Outline

1. Phân lớp câu hỏi1
2. Đặc Điểm phân lớp2

2.1Subordinate word category (Hạ bậc từ mục)2

2.2 Question Focus (Tập trung vào câu hỏi )4

2.3 Syntactic-semantic structure (cấu trúc cú pháp ngữ nghĩa)5

1. Biểu thức thông dụng (regular expression)6
2. Tổng quan bài toán phân lớp câu hỏi11
3. Kết luận12

Hình ảnh12

Tham khảo12

**1.Phân lớp câu hỏi**

*QA system (Question & Answer system): là một* hệ thống đóng vai trò phổ biến trong việc tìm kiếm thông tin chính xác và hiệu quả .Nó đưa ra câu trả lời đầy đủ và chính xác ứng với yêu cầu của người dùng và câu trả lời được thể hiện bằng ngôn ngữ tự nhiên. Người dùng nhanh chóng lấy được thông tin cần thiết thay vì tìm kiếm nó trong một khối lượng lớn các văn bản. [1]

Trong hệ thống QA bao gồm nhiều thành phần : thành phần phân tích câu hỏi , thành phần phân loại câu hỏi , thành phần truy vấn dựa trên những tập tài liệu liên quan đến câu truy vấn , …

Như trong các tài liệu tham khảo đã đề cập tới , việc phân lớp câu hỏi đóng vai tròn quan trọng trong một hệ thống QA [1-4] . Trong việc trả lời câu hỏi , nó loại bỏ đi những câu trả lời không liên quan hay phù hợp với câu hỏi . Ngoài ra , việc phân loại nhằm giúp xác định kiểu trả lời cho phù hợp với từng loại câu hỏi.

Ví dụ : “Who is the author of Harry Potter ? “ là một loại câu hỏi về con người (tác giả) . Câu trả lời mặc nhiên phải là một danh từ riêng liên quan đến một người nào đó (tức tên tác giả của cuốn Harry Potter) “J. K. Rowling is …”

Hiện có nhiều hệ thống QA system trên thế giới như :

+ NSIR : một hệ thống hỏi đáp phát triển từ CLAIR – một nhóm thuộc đại học Michigan . Hệ thống câu hỏi mẫu được lấy từ TREC . (<http://tangra.si.umich.edu/clair/NSIR/html/html/about.html>)

+AnswerBus : là một hệ thống QA system cho nhiều lĩnh vực , hỗ trợ những câu hỏi cho nhiều thứ tiếng như Anh, Đức, Pháp , Tây Ban Nha,… câu trả lời là những đường link đến câu trả lời chứ không đưa ra một đáp áp cụ thể . (<http://www.answerbus.com/about/index.shtml>)

**2.Các phương pháp phân lớp**

Theo như bài báo [2] nói tới , việc tiếp cận phân lớp câu hỏi có thể phân biệt qua ba nhóm chính : dựa trên những nguyên tắc (rule-based) , mô hình hóa ngôn ngữ (language modeling) và dựa trên máy học ( machine learning based)

Đối với các tiếp cận dựa trên những nguyên tắc , những luật văn phạm viết tay và một tập các biểu thức thông dụng được tạo ra phục vụ cho việc phân tích cú pháp và quyết định kiểu câu trả lời . Nhưng nó có một số hạn chế như sau :

+ Sự phân lớp dựa trên luật viết tay khó và tốn nhiều thời gian xử lý

+ Có sự giới hạn về mức độ bao quát và phức tạp trong việc mở rộng phạm vi của các loại câu trả lời .

+Khi có một nguyên tắc mới ,nhiều luật trước đó phải được chỉnh sửa lại hoặc viết lại hoàn toàn.

**3.Đặc điểm phân lớp (Classifying Features)**

Theo các tài liệu tham khảo [1][4] thì ta có một số các đặc điểm sau hỗ trợ cho việc phân lớp câu hỏi .

**+Bag-of-word:** Trong phương pháp này mỗi phần tử từ vựng trong câu hỏi được coi là một đặc điểm phân lớp . Với phương pháp này mối quan hệ giữa các từ trong câu sẽ bị bỏ sót .

**+ WordNet Synsets:**

**+ N-gram:**

**+ Dependency Structure:**

**+ NER ( Named Entity Recognize):**

Ngoài ra , bài báo còn đề xuất ra 3 đặc điểm mới là : Subordinate word category , Question Focus và Syntactic-Semantic Structure.

**3.1 Subordinate word category (Hạ bậc từ mục)**

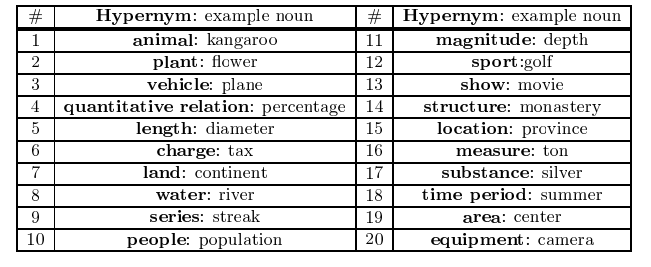
Mục đích của việc hạ bậc từ mục là đưa ra những ý nghĩa khái quát hơn từ những danh từ rút ra trong câu hỏi . Một antology là WordNet được sử dụng dựa trên mối quan hệ khái quát (hypernym) , từ đó xác định ý nghĩa chung của một danh từ rút ra trong câu hỏi của người dùng .

Nhiệm vụ của antology Wordnet là lấy ra được nghĩa khái quát (hypernym) của mỗi danh từ lấy ra từ câu hỏi . Các nghĩa khái quát đó nằm ở một mức độ trừu tượng khác nhau , từ cụ thể cho đến cái chung nhất (được liệt kê ở bảng dưới ).

Các bước được thực hiện như sau :

* Rút ra những danh từ có trong câu hỏi
* Nếu rút được , tìm một tập những từ khoái quát từ mỗi danh từ .
* Giữ thứ tự của tập này theo thứ tự mức độ trừu tượng , chọn ra từ khái quát nào có thứ tự cao nhất trong danh sách .
* Xem từ khái quát tìm được là một đặc điểm của câu hỏi (Trong trường hợp làm giàu dữ liệu thì thêm nó vào nơi lưu trữ (feature space)

Bảng thứ tự trừu tượng được liệt kê ở bảng bên dưới bao gồm 20 mục sắp xếp theo mức độ đã được đánh số .

Hình 1 – Bảng 20 hypernym trong WordNet sắp xếp theo thứ tự trừu tượng

**3.2 Question Focus (Tập trung vào câu hỏi )**

Mục đích của đặc điểm này là rút ra được từ trọng tâm của câu hỏi . Từ này làm rõ ràng cho câu hỏi và nhấn mạnh kiểu câu trả lời mong đợi .

Ví dụ : What state has the most Indians ? => từ trọng tâm của câu hỏi này là “state” . Do cấu trúc thông thường của câu hỏi what thì danh từ đứng sau chữ what thường nắm phần lớn thông tin của câu hỏi . Đồng thời câu trả lời được gợi ý dựa trên từ trọng tâm sẽ là “the state is ….”

Dựa trên những biểu thức thông dụng ( regular expression) , các câu hỏi lập đi lập lại theo một cách hỏi được biểu diễn bởi các khuôn mẫu (Pattern) . Theo khuôn mẫu đó , từ trọng tâm sẽ được xác định . Ứng với từng khuôn mẫu khác nhau , từ trọng tâm sẽ nằm ở những vị trí khác nhau trong câu . Khi có được từ trọng tâm, ta có thể biết được câu hỏi đó thuộc phân lớp nào dựa vào việc dùng một số ontology để nhận biết (WordNet).

Các biểu thức được tạo ra dựa trên kiến thức thông thường và phân tích từ một tập các câu hỏi. Tập này có thể lấy từ thư viện của TREC (như trong tài liệu là TREC-8, TREC-9, TREC-10, TREC-2002, TREC-2003 và hiện tại đã có TREC-2007).

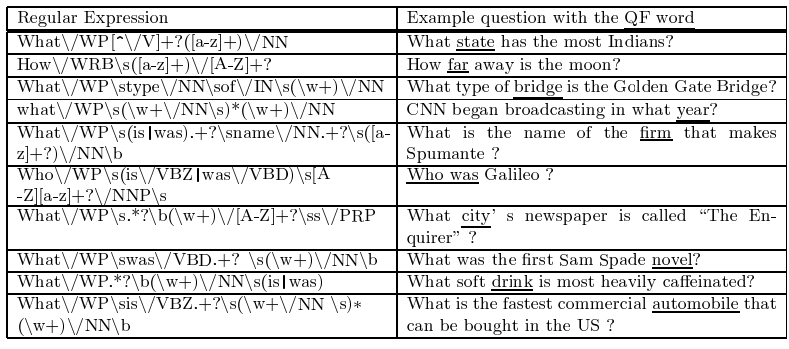
Các bước thực hiện được đề ra như sau :

+Từ một danh sách các biễu thức thông dụng , tìm một biểu thức thích hợp với câu hỏi đưa ra ( Lưu ý câu hỏi này đã được gán nhãn từ loại –POS tagger ). Nếu không tìm thấy một biểu thức nào phù hợp thì đặc điểm này không thể áp dụng .

+ Dựa vào biểu thức tìm được , rút ra từ trọng tâm của câu hỏi .

+ Xem từ mới lấy được là một đặc điểm để phân lớp câu hỏi ( trong trường hợp làm giàu dữ liệu thì đưa từ đó vào không gian lưu trữ ).

Dưới đây là một số biểu thức mà bài báo trên đã đưa ra [2] :



Hình 2 – Một số mẫu thông dụng để nhận biết từ trọng tâm .

Giải thích một số từ trong bảng Regular Expression trên .Các nhãn bên dưới là các nhãn từ loại được gán cho các từ trong câu hỏi (Quy tắc đặt tên này đựa trên tập nhãn của PannTreeBank <http://www.mozart-oz.org/mogul/doc/lager/brill-tagger/penn.html> ) :

* WP : wh-pronoun : bao gồm các từ như ‘what’, ‘who’, and ‘whom’.
* NN : noun : danh từ nói chung
* WRB : wh-adverb bao gồm ‘how’,’where’,’why’,’when’.
* IN - preposition or subordinating : giới từ
* VBZ, VBD : các loại động từ .
* PRP : unknown-, but probably possessive pronoun : đại từ

Đó là những từ viết tắt được định nghĩa . Để tìm hiểu thêm có thể tham khảo tại phần tham khảo [7].

**3.3 Syntactic-semantic structure (cấu trúc cú pháp ngữ nghĩa)**

Đặc điểm này dựa trên cấu trúc ngữ nghĩa của câu hỏi . Các câu hỏi có cùng cấu trúc như thế thông thường sẽ có cùng một mục đích để hỏi , tức là cùng hỏi về một vấn đề : định nghĩa , nơi chốn , thời gian ….

Một ví vụ về câu hỏi “What is/are <noun>” . Thông thường kết quả trả về là một định nghĩa . Một câu khác như “where is the <noun>” thì tất nhiên câu hỏi đang đặt vấn đề về nơi chốn . Với đặc điểm này ta có thể nhanh chóng kết luận được câu hỏi đang đề cập đến vấn đề gì đựa trên những cấu trúc thông dụng .

**4. Biểu thức thông dụng (regular expression)**

Biểu thức thông dụng –regular expression (viết tắt là regexp hay regex)- hay còn được gọi là khuôn mẫu (Pattern) dùng để biểu diễn một loạt các ký tự dựa theo những nguyên tắc luận lý .Nó được tạo ra dựa trên các đặc tính thông dụng của mỗi chuỗi nằm trong tập văn bản.[9]

Việc tạo ra các biểu thức nhằm để so sánh, hay nói đúng hơn là so khớp nó với một đoạn văn bản hoặc một chuỗi nào đó . Công việc mà regex sẽ làm là tìm kiếm xem vị trí của chuỗi nào trong văn bản phù hợp với điều kiện mà biểu thức đề ra .

VD : một chuỗi “car” có thể xuất hiện trong các trường hợp sau “car”,”cartoon”,”bicarbonate”. Đây chỉ là một ví dụ đơn giản , nó có thể làm được nhiều hơn thế .

Việc tạo ra các biểu thức giúp ích rất nhiều trong việc tìm kiếm một câu, một chữ trong văn bản , hoặc thực hiện thao tác như đếm tầng số xuất hiện chúng .Nhưng để tạo ra những biểu thức này cần nắm được những cú pháp về regular expression.

VD :

* một biểu thức “A\*B” thì sẽ so khớp với các chuỗi như “AB”,”AAB”,”AAAB” hoặc bất kỳ chuỗi nào bắt đầu với A và kết thúc với B .
* Biểu thức “(dog|cat)” so khớp với các chuỗi “dog” và “cat”.

***4.1 Các cú pháp cơ bản trong regular expression:*** [8]

|  |  |  |
| --- | --- | --- |
| Cú pháp | Ý nghĩa | Ví dụ |
| ***.*** | Bất kỳ một ký tự nào | "do." -> "dog", "dot", "doe” ,... "d..r" -> "door","deer",… |
| \* | Không hoặc có bất kỳ ký tự nào trước đó | "do.\*" -> "dog", "done", "doppleganger",…  "to\*" -> "to" ,"too" |
| + | Một hoặc bất kỳ ký tự nào trước đó | "fre+.." -> "freak", "freeze", "fresh" |
| ? | Không có hoặc có một ký tự trước đó | "ton?e" -> "toe" ,"tone" |
| () | Nhóm lại | (dog|cat) -> "dog" ,"cat" |
| [] | Bất kì ký tự nào trong tập | "ta[pb]" -> "tap" , "tab"  "r[aeiou]t" -> "rat", "ret", "rot", "rut" |
| [^] | Không chứ bất kì ký tự nào trong tập | "t[^aeiou]+.\*s" -> "thanks", "this", "trappings", ... |
| {min,max} | Phạm vi xuất hiện | [a-z]{3} : ba ký tự thường liên tiếp  [0-9]{3} : ba chữ số liên tiếp  [A-Z]{2,5} : các chữ cái in hoa xuấ hiện liên tiếp ở thời điểm 2,3,4,5 |

Ngoài các cú pháp cơ bản trên , regular expression còn có các lớp ký tự được gọi là Character Class mà ta rất thường sử dụng .

|  |  |
| --- | --- |
| Character Classes | Ý nghĩa |
| \d | Bất kì ký tự nào thuộc 0-9 |
| \D | Bất kì ký tự nào không thuộc 0-9 , ngược lại với \d |
| \w | Bất kì ký tự nào cả chữ lẫn số [a-zA-Z0-9] |
| \W | Ngược lại với \w |
| \s | Khoảng trắng |
| \S | Ngược lại với \s |

Vậy ta có thể thấy rằng :

* \D tương đương [^\d]
* \W tương đương [^\w]
* \S tương đương [^\s]

Ngoài ra còn có một số ký hiệu khác gọi là Anchor Sequences (tạm dịch là “đánh dấu trình tự” )

|  |  |
| --- | --- |
| Anchor Sequences | Ý nghĩa |
| ^ | Bắt đầu của một chuỗi ký tự |
| $ | Kết thúc chuỗi ký tự |
| \b | Biên (giới) của từ |
| \B | Ngược lại , bất kỳ đâu trừ biên của từ |

***4.2 Regular Expression trong Java :***

Trong java hỗ trợ java.util.regex API để thao tác với regular expression. Regular expression được sử dụng hầu hết trong nhiều ngôn ngữ như Perl, Tcl, Python,PHP,…Nhưng trong API của Java , thư viện java.util.regex hầu hết giống như trong ngôn ngữ Perl [9]

Trong gói thư viện java.util.regex cung cấp 3 class để xây dựng một regular expression hay pattern:

* **Pattern** : đối tượng thể hiện cho một biểu thức . không có hàm khởi tạo mặc định mà phải thông qua một phương thức *compile* trả về một đối tượng Pattern . Tham số truyền vào sẽ là một biểu thức regular expression
* **Matcher** :là đối tượng đóng vai trò thực hiện những hoạt động giữa chuỗi nhập vào với Pattern **.** Matcher cũng không có phương thức khởi tạo mà thông qua phương thức *matcher* của đối tượng Pattern.
* **PatternSyntaxException** : là một đối tượng để bắt lấy những lỗi trong cú pháp của regular expression.

Ví dụ :

|  |
| --- |
| public class SampleRegex  {  public static void main(String[] params)  {  Pattern pattern = Pattern.compile("(.\*):(.\*)");  Matcher matcher = pattern.matcher(params[0]);  if(matcher.matches())  {  System.out.print("Key:");  System.out.println(matcher.group(1));  System.out.print("Value:");  System.out.println(matcher.group(2));  }  else  System.out.print("No match");  }  } |

*java SampleRegex "domain: developer.com"*

Kết quả in ra khi ra chạy dòng lệnh trên là :

*Key: domain*

*Value: developer.com*

***4.3 Áp dụng regular expression để tạo các pattern cho câu hỏi :***

Với những lý thuyết về regular expression (regex) như trên thì việc tạo một pattern cho một câu hỏi là việc khá dễ dàng . Với một loại câu hỏi lặp đi lặp lại một cấu trúc thì ta có thể áp dụng một biểu thức chung cho những câu hỏi như thế .

Các biểu thức thông dụng được nêu ra ở hình 2 áp dụng cho một số loại câu hỏi đã được gán nhãn từ (POS tagger).Ta có thể xem xét những biểu thức regex đó và phân tích một số biểu mẫu :

**Ví dụ 1 :**

“How\/WRB\s([a-z]+)\/[A-Z]+?” tương ứng khớp với câu “How far away is the moon?”

Phân tích biểu mẫu trên :

* How : là một loại từ thuộc WRB [7].
* \s : tương ứng là một khoảng trắng/
* [a-z] : là ký tự trong khoảng từ a tới z . ký tự thường không viết hoa.
* ([a-z]+) : là một chữ bắt đầu với một ký tự thường.
* +? : bất kỳ chuỗi nào đứng sau.

Vậy regex của câu trên là “How\s([a-z]+)+?”

Đồng thời có dạng : “WRB [A-Z]+“ có nghĩa rằng sau một từ để hỏi (WRB : How ) tồn tại một chữ bất kỳ : How many …,How far … , How much …

**Ví dụ 2 :**

“What\/WP\stype\/NN\sof\/IN\s(\w+)\/NN” sẽ so khớp với câu “What type of bridge is the Golden Gate Beridge”

Phân tích biểu mẫu trên :

* What : một từ thuộc WP [7]
* \s : khoảng trắng
* type, of : các từ xuất hiện trong câu. Type được xem là một NN , of là IN tham khảo phụ lục [7].

Như thế tương ứng với ví dụ 1 . Ta cũng có regex của câu trên là :

“What\stype\sof\s(\w+)”

Có dạng : “WP NN IN NN” có nghĩa rằng sau một từ để hỏi (WP : What) ta có một danh từ (NN) tiếp đến là một giới từ (IN : of ) và tiếp theo đó là một danh từ .ví dụ : What kinds of film ... , what type of book … , What types of aircraft …

Từ các vị dụ trên ta có thể có được một bảng như sau từ hình 2 :

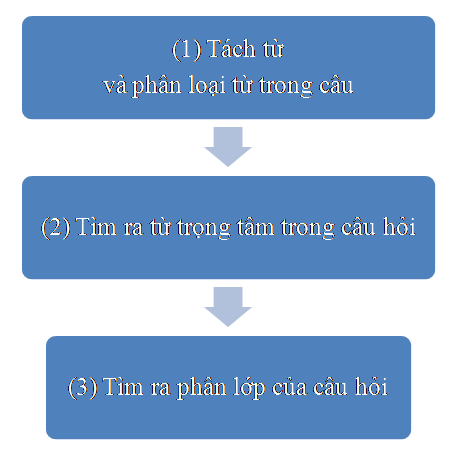
|  |  |  |
| --- | --- | --- |
| Regex | Câu mẫu | Dạng |
| What+?([a-z]+) | What state has the most Indians? | WP NN |
| How \s([a-z]+)+? | How far away the moon? | WRB [A-Z] |
| What\stype\sof\s(\w+) | What type of bridge is the Golden Gate Bridge? | WP NN IN NN |
| What \s(\w+\s)\*(\w+) | CNN began broadcasting in what year ? | WP NN (NN) |
| What\s(is|was).+?\sname.+?\s([a-z]+?)\b | What is the name of the firm that makes Spumante? | WP (is|was) NN NN |
| Who\s(is|was)\s[A-Z][a-z]+?\s | Who was Galileo? | WP (VBZ|VBD) NNP |
| What\s.\*?\b(\w+)+?\ss | What city’s newspaper is called “The Enquirer” | WP [A-Z] PRP |
| What\s was.+?\s(\w+)\b | What was the first Sam Spade novel ? | WP VBD NN |
| What.\*?\b(\w+)\s(is|was) | What soft drink is most heavily caffeinated ? | WP NN (is|was) |
| What\sis.+?\s(\w+\s)\*(\w+)\b | What is the fastest commercial automobile that can be bought in the US ? | WP VBZ NN (NN) |

**5. Tổng quan về bài toán phân lớp câu hỏi**

**Dữ liệu đầu vào :** Một câu hỏi bất kỳ (dựa trên tập câu hỏi TREC 2007).

**Dữ liệu đầu ra :** phân lớp của câu hỏi đó ( cụ thể là 25 phân lớp lexicographer của Wordnet)

Từ việc xác định dữ liệu đầu vào và đầu ra cho bài toán, ta có những bước thực hiện sau :



* 1. Từ câu hỏi đã cho tiến hành tách từ : Tách từ dựa trên khoảng trắng giữa các từ , loại bỏ các stopword (lưu ý các stopword như What,How , Where , … am,is,are,was … cần được giữ lại ).
  2. Phân loại các từ tách được theo (hay còn gọi là gán nhãn từ loại ) : động từ ,danh từ , tính từ , …..

Ở bước này ta sử dụng GATE API [11]– Một chương trình mã nguồn mở phục vụ cho việc rút trích văn bản . Trong Gate có một plugin là POS tagger dùng để gán nhãn cho từ loại .Ta sẽ sử dụng API của GATE để chạy plugin trên và gán nhãn cho các từ lấy được từ bước a .

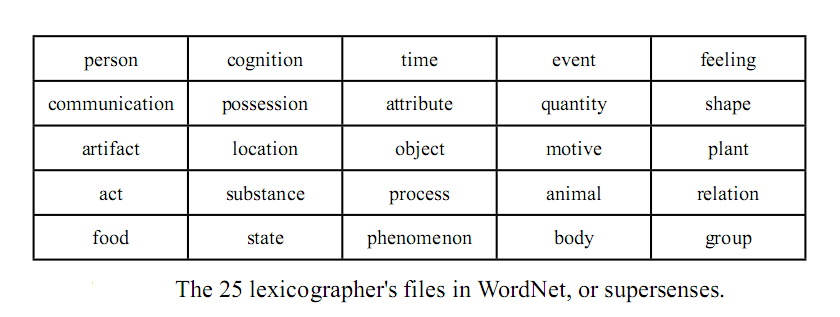
1. Tìm từ trọng tâm trong câu hỏi : dựa trên một loạt các template câu hỏi được định nghĩa trước . ( các template dựa trên việc tạo ra các Regular Expression ở mục 3)

kết hợp với phân từ loại ở bước 1 tìm ra được biểu thức regex phù hợp với câu hỏi đưa ra .

Sau đó , nếu tìm được ta sẽ rút ra được từ trọng tâm của câu hỏi .

Ngược lại , không xử lý tiếp do kho biểu thức không hỗ trợ dạng câu hỏi này hoặc tạo ra một giao diện cho người dùng cập nhật một biểu thức mới và xác định từ trọng tâm trong biểu thức mới .

1. Sau khi có được từ trọng tâm . Dùng wordnet để xác định từ trọng tâm thuộc một trong 25 lexicographer nào . Phân lớp của câu hỏi sẽ thuộc một trong 25 lexicographer này.



Hình 3 – 25 phân lớp dựa vào tập lexicographer của WordNet .

**6. Kết luận :**

Việc phân loại câu hỏi là một phần trong hệ thống QA system góp phần trong việc xác định câu trả lời và lọc các câu trả lời sau khi truy vấn .Để từ đó , ta có một câu trả lời chính xác hơn thay vì đưa ra một loạt các câu trả lời tìm kiếm được để người dùng lựa chọn. Còn việc đưa ra câu truy vấn để rút ra câu hỏi từ các tập dữ liệu (web hay corpus) như thế nào thì không nằm trong phạm vi của của phân lớp câu hỏi.

**Hình ảnh :**

Hình 1 : lấy từ Effectiveness of Combined features for machine learning based question classification (Marcin Skowron and Kenji Araki ) -2005 page 9

Hình 2 : lấy từ Effectiveness of Combined features for machine learning based question classification (Marcin Skowron and Kenji Araki ) -2005 page 10

Hình 3 : lấy từ Minimally supervised question classification and answering base on wordnet and wikipedia page 5.

**Tài liệu tham khảo:**

[1] Question Classification by Ensemble Learning (LiXin , Huang Xuan Jing ,Wu Lid-de) - 2006

[2] Effectiveness of Combined features for machine learning based question classification (Marcin Skowron and Kenji Araki ) -2005

[3] Question Classification in Social Media – 2009

[4] Minimally supervised question classification and answering base on wordnet and wikipedia\_2009

[5] Learning Surface Text Patterns for a Question Answering System 2002 (Deepak Ravichandran and Eduard Hovy)

[6] YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia

[7] <http://gate.ac.uk/releases/gate-5.1-build3431-ALL/doc/tao/splitap7.html#x32-609000G>

[8] <http://www.wdvl.com/Authoring/Languages/Perl/PerlfortheWeb/perlintro2_table1.html>

[9] <http://java.sun.com/docs/books/tutorial/essential/regex/intro.html>

[10] Xử Lý Ngôn Ngữ Tự Nhiên – 2006 , Đinh Điền . Trang 329

[11] Trang chủ của GATE <http://gate.ac.uk/>