ĐINH CÔNG THẮNG<sup>3</sup>

<sup>1</sup> Trung tâm Quản trị Hệ thống, trường Đại học Công nghiệp Thành phố Hồ Chí Minh

<sup>2</sup> Khoa Công nghệ thông tin, trường Đại học Công nghiệp Thành phố Hồ Chí Minh

<sup>3</sup> Công ty KMS Technology, Việt Nam

\*Tác giả liên hệ: [nguyenphuchung@iuh.edu.vn](mailto:nguyenphuchung@iuh.edu.vn)

Dois: <https://doi.org/10.46242/jstiuh.v61i07.4728>

**Tóm tắt.** Tìm kiếm dựa trên ngữ nghĩa thông qua thông tin truy vấn là một bước quan trọng giúp nâng cao hiệu quả truy xuất thông tin. Trong bài báo, chúng tôi đề xuất mô hình ngữ nghĩa ontology và các phương pháp tiếp cận để cải thiện mức độ liên quan trong việc truy xuất thông tin. Chúng tôi xây dựng hệ thống tìm kiếm sách điện tử cho phép người dùng tìm kiếm theo ngữ nghĩa với khoảng 360 đầu sách được lưu trữ trong ontology. Hệ thống được triển khai với sự hỗ trợ của các công nghệ Resource Description Framework (RDF) ngôn ngữ ontology cho web (OWL – Ontology Web Language). Để thực hiện truy vấn, chúng tôi đã sử dụng ngôn ngữ truy vấn SPARQL ở định dạng RDF được sử dụng để truy xuất dữ liệu đang được lưu trữ. Ứng dụng cung cấp một hệ thống mã hóa đơn giản và nhất quán hỗ trợ người sử dụng tìm kiếm và truy hồi thông tin một cách hiệu quả, tiết kiệm thời gian tìm kiếm.

**Từ khóa.** Ontology, Web ngữ nghĩa, RDF, OWL, SPARQL, ứng dụng tìm kiếm sách điện tử.

## 1 GIỚI THIỆU

Ngày nay, với sự phát triển của công nghệ thông tin song song với sự phát triển mạnh mẽ của web đã dẫn đến sự bùng nổ về dữ liệu. Mặc dù web 2.0 đã đạt được những thành tựu rất đáng kể trong việc nâng cao tính tương tác cũng như ghi nhận đặc điểm yêu cầu tìm kiếm của từng người dùng. Tuy nhiên trong xã hội thông tin đương đại nhu cầu của người dùng không dừng lại ở đó mà còn phải cải thiện được chất lượng xử lý của trang web theo yêu cầu ngày càng nâng cao. Web 3.0 (Semantic Web) ra đời nhằm đáp ứng những yêu cầu về chất lượng đó [1-3].

Công nghệ ngữ nghĩa là một phương pháp biểu diễn dữ liệu dựa trên quy tắc được sử dụng để mô tả dữ liệu và các thực thể trong thế giới thực để máy móc có thể hiểu rõ hơn về chúng. Web ngữ nghĩa phát triển bên cạnh sự phát triển mạnh mẽ của ontology [4]. Nhiều nghiên cứu cải tiến khả năng truy xuất thông tin của ontology trên Web semantic bằng sự kết hợp với Sơ đồ tri thức (Knowledge Graph) [5]. Một số nghiên cứu gần đây đã ứng dụng Web ngữ nghĩa kết hợp với ontology như trong lĩnh vực y khoa [6], công nghiệp [7] và ứng dụng trên nhiều nền tảng khác nhau [8, 9]. Ontology đóng vai trò then chốt trong việc cung cấp tài nguyên có thể truy cập để xử lý tự động bằng cách cung cấp bộ từ vựng cho việc đánh dấu ngữ nghĩa của Web ngữ nghĩa, cũng như cung cấp cấu trúc thống nhất trong việc tìm kiếm ngữ nghĩa được hiệu quả [10,11]. Nhiều mô hình, ngôn ngữ và công cụ hỗ trợ cho Web ngữ nghĩa như XML (eXtended Markup Language)[12], RDF (Resource Description Framework)[13], OWL (Web Ontology Language)[14], Jess, Jena, SWRL (Semantic Web Rule Language)[15] giúp mô tả ngữ nghĩa và lý luận về siêu dữ liệu, tài nguyên và xác định mối quan hệ giữa các thực thể có sẵn trên web, các nền tảng hỗ trợ mô hình hoá các ontology như Stanford Medical Informatics đã phát triển open-source ontology Protégé-OWL[12], W3C đã đề xuất SPARQL để đại diện cho đồ thị RDF – trong đó chứa chủ đề và vị từ [16]. Ngày nay, ontology đã trở thành một chủ đề được nghiên cứu nhiều trong các lĩnh vực thương mại điện tử, quản lý tri thức, xử lý ngôn ngữ tự nhiên và đạt được nhiều thành tựu đáng kể.

Trong bài báo, chúng tôi xây dựng hệ thống tìm kiếm sách điện tử bằng phương pháp tìm kiếm ngữ nghĩa dựa trên ontology đã được thiết kế. Phương pháp được ứng dụng trong hệ thống gồm 3 mục đích sau: Phân tích từ khóa tìm ra các mối quan hệ trong ontology, tìm kiếm ngữ nghĩa dựa trên các mối quan hệ giữa các thực thể trong ontology và xây dựng câu truy vấn Sparql phục vụ cho việc tìm kiếm ngữ nghĩa.

## 2 CÁC THUẬT TOÁN LIÊN QUAN

### 2.1 Ontology

Ontology [5,6] là tập từ vựng để mô hình hóa thế giới bên ngoài, nó đưa ra các khái niệm cơ bản và định nghĩa quan hệ giữa các khái niệm đó trong một miền lĩnh vực. Đồng thời ontology còn cung cấp các ràng buộc, là các cơ sở về ý nghĩa mong muốn của bộ từ vựng. Ontology tạo điều kiện cho việc chia sẻ kiến thức qua các hệ thống phân tán, hay nói cách khác nó hỗ trợ các hệ thống, ứng dụng tương tác với nhau bằng cách cung cấp một hệ thống hiểu biết chung. Ontology được sử dụng như là một biểu mẫu trình bày tri thức về thế giới hay một phần của nó nên ontology thường được mô tả bao gồm cá thể, lớp, thuộc tính, mối liên hệ. Sức mạnh của ontology nằm ở khả năng diễn đạt các mối quan hệ giữa các thực thể và tập hợp các quan hệ cùng nhau để mô tả ngữ nghĩa của miền tri thức. Tập các dạng quan hệ được sử dụng và cây phân loại thứ bậc của chúng thể hiện sức mạnh diễn đạt của ngôn ngữ dùng để biểu diễn ontology.

Các ontology được chia thành hai loại:

1. Ontology gọn nhẹ (Light-weight Ontologies): bao gồm phân cấp lớp hoặc phân loại các lớp, lớp con, giá trị và thuộc tính.

2. Ontology đầy đủ (Heavy-weight Ontologies): đại diện cho lĩnh vực theo cách chuyên sâu bao gồm tiên đề và ràng buộc.

Để thực hiện chức năng của web ngữ nghĩa, cần phải cung cấp một cơ chế mô tả ngữ nghĩa có cấu trúc mà máy tính có thể hiểu được và phát triển một loạt các quy tắc để suy luận tự động. Thách thức của web ngữ nghĩa là cung cấp một ngôn ngữ có thể diễn đạt dữ liệu và các quy tắc lập luận. Mô hình ngữ nghĩa ontology có ý nghĩa lớn đối với việc quản lý và tái sử dụng thông tin trong quá trình thiết kế. Ontology đóng một vai trò quan trọng trong việc đạt được khả năng tương tác giữa các tổ chức trên Web ngữ nghĩa. Do đó, nó cần thiết để thiết lập ánh xạ giữa khái niệm về các ontology khác nhau để nắm bắt sự tương ứng ngữ nghĩa giữa chúng. Tuy nhiên, việc thiết lập sự tương tác này không phải là một việc dễ dàng.

### 2.2 Mô hình ngữ nghĩa ontology

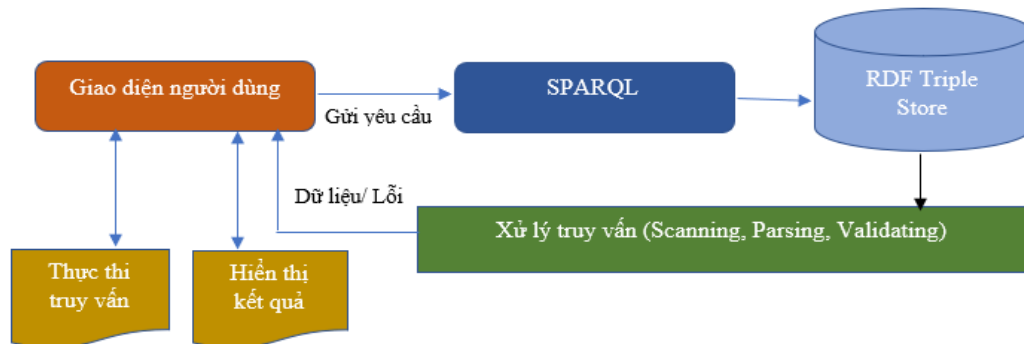
Mô hình dựa trên web ngữ nghĩa chủ yếu bao gồm XML, RDF và OWL. Mô hình dữ liệu XML có ưu điểm là định dạng lưu trữ dữ liệu, khả năng mở rộng cao, tự mô tả và khả năng tương tác. Khuyết điểm là khả năng diễn đạt những kiến thức phức tạp còn hạn chế. Do đó, XML chỉ được sử dụng ở cấp độ ngữ pháp để mô tả cấu trúc và nội dung của dữ liệu. RDF là một giải pháp cho vấn đề của các mô hình XML. RDF cung cấp một khuôn khổ chính thức cho thông tin ngữ nghĩa. Nó mạnh hơn XML trong việc biểu diễn tri thức. RDF và RDF schema (RDFS) [17] là mô hình các ngôn ngữ web được phát triển để đại diện hoặc thể hiện cho các ontology. RDF/RDFS cho phép biểu diễn một vài tri thức trong ontology và mô hình cơ bản RDF/RDFS liên quan đến tổ chức bộ từ vựng trong việc phân cấp như: các quan hệ lớp con, thuộc tính con, các ràng buộc domain và range, các thể hiện của lớp, tuy nhiên hạn chế của RDF/RDFS khi một số đặc điểm bị thiếu như: *Tách rời các lớp*, trong RDFS chỉ có thể để ở quan hệ lớp con. *Phạm vi giá trị của thuộc tính*: trong RDF Schema không thể khai báo các ràng buộc với dải giá trị chỉ áp dụng với một vài lớp nhất định. *Kết hợp các lớp với nhau*: khi xây dựng một lớp mới được tạo ra từ việc kết hợp các lớp khác sử dụng hợp hoặc giao hoặc bù, RDF Schema không cho phép định nghĩa như vậy. *Các ràng buộc tồn tại*: khi muốn các ràng buộc bao nhiêu giá trị thuộc tính phân biệt có thể có hoặc bắt buộc, kiểu ràng buộc này không thể biểu diễn trong RDF Schema được. *Các đặc điểm cụ thể của thuộc tính*: hữu ích khi một thuộc tính là bắc cầu hoặc nghịch đảo của thuộc tính khác, trong RDF Schema không biểu diễn được.

Vì những hạn chế này, ngôn ngữ ontology cho web (OWL – Ontology Web Language) ra đời đưa ra tiêu chuẩn cho ontology trong sử dụng trong web. Mục đích chính của OWL là cung cấp các chuẩn để tạo ra một nền tảng quản lý tài nguyên và chia sẻ cũng như tái sử dụng dữ liệu trên Web. OWL được thiết kế như lớp trên cùng của RDF, được sử dụng để xử lý thông tin trên web. OWL cung cấp khả năng thông dịch máy ở mức độ cao hơn so với XML và RDF. OWL mô tả ngữ nghĩa của các nguồn tri thức và mối liên hệ giữa chúng. So với XML và RDF, OWL có khả năng biểu đạt ngữ nghĩa phong phú để chia sẻ kiến thức.

### 2.3 Ngôn ngữ truy vấn SPARQL

Để truy xuất dữ liệu hiệu quả và hiệu quả từ RDF lớn như vậy, ngôn ngữ truy vấn SPARQL (Simple Protocol And RDF Query Language) được sử dụng [13]. Ngôn ngữ truy vấn SPARQL là một ngôn ngữ

được phát triển bởi nhóm RDF Data Access Working Group – một phần trong hoạt động của web ngữ nghĩa. SPARQL là một ngôn ngữ để truy cập thông tin từ các đồ thị RDF, thông qua các bộ 3 triple store thực hiện xử lý các truy vấn và trả kết quả về cho người sử dụng. Kiến trúc SPARQL được mô tả ở hình 3.

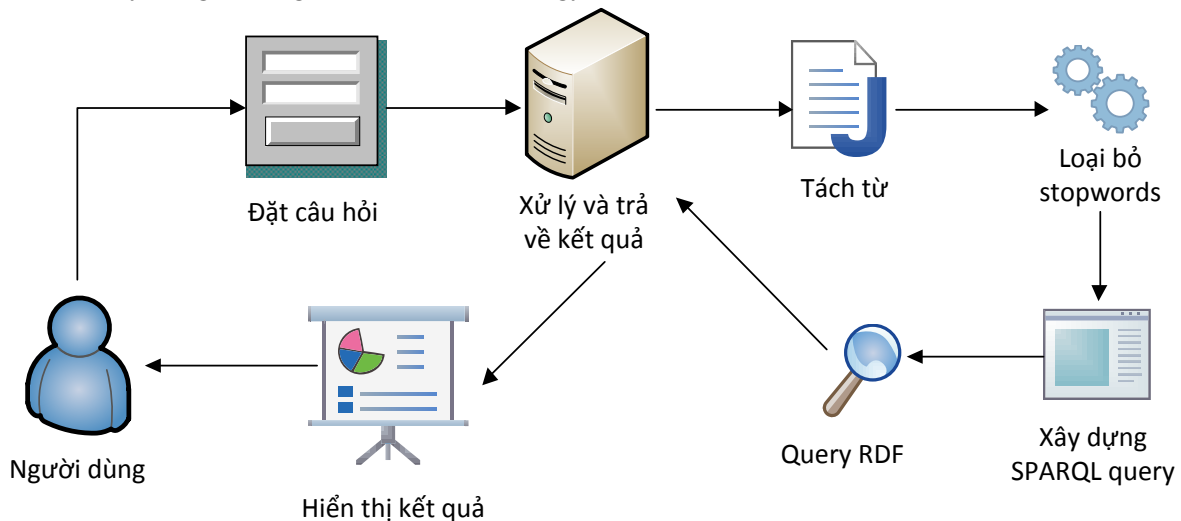


Hình 1. Kiến trúc thực thi SPARQL

Mô hình xử lý và tối ưu truy vấn cho SPARQL bao gồm các bước: đọc truy vấn từ phía người dùng, tách từ, viết lại câu truy vấn dựa vào truy vấn từ phía người dùng, tối ưu thực hiện truy vấn và trả kết quả về cho người dùng. Bước xử lý truy vấn là quá trình xử lý nội bộ được thực hiện trong công cụ truy vấn để đánh giá và chuyển đổi truy vấn, thực thi và sử dụng phương pháp viết lại nếu cần thiết mà không làm thay đổi kết quả truy vấn. Bước tối ưu hóa truy vấn trình bày các quy tắc để viết lại truy vấn bằng cách chuyển đổi cấu trúc thực thi một truy vấn. Ngôn ngữ truy vấn SPARQL được sử dụng để tăng tốc và triển khai các ứng dụng web 3.0 và cung cấp tiêu chuẩn của web.

## 2.4 Mô hình xây dựng chương trình dựa trên ontology

Mô hình xây dựng chương trình dựa trên ontology được mô tả ở hình 2.



Hình 2. Mô hình tổng quan các bước xây dựng chương trình

Mô hình bao gồm các bước:

**Tách từ:** trong bài báo chúng tôi sử dụng phương pháp tách từ sử dụng vnTokenizer đây là công cụ tách từ tự động cho tiếng Việt được viết bằng ngôn ngữ java. Công cụ này được xây dựng sử dụng kết hợp từ điển (từ điển tiếng Việt được lấy từ đề tài VLSP) và giải thuật N-gram, trong đó mô hình N-gram được huấn luyện sử dụng Treebank tiếng Việt (70,000 câu đã được tách từ). VietTreeBank là kho ngữ liệu câu đã được chú giải ngữ pháp.

Chúng tôi đánh giá mô hình bằng cách chia bộ dữ liệu VietTreeBank [18] đã được chuyển đổi bao gồm

## XÂY DỰNG ỨNG DỤNG TÌM KIẾM SÁCH...

10.350 câu thành 2 phần ngẫu nhiên theo tỉ lệ là 7:3. Phần chiếm 70% sử dụng làm tập dữ liệu để MaltParser học, phần 30% sử dụng để MaltParser phân tích dựa vào tập dữ liệu vừa học được. Sau đó tiến hành đo độ chính xác của tập dữ liệu vừa được phân tích với bộ dữ liệu gốc. Sự so sánh sẽ dựa trên hướng của quan hệ và nhãn của quan hệ. MaltParser đánh giá bằng cách sử dụng lược đồ xác thực chéo k lần (k- fold cross validation scheme). Trong bài báo, chúng tôi sử dụng lược đồ xác thực chéo 5 lần, kết quả đánh giá thể hiện trong bảng 1.

Bảng 1. Kết quả đánh giá

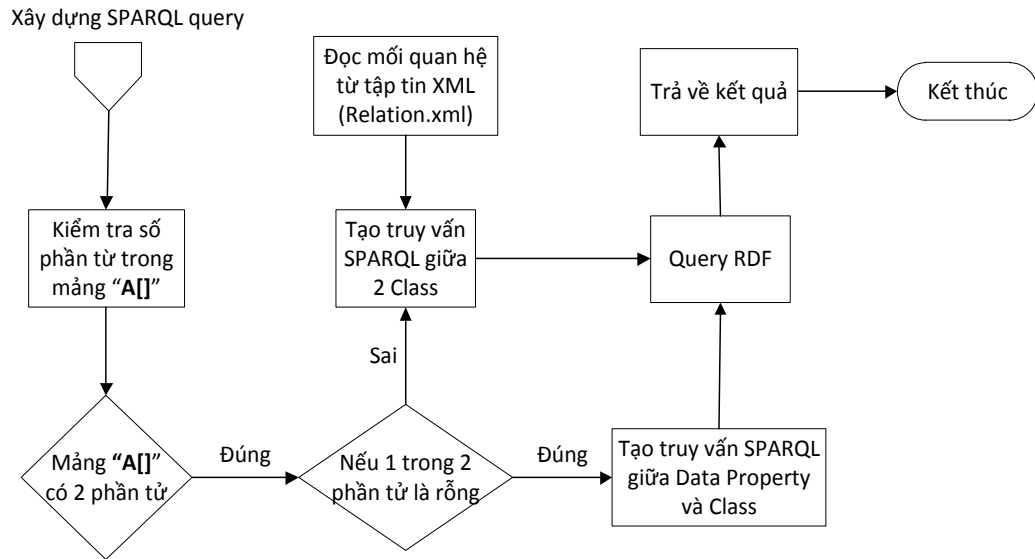
TT	Tập học (câu)	Tập thử nghiệm (câu)	Số câu đúng	Độ chính xác (%)
1	7245	3105	2170	69,88
2	7240	3110	2172	69,84
3	7248	3102	2165	69,79
4	7244	3106	2169	69,83
5	7250	3100	2164	69,80
		Trung bình		69,82

**Loại bỏ stopword:** Để việc truy vấn dữ liệu ngữ nghĩa được hiệu quả thì ngoài việc tách từ thì việc loại bỏ stopword là một phần quan trọng trong việc xác định những thành phần chủ yếu nhằm đem lại hiệu quả trong quá trình truy vấn dữ liệu, do những từ này không có ý nghĩa gì trong việc xử lý mà nó còn có thể gây nhiễu cho việc phân tách từ. Chẳng hạn từ ví dụ trên “thủ đô bagdad của nước nào?” sau khi tách từ và loại bỏ những từ gây nhiễu chúng ta sẽ có được những từ sau: “*thủ đô*”, “*bagdad*”, “*nước*”, những từ như “*của*”, “*nào*” và dấu chấm hỏi “*?*” là những từ gây nhiễu đã được loại bỏ.

**Xác định ngữ nghĩa các lớp:** Sau khi công đoạn tách từ và loại bỏ stopword, chúng tôi tiến hành việc xác định xem những từ loại được tách này thuộc là những từ thuộc lớp hay thuộc tính nào. Thông thường, trong câu truy vấn người dùng tìm kiếm luôn kèm theo thông tin cần tìm kiếm, do đó có thể xác định từ đó thuộc loại nào trong ontology, chẳng hạn như “*thủ đô*”, “*bagdad*”, “*nước*” khi tìm kiếm những từ này trong ontology chúng ta sẽ thấy “*thủ đô*” là tên của lớp, “*bagdad*” là thể hiện của lớp “*thủ đô*” và “*nước*” là từ loại thuộc lớp “*quốc gia*”.

Tiếp theo các cụm từ này sẽ được duyệt qua để kiểm tra xem chúng có thuộc lớp nào hay không. Nếu thuộc lớp mà chưa tồn tại trong mảng A thì thêm vào mảng A, nếu cụm từ này không thuộc lớp nhưng nó thuộc Data Property thì kiểm tra xem chúng đã tồn tại trong mảng A chưa, nếu chưa tồn tại thì thêm vào mảng A. Ví dụ, duyệt qua cụm từ: *thủ đô*, *bagdad*, *nước*, ta có *thủ đô* thuộc Class và chưa có phần tử nào trong mảng A nên ta thêm vào mảng:  $A = \{thủ\ đô\}$ , *bagdad* thuộc lớp quốc gia và trong mảng A vẫn chưa có quốc gia nên ta thêm quốc gia vào mảng A, đồng thời gán  $PT = bagdad$ , khi đó ta có  $A = \{thủ\ đô, quốc\ gia\}$ ,  $PT = bagdad$ , nước là từ đại diện cho quốc gia nhưng vì quốc gia đã tồn tại trong mảng A nên ta sẽ bỏ qua từ này. Kết quả cuối cùng ta sẽ có, mảng A gồm 2 phần tử, biến PT chứa 1 giá trị:  $A = \{thủ\ đô, quốc\ gia\}$ ,  $PT = bagdad$ . Tìm kiếm quan hệ các lớp trên ontology

**Xây dựng truy vấn Sparql:** Mô hình xây dựng truy vấn được mô tả ở hình 3.



Hình 3. Xây dựng truy vấn Sparql

Để tạo truy vấn SPARQL chúng tôi tiến hành kiểm tra số phần tử trong mảng A, nếu số phần tử trong mảng A = 2 thì ta kiểm tra 1 trong 2 phần tử này có phần tử nào là rỗng hay không, nếu có 1 trong 2 phần tử này là rỗng thì ta xây dựng SPARQL giữa Data Property và Class. Đối với câu hỏi “Dân số Việt Nam là bao nhiêu?” thông qua tách từ ta sẽ được: “Dân số”, “Việt Nam”, “là”, “bao nhiêu”, “?”, lọc bỏ những từ không liên quan ta sẽ được “Dân số”, “Việt Nam”, duyệt qua 2 phần tử này thì “Dân số” là Data Property nên ta gán biến “DT” = “Dân số” đồng thời thêm vào mảng A giá trị rỗng, A = { “” }, phần tử “Việt Nam” thuộc lớp “Quốc gia” và “Quốc gia” chưa có phần tử nào trong mảng A, nên ta sẽ thêm “Quốc gia” vào mảng A đồng thời gán PT = “Việt Nam”, khi đó chúng ta sẽ có: A = { “”, “Quốc Gia” }, DT = “Dân số”, PT = “Việt Nam”. Khi đó SPARQL giữa Data Property và Class sẽ như sau:

```

Prefix rdf: <http://www.owl-ontologies.com/thegioi.owl#>
Select ?ketqua
Where
{
    ?rdf:việt_nam    ?rdf:dân_số        ?ketqua
}
  
```

Ngược lại khi mảng A chứa giá trị là 2 lớp thì chúng ta sẽ tìm mối quan hệ giữa 2 lớp để xây dựng câu truy vấn SPARQL. Nếu giữa 2 lớp chỉ có một mối quan hệ thì ta sẽ gán mối quan hệ vào phần tử quan hệ:

```

Prefix rdf: <http://www.owl-ontologies.com/thegioi.owl#>
Select ?ketqua
Where
{
    ?rdf:bagdad        ?rdf:là_thủ_đô_của    ?ketqua
}
  
```

Vậy kết quả Bagdad là thủ đô của Iraq. Nếu giữa 2 lớp có nhiều hơn một mối quan hệ, chúng ta sẽ duyệt qua các mối quan hệ.

```

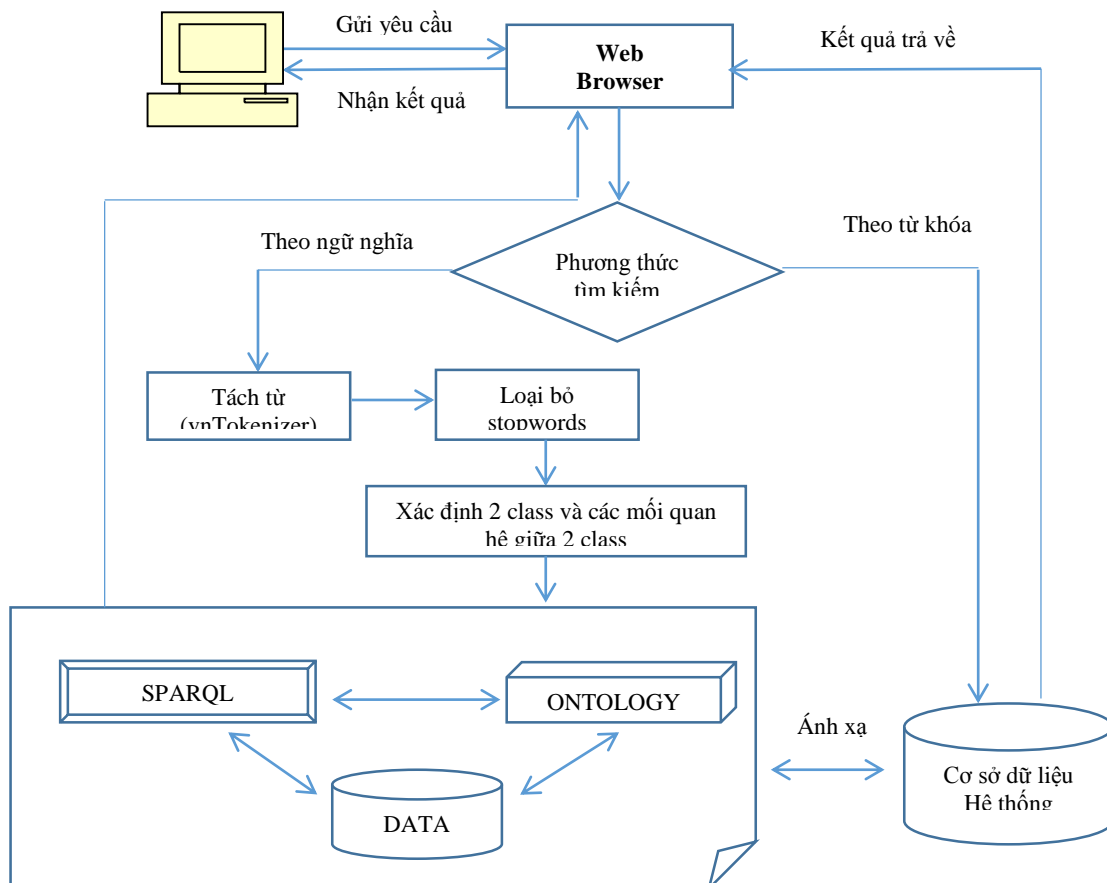
Prefix rdf: <http://www.owl-ontologies.com/thegioi.owl#>
Select ?ket_qua_2
Where
{
    ?rdf:hà_nội        ?rdf:thuộc_quốc_gia    ?ket_qua_1 .
    ?ket_qua_1        ?rdf:thuộc_khu_vực    ?ket_qua_2 .
}
  
```

Kết quả là Hà Nội thuộc khu vực Châu Á.

### 3 KẾT QUẢ THỰC NGHIỆM

#### 3.1 Mô hình tổng quan hệ thống website tìm kiếm sách điện tử

Trong bài báo, chúng tôi xây dựng ứng dụng website tìm kiếm sách điện tử (xem website tại: <http://phuchung.apphb.com>). Ứng dụng này được xây dựng với các tính năng: Cho phép người dùng tìm kiếm theo ngữ nghĩa với khoảng 360 đầu sách được lưu trữ trong ontology, Chức năng Autocomplete hỗ trợ tìm kiếm chính xác hơn, tìm kiếm theo keyword để có thể so sánh kết quả tìm kiếm. Mô hình hệ thống được mô tả ở hình 4.



Hình 4. Mô hình tổng quan của hệ thống tìm kiếm

Để xây dựng được website tìm kiếm sách điện tử chúng tôi thực hiện qua các bước sau:

**Tạo các lớp và các ràng buộc:** Tại giao diện của Protégé ta chọn tab OWL Classes, mặc định mỗi ontology mới tạo có lớp cha là Thing. Để tạo một lớp mới ta sử dụng chức năng Create subclass sẽ tạo ra lớp con ta sử dụng chức năng Create subclass. Trong Ontology về sách điện tử thông tin của các lớp được mô tả như hình 5(a) và trong bảng 2.

Bảng 2. Mô tả thông tin các Class trong Ontology

Stt	Tên lớp	Lớp cha	Mô tả
1	Sách	Thing	Mô tả thông tin của một đối tượng sách
2	Lĩnh vực	Thing	Mô tả các thông tin về một số lĩnh vực của sách
3	Danh mục	Thing	Danh mục của một sách và của một lĩnh vực
4	Lĩnh vực nghiên cứu	Thing	Lĩnh vực nghiên cứu của tác giả
5	Ngôn ngữ	Thing	Ngôn ngữ cho đối tượng sách
6	Tác giả	Thing	Thông tin tác giả của sách
7	Upload	Thing	Thông tin về thành viên upload sách

**Tạo các thuộc tính và quan hệ:** Các thuộc tính và quan hệ trong Ontology được thể hiện trong phần Property và Datatype của protégé. Các thuộc tính được mô tả ở hình 5(b) các quan hệ được mô tả ở bảng 3, các Datatype được mô tả ở bảng 4.

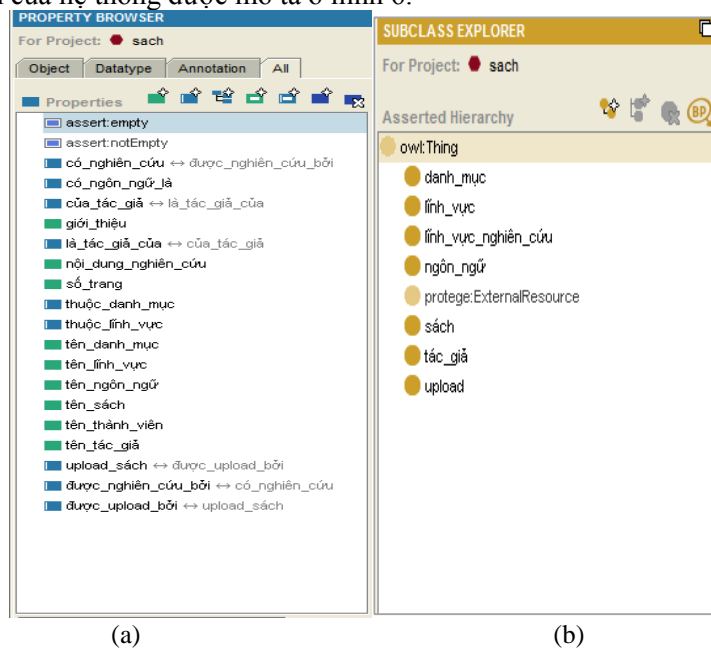
Bảng 3. Bảng mô tả quan hệ giữa các lớp

STT	Tên quan hệ	Domain	Range	Inverse
1	có nghiên cứu	Tác giả	Lĩnh vực nghiên cứu	Được nghiên cứu bởi
2	Có ngôn ngữ là	Sách	Ngôn ngữ	
3	Của tác giả	Sách	Tác giả	Là tác giả của
4	Là tác giả của	Tác giả	Sách	Của tác giả
5	Thuộc danh mục	Sách	Danh mục	
6	Thuộc lĩnh vực	Danh mục	Lĩnh vực	
7	Upload sách	Upload	Sách	Được upload bởi
8	Được nghiên cứu bởi	Lĩnh vực nghiên cứu	Tác giả	Có nghiên cứu
9	Được upload bởi	Sách	Upload	Upload sách

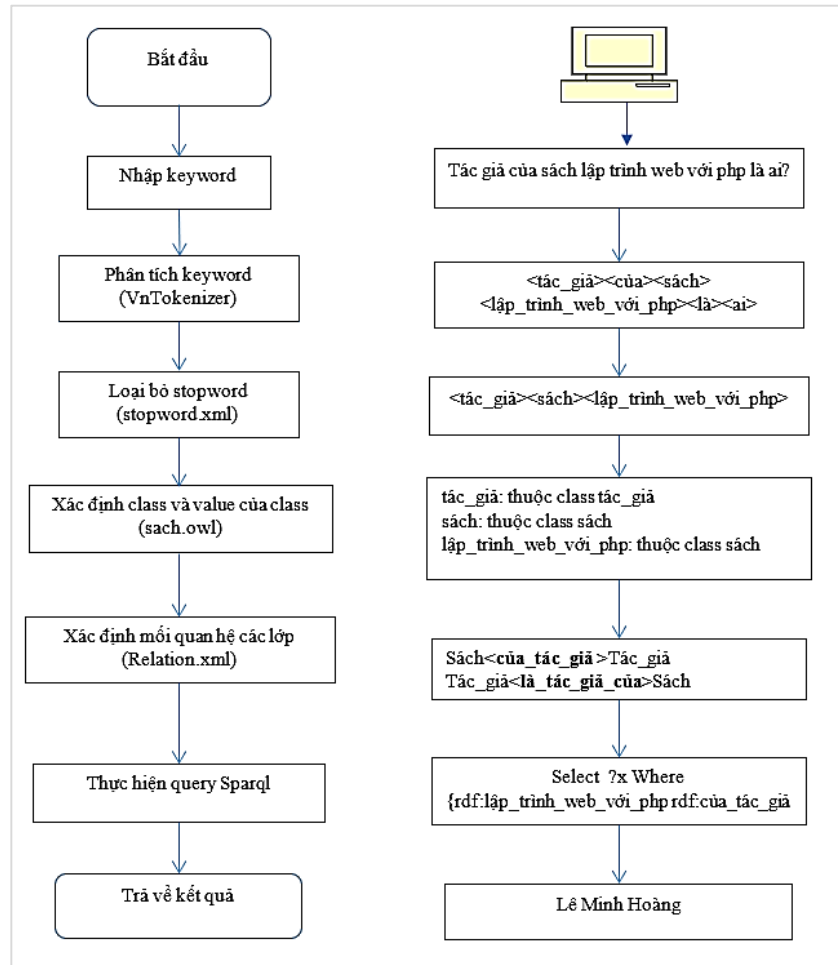
Bảng 4. Bảng mô tả datatype của Ontology

STT	Tên	Domain	Range
1	Giới thiệu	Sách	String
2	Nội dung nghiên cứu	Lĩnh vực nghiên cứu	String
3	Số trang	Sách	Int
4	Tên danh mục	Danh mục	String
5	Tên lĩnh vực	Lĩnh vực	String
6	Tên ngôn ngữ	Ngôn ngữ	String
7	Tên sách	Sách	String
8	Tên thành viên	Upload	String
9	Tên tác giả	Tác giả	String

Quá trình tìm kiếm của hệ thống được mô tả ở hình 6.



Hình 5: (a) các lớp trong Ontology, (b) các Object property và Datatype của Ontology



Hình 6. Quá trình tìm kiếm của hệ thống

### 3.2 Kết quả thực nghiệm chương trình

Với phương pháp được đề xuất ở phần trên hệ thống tìm kiếm có thể đáp ứng được một số lượng lớn hỏi nhập vào. Cụ thể cách tính như sau:

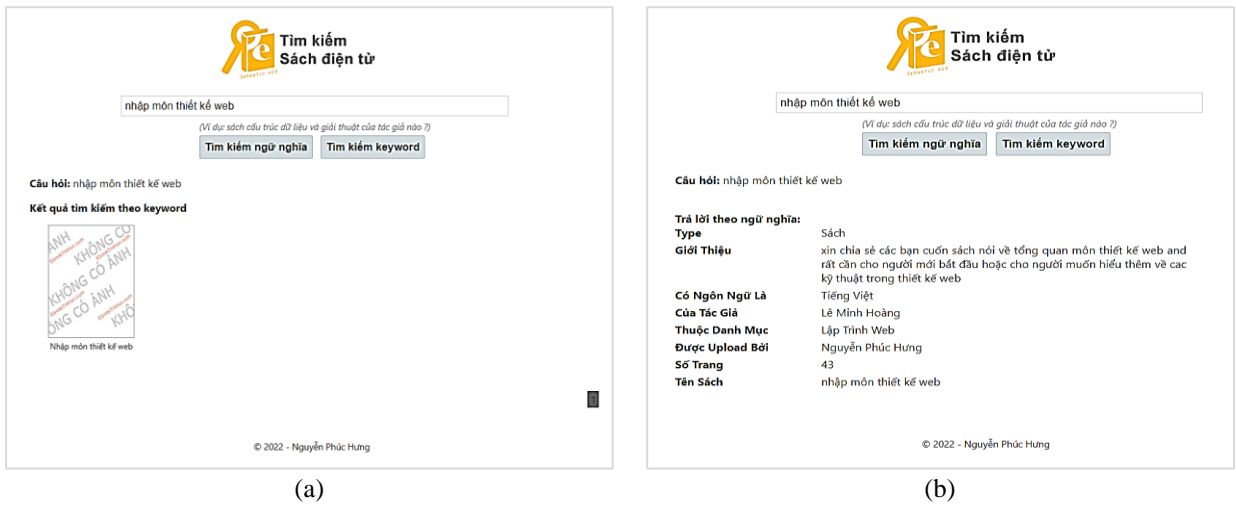
Trường hợp câu hỏi xác định 2 lớp trong Ontology: Số lớp trong ontology sách điện tử là 7 lớp. Số mối quan hệ các lớp là 9 quan hệ. Dữ liệu cho lớp trung bình là 100 giá trị. Như vậy số câu có thể tạo ra từ dữ liệu và mối qua hệ giữa các lớp trung bình là  $9 \times 100 = 900$  câu trả lời. Theo như nguồn về các loại câu đồng nghĩa trong tiếng Việt ta có 26 kiểu Câu Đồng Nghĩa. Từ 900 câu suy luận đồng nghĩa ta có  $900 \times 26 = 23,400$  câu.

Trường hợp câu hỏi dạng liệt kê thông tin (chỉ có thông tin phần tử cụ thể nào đó của 1 class): kết quả số câu trả lời được tính trung bình từ dữ liệu trên như sau:  $7 \times 100 \times 26 = 18,200$  câu trả lời. Tổng số câu trả lời với số liệu đã cho là  $23,400 + 18,200 = 41,600$  câu trả lời.

Để đánh giá hiệu quả của hệ thống, chúng tôi so sánh tìm kiếm ngữ nghĩa và tìm kiếm bằng từ khoá. Về tốc độ tìm kiếm, tốc độ tìm kiếm theo ngữ nghĩa hạn chế hơn so với tìm kiếm theo keyword thông thường, bởi các bước tìm kiếm bằng ngữ nghĩa phức tạp hơn so với tìm kiếm bằng từ khoá. Tuy nhiên, tìm kiếm ngữ nghĩa mang lại kết quả tìm kiếm phù hợp hơn nhiều so với tìm kiếm theo từ khoá, kết quả được kiểm chứng như sau:

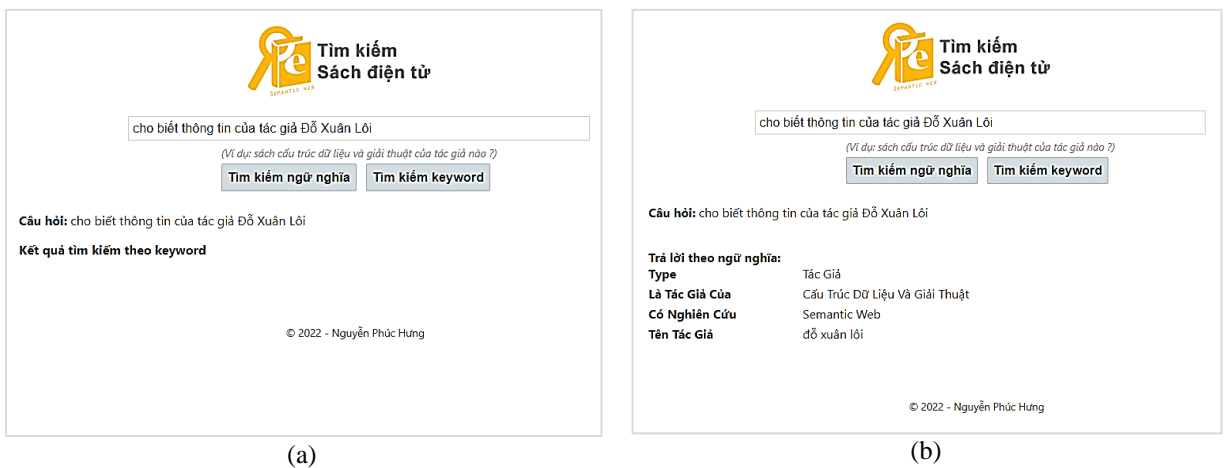
**Đánh giá tìm kiếm trong cùng hệ thống:** Kết quả tìm kiếm bằng từ khoá cho thấy tìm kiếm mang tính tổng quát, nội dung tìm kiếm không cụ thể dẫn đến thông tin khá nhiều không được định hướng rõ ràng cho người dùng. Trong khi đó, tìm kiếm dựa trên ngữ nghĩa cho thấy được kết quả cụ thể, chính xác từng đối tượng, giúp người dùng dễ dàng tiếp cận với thông tin cần tìm. Đối với việc tìm kiếm bằng các từ khoá đơn giản kết quả được so sánh được mô tả ở hình 7.





Hình 7. Tìm kiếm trên theo từ khoá, (a) Tìm kiếm keyword, (b) Tìm kiếm ngữ nghĩa

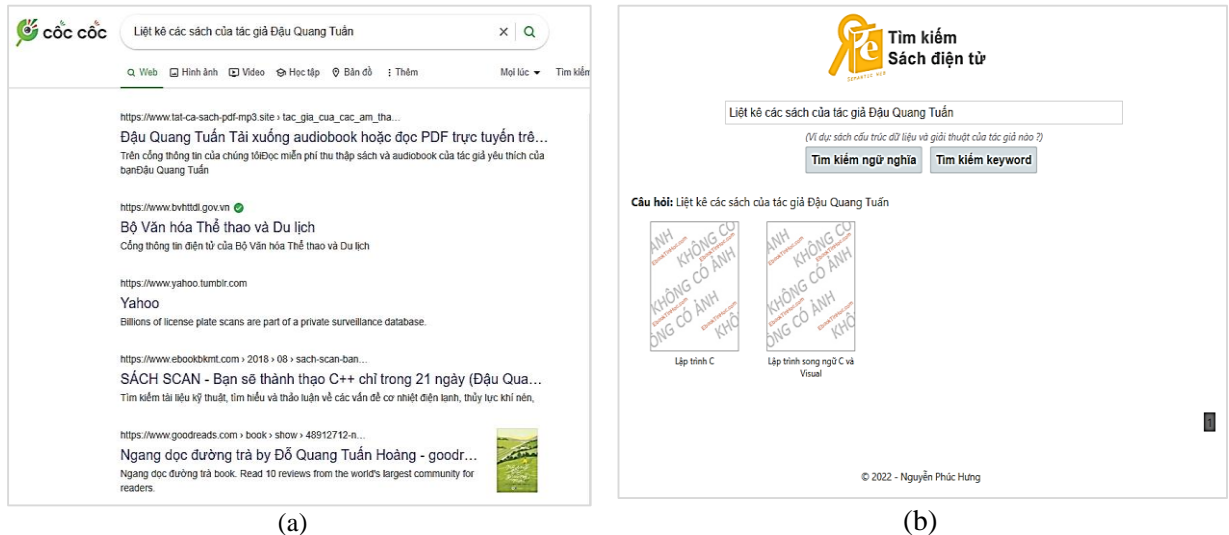
Đối với các câu hỏi phức tạp, tìm kiếm bằng từ khoá không thể trả về thông tin mong muốn, trong khi đó tìm kiếm ngữ nghĩa đã truy xuất được các thông tin quan trọng và đưa ra được thông tin mong muốn. Kết quả được mô tả ở hình 8.



Hình 8: Tìm kiếm theo câu hỏi, (a) Tìm kiếm keyword, (b) Tìm kiếm ngữ nghĩa

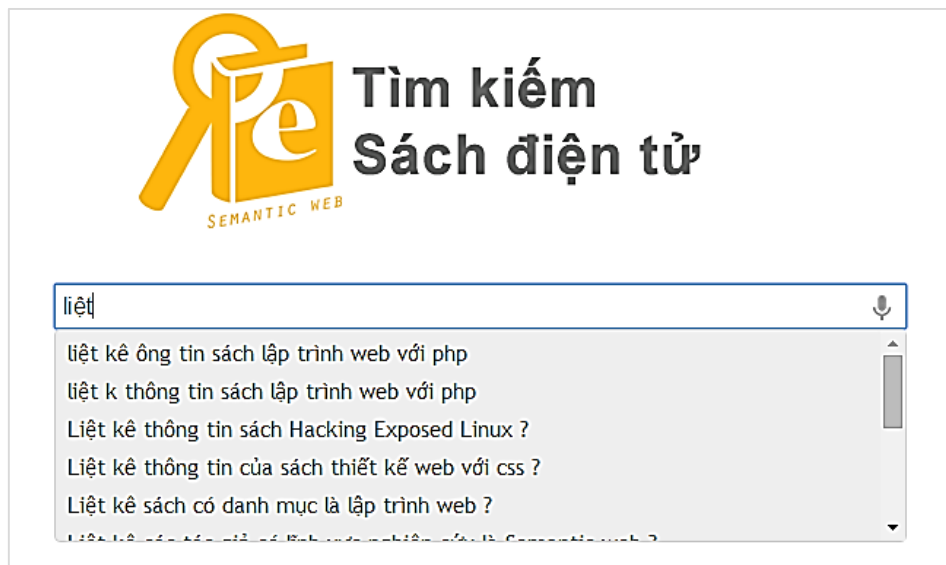
**Đánh giá so với các công cụ tìm kiếm khác:** Đối với các công cụ tìm kiếm dữ liệu trung tâm phổ biến khác, việc tìm kiếm thường nhận được kết quả với rất nhiều thông tin từ internet, trong đó có những thông tin không phục vụ cho yêu cầu tìm kiếm cụ thể, do đó người dùng mất nhiều thời gian để chọn lọc thông tin phù hợp. Trong khi đó website tìm kiếm sách điện tử hoạt động trên cấu trúc dữ liệu chuyên về sách điện tử hỗ trợ người dung tiếp cận đối tượng thông tin cụ thể, hiệu quả, tiết kiệm thời gian tìm kiếm. Kết quả so sánh với công cụ khác được mô tả ở hình 9.

## XÂY DỰNG ỨNG DỤNG TÌM KIẾM SÁCH...



Hình 9: (a) Tim kiếm với công cụ Coccoc.com, (b) Tim kiếm với website sách điện tử

**Tính tiện lợi hỗ trợ tìm kiếm:** Website sử dụng công nghệ Ajax autocomplete mang đến sự tiện dụng và thân thiện cho người dùng, không cần phải nhớ chính xác từ cần tìm. Công nghệ được tích hợp vào website như hình 10.



Hình 10: Autocomplete trong website tìm kiếm

## 4 KẾT LUẬN

Trong bài báo, chúng tôi đã sử dụng nền tảng PROTÉGÉ-OWL để tạo ra các ontology cho hệ thống tìm kiếm sách điện tử bao gồm 360 loại sách khác nhau. Các ontology OWL này bao gồm các lớp, thuộc tính và mối quan hệ của chúng với nhau. Chúng tôi đã sử dụng ngôn ngữ truy vấn SPARQL ở định dạng RDF để truy xuất dữ liệu đang được lưu trữ. Hệ thống cho phép phân tích từ khóa tìm ra các mối quan hệ trong ontology, tìm kiếm ngữ nghĩa dựa trên các mối quan hệ giữa các thực thể trong ontology và xây dựng câu truy vấn SPARQL phục vụ cho việc tìm kiếm ngữ nghĩa, hỗ trợ người sử dụng tìm kiếm và truy hồi thông tin một cách hiệu quả. Kết quả so sánh tìm kiếm dựa trên ngữ nghĩa và từ khóa cho thấy, tìm kiếm dựa trên ngữ nghĩa đạt hiệu quả cao hơn trong việc tìm kiếm những đối tượng cụ thể, đối với những câu hỏi phức tạp, tìm kiếm ngữ nghĩa truy xuất thông tin và cho kết quả tìm kiếm chính xác. Hơn nữa, việc tìm kiếm những đối tượng và thông tin cụ thể của website sách điện tử cũng giúp người dùng nhanh chóng tìm

được đối tượng mong muốn, tiết kiệm được thời gian tìm kiếm. Trong tương lai, chúng tôi sẽ bổ sung nguồn dữ liệu phong phú, tối ưu mô hình ngữ nghĩa để phục vụ cho việc tìm kiếm đa dạng hơn.

## TÀI LIỆU THAM KHẢO

- [1] A. Khamparia and B. Pandey, "Comprehensive analysis of semantic web reasoners and tools: a survey," *Education and Information Technologies* 22, 6, 2017, pp. 3121–3145. <https://doi.org/10.1007/s10639-017-9574-5>
- [2] A. Takhom, D. Leenoi, P. Soomjinda, S. Usanavasin, P. Boonkwan and T. Supnithi, "A Supportive Environment for Knowledge Construction based on Semantic Web Technology: A Case Study in a Cultural Domain," in *2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2019, pp. 1-4, doi: 10.1109/iSAI-NLP48611.2019.9045538.
- [3] O. Ochoa, M. Rodney and N. Del Rio, "Accessing Provenance Records in Semantic Web Services," in *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, 2020, pp. 141-144, doi: 10.1109/ICSC.2020.00027.
- [4] L. Ding, P. Kolari, Z. Ding, S. Avancha, "Using Ontologies in the Semantic Web: A Survey," In *Sharman, R., Kishore, R., Ramesh, R. (eds) Ontologies. Integrated Series in Information Systems*, vol 14. Springer, Boston, MA, 2007, [https://doi.org/10.1007/978-0-387-37022-4\\_4](https://doi.org/10.1007/978-0-387-37022-4_4).
- [5] S. Kakad and S. Dhage, "Semantic Web: Knowledge Graph of Digital Minimalism Ontology," *2022 3rd International Conference for Emerging Technology (INCET)*, 2022, pp. 1-6, doi: 10.1109/INCET54531.2022.9824657.
- [6] Z. Fayçal and T. Abdelkamel, "Building a semantic web services ontology in the pharmaceutical field using the OWL-S Language," *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*, 2021, pp. 1-8, doi: 10.1109/ICISAT54145.2021.9678408.
- [7] E. Cibil and M. Komesli, "Implementation of Semantic Web in the Companies: Human Resources Ontology," *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 2021, pp. 260-264, doi: 10.1109/UBMK52708.2021.9558937.
- [8] M. Lian, D. Yang, Z. Yin, J. Liu, M. Li and A. Chai, "A Semantic Web Service Oriented Middleware Framework for Internet of Things," in *2020 13th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2020, pp. 553-557, doi: 10.1109/ICICTA51737.2020.00122.
- [9] I. Rustam, A. Fatima, M. Asfand-e-Yar, A. S. Akram and S. Mehmood, "An Ontology Model Approach for Cloud Service Level Agreement (SLAs)," in *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, 2022, pp. 1-5, doi: 10.1109/ICBATS54253.2022.9759001
- [10] A.A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne and E. Motta, "The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas," in: *The Semantic Web – ISWC 2018. ISWC 2018. Lecture Notes in Computer Science()*, vol 11137. Springer, Cham, 2018, [https://doi.org/10.1007/978-3-030-00668-6\\_12](https://doi.org/10.1007/978-3-030-00668-6_12)
- [11] A.G. Nuzzolese, A.L. Gentile, V. Presutti, A. Gangemi, H. Sack, G. Rizzo, N. Steinmetz, D. Mladenici, S. Auer, C. Lange, "Semantic web conference ontology - a refactoring solution," in *ESWC 2016. LNCS*, vol. 9989, pp. 84–87. Springer, Cham, 2016, [https://doi.org/10.1007/978-3-319-47602-5\\_18](https://doi.org/10.1007/978-3-319-47602-5_18)
- [12] T. Bray, J. Paoli, Sperberg-McQueen, C.M., E. Maler, and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", in *W3C Recommendation*, 2008.
- [13] V. Christophides, "Resource Description Framework (RDF) Schema (RDFS), " in: *LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA, 2009. https://doi.org/10.1007/978-0-387-39940-9\_1319*

- [14] P. Hitzler, M. Krötzsch., B. Parsia, P.F Patel-Schneider, and S. Rudolph, "OWL 2 Web Ontology Language Primer (Second Edition)," 2012.
- [15] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosf, B., Dean, "M.: SWRL: A semantic web rule language combining OWL and RuleML," *W3C member submission, World Wide Web Consortium*, 2004.
- [16] L. R. Daniel, K. Holger, W. F. Ray, O. Dameron, M. A. Musen, "Protégé-OWL: Creating Ontology-Driven Reasoning Applications with the Web Ontology Language," in *AMIA 2005, American Medical Informatics Association Annual Symposium*, Washington, DC, USA, October 22-26, 2005.
- [17] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," in W3C Recommendation, 2008.
- [18] Nassar, Nefretiti, and Mark Austin. "Model-Based Systems Engineering Design and Trade-Off Analysis with RDF Graphs." *Procedia Computer Science 16*, pp: 216–25, 2013, doi:10.1016/j.procs.2013.01.023.
- [19] Luong Nguyen Thi, Linh Ha My, H. Nguyen Viet, H. Nguyen Thi Minh and P. Le Hong, "Building a treebank for Vietnamese dependency parsing," in *The 2013 RIVF International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*, 2013, pp. 147-151, doi: 10.1109/RIVF.2013.6719884.

## AN ONTOLOGY MODEL APPROACH FOR E-BOOK SEARCH APPLICATION

NGUYEN PHUC HUNG<sup>1\*</sup>, DANG THI PHUC<sup>2</sup>, DINH CONG THANG<sup>3</sup>

<sup>1</sup> *Center of System Management, Industrial University of Ho Chi Minh City*

<sup>2</sup> *Faculty of Information Technology, Industrial University of Ho Chi Minh City*

<sup>3</sup> *KMS Technology Company, Viet nam*

\**Corresponding author: nguyenvphuchung@iuh.edu.vn*

**Abstract.** Semantic search through query information is an important step to improve information retrieval efficiency. In this paper, we propose an ontology model and approaches to improve the relevancy in information retrieval. We built an e-book search system that allows users to search based on semantics with about 360 titles stored in the ontology. The system is implemented with the support of Resource Description Framework (RDF) technologies and ontology Web Language (OWL) technologies. For executing query, we used the SPARQL query language in RDF format which is used to retrieve the data that is being stored. The application provides a simple and consistent encryption system that helps users to search and retrieve information efficiently, saving search time.

**Keywords.** Ontology, Semantic Web, RDF, OWL, SPARQL, e-book search application

*Ngày gửi bài: 21/07/2022*

*Ngày chấp nhận đăng: 16/11/2022*