Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9)"; Cần Thơ, ngày 4-5/8/2016
DOI: 10.15625/vap.2016.00042

MÔ HÌNH TRUY VẤN DỮ LIỆU TÙY CHỌN DỰA TRÊN NGỮ NGHĨA CỦA CÂU TRUY VẤN

Nguyễn Thanh Tuấn[1], Hoàng Thị Thanh Hà[2], Nguyễn Văn Tin[3], Lê Thành Nhân[4]

[1] Trường Đại học Sư phạm, Đại học Đà Nẵng
[2] Trường Đại học Kinh tế, Đại học Đà Nẵng
[3] Đại học Đà Nẵng
[4] Đại học Nice Sophia-Antipolis
nttuan@ued.udn.vn, ha.htt@due.edu.vn, vantinqng@gmail.com, Nhan.LE-THANH@unice.fr

TÓM TẮT— *Hiện nay, cùng với sự mở rộng của World Wide Web là sự tăng trưởng nhanh chóng của dữ liệu. Nguồn dữ liệu khổng lồ hiện nay vẫn được lưu trữ trong các mô hình cơ sở dữ liệu khác nhau, phổ biến là các mô hình: cơ sở dữ liệu quan hệ (relational database), cơ sở dữ liệu ngữ nghĩa,... Sự phong phú của các loại cơ sở dữ liệu gây khó khăn cho người dùng có thể theo kịp với những phát triển của ngôn ngữ truy vấn trong việc truy cập vào các kho dữ liệu mà họ cần tra cứu. Bài báo này trình bày hướng tiếp cận cho phép người sử dụng có thể lựa chọn và truy vấn được dữ liệu mong muốn từ hai loại cơ sở dữ liệu phổ biến hiện nay là cơ sở dữ liệu ngữ nghĩa (RDF database), cơ sở dữ liệu quan hệ (relational database). Dựa trên ngữ nghĩa của câu truy vấn bằng ngôn ngữ tự nhiên và loại cơ sở dữ liệu lựa chọn, hệ thống sẽ đề nghị ra một câu truy vấn tương ứng cho phép người dùng kiểm tra trước khi đồng ý thực thi.*

Từ khóa— *RDF, SQL, SPARQL, chuyển đổi câu truy vấn cơ sở dữ liệu.*

I. GIỚI THIỆU

Hiện nay, World Wide Web đang chứa một khối lượng dữ liệu khổng lồ. Với thế hệ Web 2.0 hiện tại, nhu cầu về tốc độ xử lý đáp ứng yêu cầu của người dùng, đồng thời việc khai phá nội dung Web đã và đang gặp nhiều khó khăn. Vì vậy, thế hệ Web 3.0 (Web ngữ nghĩa) đã ra đời nhằm giải quyết các vấn đề trên. Web ngữ nghĩa là bước tiến lớn so với kỹ thuật của Web hiện tại về khả năng làm việc với thông tin thay vì chỉ đơn thuần là lưu trữ, cùng với xử lý ngôn ngữ tự nhiên, trí thông minh nhân tạo,… đã giúp cho web có thể hiểu những gì chúng ta nghĩ.

Song song với việc thay đổi cấu trúc mô tả dữ liệu cho Web ngữ nghĩa, nguồn dữ liệu khổng lồ hiện nay vẫn được lưu trữ trong các mô hình cơ sở dữ liệu (CSDL) khác nhau, phổ biến là các mô hình: CSDL quan hệ (relational database), CSDL hướng tài liệu (document-oriented database), CSDL ngữ nghĩa (RDF Database)… Sự phong phú của các loại CSDL gây khó khăn cho người dùng có thể theo kịp với những phát triển của ngôn ngữ truy vấn trong việc truy vập vào các kho dữ liệu mà họ cần tra cứu.

Để giúp người dùng bằng ngôn ngữ tự nhiên (NNTN) có thể truy vấn một cách dễ dàng với các mô hình CSDL phổ biến hiện nay và thế hệ cơ sở dữ liệu mới là CSDL ngữ nghĩa, đã có nhiều nghiên cứu xây dựng hệ thống hỗ trợ việc chuyển đổi câu truy vấn tùy thuộc vào nhu cầu của người dùng vào từng mô hình CSDL. Một số mô hình đã được phát triển để giúp chuyển đổi câu truy vấn bằng NNTN sang câu truy vấn SQL[1] (ngôn ngữ truy vấn phổ biến của CSDL quan hệ) cũng như SPARQL[2] (ngôn ngữ truy vấn cho CSDL ngữ nghĩa) đã được công bố trong nhiều công trình trong và ngoài nước.

Bài báo này trình bày hướng tiếp cận mới cho phép người sử dụng có thể lựa chọn và truy vấn được dữ liệu mong muốn từ hai loại CSDL phổ biến hiện nay là CSDL ngữ nghĩa (RDF database), CSDL quan hệ (relational database), dựa trên ngữ nghĩa của câu truy vấn bằng NNTN và loại CSDL lựa chọn, hệ thống sẽ đề nghị ra một câu truy vấn tương ứng cho phép người dùng kiểm tra trước khi đồng ý thực thi.

II. TRUY VẤN CƠ SỞ DỮ LIỆU BẰNG NGÔN NGỮ TỰ NHIÊN

A. Tổng quan về các mô hình chuyển đổi giữa các loại cơ sở dữ liệu

Một lĩnh vực có thể áp dụng hiệu quả những kết quả của Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) đó là các hệ truy vấn CSDL, vì các CSDL thường phủ một miền tri thức đủ nhỏ nên trong một số tình huống, sự nhập nhằng trong ngôn ngữ tự nhiên có thể giải quyết được. Các hệ thống ngôn ngữ tự nhiên dùng cho truy vấn CSDL được gọi chung là *Giao diện ngôn ngữ tự nhiên đối với CSDL* (Natural Language Interface to Database). Giao diện ngôn ngữ tự nhiên đã xuất hiện từ những năm 60, 70 của thế kỷ trước [1]. Một số hệ thống tiêu biểu đã được phát triển như là:

- **LUNAR** là một giao diện NNTN cho cơ sở dữ liệu mang các phân tích hóa học của các mẫu đá mặt trăng mang về từ Apollo-11 bởi William A. Woods[3].

[1] https://msdn.microsoft.com/en-gb/library/windows/desktop/ms714670(v=vs.85).aspx
[2] https://www.w3.org/TR/rdf-sparql-query/
[3] http://parsecraft.com/

- **LIFER / LADDER** là một trong những hệ thống xử lý ngôn ngữ CSDL đầu tiên. Nó được thiết kế như một giao diện NNTN đến một cơ sở dữ liệu của thông tin về tàu Hải quân Hoa Kỳ. Hệ thống này, được sử dụng một ngữ nghĩa ngữ pháp để phân tích câu hỏi và truy vấn một phân phối cơ sở dữ liệu. [2].

Ngoài ra còn nhiều hệ thống khác đã được nghiên cứu phát triển, thông tin về các hệ thống này có thể được tìm thấy trong [3]. Cũng đã có một số nghiên cứu trong việc truy vấn bằng ngôn ngữ tiếng Việt [4], tuy nhiên kết quả chỉ truy vấn trên CSDL quan hệ và chưa thấy nhiều ứng dụng rộng rãi.

Việc sử dụng giao diện tìm kiếm bằng ngôn ngữ tự nhiên hiện nay trong các hệ thống điều kiển bằng giọng nói, trợ lý ảo hay các ứng dụng nhúng có thể tìm thấy trên nền tảng của các công ty công nghệ lớn như Wolfram Alpha[4] của Wolfram, Siri[5] của Apple, Alexa[6] của Amazon…

Các ứng dụng của giao diện tìm kiếm bằng NNTN đều có một phần quan trọng là nền tảng truy vấn cơ sở dữ liệu. Các nền tảng tiêu biểu mà chúng tôi trình bày ở đây đó là:

- **Quepy**[7] là một nền tảng viết bằng python sử dụng để chuyển đổi truy vấn bằng NNTN sang một ngôn ngữ truy vấn cơ sở dữ liệu. Nó có thể dễ dàng tùy chỉnh đến các loại câu hỏi khác nhau trong NNTN và các truy vấn cơ sở dữ liệu. Vì vậy, ta có thể xây dựng hệ thống sử dụng NNTN để truy xuất cơ sở dữ liệu. Hiện nay Quepy cung cấp hỗ trợ cho các ngôn ngữ SPARQL và MQL[8].

- **Pythia**[9] một hệ thống trả lời tự động dựa trên ontology [5]. Nó cấu trúc hóa các thành phần có nghĩa bằng cách sử dụng từ vựng phù hợp với từ vựng của ontology đã cho. Bằng cách làm như vậy dựa trên phân tích sâu ngôn ngữ, cho phép xây dựng các truy vấn hình thức từ những câu hỏi phức tạp bằng NNTN.

- **KUERI**[10] một nền tảng cho phép tích hợp vào ứng dụng khả năng truy cập và tìm kiếm dữ liệu bằng NNTN. Nó cho phép tạo các truy vấn SQL hoặc JSON trên nhiều dạng CSDL khác nhau.

- **EasyQuery**[11] một thư viện cho nền tảng .Net cho phép xây dựng các giao diện truy vấn bằng NNTN, thư viện này làm chỉ làm việc với CSDL quan hệ.

Xét trên các tiêu chí như: vấn đề cần giải quyết, loại CSDL truy vấn và ngôn ngữ truy vấn được sinh ra, **Bảng 1** sẽ so sánh về các nền tảng này.

Bảng 1. So sánh các nền tảng tiêu biểu truy vấn CSDL bằng NNTN

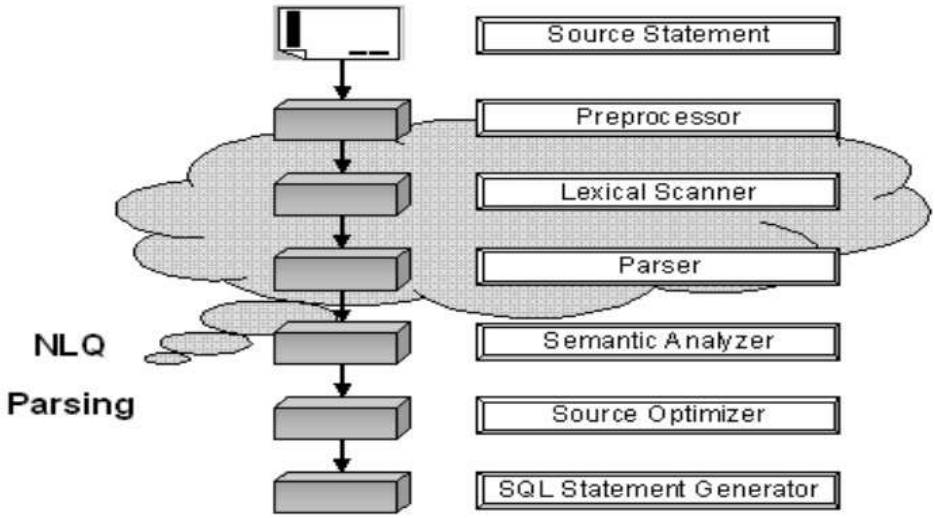
Các nền tảng	Vấn đề giải quyết	Loại CSDL truy vấn	Ngôn ngữ truy vấn
Quepy	Chuyển đổi truy vấn bằng NNTN sang ngôn ngữ truy vấn CSDL	Ngữ nghĩa	SPARQL, MQL
Pythia	Xây dựng hệ thống trả lời tự động	Ngữ nghĩa	SPARQL
KUERI	Nền tảng tích hợp khả năng truy cập và tìm kiếm bằng NNTN.	Quan hệ, NoSQL[12]	SQL, JSON[13]
EasyQuery	Thư viện xây dựng giao diện truy vấn bằng NNTN	Quan hệ	SQL

Như vậy có thể thấy, các nền tảng được phát triển đều hướng đến một loại CSDL là quan hệ hay ngữ nghĩa, trừ KUERI có thể sử dụng với CSDL noSQL, hiện tại vẫn chưa thấy hệ thống nào cho phép truy vấn cả cơ sở dữ liệu ngữ nghĩa và cơ sở dữ liệu quan hệ.

Trong hầu hết các hệ thống chuyển đổi câu truy vấn bằng NNTN sang một câu truy vấn tương ứng trong ngôn ngữ truy vấn, giải pháp xử lý cơ bản là giống nhau, nó phải trải qua rất nhiều bước:

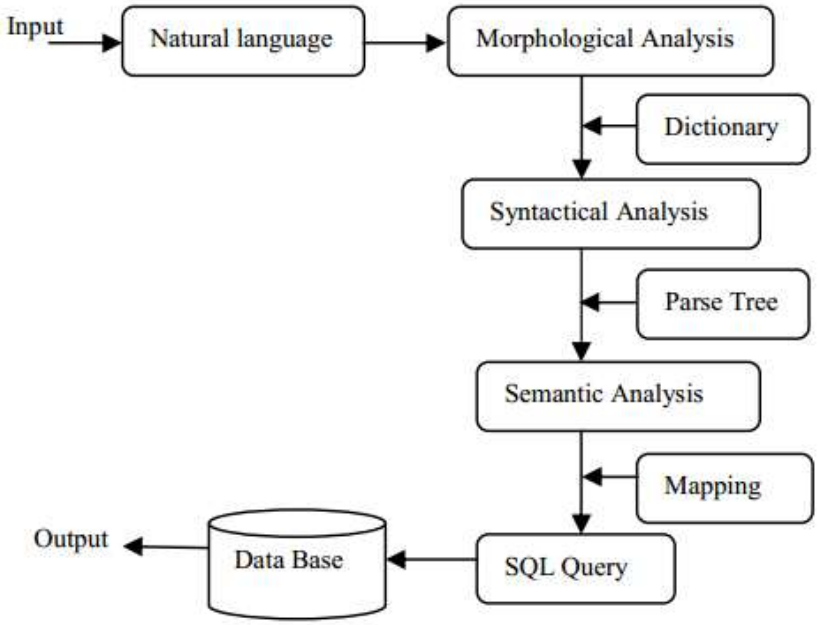
- Gắn thẻ từ loại.
- Kiểm tra ngữ pháp.
- Sinh câu truy vấn.
- Thu thập dữ liệu.

[4] https://www.wolframalpha.com/
[5] http://www.apple.com/ios/siri/
[6] https://developer.amazon.com/public/solutions/alexa
[7] http://quepy.machinalis.com/
[8] http://wiki.freebase.com/wiki/MQL
[9] https://github.com/cunger/pythia
[10] http://simpleql.com/
[11] http://devtools.korzh.com/easyquery/
[12] https://www.mongodb.com/nosql-explained
[13] http://www.json.org/



Hình 1. Kiến trúc của hệ thống NLDBi [6]

Trong kiến trúc hệ thống Natural language database interface (NLDBi) được phát triển như **Hình 1**, mô tả cách bố trí các quá trình bao gồm việc chuyển đổi truy vấn NNTN thành câu truy vấn cú pháp SQL để thực hiện trên hệ quản trị CSDL quan hệ. Toàn bộ quá trình liên quan đến việc gắn thẻ cho câu đầu vào, áp dụng ngữ pháp và đại diện ngữ nghĩa để tạo ra cây phân tích cú pháp, phân tích cây phân tích cú pháp sử dụng ngữ pháp và dịch các lá của cây để tạo ra câu truy vấn SQL tương ứng.

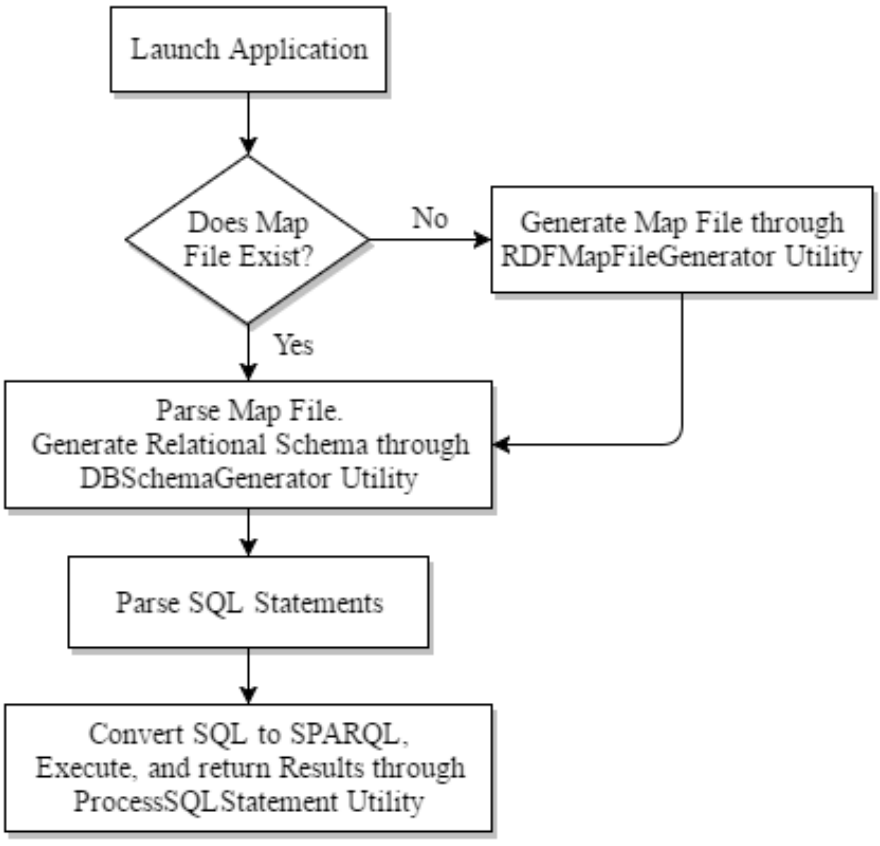


Hình 2. Cấu trúc hệ thống trong nghiên cứu [7]

Trong nghiên cứu [7] các tác giả đã đề xuất hệ thống sinh và thực thi SQL từ NNTN có cấu trúc như **Hình 2**, cũng gồm các bước gần giống với hệ thống NLDBi:

- Phân tích hình thái học: các từ cá thể được phân tích thành các thành phần và các thẻ.
- Phân tích cú pháp: trình tự tuyến tính cả các từ được chuyển thành cấu trúc miêu tả sự liên quan giữa các từ.
- Phân tích ngữ nghĩa: các cấu trúc được tạo ra bởi phân tích cú pháp được gán nghĩa.
- Câu truy vấn sẽ được ánh xạ từ kết quả của các bước trên để tạo câu truy vấn SQL.

Hiện nay, các nghiên cứu biến đổi SQL sang SPARQL còn rất nghèo nàn, chưa hỗ trợ miễn phí cho người dùng. Trong [8], các tác giả đã xây dựng một nền tảng gọi tên là R2D, cung cấp một giao diện quan hệ sang lưu trữ dữ liệu dưới hình thức của bộ ba RDF. Trình tự triển khai thủ tục dịch SQL sang SPARQL của nền tảng R2D được minh họa trong **Hình 3**.



Hình 3. Trình tự triển khai thủ tục dịch SQL sang SPARQL trong R2D [8]

Thuật toán phân tích cú pháp file ánh xạ và tạo lược đồ quan hệ thông qua tiện ích được cài đặt sẵn, sau đó phân tích câu truy vấn SQL và chuyển đổi SQL sang SPARQL.

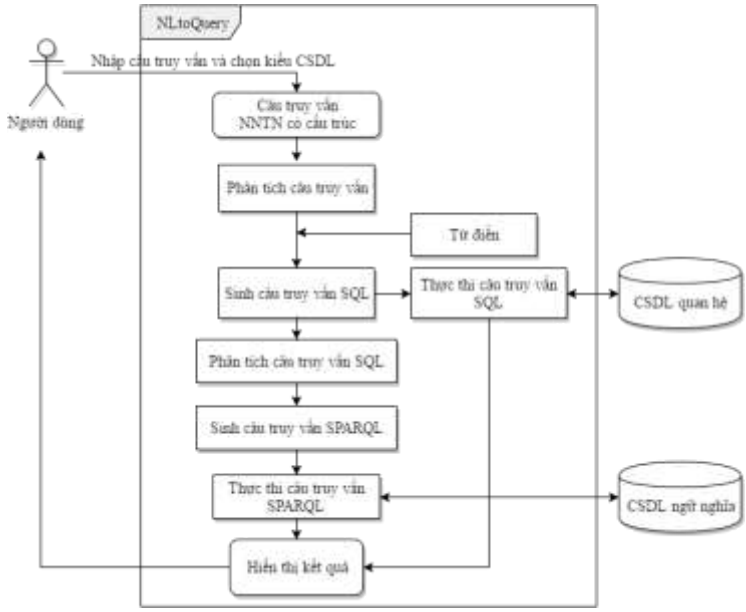
Như vậy, cả mô hình [1] và [2] trên đều có chung một mục đích là chuyển đổi từ câu truy vấn bằng NNTN sang câu truy vấn các mô hình CSDL phổ biến nhất hiện nay như là CSDL quan hệ, CSDL ngữ nghĩa. Các mô hình chuyển đổi này đều tiến hành xử lý phân tích cú pháp NNTN, sau đó phân tích về ngữ nghĩa và chuyển đổi về câu truy vấn CSDL quan hệ SQL. Còn mô hình trong [3] thì tập trung vào việc chuyển đổi từ câu truy vấn SQL thành câu truy vấn SPARQL dựa vào bản đồ (Map). Có thể thấy hiện tại chưa có hệ thống hay nền tảng nào cho phép chuyển đổi câu truy vấn bằng NNTN sang cả truy vấn CSDL quan hệ và CSDL ngữ nghĩa. Phần tiếp theo, bài báo sẽ đề xuất mô hình chuyển đổi câu truy vấn từ NNTN sang câu truy vấn SQL và câu truy vấn SPASQL.

B. Mô hình chuyển đổi câu truy vấn bằng NNTN sang CSDL quan hệ và CSDL ngữ nghĩa

Dựa vào tìm hiểu đã mô tả ở các phần trước, phát biểu của bài toán đặt ra và các nghiên cứu đã được công bố, có thể thấy rằng mô hình hệ thống cần phải có các bước chính như sau:

- Phân tích câu truy vấn NNTN có cấu trúc: phân tích câu truy vấn bằng NNTN mà người dùng nhập vào để xác định các từ khóa, từ loại của từ.
- Chuyển đổi thành câu truy vấn SQL: từ phân tích ở bước trên và dựa vào kho từ điển, tiến hành phân tích ngữ nghĩa để xác định các thành phần của câu truy vấn. Từ đó tạo thành câu truy vấn SQL.
- Phân tích câu truy vấn SQL.
- Chuyển đổi thành câu truy vấn SPARQL.
- Kết nối với cơ sở dữ liệu tùy chọn và thực hiện câu truy vấn, trả về kết quả cho người dùng.

Vì vậy, chúng tôi đã xây dựng một mô hình hệ thống tổng quan như **Hình 4**:



Hình 4. Mô hình hệ thống

Mô hình hệ thống tổng quan kế thừa những bước cơ bản của hai quy trình trong các hệ thống đã được xây dựng trước đó:

- Quy trình chuyển đổi câu truy vấn bằng NNTN có cấu trúc sang câu truy vấn SQL: phân tích câu, cú pháp, phân tích ngữ nghĩa và xác định loại câu truy vấn, các thành phần của câu truy vấn đó, từ đó tạo thành câu truy vấn SQL.
- Quy trình từ câu truy vấn SQL sang câu truy vấn SPARQL: phân tích câu truy vấn SQL, ánh xạ sang câu truy vấn SPARQL.
- Đồng thời tích hợp hai quy trình vào một mô hình để giải quyết mục đích là truy vấn CSDL tùy chọn dựa trên ngữ nghĩa của câu truy vấn.

C. Chuyển đổi truy vấn bằng NNTN sang câu truy vấn SQL và câu truy vấn SPARQL

1. Cấu trúc câu truy vấn bằng NNTN và kỹ thuật xử lý

Câu truy vấn là một câu tự nhiên bình thường có chứa các từ khóa được chứa trong ngoặc kép “”. Các từ khóa này là mô tả bằng NNTN cho: tên bảng, tên thuộc tính, giá trị của thuộc tính trong CSDL.

Ví dụ:

choose "Last Name" in "Contact" with "Last Name" like "t".

Các bước xử lý câu truy vấn bằng NNTN có cấu trúc:

- Tách các từ và từ khóa trong câu truy vấn bằng NNTN có cấu trúc: dựa vào phương pháp tách từ trong NLP và các từ khóa chứa trong dấu ngoặc kép.

- Xác định từ loại của từ: dựa vào gán nhãn từ loại trong NLP.

2. Chuyển đổi sang câu truy vấn SQL

Quá trình chuyển đổi câu truy vấn NNTN sang câu truy vấn SQL dựa vào việc phân tích ngữ nghĩa ở mức độ ngữ nghĩa từ vựng, biểu hiện các ý nghĩa của những từ thành phần.

Bảng 2. Từ khóa và kết quả tương ứng xác định loại câu truy vấn

Từ khóa	*Từ khóa tương ứng trong SQL*
“delete”, “erase”, “cancel”, “obliterate”, “drop”, “destroy”, “demolish”	DELETE
“select”, “choose”, “find”, “want”, “opt”, “determine”, “prefer”, “what”, ”list”, “display”	SELECT
“insert”, “add”, “append”, “enter”, “include”, “put”, “initialize”,”fill”, “introduce”	INSERT
“update”, “amend”, “modernize”, “refresh”, “refurbish”, “rejuvenate”, “renew”, “renovate”, “restore”, “revise”	UPDATE

Bước 1: Xác định câu truy vấn bằng NNTN thuộc câu truy vấn loại nào: SELECT, DELETE, UPDATE, INSERT. Việc xác định này nhờ vào việc gắn thẻ từ loại cho từ và truy vấn từ điển.

Bước 2: Xác định thành phần của câu truy vấn bằng NNTN có cấu trúc tương ứng với cấu trúc câu truy vấn SQL

- Câu truy vấn SELECT

Xác định các thành phần của câu SELECT

Bảng 3. Từ khóa và kết quả tương ứng xác định thành phần câu SELECT

Từ khóa	*Từ khóa tương ứng trong SQL*
"select", "choose", "find", "want", "opt", "determine", "prefer", "what", "list", "display"	Select
"from", "of", "in", "source"	From
"filter", "for", "during", "with", "where"	Where

Xác định các toán tử so sánh ở mệnh đề WHERE

Bảng 4. Từ khóa và kết quả tương ứng xác định toán tử so sánh

Từ khóa	*Giá trị tương ứng trong SQL*
"equal to", "is", "="	=
"difference", "<>"	<>
"less than", "<"	<
"greater than", ">"	>
"less than or equal to", "<=", "=<"	<=
"greater than or equal to", ">=", =>"	>=
"and"	AND
"or"	OR
"like", "contain", "include"	LIKE

Bước 3: Tổng hợp các thành phần xác định được của câu truy vấn và sinh ra câu truy vấn SQL hoàn chỉnh.

3.Chuyển đổi sang câu truy vấn SPARQL

Bước 1: Câu truy vấn SQL được phân tích để xác định các bảng, trường dữ liệu, và mệnh đề WHERE và GROUP BY nếu có.

Bước 2: Xác định các tiền tố PREFIXE của SPARQL dựa vào các bảng đã xác định được trong bước 1.

Bước 3: Mệnh đề SELECT SPARQL được tạo ra bằng cách thêm một biến cho mọi trường dữ liệu (bao gồm các trường tập hợp và các trường trong mệnh đề WHERE của SQL) trong danh sách SELECT SQL.

Bước 4: Mệnh đề WHERE SPARQL được tạo ra bằng cách mỗi "biến" sẽ được định dạng là "?biến", hoặc nếu chúng có biệt danh riêng thì sẽ được định dạng kiểu "?biệt danh_biến" trong câu truy vấn SPARQL. Ví dụ:

WHERE prename = "Nguyễn"

Được chuyển đổi thành

FILTER (?prename = "Nguyễn")

WHERE name = prename

Được chuyển đổi thành

FILTER (?name = ?prename)

Toán tử AND và OR được sử dụng để lọc các bản ghi dựa trên nhiều điều kiện, sẽ được thay thế bằng "&&" và "| |" trong SPARQL.

WHERE name = 'Tin' AND prename = 'Nguyễn Văn' OR homeaddress = 'Phổ Văn - Đức Phổ - Quảng Ngãi'

Được chuyển đổi thành

FILTER (?name = "Tin" && ?prename = "Nguyễn Văn" || ?homeaddress = "Phổ Văn - Đức Phổ - Quảng Ngãi")

Nếu câu truy vấn có sử dụng toán tử LIKE, cần tạo biểu thức *regex*:

WHERE name LIKE 't'

Được chuyển đổi thành

FILTER regex(?name, "t")

Bước 5: Chuyển đổi các câu lệnh GROUP BY, ORDER BY, LIMIT

Bước 6: Tích hợp các kết quả từ bước 2 đến bước 5 để tạo thành câu truy vấn SPARQL hoàn chỉnh.

III. CÀI ĐẶT VÀ THỬ NGHIỆM

Trong cài đặt thử nghiệm mô hình đề xuất, chúng tôi sử dụng các công cụ Framework Apache OpenNLP dùng để hỗ trợ trong việc xử lý ngôn ngữ tự nhiên và General SQL Parser giúp phân tích cú pháp câu truy vấn SQL.

A. Giới thiệu về framework Apache OpenNLP và General SQL Parser

1. Framework Apache OpenNLP

OpenNLP được bắt đầu phát triển vào năm 2000 bởi Jason Baldridge và Gann Bierner. Các thư viện Apache OpenNLP[14] là một thư viện Java phục vụ cho việc xử lý văn bản NNTN. Nó bao gồm nhiều thành phần hỗ trợ các nhiệm vụ NLP phổ biến nhất, ví dụ như bộ tách từ, phân chia câu, gán nhãn từ loại, trích xuất thực thể có tên, phân tích cú pháp, phân giải đồng tham chiếu. Những nhiệm vụ này rất cần thiết để xây dựng các dịch vụ xử lý văn bản nâng cao hơn. OpenNLP cũng bao gồm các dữ liệu ngẫu nhiên cực đại và thuật toán Perceptron dựa trên học máy. OpenNLP hỗ trợ các xử lý NLP: Bộ tách từ, phân chia câu, gán nhãn từ loại, nhận dạng thực thể có tên, phân tích cú pháp, phân giải đồng tham chiếu.

2. General SQL Parser

Gudu Software đã phát triển bộ phân tích cú pháp SQL giúp đơn giản hóa việc giải mã ngữ pháp SQL và giúp ứng dụng được cập nhật với phiên bản mới nhất của chương trình cơ sở dữ liệu. General SQL Parser[15] là một gói phần mềm cho phép thêm chức năng SQL mạnh mẽ cho ứng dụng. General SQL Parser cung cấp bộ công cụ SQL tùy chỉnh cho nhiều cơ sở dữ liệu sử dụng rộng rãi, trong đó có cú pháp truy vấn cho các hệ cơ sở dữ liệu như SQL Server, Oracle, DB2, MySQL, PostgreSQL, Access.

General SQL Parser giải quyết được nhiều vấn đề chung và khó khăn trong phát triển SQL, bao gồm:

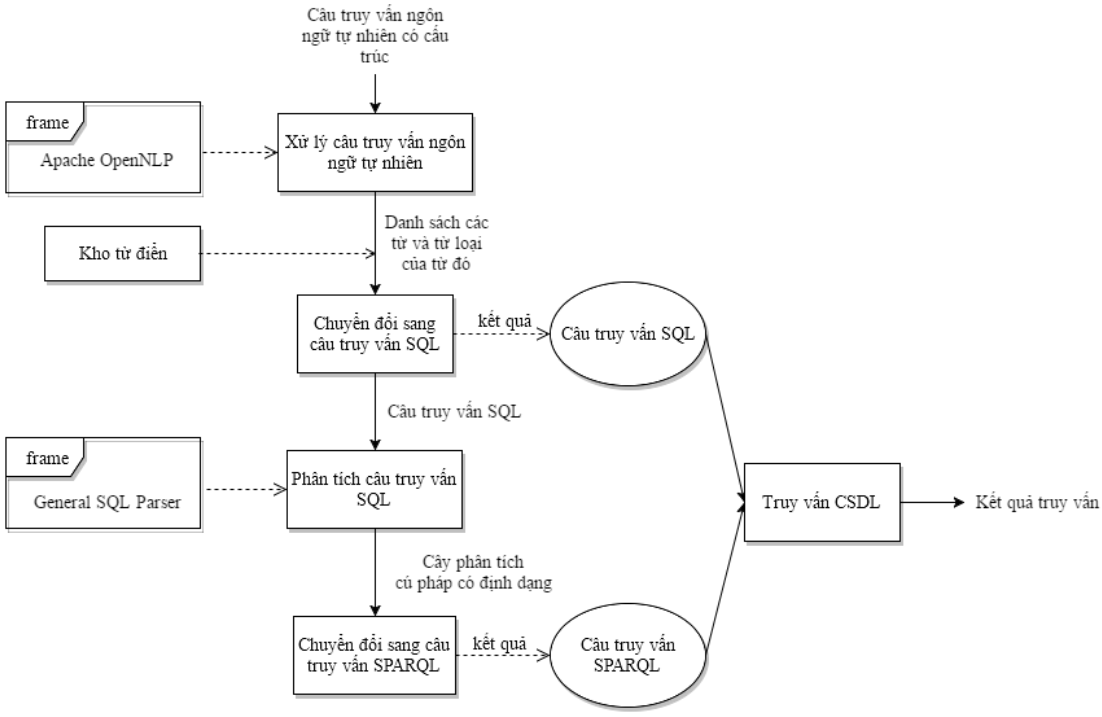
- Kiểm tra cú pháp SQL ngoại tuyến, do đó có thể xác nhận cú pháp mà không cần kết nối cơ sở dữ liệu.
- Các định dạng SQL với hơn 100 định dạng tùy chọn tùy biến cao.
- Thực hiện phân tích chuyên sâu các tập lệnh SQL, bao gồm phân tích cú pháp chi tiết câu SQL thành cấu trúc nút.
- Cung cấp truy cập đầy đủ đến các cây cú pháp truy vấn SQL, bao gồm tìm kiếm, sửa đổi và viết lại các phân đoạn SQL.
- Ngăn chặn các cuộc tấn công SQL Injection.
- Dịch SQL giữa các cơ sở dữ liệu khác nhau.

General SQL Parser bao gồm đầy đủ các gói thư viện có sẵn cho các môi trường phát triển như .NET, Java, C/C++, COM, VCL.

B. Cài đặt mô hình đã đề xuất

Dựa vào mô hình hệ thống tổng quan ở phần III và phần giới thiệu các framework hỗ trợ nhiệm vụ chuyên biệt trong hệ thống, chúng tôi xây dựng nên một sơ đồ cài đặt giải pháp chi tiết như **Hình 5.** Hệ thống bao gồm các module xử lý truy vấn bằng NNTN. Trong module này, chúng tôi sử dụng framework Apache OpenNLP hỗ trợ quá trình xử lý NNTN và đầu ra là danh sách các từ và từ loại của từ đó. Module thứ hai là xây dựng câu truy vấn SQL dựa vào các từ, từ loại của từ, từ khóa đã xác định ở module 1, từ đó phân tích ngữ nghĩa của câu truy vấn bằng NNTN để xác định các thành phần và sinh ra câu truy vấn SQL. Module thứ ba sử dụng framework General SQL Parser để phân tích câu truy vấn SQL thành cây phân tích cú pháp có định dạng, là đầu vào của module thứ tư để sinh ra câu truy vấn SPARQL.

[14] https://opennlp.apache.org
[15] http://www.sqlparser.com/



Hình 5. Sơ đồ cài đặt mô hình chi tiết

Khi người dùng nhập câu vấn bằng NNTN, hệ thống sẽ xử lý câu truy vấn nhờ sự giúp đỡ của framework Apache OpenNLP để đưa ra danh sách các từ và từ loại của từ đó. Dùng bộ từ điển để ánh xạ xác định thành phần của câu truy vấn, từ đó tổng hợp các thành phần thành câu truy vấn SQL. Nếu người dùng chọn truy vấn dữ liệu RDF, hệ thống sẽ dùng framework General SQL Parser để phân tích câu truy vấn SQL thành dạng cây phân tích cú pháp có định dạng, từ đó chuyển sang câu truy vấn SPARQL.

Từ sơ đồ cài đặt chi tiết, tiến hành xây dựng chương trình bằng Java với các hàm thực hiện các công việc cụ thể. Một số hàm chính được cài đặt được giới thiệu và mô tả ngắn gọn trong **Bảng 5**.

Bảng 5. Mô tả một số hàm chính được cài đặt trong chương trình

CÁC HÀM CHÍNH ĐƯỢC CÀI ĐẶT TRONG CHƯƠNG TRÌNH	
Hàm, đầu vào và đầu ra của hàm	**Mô tả ngắn gọn về hàm**
Generate **Đầu vào**: Câu truy vấn NNTN có cấu trúc và loại câu truy vấn tùy chọn **Đầu ra**: Câu truy vấn tương ứng	Hàm có nhiệm vụ phân tích câu truy vấn NNTN có cấu trúc thành các từ và xác định từ loại của chúng. Sau đó gọi các hàm trong class GenerateSQL và GenerateSPARQL để lấy kết quả là câu truy vấn tương ứng.
querySelect **Đầu vào:** Danh sách các từ và từ loại của chúng. **Đầu ra:** Câu truy vấn Select SQL	Hàm có nhiệm vụ sinh ra câu truy vấn Select SQL dựa vào phương pháp ánh xạ danh sách từ vào các từ điển đã được định nghĩa trước trong các file XML để xác định các thành phần của câu truy vấn.
queryInsert **Đầu vào:** Danh sách các từ khóa. **Đầu ra:** Câu truy vấn Insert SQL	Hàm có nhiệm vụ sinh câu truy vấn Insert SQL.
querySPARQL **Đầu vào:** Câu truy vấn Select SQL **Đầu ra:** Câu truy vấn Select SPARQL	Hàm processQuery sử dụng General SQL Parser để phân tích cú pháp câu SQL thành cây cú pháp. Sau đó dựa vào số bảng trong mệnh đề FROM SQL để gọi các hàm *simpleLightSelectWhere* cho một bảng và *complexLightSelectWhere* cho nhiều bảng.

C. Kết quả thực nghiệm hệ thống

1. Cơ sở dữ liệu dùng cho thực nghiệm

CSDL được xây dựng đơn giản để kiểm tra việc chuyển đổi câu truy vấn tự nhiên có cấu trúc sang câu truy vấn SQL và SPARQL. Mục đích CSDL là lưu trữ thông tin của một liên hệ (contact) chứa các thông tin gồm họ tên, địa chỉ và số điện thoại nhà, địa chỉ và số điện thoại nơi làm việc, địa chỉ mail.

▪ Cơ sở dữ liệu quan hệ

CSDL quan hệ được xây dựng với tên bảng là đối tượng liên hệ (contact), với các thuộc tính có khóa chính là id, bảng **contact** được biểu diễn như sau:

contact (*id, name, prename, homeaddress, hometelephone, workaddress, worktelephone, email)*

▪ Cơ sở dữ liệu ngữ nghĩa

CSDL ngữ nghĩa được xây dựng tương tự như của CSDL quan hệ gồm class **contact**, với các thuộc tính dữ liệu (data properties) tương tự thuộc tính được biểu diễn ở CSDL quan hệ. Một đối tượng contact sẽ được lưu trong file dữ liệu dưới dạng như sau:

```
<owl:NamedIndividual rdf:about="&contact;contact1">
  <rdf:type rdf:resource="&contact;Contact"/>
  <hometelephone>01656558000</hometelephone>
  <worktelephone>01234757113</worktelephone>
  <name>Tin</name>
  <prename>Nguyễn Văn</prename>
  <id>1</id>
  <homeaddress>Phổ Văn - Đức Phổ - Quảng Ngãi</homeaddress>
  <workaddress>Liên Chiểu - Đà Nẵng</workaddress>
  <email>vantinqng@gmail.com</email>
</owl:NamedIndividual>
```

Ngoài ra, chúng tôi xây dựng thêm bộ từ điển mô tả cho từng thuộc tính của CSDL giúp cho người dùng dễ dàng truy vấn dữ liệu bằng NNTN.

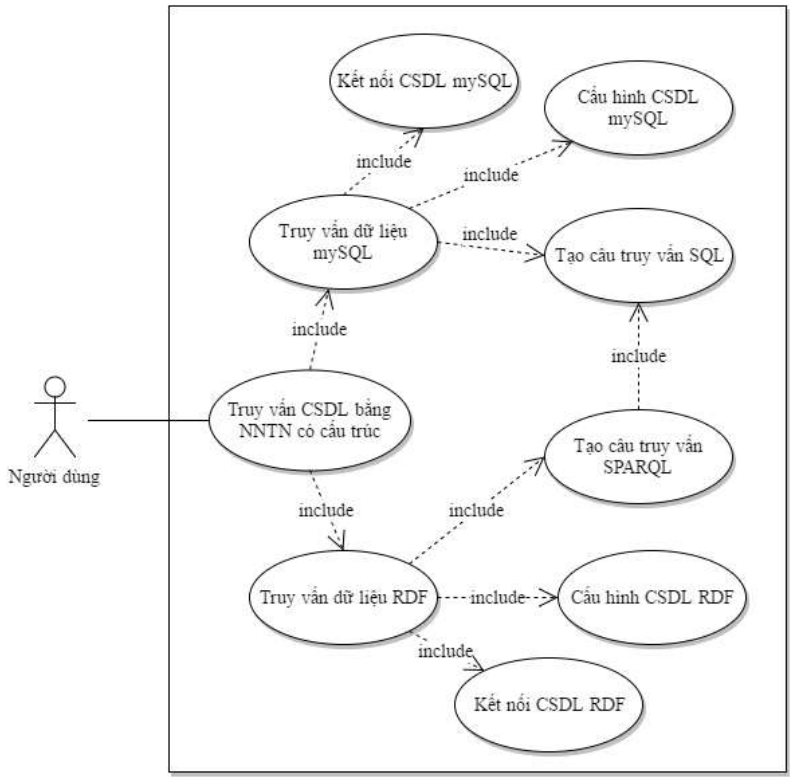
Bảng 6. Từ khóa và thuộc tính tương ứng trong CSDL

Từ khóa	*Thuộc tính tương ứng trong CSDL*
Last Name	name
First Name	prename
Home address	homeaddress
Home phone number	hometelephone
Work address	workaddress
Work phone number	worktelephone
Email	email

*2. Xây dựng hệ thống thực nghiệm mô hình **NL2SQL-SPARQL***

Chúng tôi xây dựng hệ thống ***NL2SQL-SPARQL*** gồm các ca sử dụng như **Hình 6**.

- Truy vấn CSDL bằng NNTN có cấu trúc: khi người dùng muốn truy vấn dữ liệu, người dùng nhập câu truy vấn bằng NNTN và chọn mô hình CSDL.
- Truy vấn dữ liệu mySQL: khi người dùng muốn truy vấn CSDL mySQL.
- Truy vấn dữ liệu RDF: khi người dùng muốn truy vấn CSDL RDF.
- Cấu hình CSDL mySQL: muốn kết nối tới CSDL mySQL, người dùng cần cấu hình các thông số như: hostname, username, password, database name
- Cấu hình CSDL RDF: cấu hình đường dẫn tới tập tin lưu trữ CSDL RDF.
- Tạo câu truy vấn SQL: ca sử dụng thực hiện phân tích câu truy vấn NNTN có cấu trúc để sinh câu truy vấn SQL.
- Tạo câu truy vấn SPARQL: ca sử dụng thực hiện phân tích câu truy vấn SQL để sinh câu truy vấn SPARQL.
- Kết nối CSDL mySQL: thực hiện kết nối tới CSDL mySQL dựa trên các thông tin đã được cấu hình.
- Kết nối CSDL RDF: thực hiện kết nối tới CSDL RDF dựa trên đường dẫn đã được cấu hình.



Hình 6. Sơ đồ ca sử dụng của chương trình

Giao diện chính của ***NL2SQL-SPARQL*** như trong **Hình 7** chứa:

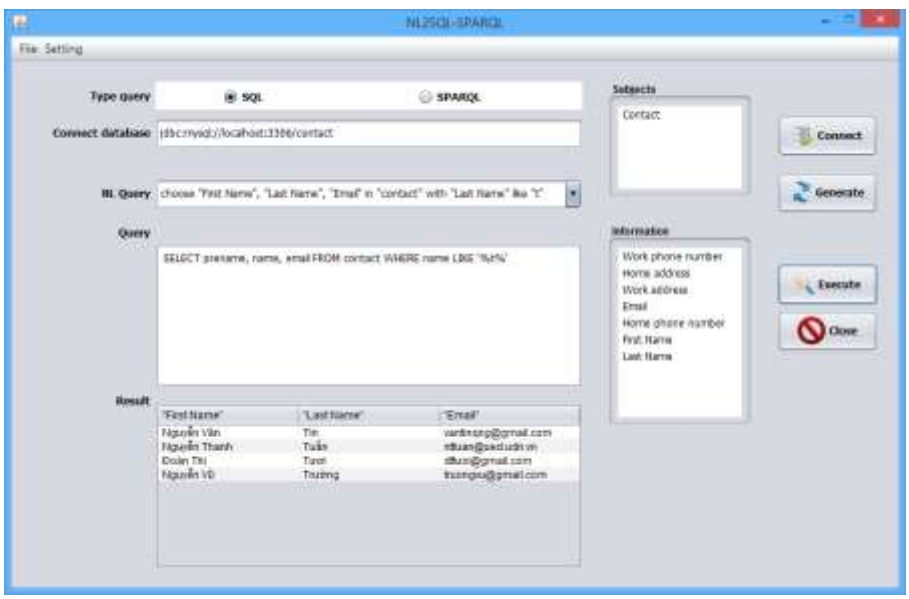
+ Một nhóm nút cho phép chọn kết quả sinh ra của câu truy vấn NNTN là SQL hoặc SPARQL, khi chọn một trong hai nút sẽ hiện ra thông tin cấu hình CSDL tương ứng;

+ Giao diện nhập câu truy vấn bằng NNTN, có chứa một số mẫu câu truy vấn có sẵn. Bên cạnh có một danh sách chứa các từ khóa gợi ý cho người dùng để nhập câu truy vấn bằng NNTN;

+ Giao diện chứa câu kết quả câu truy vấn được sinh ra;

+ Bảng chứa kết quả truy vấn CSDL;

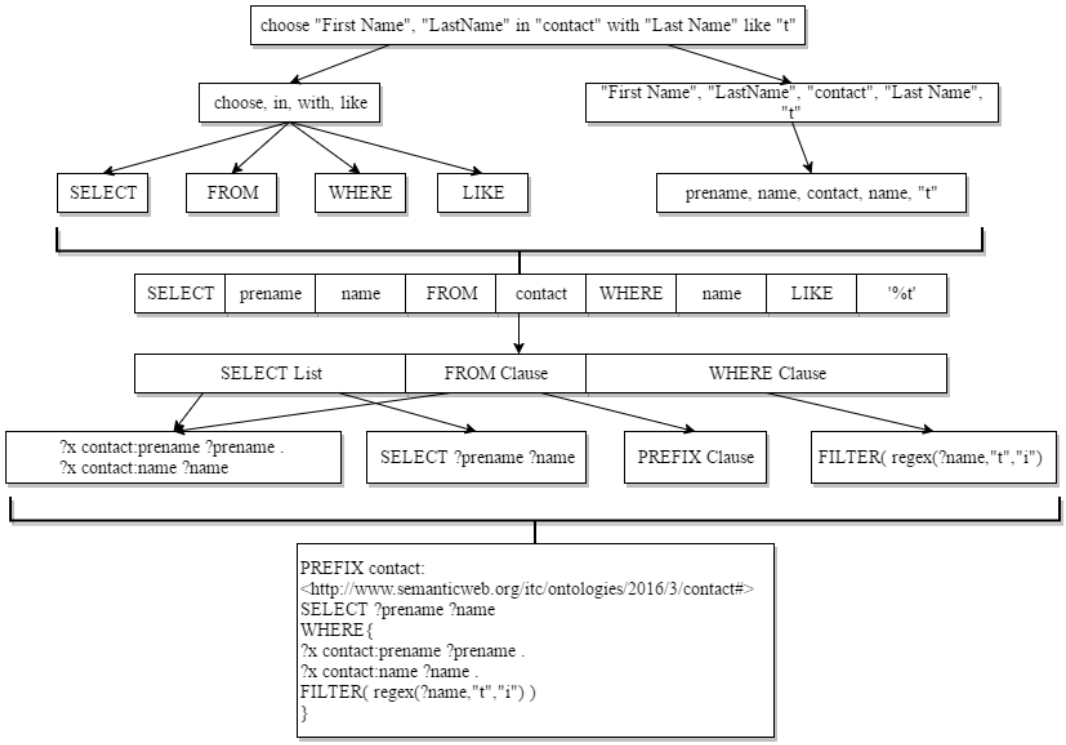
+ Các nút thực thi.



Hình 7. Giao diện chính của hệ thống NL2SQL-SPARQL và kết quả thực nghiệm

Hình minh họa kết quả của từng bước xử lý của hệ thống ***NL2SQL-SPARQL*** với ví dụ:

choose "First Name", "Last Name" in "contact" with "Last Name" like "t"



Hình 8. Sơ đồ biểu diễn kết quả từng bước xử lý của hệ thống

3. Một vài kết quả thử nghiệm

Để đánh giá khả năng thực thi của chương trình đã xây dựng, chúng tôi đã đưa ra một vài ví dụ để chương trình chuyển đổi câu truy vấn và thực hiện truy vấn lấy kết quả từ CSDL. Sau khi thực hiện những ví dụ bằng **NL2SQL-SPARQL**, chúng tôi đã tổng hợp các kết quả ở **Bảng 7**.

Bảng 7. Minh họa kết quả truy vấn của chương trình

Câu truy vấn NNTN	*Câu truy vấn SQL*	*Câu truy vấn SPARQL*	*Kết quả*

list "First Name", "Last Name" in "contact"	SELECT prename, name FROM contact	SELECT ?prename ?name
		WHERE{
		?x contact:prename ?prename .
		?x contact:name ?name .
		}
			"First Name" / "Last Name"
			Lê Hồng / Ân
			Nguyễn Văn / Nhẫn
			Nguyễn Vũ / Trường
			Đoàn Thị / Tươi
			Trương Minh / Ngà
			Huỳnh Thị Lệ / Chi
			Nguyễn Thanh / Tuấn
			Nguyễn Văn / Tin

find "Last Name" and "First Name" in "contact" with "Last Name" = "Tin" or "Last Name" = "Chi"	SELECT name, prename
	FROM contact
	WHERE
	name = 'Tin' OR name = 'Chi'
		SELECT ?name ?prename
		WHERE{
		?x contact:name ?name .
		?x contact:prename ?prename .
		FILTER(?name="Tin" || ?name="Chi")
		}
			"Last Name" / "First Name"
			Chi / Huỳnh Thị Lệ
			Tin / Nguyễn Văn

choose "First Name", "Last Name" in "contact" with "Last Name" like "t"	SELECT prename, name
	FROM contact
	WHERE name LIKE '%t%'
		SELECT ?prename ?name
		WHERE{
		?x contact:prename ?prename .
		?x contact:name ?name .
		FILTER(regex(?name,"t","i"))
		}
			"First Name" / "Last Name"
			Nguyễn Vũ / Trường
			Đoàn Thị / Tươi
			Nguyễn Thanh / Tuấn
			Nguyễn Văn / Tin

IV. KẾT LUẬN

Như vậy, bài báo này đã đề xuất mô hình chuyển đổi câu truy vấn CSDL bằng NNTN thành câu truy vấn CSDL quan hệ hoặc câu truy vấn CSDL ngữ nghĩa. Mô hình này sử dụng các bộ công cụ Apache OpenNLP để hỗ trợ trong việc thực hiện các bước phân tích NNTN. Sau đó, chúng tôi sử dụng kỹ thuật từ điển để tìm ra các tự tương ứng để xây dựng câu truy vấn SQL cho CSDL quan hệ. Trong trường hợp người dùng muốn chuyển đổi câu truy vấn NNTN thành câu truy vấn CSDL ngữ nghĩa SPARQL, chúng tôi sử dụng bộ tích hợp General SQL Parser để phân tích câu truy vấn SQL, sau đó xây dựng câu truy vấn cho CSDL ngữ nghĩa. Mô hình này đã được thực nghiệm bằng hệ thống ***NL2SQL-SPARQL*** . Bước đầu ***NL2SQL-SPARQL*** đã thử nghiệm thành công việc sinh câu truy vấn chọn (select) sang mô hình CSDL quan hệ và sang CSDL ngữ nghĩa. Tuy nhiên hiệu quả của hệ thống phụ thuộc rất nhiều vào từ vựng trong từ điển, đây là khó khăn lớn nhất và là vấn đề cơ bản của bất kỳ hệ thống xử lý ngôn ngữ tự nhiên nào. Việc mở rộng từ điển và phân tích sâu hơn ngữ nghĩa của câu truy vấn bằng NNTN sẽ là hướng nghiên cứu tiếp theo của bài báo. Ngoài ra, hệ thống chỉ mới thử nghiệm ở câu lệnh *select*. Hướng sắp tới, chúng tôi sẽ mở rộng từ điển và áp dụng ***NL2SQL-SPARQL*** cho các loại câu truy vấn còn lại. Cuối cùng, việc mở rộng mô hình để hỗ trợ thêm nhiều loại mô hình CSDL khác ngoài CSDL quan hệ và CSDL ngữ nghĩa cũng là một trong những hướng phát triển khác của chúng tôi.

TÀI LIỆU THAM KHẢO

[1] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, "Natural Language Interfaces to Databases - An Introduction," *Journal of Natural Language Engineering*, no. 709, p. 50, 1995.

[2] R. Alexander, P. Rukshan, and S. Mahesan, "Natural Language Web Interface for Database (NLWIDB)," no. July, pp. 6–7, 2013.

[3] D. T. Jayakumar, Naveenkumar, Raj, SDjoshi, "A Survey of Natural Language Query Builder Interface to Database," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 9, pp. 62 – 70, 2012.

[4] N. K. Anh and P. T. T. Hoài, "Truy vấn ngôn ngữ tự nhiên không hoàn chỉnh đối với các cơ sở dữ liệu quan hệ," 2006, pp. 117–122.

[5] C. Unger and P. Cimiano, "Pythia: Compositional meaning construction for ontology-based question answering on the semantic web," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6716 LNCS, pp. 153–160.

[6] P. Toge, P. Gargote, and P. S. Bajpai, "Natural Language to Relational Query by Using Parsing Compiler," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 3, pp. 485–494, 2015.

[7] S. Kaur and R. S. Bali, "SQL generation and execution from natural language," *International Journal of Computing & Business Research*, 2012.

[8] S. Ramanujam, A. Gupta, L. Khan, B. Thuraisingham, and S. Seida, "R2D: A framework for the relational transformation of RDF data," *International Journal of Semantic Computing*, vol. 3, no. 4, 2009.

MODEL FOR QUERY OFTIONAL DATA BASED ON THE SEMANTICS OF THE QUERY

Nguyen Thanh Tuan, Hoang Thi Thanh Ha, Nguyen Van Tin, Le Thanh Nhan

ABSTRACT— *Currently, along with the expansion of the World Wide Web is the rapid growth of data. Huge data source is still stored in the various database models: relational database, RDF database, ... The abundance of all kinds of database makes it difficult for users to keep up with the development of a query language for accessing databases in which they want to search. This paper presents the approach allows the user can select and query the desired data from two types of popular database are the RDF database and the relational database. Based on the semantics of the query in natural language and selected database type, the system will suggest a corresponding query allows users to check before agreeing to execute.*