1. **Xây dựng File miêu tả Thông Tin Database và các mối quan hệ giữa thực thể**

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

Hơn nữa, tên của các table 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, table chứa thông tin để trả lời câu hỏi vừa rồi là dblp\_pub\_new, đây là một từ hoàn toàn không có nghĩa bởi nó được đặt bởi người tạo ra database. Vì thế việc tận dụng các từ điển đồng nghĩa để nhận dạng trở nên khó khăn. Nhưng nếu ta đặt tên table *dblp\_pub\_new* là một tên 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 trong table và việc ánh xạ giữa các table nếu các table đó có quan hệ với nhau. Điều này sẽ giúp ích nhiều trong việc phát sinh câu truy vấn SQL tự động với một câu hỏi được đưa ra và các đối tượng trong câu hỏi đó những mối quan hệ ánh xạ với nhau trong database.

Chính vì thế, nhóm đề xuất một lượt 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 là dblp, với các thông tin sau:

Table: dblp\_author\_ref\_new

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu | Chú thích |
| pub\_id | Int | Khóa chính của dblp\_pub\_new |
| Author | varchar(50) | Tên tác giả |
| Editor | int(1) | Là true nếu tác giả đó là chủ bút |
| author\_num | int(3) | Số lượng tác giả |

Table: dblp\_pub\_new

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu | Chú thích |
| Id | Int | Khóa chính |
| dblp\_key | varchar(150) | dblp\_key của bài viết |
| Title | Longtext | Tiêu đề |
| Source |  | Nguồn bài viết |
| …. | …. | …. |
| Year |  | Năm xuất bản |
| Type |  | Loại bài viết |
| Publisher | varchar(150) | Tên nhà xuất bản |

Table: dblp\_ref\_new

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu | Chú thích |
| Id | int(8) | Khóa chính của dblp\_pub\_new |
| dblp\_key | varchar(150) | Dblp\_key của nhà xuất bản |

Từ thông tin trên, ta có thể xây dựng file XML sau:

<database>

<table table-alias="publication"> <!-- Tên do người dùng đặt thay thế cho dblp\_pub\_new -->

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

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

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

<relation>

<relation-name>has author,is written</relation-name> <!--Thiết lập các loại quan hệ chính sẽ xuất hiện trong câu hỏi -->

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

<mapping-table>pub\_author</mapping-table> <!-- Thông tin ánh xạ -->

</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>

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

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

<name>pub\_author</name>

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

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

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

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

</mapping-table>

</database>

1. **Các bước đề xuất**

1 .TIỀN XỬ LÍ CÂU HỎI

2.LIỆT KÊ CÁC BỘ BA QUAN HỆ VỀ TỪ

3.NHẬN DIỆN 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**

Theo ý kiến đề xuất của nhóm, trong câu, thực thể đặt tên là những từ được nhóm quy định bởi quy tắc sau:

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

Ví dụ: Mark Twain,John,…là những từ (cụm từ) dùng để xác định thực thể đặt tên.

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

Ví dụ: Who wrote “Active Database Systems”? Như vậy cụm “Active Database Systems” là một cụm từ dùng để xác định thực thể đặt tên.

Việc định dạng thực thể đặt tên sẽ giúp cho việc xử lí 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, ANNIE POS TAGGER … 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 cụm danh từ của thực thể đặt tên**.** 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 <’> đã nêu mà thay vào 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:

WP VBZ DT NN IN *NN* DT NNP NNP NNP IN DT JJ 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, nếu việc gán nhãn được cải thiện với kết quả :

WP VBZ DT NN IN “DT NNP NNP NNP IN DT JJ NNP” ? thì kết quả này cũng đã làm mất ý nghĩa là cụm danh từ của chuỗi ***“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 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 bằng công thức : NER+MÃ SỐ. Mã số 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:

* Đã tiền xử lý, đặt “***A Distributed Object Manager for the Smalltalk-80 System.***” là NER001 : *Who is the author of NER001 ?*

*WP VBZ DT NN IN* NNP ?

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

Việc xử lý Pos Tagger sẽ giúp ta dễ dàng xác định từ loại trong câu, từ đó dễ dàng ánh xạ với các bộ từ điển ngữ nghĩa như Wordnet. Hơn nữa nó còn giúp ta xác định các từ quan hệ và các từ mà sau này 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 NNS 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 khi xác định động từ trong câu thì các kết quả ở trên vẫn chưa hề chính xác về mặt ngữ nghĩa của nó.

Ví dụ: động từ “look up” là sự kết hợp giữa động từ “look” và giới từ “up”. Tuy nhiên, nếu chỉ dựa vào các bước xử lý ở trên để xác định động từ thì kết quả cho ra lại là “look”. Điều này không thật sự chính xác.

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 mẫu câu về dạng hiện tại đơn. Cụ thể nhóm sẽ liệt kê tất cả các mẫu câu và ánh xạ thành mẫu câu hiện tại đơn tương ứng. Đồng thời nhóm đề xuất dùng công cụ RitaWordnet để xác định động từ ở dạng nguyên mẫu. Sau đây là những mẫu câu và ánh xạ hiện tại đơn tương ứng:
* 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.

**2. Liệt kê các bộ 3 quan hệ về từ**

Ta cần liệt kê tất cả các bộ ba quan hệ <đối tượng 1, từ quan hệ, đối tượng 2>, điều này sẽ hỗ trợ việc tìm bộ ba <thực thể(tên lớp), loại quan hệ, thực thể (thuộc tính của lớp)> sau này.

Việc tìm bộ 3 sẽ dựa vào các cấu trúc sau (nhóm tự đề xuất) :

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)>

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 quan hệ về từ bằng cách chọn lọc tất cả các bộ ba có mối quan hệ giống với từ quan hệ (nếu là động từ sẽ dùng từ điển Wordnet để so sánh ở dạng đồng nghĩa và so sánh cả dạng chủ động và bị động của từ quan hệ đó), sẽ xuất hiện các trường hợp sau:

* Nếu tồn tại duy nhất một bộ 3:

+ Nếu từ quan hệ được tìm thấy ở mối quan hệ ngược (reversed-relation) nghĩa là loại thực thể của 2 đối tượng trong bộ ba quan hệ <đối tượng 1, từ quan hệ, đối tượng 2> sẽ có thứ tự ngược với thứ tự trong bộ ba quan hệ <thực thể(tên lớp), loại quan hệ, thực thể (thuộc tính của lớp)> tìm thấy.

Ví dụ: Who published “Active Database Systems”từ quan hệ sau các bước trên sẽ là publish, bộ 3 quan hệ <đối tượng 1, từ quan hệ, đối tượng 2> sẽ là <Who,publish, “Active Database Systems”>. Khi ánh xạ vào XML sẽ tìm thấy một bộ ba duy nhất trong quan hệ đảo là <Book,publish,publisher>. Như vậy “Active Database Systems” được nhận diện là Book chứ không phải là publisher

+ Nếu từ quan hệ không được tìm thấy ở mối quan hệ đảo thì bộ ba<thực thể(tên lớp), loại quan hệ, thực thể (thuộc tính của lớp)> sẽ có thứ tự giống với bộ ba quan hệ <đối tượng 1, từ quan hệ, đối tượng 2>

+Đi đến bước 4.

* Nếu tồn tại nhiều kết quả, thì đi qua bước 3.3 nhưng tập miền các bộ 3 <thực thể(tên lớp), loại quan hệ, thực thể (thuộc tính của lớp)> cần kiểm tra của bước này sẽ là miền kiểm tra ở các bước sau.
* Nếu không tồn tại kết quả nào, vẫn tiếp tục qua bước 3.3 nhưng miền các bộ 3 <thực thể(tên lớp), loại quan hệ, thực thể (thuộc tính của lớp)> cần kiểm tra của các bước sau chính là tòan bộ các bộ ba <thực thể(tên lớp), loại quan hệ, thực thể (thuộc tính của lớp)> có trong XML.

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

**3.2.1. Phương pháp nhận diện thực thể đặt tên**

Nếu đứng trước hoặc đứng sau thực thể đặt tên là một thực thể không đặt tên mà không có từ quan hệ, thì 2 thực thể có cùng chung một phân lớp. Như vậy ta chỉ cần xác định thực thể chưa đặt tên.

Ví dụ: Who write book “Active Database Systems”? Với câu hỏi này thì book và “Active Database Systems.” có cùng phân lớp.

Nếu đứng trước thực thể đặt tên không có thực thể chưa đặt tên nào,

* Ta chỉ cần ánh xạ dữ liệu của thực thể đặt tên đó vào database. Việc này sẽ được ưu tiên cho các thuộc tính đại diện của table (Như mô hình XML ở trên, thuộc tính đại diện của Author sẽ là Name) rồi mới đến các thuộc tính khác. Nếu thực thể đó là 1 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)... Nếu thực thể đặt tên là thuộc tính đại diện của table đó thì phân lớp của thực thể đặt tên chính là table, ngược lại phân lớp của thực thể đặt tên chính là thuộc tính.

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**. Hơn nữa **title** là thuộc tính đại diện của table **Publication**. Như vậy phân lớp tương ứng của cụm từ *“*Active Database Systems*”* là **Publication**.

**3.2.2. Phương pháp nhận diện thực thể chưa đặt tên**

Việc nhận dạng thực thể chưa đặt tên sẽ giúp ta tìm ra được loại thực thể mà Database đã tồn tại gần hoặc cùng nghĩa nhưng khác từ. Ví dụ:

Từ “writer” sẽ đồng nghĩa với từ “author” . Nếu như không nhận dạng thì ta sẽ không trả lời được những câu hỏi liên quan đến writer.

Nhóm đề xuất sử dụng công cụ Similarity Wordnet để nhận dạng thực thể chưa đặt tên. Similarity Wordnet là một công cụ dùng để đánh giá độ tương đồng về nghĩa giữa 2 danh từ.Nếu giữa 2 danh từ cùng nghĩa thì mức đánh giá giữa 2 danh từ đó là 1,mức thấp nhất là 0 tức là 2 danh từ đó không có liên quan gì về nghĩa với nhau. Như vậy, với cách này đầu tiên ta chỉ cần tìm các table mà tên của nó có độ tương đồng về nghĩa với thực thể không đặt tên trong mức [0.9,1].Nếu xuất hiện thì thực thể không đặt tên đó là phân lớp là table có độ tương đồng về nghĩa cao nhất.Ngược lại thì ta tìm theo tên của tất cả các thuộc tính của các lớp, nếu tìm được thì phân lớp của nó là thuộc tính tìm thấy có đó tương đồng về nghĩa cao nhất. của danh ta có thể chọn lớp thực thể trong Database có độ tương đồng về nghĩa lớn nhất để làm lớp cho thực thể chưa đặt tên.

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 Database đã có chứa một lớp có tên là Author.

Bằng công cụ Similarity Wordnet, ta có thể chọn lớp cho từ *writer* là *author* thay vì là *Book* hay bất kỳ tên table nào khác trong Database.

**3.2.3. Phương pháp nhận diện lớp của câu hỏi:**

- Đối với câu hỏi What:

+ Với dạng câu hỏi *What do/does +Noun + mean?* Thì lớp câu hỏi chính là lớp của danh từ.

+ Với dạng câu hỏi *What is/are +Noun+ stand/short for?* Thì lớp câu hỏi là lớp có độ tương đồng về nghĩa lớn nhất với từ “name”.

+ Với dạng câu hỏi : *What is/are + Noun?* thì lớp câu hỏi chính là lớp của danh từ .

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

+ Với những trường hợp còn lại, thì việc xác định lớp của What phải dựa vào lớp của danh từ có quan hệ với What và từ quan hệ giữa chúng và tiến hành sao khớp 2 thông tin đã biết đó vào Ontology để tìm lớp của What.

-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)

Ngược lại thì lớp của câu hỏi chính là thuộc tính của danh từ. Bằng cách dùng từ điển Wordnet, ta có thể tìm ra danh từ tương ứng của tính từ (Adj), đồng thời kết hợp với công cụ similarity WordNet để tìm tên thuộc tính có độ tương đồng về nghĩa với danh từ tương ứng vừa tìm được lớn nhất.

**3.3. Nhận diện thực thể đối tượng**

Thực thể đối tượng theo nhóm đề ra là thực thể đứng đằng sau trong bộ ba quan hệ về từ <đối tượng 1, từ quan hệ, đối tượng 2>. Ngược với nó là thực thể chủ thể.

Bằng các phương pháp nhận diện thực thể đã nêu ở trên, ta có thể nhận diện được thực thể đó là thuộc tính hay table nào.

Ví dụ: Who published “Active Database Systems”. Với ví dụ này, thực thể đối tượng là thực thể “Active Database Systems”. Vì đây là thực thể đặt tên nên ta sẽ dùng phương pháp nhận diện thực thể đặt tên ở trên và tìm ra được phân lớp của thực thể trên là table *Book*.

**3.4. Nhận diện table hay thuộc tính của thực thể**

Nhóm đề xuất cách nhận biết table hay thuộc tính của một thực thể như sau:

* Trong bộ ba quan hệ <đối tượng 1, từ quan hệ, đối tượng 2> , nếu giữa hai thực thể có từ quan hệ là “*has*” thì thực thể đầu là tên table, thực thể sau là thuộc tính.

Ví dụ: What is author of “Active Database Systems”? có bộ 3 quan hệ từ sau <“Active Database Systems”,has, “author”>,Thực thể author là thuộc tính của thực thể “Active Database Systems”.

Ví dụ: Which book has publisher ACM?có bộ 3 quan hệ từ <book,has,publisher>. Như vậy “book” là table và “publisher” là thuộc tính của “book”.

* Nếu thực thể đối tượng là table thì thực thể chủ thể là thuộc tính của table đó .Ví dụ: Who published “Active Database Systems” ? Có bộ 3 quan hệ từ <Who,publish, Active Database Systems >

Thực thể sau được nhận diện là table Book. Như vậy thực thể đầu là thuộc tính của table đó. Ở đây là publisher, từ quan hệ pubished được chuyển thành “is published”.

+ Thực thể đứng sau là thuộc tính thì thực thể đầu chính là table.

Ví dụ: which books has publisher ACM ?

Thực thể thứ 2 được xác định là thuộc tính, vậy thực thể đầu chính là table.

**3.5. Nhận diện thực thể chủ thể**

Thực thể chủ thể được xác định với vị trí ngược với thực thể đối tượng. Thực thể chủ thể sẽ dễ dàng nhận diện hơn nếu ta biết được thực thể đối tượng vì nó đã giúp ta loại bỏ khá nhiều các tập miền tìm kiếm.

Nếu thực thể đối tượng là một thuộc tính thì ta dễ dàng nhận diện thực thể chủ thể chính là lớp chứa thuộc tính đó.

Ví dụ: which books has publisher ACM ?

Ở đây publisher là thuộc tính của table *publication*. Suy ra phân lớp của thực thể chủ thể là *publication*.

Nếu thực thể đối tượng là tên lớp, thì thực thể chủ thể chính là một trong các thuộc tính của lớp đó. Bằng các phương pháp xác định thực thể đã nêu, ta có thể nhận diện được thực thể chủ thể.

**4. Xác định quan hệ ẩn**

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 table thì thực thể thứ 2 là thuộc tính đại diện cho table đó.

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 lớp thì sẽ xuất hiện một quan hệ ẩn giữa lớp và thuộc tính đại diện cho lớp đó.

Ví dụ: Who write “Active Database Systems” . Ở đây, “Active Database Systems” được xác định là lớp 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 table và tìm ra các mối quan hệ ẩn. Tiếp theo từ các bộ ba <thực thể(tên table),tên quan hệ, thực thể (tên thuộc tính), 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 table:

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.

1. 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