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 thông tin trong một khối lượng lớn các văn bản.

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

Trong các thành phần nêu trên , thành phần phân lớp câu hỏi đóng vai trò quan trọng trong một hệ thống QA [1-4]. Người dùng đưa vào hệ thống một câu hỏi, câu hỏi sau đó sẽ được phân tích để tìm từ khóa tìm kiếm hoặc định dạng thành một loại câu truy vấn xuống cơ sở dữ liệu chứa thông tin. Kết quả trả về của quá trình đó không thể trả về ngay lập tức một câu trả lời duy nhất và chính xác. Trên thực tế nó sẽ đưa ra một tập các câu trả lời đề cử có liên quan đến vấn đề được hỏi. Vậy, công việc chính của phân lớp câu hỏi là lọc và loại bỏ những câu trả lời nào không phù hợp, hơn thế nữa có thể dựa trên một số tiêu chí để đánh giá xếp hạng các câu trả lời theo mức độ liên quan đến câu hỏi.

Ngoài ra, phân lớp câu hỏi còn có thể giúp ta xác định được kiểu câu trả lời một khi ta biết được phân lớp của nó. Việc này giúp hệ thống đưa ra một câu trả lời gần gũi với ngôn ngữ tự nhiên.

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 …”

Thông thường các câu hỏi được phân lớp theo 6 mục thô và 50 mục mịn (Bảng 1) . Các mục này được dưa ra bởi Li và Roth (2002). Các bài báo nghiên cứu sau này về phân lớp câu hỏi cũng đựa trên các mục này để nghiên cứu và đánh giá [1,2,12,14].

|  |  |
| --- | --- |
| Mục thô | Mục mịn |
| ABBR | abbreviation,expansion |
| DESC | definition, description, manner, reason |
| ENTY | animal, body, color, creation, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, subtance, symbol, technique, term, vehicle, word |
| HUM | description, group, individual, title |
| LOC | city, country, mountain, other, state |
| NUM | Code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight |

Bảng 1 – Phân lớp 6 mục thô và 50 mục mịn

Hiện có một số hệ thống QA system được biết tới 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>)

1. **Các phương pháp phân lớp**

Việc tiếp cận phân lớp câu hỏi có thể phân biệt qua ba nhóm chính : dựa trên luật (rule-based) , mô hình hóa ngôn ngữ (language modeling) và dựa trên máy học ( machine learning based). Đó là ba hướng tiếp cận chính hiện nay [2]

*Đối với các tiếp cận dựa trên luật :* việc phân lớp câu hỏi dựa vào một số các luật bằng tay. Các luật này có được là do đề xuất từ các chuyên gia. Đối với cách tiếp cận này , một loạt các biểu thức thông dụng (regular expression) được tạo ra để so khớp với câu hỏi từ đó quyết định phân lớp của câu hỏi và loại câu trả lời . Nhưng nó có một số hạn chế như sau :

+ Sự phân lớp dựa trên các luật viết gặp nhiều khó khăn và tốn nhiều thời gian xử lý . Do dựa trên kiến thức chủ quan của con người trên một tập dữ liệu câu hỏi .

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

*Trong cách tiếp cận máy học :* kiến thức chuyên môn được thay thế bằng một tập lớn các câu hỏi được gán nhãn kết quả sẵn (một tập dữ liệu mẫu ) . Từ tập này , classifier (bộ phân lớp) sẽ được học một cách có kiểm soát hoặc không kiểm soát trên tập mẫu. Các thuật toán máy học thường dùng bao gồm : Mạng nơ-ron (Neural NetWork), tính xác suất Naïve Bayes, Maximum Entropy , cây quyết định (decision Tree) , lân cận (Nearest-Neighbors), Sparse Network of Winnows(SNoW), Support Vector machine(SVM) ... Với cách tiếp cận máy học , ta dễ dàng thấy nó sẽ giải quyết được nhiều hạn chế từ cách tiếp cận dựa trên những luật . Những thuận lợi trong cách tiếp cận này gồm :

+ Thời gian tạo dựng ngắn , không tốn thời gian để đề ra các luật.s

+ Bộ phân lớp được tạo ra tự động thông qua việc học từ một tập dữ liệu huấn luyện ; việc cung cấp các luật giờ không cần thiết nữa .

+ Mở rộng độ bao phủ : bằng cách thu được từ các ví dụ huấn luyện .

+ Nếu có nhu cầu , bộ phân loại có thể tái cấu trúc lại (học lại) một cách linh hoạt để phù hợp với quy luật mới.

Hiện tại , bộ phân lớp thường được sử dụng trong phương pháp máy học là Support Vector Machine . Dựa trên kiểm thử và đánh giá với một số phương pháp máy học khác Zhang và Lee [12] cho thấy Support Vector Machine có phần vượt trội hơn so với các phương pháp máy học khác.

***Nearest-Neighbors***

Ý tưởng của thuật toán này là dựa trên mức độ giống nhau nhất giữa một trường hợp chưa được gán nhãn phân lớp với các thực thể trong tập dữ liệu học ( đã được gán nhãn ) . Tức là một câu hỏi cần Phân lớp sẽ được đem ra so sánh với các thực thể trong tập huấn luyện dựa trên các đặt trưng . Câu hỏi đó có độ tương tự gần giống với thực thể nào nhất sẽ lấy nhãn của thực thể đó . Điều này , đơn giản chỉ là so sánh sự trùng lặp các đặc trưng giữa hai câu hỏi .

***Naive Bayes***

Là mô hình tiếp cận cho việc Phân lớp dựa trên định lý sác xuất Bayes . Mô hình này giả định rằng các đặc trưng tồn tại độc lập trong một phân lớp của câu hỏi , vì thế sác xuất của chúng cũng độc lập . Mô hình sẽ tính sác xuất của các đặc trưng trong câu hỏi và xem tỉ lệ sác xuất nào cao nhất để phân lớp câu hỏi đó. Công thức tính sác xuất có điều kiện của Bayes như sau:

P(A/B) = P(B/A) x P(A) / P (B)

Xác xuất của biến cố A khi tồn tại một biến cố B . Tức , với một đặc trưng B và phân lớp A, ta tính sác xuất của *P(A) trên tập dữ liệu huấn luyện = số trường hợp là A / tổng số các trường hợp* . Tiếp theo tính xác xuất của đặc trưng B trên phân lớp A : *P(B/A) = số đặc trưng B xuất hiện trong A / Số trường hợp là A* .Xác xuất của P(B) là 1. ( Tham khảo thêm ví dụ tại <http://www.statsoft.com/textbook/naive-bayes-classifier/> )

***Decision Tree***

Cây quyết định sẽ mô tả một cấu trúc cây , trong đó , các lá đại diện cho các Phân lớp còn cành đại diện cho các thuộc tính kết hợp dẫn đến Phân lớp đó . Các dữ liệu đưa vào học có dạng :

(x,y) = (x1,x2,x3...xk,y)

Trong đó y là giá trị mà chúng ta hướng tới để Phân lớp . Tập các biến x1,x2,x3...xk là các thuộc tính mà ta dựa vào để thực hiện việc Phân lớp .

Vd : Một tập dữ liệu mẫu về thói quen đi chơi cuối tuần :

|  |  |  |  |
| --- | --- | --- | --- |
| Set of Atributes (x1,x2,x3) | | | Variable y |
| Weekend | (Example)WeatherParentsMoney | Decision | (Category) |
| Sunny | Yes | Rich | Cinema |
| Sunny | No | Rich | Tennis |
| Windy | Yes | Rich | Cinema |
| Rainy | Yes | Poor | Cinema |

Sau khi có một tập dữ liệu mẫu , một số thuật toán được áp dụng để xây dựng cây quyết định như ID3 của Quinlan (ý tưởng chung là chọn thuộc tính sau cho nó phân các thực thể thành các tập con (các nhánh) sao cho có nhiều thực thể cùng loại nhất ) . (tham khảo thêm tại <http://www.doc.ic.ac.uk/~sgc/teaching/v231/lecture11.html> )

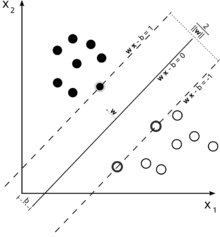
***Support Vector Machine***

Phương pháp này dựa trên ý tưởng từ một tập huấn luyện cho trước, dùng một mặt phẳng để chia các điểm trong không gian thành hai lớp riêng biệt. Trong đó các điểm là các đặc trưng mà ta rút ra được trong quá trình huấn luyện.

Khoảng cách giữa mặt phẳng tới điểm dữ liệu gần nhất của một phân lớp sẽ quyết định chất lượng của mặt phẳng . Khoảng cách này gọi là biên.

Nếu chúng ta có một tập dữ liệu :

\mathcal{D} = \{ (\mathbf{x}_i, c_i)|\mathbf{x}_i \in \mathbb{R}^p, c_i \in \{-1,1\}\}_{i=1}^n

[](http://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png)Là một tập các vector xi trong đó ci sẽ có giá trị là -1 hoặc 1. Chúng ta sẽ có được một phương trình mặt phẳng :

\mathbf{w}\cdot\mathbf{x} - b=0,\,

Vector w là một vector pháp tuyến: nó vuông góc với siêu phẳng. Biến xác định độ lệch của siêu phẳng từ gốc theo vector pháp tuyến.

Ta có thêm hai mặt phẳng tại điểm biên :

\mathbf{w}\cdot\mathbf{x} - b=1\, và \mathbf{w}\cdot\mathbf{x} - b=-1.\,

Bằng phương pháp hình học, người ta đã tính được khoảng cách giữa hai mặt phẳng này là . Vì thế, để cực đại khoảng cách biên thì phải phải cực tiểu hóa ||w||.

Chúng ta có ràng buộc sau :

\mathbf{w}\cdot\mathbf{x}_i - b \ge 1\qquad\text{ for }\mathbf{x}_i  thuộc về phân lớp thứ 1

\mathbf{w}\cdot\mathbf{x}_i - b \le -1\qquad\text{ for }\mathbf{x}_i  thuộc về phân lớp thứ 2

Có thể viết lại : c_i(\mathbf{w}\cdot\mathbf{x}_i - b) \ge 1,  \quad \text{ for all }  1 \le i \le n.\qquad\qquad(1)

Cuối cùng bài bài toán đưa về việc cực tiểu hóa *w* và tìm *b* sao cho thỏa điều kiện (1) .

1. **Đặc trưng phân lớp**

Trong phương pháp tiếp cận máy học, từ tập dữ liệu có sẵn ta rút ra những đặc trưng phân lớp đề từ đó đưa ra huấn luyện. Các đặc trưng này đơn giản chỉ là một hoặc nhiều từ nằm đâu đó trong câu hỏi. Chúng không quyết định câu hỏi đó thuộc về phân lớp nào, chỉ là cơ sở để qua qua trình học dự đoán một câu hỏi thuộc về một phân lớp.

Trong phân lớp câu hỏi, các đặc trưng này đều là các đặc trưng nhị phân. Điều này có nghĩa là giá trị của một đặc trưng là 1hoặc -1: 1 tức là đặc trưng đó thuộc về phân lớp đó và ngược lại là -1. Vì thế, các đặc trưng khi rút được ra từ một câu hỏi để huấn luyện đều có giá trị nhị phân là 1.

Có nhiều loại đặc trưng cho bài toán phân lớp câu hỏi, có một số đặc trưng thông dụng và thường được sử dụng. Một số khác do một số tác giả đề xuất, sau đó đánh giá và kiểm thử để biết được mức độ chính xác của bộ phân lớp khi áp dụng đặc trưng đó. Các đặc trưng đó sẽ được trình bày rõ hơn ở phần bên dưới.

* 1. ***Bag of words***

BOW là đặc trưng thường được sử dụng nhất. Vì nó khá đơn giản và dễ hiểu. Các từ vựng trong một câu hỏi đều được coi như là đặc trưng phân lớp. Một câu hỏi được coi như là một túi chứa các đặc trưng.

Thông thường các stopwords như : what, is ... có ít vai trò trong việc tìm kiếm dữ liệu vì chủ yếu việc này dựa trên các key-word. Nhưng đối với phân lớp câu hỏi thì các stop words đóng vai trò quan trọng và cần được giữ lại. Điều này có lẽ quá rõ ràng vì đối với một câu hỏi thì các stop words như *What, Who, When, Where...* góp phần lớp trong việc quyết định phân lớp.(danh sách các stop words có thể tham khảo tại phục lục A ).

* 1. ***N-gram***

Câu hỏi được coi như là một danh sách các từ liên tiếp nhau. Dựa trên ý tưởng như thế, ta có 3 loại n-gram thường được sử dụng là unigram, bigram, trigram.

+unigiram : tương tự bag of words.

+bigram : lấy lần lượt 2 từ liên tiếp nhau trong câu.

+trigram : lấy lần lượt 3 từ liên tiếp nhau trong câu.

* 1. **Tận dụng tính năng ngữ nghĩa của Wordnet**

Wordnet là một kho từ điển ngữ nghĩa của tiếng Anh. Các danh từ, động từ, tính từ được nhóm thành các các nhóm từ đồng nghĩa gọi là synset. Ngoài ra, còn kể đến cấu trúc từ bao hàm hypernym của Wordnet, một từ có liên quan đến một từ khác mang một nghĩa bao hàm rộng hơn.

Ví dụ cho một cấu trúc hypernym :

*dog, domestic dog, Canis familiaris*

*=> canine, canid*

*=> carnivore*

*=> placental, placental mammal, eutherian, eutherian mammal*

*=> mammal*

*=> vertebrate, craniate*

*=> chordate*

*=> animal, animate being, beast, brute, creature, fauna*

*=> ...*

Như ví dụ trên thì {dog, domestic dog, Canis familiaris} là một tập synset. Hypernym của *dog* là *canine.*Hypernym của *canine* là *carnivore.* Cấu trúc cứ thế đi từ nghĩa cụ thể đến nghĩa khái quát hơn :

*=> canine => carnivore => ... => animal => ...*

Do cách thức tổ chức từ vựng của Wordnet mà nó được sử dụng nhiều trong xử lý ngôn ngữ tự nhiên. Hypernym là một tính năng mà được sử dụng nhiều trong việc chọn làm đặc trưng phân lớp . Cách tận dụng hypernym của wordnet sẽ được trình bày ở mục e.

* 1. **Các đặc trưng khác**

# Phụ lục

# A. Danh sách các stop words

Bảng danh sách các stopword trong tiếng anh :

(lấy từ trang <http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/> )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | List of stop words | | | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 | a  about  above  across  after  afterwards  again  against  all  almost  alone  along  already  also  although  always  am  among  amongst  amoungst  amount  an  and  another  any  anyhow  anyone  anything  anyway  anywhere  are  around  as  at  back  be  became  because  become  becomes  becoming  been  before  beforehand  behind  being  below  beside  besides  between  beyond  bill  both  bottom  but  by  call  can  cannot  cant  co  computer  con  could  couldnt  cry  de  describe  detail  do  done  down  due  during  each  eg  eight  either  eleven  else | elsewhere  empty  enough  etc  even  ever  every  everyone  everything  everywhere  except  few  fifteen  fify  fill  find  fire  first  five  for  former  formerly  forty  found  four  from  front  full  further  get  give  go  had  has  hasnt  have  he  hence  her  here  hereafter  hereby  herein  hereupon  hers  herse”  him  himse”  his  how  however  hundred  i  ie  if  in  inc  indeed  interest  into  is  it  its  itse”  keep  last  latter  latterly  least  less  ltd  made  many  may  me  meanwhile  might  mill  mine  more | moreover  most  mostly  move  much  must  my  myse”  name  namely  neither  never  nevertheless  next  nine  no  nobody  none  noone  nor  not  nothing  now  nowhere  of  off  often  on  once  one  only  onto  or  other  others  otherwise  our  ours  ourselves  out  over  own  part  per  perhaps  please  put  rather  re  same  see  seem  seemed  seeming  seems  serious  several  she  should  show  side  since  sincere  six  sixty  so  some  somehow  someone  something  sometime  sometimes  somewhere  still  such  system  take  ten  than  that | the  their  them  themselves  then  thence  there  thereafter  thereby  therefore  therein  thereupon  these  they  thick  thin  third  this  those  though  three  through  throughout  thru  thus  to  together  too  top  toward  towards  twelve  twenty  two  un  under  until  up  upon  us  very  via  was  we  well  were  what  whatever  when  whence  whenever  where  whereafter  whereas  whereby  wherein  whereupon  wherever  whether  which  while  whither  who  whoever  whole  whom  whose  why  will  with  within  without  would  yet  you  your  yours  yourself  yourselves |

# B. Biểu thức thông dụng (regular expression)

Theo định nghĩa của Java: regular expression là một cách để miêu tả một tập các chuỗi dựa trên những đặc điễm của mỗi chuỗi trong tập. Nó được sử dụng để tìm kiếm, chỉnh sửa và thao tác trên văn bản và dữ liệu. (<http://java.sun.com/docs/books/tutorial/essential/regex/intro.html> )

Theo định nghĩa của Wikipedia: regular expression là một chuỗi được dùng để miêu tả hoặc so khớp với một tập các chuỗi khác dựa trên những luật cú pháp. (<http://en.wikipedia.org/wiki/Regular_expression_examples> ).

Theo ngôn ngữ Perl : regular expression là những cú pháp , thực thi trong Perl và các môi trường khác , việc tạo ra nó thì khó nhưng dễ dàng trong các thao tác : so sánh chuỗi , chọn lựa chuỗi ,thay thế chuỗi và phân tích cú pháp . (<http://www.troubleshooters.com/codecorn/littperl/perlreg.htm#WhatTheyAre>)

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*

**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**](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**](http://www.wdvl.com/Authoring/Languages/Perl/PerlfortheWeb/perlintro2_table1.html)

**[9]** [**http://java.sun.com/docs/books/tutorial/essential/regex/intro.html**](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/**](http://gate.ac.uk/)

**[12] Question Classification using Support Vector Machine – 2003**

**[13] Learning Question Classifier – Li and Roth 2002**

**[14] Question classification using Head word and their Hypernyms -2008**