**PHƯƠNG PHÁP ĐỀ NGHỊ CHO VIỆC XỬ LÝ CÂU HỎI**

1. **Xây dựng File miêu tả ngữ nghĩa cho Database và các mối quan hệ giữa các bảng.**

Bởi vì thông tin trên Database chưa thật sự miêu tả rõ các mối quan hệ ngữ nghĩa giữa các bảng với nhau hoặc giữa bảng với các thuộc tính của bảng đó. Chẳng hạn, với một bảng có tên là Book và nó có thuộc tính là publisher, ta có thể định nghĩa quan hệ giữa bảng Book và thuộc tính publisher là “is published”.

Hơn nữa, tên của các bảng hay thuộc tính trong database đôi lúc được đặt bởi người dùng, điều đó gây ra sự khó khăn cho việc nhận dạng các thực thể của câu hỏi.

Ví dụ: với câu hỏi *Who write “*Active Database Systems*”*. Giả sử trong database, bảng chứa thông tin để trả lời câu hỏi vừa rồi có tên là dblp\_pub\_new. Đây là một tên được đặt bởi người tạo ra database, cho nên nó không mang một ngữ nghĩa nào hết. Vì thế, việc tận dụng các từ điển đồng nghĩa để xác định sự xuất hiện của tên bảng trong câu trở nên khó khăn. Nhưng nếu ta đặt tên bảng *dblp\_pub\_new* bằng một tên tượng trưng khác có nghĩa như *publication* thì việc nhận dạng sẽ dễ dàng hơn nhiều.

Đồng thời, cần có một thông tin để miêu tả các mối quan hệ giữa bảng và các thuộc tính trong bảng, ánh xạ giữa các bảng nếu các bảng đó có quan hệ với nhau. Điều này sẽ giúp ích nhiều trong việc phát sinh câu truy vấn SQL tự động. Từ một câu hỏi được đưa ra, dựa vào các đối tượng trong câu hỏi đó và những mối quan hệ ánh xạ của các bảng với nhau trong database, hệ thống sẽ sinh ra câu truy vấn.

Chính vì thế, nhóm đề xuất một lược đồ XML để mô tả thông tin của database và các mối quan hệ.



Trong đó: (giá trị đầu là tên của nút cha, giá trị trong dấu “[]” là tên của nút con)

* Database[database-name]: tên của database.
* database [mapping-table]: là thông tin của table ở giữa dùng để ánh xạ mối quan hệ n-n giữa 2 table với nhau.
* mapping-table [table-key]: là khóa chính của table đầu tiên.
* mapping-table [related-table-key]: là khóa chính của table quan hệ.
* mapping-table [mapping-table-name]: là tên của table ánh xạ đã có sẵn trong database.
* mapping-table [name] tên của mapping-table do người dùng đặt.
* database[table]: thông tin của table trong database
* table [primary-key]: Khóa chính của table.
* table [table-name]: Tên của table.
* table[visible]: là thuộc tính dùng để thể hiện table đó có được hiện ra cho người dùng thấy hay không.
* table[table-alias]: tên gọi khác của table đó. Ví dụ: Một table có table-name là bks. Ta có thể đặt table-alias là Books để người dùng dễ hiểu.
* table[field] thông tin thuộc tính của table.
* field[field-name] tên của thuộc tính trong database.
* field[field-alias] tên gọi khác của thuộc tính.
* field[type]: Kiểu dữ liệu của thuộc tính đó.
* field[visible] thể hiện thuộc tính đó có hiện cho người dùng thấy hay không.
* field[relation]: thể hiện mối quan hệ giữa table và thuộc tính.
* relation[relation-name] tên của mối quan hệ.
* relation[related-table] tên của table quan hệ. Thông tin này sẽ hiểu rõ hơn trong ví dụ.
* relation[reversed-relation] tên quan hệ đảo.
* relation[mapping-table] tên của table-mapping ở giữa 2 table

Ví dụ: với database của DBLP, ta có thể xây dựng file XML sau:

<database>

<!-- Tên do người dùng đặt thay thế cho dblp\_pub\_new -->

<table table-alias="publication">

<table-name>dblp\_pub\_new</table-name>

<!--Thông tin này không được hiển thị cho người dùng-->

<field type="String" field-alias="author" visible="false">

<relation>

<!--Thiết lập các loại quan hệ chính sẽ xuất hiện trong câu hỏi -->

<relation-name>has author,is written</relation-name>

<!--Thiết lập các loại quan hệ đảo sẽ xuất hiện trong câu hỏi -->

<reversed-relation>write</reversed-relation>

<!-- Thông tin ánh xạ -->

<mapping-table>pub\_author</mapping-table> </relation>

</field>

<field type="String" field-alias="title" visible="true">

<relation>

<relation-name>has title</relation-name>

<!-- Vì thuộc tính này không được ánh xạ thông qua table nào khác nên nó sẽ không có mapping-table hoặc related-table -->

</relation>

<field-name>title</field-name>

</field>

<field type="String" field-alias="source" visible="true">

<relation>

<relation-name>has source</relation-name>

</relation>

<field-name>source</field-name>

</field>

<!-- Các thuộc tính khác -->

<!-- ................... -->

<field type="Int" field-alias="year" visible="true">

<relation>

<relation-name>has year,in,since</relation-name>

</relation>

<field-name>year</field-name>

</field>

<field type="String" field-alias="type" visible="true">

<relation>

<relation-name>has type</relation-name>

</relation>

<field-name>type</field-name>

</field>

<field type="String" field-alias="publisher" visible="true">

<relation>

<relation-name>has publisher,is published</relation-name>

<reversed-relation>publish</ reversed-relation >

</relation>

<field-name>publisher</field-name>

</field>

<primary-key>id</primary-key>

</table>

<!-- Các table khác -->

<!-- ................... -->

<database-name>dblp</database-name>

<mapping-table>

<!-- Tên table mapping là dblp\_author\_ref\_new trong database -->

<mapping-table-name>dblp\_author\_ref\_new</mapping-table-name>

<!-- Tên table mapping do người dùng đặt -->

<name>pub\_author</name>

<!—Khóa chính của table dblp\_pub\_new tương ứng trong table dblp\_author\_ref\_new-->

<table-key>pub\_id</table-key>

<related-table-key>author</related-table-key>

</mapping-table>

</database>

1. **Các bước xử lý câu hỏi**

1 .TIỀN XỬ LÍ CÂU HỎI

2.LIỆT KÊ CÁC BỘ BA QUAN HỆ VỀ TỪ

3.NHẬN DIỆN LẠI THỰC THỂ

4.NHẬN DẠNG QUAN HỆ ẨN

5.SINH CÂU TRUY VẤN

**1. Tiền xử lý câu hỏi:**

**1.1. Định dạng lại thực thể đặt tên**

Thực thể đặt tên là những thành phần của văn bảng thuộc vào những mục sau: tên người, tên tổ chức, địa điểm, thời gian, số lượng, phần trăm … (<http://en.wikipedia.org/wiki/Named_entity_recognition>)

Theo ý kiến đề xuất của nhóm, trong câu, thực thể đặt tên là những từ hoặc cụm từ được xác định bởi quy tắc sau (không xét đến các thực thể có kiểu dữ liệu là số và ngày tháng):

* Những từ viết hoa hoặc những cụm từ viết hoa liên tiếp.

Ví dụ: Mark Twain, John, ...

* Những từ hoặc cụm từ trong dấu ngoặc kép <“>.

Ví dụ: “Active Database Systems”.

* Những từ viết hoa tòan bộ

Ví dụ: ACM, IEEE …

Việc định dạng thực thể đặt tên sẽ giúp cho việc xử lí gán nhãn từ lọai (Pos Tagger) sẽ tốt hơn. Xử lí Pos Tagger là quá trình xác định từ loại của một từ trong câu. Hiện nay, có các nhiều công cụ để giúp ta có thể xử lý Pos Tagger được đưa ra như OPENNLP[[1]](#footnote-2), ANNIE POS TAGGER[[2]](#footnote-3) … Mặc dù các công cụ đó đã giúp người dùng có thể gán nhãn hầu như chính xác các từ loại trong câu tiếng anh, tuy nhiên với những câu có những thực thể đặt tên là một cụm danh từ thì việc gán nhãn từ loại từng loại từ trong câu lại không chính xác về mặt ý nghĩa của cụm từ đó**.** Điển hình: Các công cụ đó chưa có thể hiểu được những thực thể đặt tên là những cụm từ viết hoa liên tiếp hay những cụm từ trong dấu <”> hoặc <’> như đã nêu mà thay vào đó là hiểu những từ riêng lẽ trong những cụm từ đó.

Chẳng hạn với câu hỏi:

*Who is the author of* ***“A Distributed Object Manager for the Smalltalk-80 System”****?*

*Thì kết quả xử lý Pos Tagger với công cụ OpenNLP sẽ cho ra kết quả như sau:*



Với kết quả trên thì kết quả của câu trên sẽ là:

*Who(*WP) *is(*VBZ) *the(DT) author(NN) of(IN) “(NN)* ***A(DT) Distributed(NNP) Object(NNP) Manager(NNP) for(IN) the(DT) Smalltalk-80(JJ) System(NNP) ”(NNP) ?***

Trong đó:

WP: Đại từ câu hỏi.

VBZ: Động từ số ít.

JJ: Tính từ

NN: danh từ số ít

NNP:Danh từ viết hoa số ít.

DT: từ hạn định

IN: Giới từ.

Công cụ ANNIE POS TAGGER sẽ cho kết quả tương tự với khác biệt là dấu ” thứ 2 được gán nhãn là NN và xuất hiện nhãn NNP CD thay vì JJ như ở OpenNLP.



Với câu trên thì dấu “ đã được gán nhãn là NNP hoặc NN. Điều này thật sự không chính xác. Hơn nữa, kết quả này cũng đã làm mất đi ý nghĩa của cụm từ ***“A Distributed Object Manager for the Smalltalk-80 System”***

Vì thế, nhóm đề xuất quy tắc định dạng lại thực thể đặt tên như sau:

Thay những cụm từ của những thực thể đặt tên đã xác định bằng quy tắc trước đó bằng một dãy các ký tự bắt đầu bằng chữ hoa tương ứng với mã số kèm theo. Như vậy, các công cụ xử lí Pos Tagger sẽ hiểu những cụm từ đó là các NNP.

Cụ thể, các thực thể đặt tên đó sẽ thay theo mẫu : NER+SỐ THỨ TỰ. Số thứ tự này và thực thể đặt tên sẽ được lưu vết lại để xác định thực thể đặt tên tương ứng với mã số.

Ví dụ: Với câu hỏi *Who is the author of* ***“A Distributed Object Manager for the Smalltalk-80 System.”****?*, kết quả xử lí Pos Tagger (bằng OpenNLP) như sau:

* Qua xử lý đặt “***A Distributed Object Manager for the Smalltalk-80 System.***” là NER001
* *Kết quả gán nhãn từ lọai: Who(WP) is(VBZ) the(DT) author(NN) of(IN) NER001(NNP) ?*

**1.2. Xây dựng danh sách Pos Tagger**

Việc xử lý Pos Tagger sẽ giúp ta dễ dàng xác định vai trò các từ trong câu. Từ đó, ta dễ dàng ánh xạ với các bộ từ điển ngữ nghĩa như Wordnet để so sánh đối chiếu, hay tìm đồng nghĩa, tìm khái niệm, tổng quá hóa... Hơn nữa nó còn giúp ta xác định các từ quan hệ và các từ mà các bước sau dùng để xác định thực thể chưa đặt tên.

Ở bước này, Nhóm sử dụng công cụ OpenNLP để gán nhãn từ loại và các nhãn NNP có giá trị là NER+Mã Số sẽ được gán lại giá trị đã lưu vết ở bước trên.

**1.3. Tối ưu hóa danh sách Pos Tagger**

Mặc dù đã tiền xử lí ở bước trên, nhưng các động từ trong câu ở các kết quả ở trên vẫn cần được xác định thêm. Do đôi khi trong câu xuất hiện các ngữ động từ.

Ví dụ: động từ “look up” là sự kết hợp giữa động từ “look” và giới từ “up”. Với kết quả ở các bước xử lý trên thì “look” và “up” là hai từ riêng lẻ.

Do vậy, nhóm đề xuất phương pháp tối ưu hóa danh sách Pos Tagger cùng với sự hỗ trợ của bộ từ điển Wordnet để xác định nghĩa của từ, với các bước sau:

Chuyển các động từ về dạng đơn giản. Trong câu, các động từ thường được chia bởi nhiều thì khác nhau. Động từ nên chuyển thành dạng đơn giản (nguyên mẫu) để tiện cho việc xử lý. Đồng thời nhóm đề xuất dùng công cụ RitaWordnet để chuyển động từ về dạng nguyên mẫu. Sau đây là những quy tắc đề xuất để đơn giản hóa cụm động từ:

* VBP/VBZ(1) + VBN(2) 🡪 VBP(2).

Ví dụ: has written 🡪 write.

* VBP/VBZ (1) + VBN (2) + VBN (3) 🡪 VBZ(2) + VBP(3)

Ví dụ : has been written 🡪 is write

* VBP/VBZ (1) + VBN (2) + VBG (3) 🡪 VBP(3)

Ví dụ: has been writing 🡪 write.

* VBP/VBZ(1) + VBG(2) 🡪 VBP(2).

Ví dụ: is writing 🡪 write.

* VBP/VBZ(1) + VBN(2) 🡪 VBZ(1) + VBP(2) (VBP/VBZ là động từ TOBE)

Ví dụ: are written 🡪 is write

* VBP/VBZ(1) + VBG(2) + VBN(3) 🡪 VBZ(1) + VBP(3)

Ví dụ: is being written 🡪 is write

* VBD(1) 🡪 VBP(1).

Ví dụ: wrote🡪 write.

* VBD(1) + VBG(2) 🡪 VBP(2).

Ví dụ: was writing 🡪 write.

* VBD(1) + VBN(2) 🡪 VBZ(1) + VBP(2)

Ví dụ: was written 🡪 is write

* VBD(1) + VBG(2) + VBN(3) 🡪 VBZ(1) + VBP(3)

Ví dụ: was being written 🡪 is write.

* MD(1) + VBN(2) 🡪 VBP(3)

Ví dụ: can write 🡪 write.

* MD(1) +VB(2) + VBN(2) 🡪VBZ(2) + VBP(3)

Ví dụ: can be written 🡪 is write

* Nếu

+ *Động từ* + *Giới từ* là một cụm từ có nghĩa:

Động từ = Động từ + Giới từ

+ Ngược lại: Động từ = Động từ.

Như vậy, nếu một câu hỏi tương tự với từ “search” được thay thế cho từ “look up” thì với việc sử dụng từ điển WordNet thì ngữ nghĩa câu hỏi không thay đổi. Ngòai ra, các mạo từ như: the, a, an được lọai bỏ khỏi danh sách do chúng không có ý nghĩa gì trong thao tác truy vấn.

**2. Liệt kê các bộ 3 quan hệ về từ**

Đối với các cách xác định bộ ba của một số nghiên cứu đã nêu, ta có thể dung một từ điển như Wordnet để xác định các thực thể đặt tên, không đặt tên, động từ quan hệ. Ngòai ra, công cụ Gate – một công cụ xử lý văn bản – cũng hỗ trợ tốt cho vấn đề xác định thực thể đặt tên dựa trên các tập luật mà Gate đề ra. Hơn thế nữa, Gate còn hỗ trợ các plugin như Gazeteer và ONCAT để hỗ trợ thêm cho việc nhận dạng thực thể.

Theo cách tạo đồ thị ý niệm của Wael Salloum, một số dạng mẫu của câu được xác định trong quá trình hình thành đồ thị. Những mẫu này dựa trên cây cú pháp của câu sau khi được phân tích. Ví dụ: “The human liver secretes the bile” thì có mẫu NP-VP-NP (Noun Phrase - Verb Pharse – Noun Pharse) tương ứng với

(The(DT) human(ADJ) liver(NN)) (NP)- (secretes(VBZ))(VP) - (the(DT) bile (NN))(NP)

Ở đây ta cũng liệt kê bộ ba theo những mẫu quy tắc từ lọai như sau:

NN/WP(1) + VBP + NNP/NN(2) 🡪 <NN/WP(1),VBP,NNP/NN(2)>

NN/WP(1) +VBZ + VBP + NNP/NN(4) 🡪 <NN/WP(1),VBZ + VBP,NNP/NN(2)>

NN/NNP(1)’s NN(2) 🡪 <NN/NNP(1),has,NN(2)>

NN(1) +of+ NN/NNP(2) 🡪 <NN/NNP(2),has,NN(1)>

NN/NNP(1) + IN + NN/NNP/DT(2) 🡪<NN/NNP/DT(1),IN,NN(2)>l

NN/NNP(1) + IN(2) + NN/NNP/DT(3) + IN(4) + NN/NNP/DT(5) 🡪<NN/NNP/DT(1),IN(1),NN(3)>, <NN/NNP/DT(1),IN(4),NN(5)>

**3.Nhận diện thực thể**

**3.1. Nhận diện tất cả các thực thể dựa vào từ quan hệ:**

Ta dựa vào từ quan hệ trong bộ ba được liệt kê ở bước trên. Từ đó sẽ được đem tra vào các từ quan hệ được liệt kê cho mỗi thuộc tính ở tập tin cấu hình ngữ nghĩa nêu ở mục trên. Trong trường hợp không tìm thấy, ta có thể kết hợp với từ điển đồng nghĩa của Wordnet để hỗ trợ tìm kiếm tốt hơn.

Ta có các trường hợp sau:

* Nếu tìm thấy một trường hợp duy nhất so khớp với từ quan hệ:

+ Nếu từ quan hệ được tìm thấy ở mối quan hệ ngược thì thứ tự quan hệ trong tập tin cấu hình sẽ đảo ngựơc lại.

Ví dụ: Who published “Active Database Systems”. Từ quan hệ sau các bước trên sẽ là publish, bộ 3 liệt kê được là <Who, publish, “Active Database Systems”>. Khi ánh xạ vào XML sẽ tìm thấy một bộ ba duy nhất trong là <Book, publish, publisher>. Như vậy ta chỉ cần đảo ngược lại là <publisher, publish, book>

+ ngược lại thứ tự sẽ đựơc giữ nguyên.

* Nếu tồn tại nhiều kết quả, thì dùng các phương pháp ở mục sau để nhận diện các thực thể, lọai bỏ các bộ ba không phù hợp.
* Nếu không tồn tại kết quả nào, kiếm từ đồng nghĩa với từ quan hệ, lặp lại bước tìm kiếm.

**3.2. Các phương pháp nhận diện thực thể**

**3.2.1.Nhận diện thực thể đặt tên**

Ta chỉ cần ánh xạ giá trị của thực thể đặt tên đó vào cơ sở dữ liệu. Với trường hợp ở trên, đối với một tập các bộ ba có từ quan hệ giống nhau. Ta chỉ cần lấy giá trị của thực thể đặt tên ánh xạ vào miền giá trị của thuộc tính đã biết trong bộ ba. Nếu tồn tại, bộ ba đó được giữ lại, ngược lại bộ ba bị lọai bỏ.

Nếu không xác định được thuộc tính, việc ánh xạ này sẽ được ưu tiên cho các thuộc tính đại diện của cho tên bảng (Như mô hình XML ở trên, thuộc tính đại diện của Author sẽ là Name) rồi mới đến các thuộc tính khác.

Nếu giá trị của thực thể có kiểu dữ liệu là chuỗi(String) thì chỉ tìm ở các thuộc tính có kiểu dữ liệu là chuỗi(String), nếu kiểu dữ liệu là số(Integer) thì sẽ tìm ở những thuộc tính có kiểu dữ liệu là số(Integer)...

Ví dụ: Với câu hỏi : “*Who composed* “Active Database Systems” *?*”. Từ quan hệ đứng trước nó là “compose” khi ánh xạ vào các thông tin của database thì ta sẽ tìm thấy table chứa nó là **Publication** và thuộc tính là **title**.

**3.2.2. Nhận diện thực thể chưa đặt tên**

Ngược lại với thực thể đặt tên là thực thể không đặt tên, nó thường là các danh từ xuất hiện trong câu. Đối với phạm vi để tài, thực thể không đặt tên được nhận biết để đánh dấu sự xuất hiện của những bảng trong cơ sở dữ liệu.

Tuy nhiên, trong câu hỏi tự nhiên, người dùng có thể tùy ý nhập vào một từ nào đó có cùng nghĩa với từ mình đã xác định (như tên của bảng, thuộc tính). Ví dụ:

Từ “writer” sẽ đồng nghĩa với từ “author” . Nếu như không nhận biết chúng là hai từ đồng nghĩa trả lời được những câu hỏi liên quan đến writer.

Nhóm đề xuất sử dụng công cụ Similarity Wordnet[[3]](#footnote-4) để hỗ trợ cho quá trình nhận dạng thực thể chưa đặt tên. Similarity Wordnet là một công cụ có thể dùng để đánh giá độ tương đồng về nghĩa giữa hai danh từ. Nếu giữa 2 danh từ hoàn tòan cùng nghĩa, thì mức đánh giá giữa hai danh từ đó là 1. Mức thấp nhất là 0, tức là hai danh từ đó không có liên quan gì về nghĩa với nhau.

Như vậy, với cách này đầu tiên ta chỉ cần tìm các bảng mà tên của nó có độ tương đồng về nghĩa với thực thể không đặt tên trong mức. Theo nhóm, thì chỉ xác định những từ có độ tương đồng trên 0.5.

Theo một thứ tự ưu tiên, đầu tiên sẽ dùng công cụ so sánh với các tên bảng trong cơ sở dữ liệu. Nếu kết quả đều không thỏa, ta so sánh tiếp với các thuộc tính trong các bảng. Nếu kết quả vẫn tiếp tục không thỏa, ta chọn thuộc tính hay tên bảng có độ tương đồng cao nhất.

Ví dụ: Với câu hỏi *Who is the writer of “*Active Database Systems*” ?*

Thực thể chưa đặt tên ở đây (ngoại trừ từ what) là writer. Giả sử trong cơ sở dữ liệu đã có chứa một bảng có tên là Author.

Bằng công cụ Similarity Wordnet, ta có thể nhận biết được từ *writer* là *author* thay vì thay vì tên bảng nào khác trong cơ sở dữ liệu.

**3.2.3. Nhận diện lớp của câu hỏi:**

Với câu hỏi xuất hiện các từ hỏi như Who, What, When, Where, … thì ta áp dụng một số cách nhận diện theo tài liệu [5]. Nếu không, mặc định phân lớp là các đối tượng bài báo.

- Đối với câu hỏi What:

+ Với dạng câu hỏi *What+ Noun+….?* Thì lớp câu hỏi chinh là lớp của danh từ.

-Đối với câu hỏi Who:

+Với dạng câu hỏi *Who is Noun…?* Thì lớp của câu hỏi chính là lớp của Noun

+Với dạng câu hỏi Who do/does +Noun + Verb thì Lớp của câu hỏi Who chỉ có thể liên quan đến Person hoặc Group, vì thế ta tìm tất cả những lớp có độ tương đồng về nghĩa với “Person” và “Group” đồng thời có mối quan hệ liên quan với động từ Verb.

-Đối với câu hỏi Which:

+ Với dạng câu hỏi : Which is/are+Noun…? Thì lớp câu hỏi chính là lớp của Noun

+ Với dạng câu hỏi : Which + Noun …? Thì lớp câu hỏi chinh là lớp của Noun.

- Đối với câu hỏi How:

+Với dạng câu hỏi : How +adj+Noun?

Nếu dạng câu hỏi ở dạng How many +Noun thì lớp của câu hỏi được quy thành COUNT(lớp của Noun)

**4. Xác định quan hệ ẩn**

Như trong nghiên cứu của tác giả Cao Duy Trường, trong quá trình hình thành đồ thị ý niệm, có một số quan hệ ẩn chưa được xác định. Lý do là người dùng không nhập vào từ quan hệ. Các quan hệ đó được ngầm hiểu dựa trên ngôn ngữ con người mà máy không thể hiểu được.

Ví dụ: “Which books were published by ACM, 2009 ?” thì từ quan hệ “in” không được đề cập tới. Đơn giản người dùng chỉ đề cập tới nhà xuất bản là ACM và con số 2009. Con số 2009 phải được hiểu là năm mà nhà xuất bản ACM phát hành các cuốn sách đó.

Vì một số trường hợp giữa 2 thực thể trong câu hỏi đứng liền kề nhau nhưng giữa chúng không có từ quan hệ nào, vì thế ta cần xác định quan hệ ẩn giữa chúng.

Quan hệ ẩn còn được xác định bởi ràng buộc sau:

* 2 thực thể liền kề nhau, thực thể đầu tiên là một thực thể chưa đặt tên, thực thể thứ hai là thực thể đặt tên. Nếu thực thể thứ nhất là tên bảng thì thực thể thứ 2 là thuộc tính đại diện cho bảng đó. Thuộc tính đại diện là thuộc tính được quy định cho mỗi bảng. Mỗi bảng chỉ có duy nhất một thuộc tính đại diện. (Ví dụ: bảng Author có thuộc tính đại diện là Author\_name. Bảng Publication có thuộc tính đại diện là title.)

Ví dụ: *Who is the author of the publication “*Active Database Systems*”?* Như vậy giữa thực thể *publication* và “Active Database Systems” có mối quan hệ ẩn là “has title”

- Thực thể đặt tên không có thực thể nào đứng trước nó và được nhận diện là thuộc tính của bảng thì sẽ xuất hiện một quan hệ ẩn giữa tên bảng và thuộc tính đại diện cho bảng đó.

Ví dụ: Who write “Active Database Systems” . Ở đây, “Active Database Systems” được xác định là thuộc tính title của bảng publication, như vậy sẽ xuất hiện quan hệ ẩn <Publication, has title, title>

**5.Tạo câu truy vấn**

Ở các bước trên, nhóm đã đề ra các phương pháp để nhận diện thực thể, nhận biết thuộc tính hay tên bảng và tìm ra các mối quan hệ ẩn. Tiếp theo, từ các bộ ba, ta phát sinh ra câu truy vấn.

* **Đối tượng truy vấn:**

Nếu Thực thể câu hỏi là thuộc tính và thuộc tính đó không ánh xạ đến table nào

(Dựa vào file cấu trúc XML ở trên) hoặc thực thể là tên bảng:

Ta có câu truy vấn: SELECT (tên table đã ánh xạ).\*

Ngược lại thì ta truy xuất đến thuộc tính với table tương ứng.

Ta có câu truy vấn: SELECT tên\_table\_của\_thuộc\_tính.tên thuộc tính

Ví dụ: Với câu hỏi : Who write “Harry Potter”?

Thực thể Who được xác định là thuộc tính author của table Book. Và author lại ánh xạ đến một table khác là author.

Ta có một mẫu của câu truy vấn như sau:

Select author.\*

Ví dụ: Với câu hỏi: Who published “Active Database Systems”?

Thực thể Who được xác định là thuộc tính publisher của Book,và nó không ánh xạ đến bất cứ table nào. Do đó ta có câu truy vấn:

Select Book.publisher

* **Nguồn truy vấn:**

Ta cần lập danh sách tất cả các tên table từ danh sách các bộ ba, nếu có thuộc tính có mapping-table hoặc related-table thì thêm mapping-table hoặc related-table đó vào nguồn truy xuất.

Ví dụ: Với câu hỏi: Who write “Active Database Systems”

Ta có bộ 3 quan hệ < Book , write, Who:author>,<Book,has title, “Harry Potter”:title>

Ở đây, who là thuộc tính author của Book, và nó có related-table là author, mapping-table là ref\_author\_book.

Nên ta có được mẫu truy vấn sau:

Select author.\*

From Author,Book,ref\_author\_book

* **Điều kiện truy vấn:**

+ Nếu một thuộc tính có related\_table và có mapping table(quan hệ n-n), ta có thêm điều kiện sau(dựa vào thông tin tại XML):

Table\_chứa\_thuộc\_tính\_đó.primary\_key =

[mapping\_table]mapping-table-name.table-key and

[mapping\_table]mapping-table-name.related-table-key = related-table.primary-key

+ Nếu thuộc tính chỉ có mỗi mapping-table và không có related-table đồng thời thuộc tính đó có giá trị thì ta có điều kiện sau:

Table\_chứa\_thuộc\_tính\_đó.primary\_key =

[mapping\_table]mapping-table-name.table-key

and [mapping\_table]mapping-table-name.related-table-key = giá\_trị\_thuộc\_tính

+ Nếu thuộc tính chỉ có mỗi mapping-table và không có related-table đồng thời thuộc tính đó không có giá trị thì ta có điều kiện sau:

Table\_chứa\_thuộc\_tính\_đó.primary\_key =

[mapping\_table]mapping-table-name.table-key

+ Nếu một thuộc tính chỉ có mỗi related\_table (quan hệ n-1) , ta có điều kiện:

Table\_chứa\_thuộc\_tính\_đó.tên\_thuộc\_tính = related\_table.primary-key

+ Nếu thuộc tính đó đã có giá trị, ta chỉ cần thêm điều kiện:

Table\_của\_thuộc\_tính.Thuộc tính = giá trị của thuộc tính đó.

Ở ví dụ trên, với các bộ 3 đã xác định ,Who là thuộc tính author của “Harry Potter” và có mapping-table-name là ref\_author\_book, có related-table là author, đồng thời thuộc tính Book.title có giá trị là “Active Database Systems”, ta sinh được câu truy vấn sau:

Select author.\*

From Author,Book,ref\_author\_book

Where

Book.id = ref\_author\_book.book\_id and ref\_author\_book.author\_id = author.id

and Book.title= “Harry Potter”.

Ví dụ: Với câu hỏi “Who published “Active Database Systems”

Các bộ ba được tìm thấy <Book,has\_title, “Active Database Systems”:title> và <Book,is written,Who:publisher>

Ta có câu truy vấn:

Select Book.publisher

From Book

Where

book.title = “Active Database Systems”

1. **Một số ví dụ**

Các ví dụ được giải quyết với thông tin database trên dblp.

Ví dụ 1: Câu hỏi “Who published “Active Database Systems”?

Bước 1: Tiền xử lý câu hỏi

Bước 1.1: Định dạng lại thực thể đặt tên

Đầu vào : Câu hỏi.

Đầu ra: Câu hỏi mà thực thể đặt tên đã được định dạng lại

Kết quả: “Who published NER001?

(Thông tin NER001 tương ứng với “Active Database Systems” đã được lưu vết lại.)

Bước 1.2: Xây dựng danh sách Pos Tagger

Đầu vào: “Who published NER001?

Đầu ra: Danh sách Pos Tagger:

Kết quả: WP/Who VBD/published NNS/“Active Database Systems”.

Bước 1.3: Tối ưu hóa Pos Taggers

Đầu vào: Danh sách Pos Tagger ở bước 1.2

Đầu ra: Danh sách Pos Tagger đã tối ưu

Kết quả: WP/Who VBZ/publish NNS/“Active Database Systems”.

Bước 2: Liệt kê các bộ 3 quan hệ về từ

Đầu vào: danh sách Pos Tagger bước 1.3

Đầu ra: Tập các bộ ba quan hệ về từ và từ loại tương ứng

Kết quả: <Who,publish, “Active Database Systems”>

Bước 3: Nhận diện thực thể

Bước 3.1: Nhận diện các thực thể dựa vào từ quan hệ

Đầu vào: Bộ 3 quan hệ từ ở bước 2

Đầu ra: Bộ ba quan hệ <thực thể, loại quan hệ, thực thể>.

Kết quả: <“Active Database Systems”:publication,be publish,publisher>

Bước 4: Nhận dạng quan hệ ẩn

Đầu vào: Kết quả bước 3.1

Đầu ra: Bộ 3 quan hệ ẩn

Kết quả: <publication, has title, “Active Database Systems”: title>

Bước 5: Sinh câu truy vấn  
Đầu vào: Danh sách các bộ 3 quan hệ

Đầu ra: Câu truy vấn SQL

Với ví dụ trên đầu vào:

<publication, has title, “Active Database Systems”: title>

Và < publication, is published, Who:publisher>

Đối tượng truy vấn :select dblp\_pub\_new.publisher

Nguồn truy vấn:From dblp\_pub\_new

Điều kiện: Where dblp\_pub\_new.title = “Active Database Systems”

Câu truy vấn:

Select dblp\_pub\_new.publisher

From dblp\_pub\_new

Where dblp\_pub\_new.title = “Active Database Systems”

Ví dụ 2: Câu hỏi “Who wrote “Active Database Systems”?

Bước 1: Tiền xử lý câu hỏi.

Bước 1.1: Định dạng lại thực thể đặt tên

Đầu vào : Câu hỏi.

Đầu ra: Câu hỏi mà thực thể đặt tên đã được định dạng lại

Kết quả: “Who published NER001?

(Thông tin NER001 tương ứng với “Active Database Systems” đã được lưu vết lại.)

Bước 1.2: Xây dựng danh sách Pos Tagger

Đầu vào: Kết quả bước 1.1

Đầu ra: Danh sách Pos Tagger

Kết quả: WP/Who VBD/wrote NNS/“Active Database Systems”.

Bước 1.3: Tối ưu hóa Pos Taggers

Đầu vào: Danh sách Pos Tagger ở bước 1.2

Đầu ra: Danh sách Pos Tagger đã tối ưu

Kết quả: WP/Who VBZ/write NNS/“Active Database Systems”.

Bước 2: Liệt kê các bộ 3 quan hệ về từ

Đầu vào: danh sách Pos Tagger bước 1.3

Đầu ra: Tập các bộ ba quan hệ về từ và từ loại tương ứng

Kết quả: <Who,write, “Active Database Systems”>

Bước 3: Nhận diện thực thể

Bước 3.1: Nhận diện tất cả các thực thể dựa vào từ quan hệ

Đầu vào: Bộ 3 quan hệ từ ở bước 2

Đầu ra: Bộ ba <thực thể, loại quan hệ, thực thể>.

Kết quả: <“Active Database Systems”:publication,write,Who:publication.author>

Bước 4: Nhận dạng quan hệ ẩn

Đầu vào: Kết quả bước 3

Đầu ra: Bộ 3 quan hệ ẩn

Kết quả: <publication, has title, “Active Database Systems”: title>

Bước 5: Sinh câu truy vấn  
Đầu vào: Danh sách các bộ 3 quan hệ

Đầu ra: Câu truy vấn SQL

Với ví dụ trên đầu vào:

<publication, has title, “Active Database Systems”: title>

Và < publication, is written, Who:author>

Đối tượng truy vấn :select distinct dblp\_author\_pub\_ref.author

(Vì author là thuộc tính có mapping\_table và không có related-table)

Nguồn truy vấn:From dblp\_pub\_new, dblp\_author\_pub\_ref

Điều kiện: Where publication.title = “Active Database Systems”

And dblp\_pub\_new.id = dblp\_author\_pub\_ref.pub\_id

Câu truy vấn:

Select select distinct dblp\_author\_pub\_ref.author

From dblp\_pub\_new, dblp\_author\_pub\_ref

Where publication.title = “Active Database Systems”

And dblp\_pub\_new.id = dblp\_author\_pub\_ref.pub\_id

1. http://opennlp.sourceforge.net/ [↑](#footnote-ref-2)
2. http://gate.ac.uk/gate/doc/plugins.html [↑](#footnote-ref-3)
3. http://sourceforge.net/projects/wn-similarity/ [↑](#footnote-ref-4)