THIẾT KẾ VÀ XÂY DỰNG MÁY TÌM KIẾM NGỮ NGHĨA ĐỂ HỖ TRỢ CHO HỆ THỐNG HỎI ĐÁP THÔNG MINH TBT LONG AN

■ ThS. NGUYỄN MINH ĐỆ (*)

TÓM TẮT

Thiết kế và phát triển hệ thống tìm kiếm ngữ nghĩa cho hệ thống hỏi đáp thông minh là một trong các công việc thiết yếu và cần phải thực hiện liên tục việc cải tiến. Trong bài báo này thực hiện việc phân tích và đề nghị một thiết kế từ tổng thể đến chi tiết cho hệ thống tìm kiếm nói trên. hệ thống tìm kiếm ngữ nghĩa ở đây được áp dụng chuyên biệt cho hệ thống hoi đáp thông minh, là một máy tìm kiếm ngữ nghĩa. Kiến trúc nền tảng của máy tìm kiếm ngữ nghĩa được thiết kế chuyên biệt và có các thành phần chính bên trong, gồm có 3 phần: a) Phần phân lớp cho câu hoi sẽ dựa trên cách tiếp cận theo hướng máy học (hướng tiếp cận này đều phù hợp với các hệ thống nho đến các hệ thống lớn), cu thể áp dụng thuật toán Support Vector Machines (SVM). b) Phần xây dựng cơ sở dữ liệu tri thức (ngữ nghĩa) sẽ được thực hiện song song với việc mô tả các tài nguyên thông tin có ngữ nghĩa (Ontology): c) Tìm kiếm trên mạng ngữ nghĩa.

***Từ khóa**. Tìm kiếm ngữ nghĩa; Máy tìm kiếm; Phân lớp câu hỏi, Máy học; Support Vector Machines (SVM); Cơ sở dữ liệu tri thức (ngữ nghĩa); Tài nguyên thông tin có ngũ nghĩa (Ontology).*

SUMMARY

Design and develope Semantic Search System for Smart Answer-Question System[1] is one of essential tasks and have to perform continuously improvements. In this article, we performed analysises and proposed a design from the overall to the details for this Search System. Semantic Search System is applied specially to Smart Answer-Question System[1], is Semantic Search Engine. The fundamental archuecture of Semantic Search Engine was specially designed and has the following main components, consists of 3 components: a) Question Classification will be based Machine Learning method (this approach is suitable with all systems from small to large), applying SVM Algorithm; b) The construction of the knowledge database (Ontology) will performed in parallel with the description of semantic information resources (Ontology); c) Searching on the semantic network

***Key words**. Semantic Search; Search Engine; Question Classification; Machine Learning; Support Vector Machines (SVM); Knowledge Database; Ontology.*

1. Mở đầu

Đề tài "*Nghiên cứu và xây dựng Hệ hỏi đáp thông minh cho thông tin về Hàng rào Kỹ thuật trong Thương mại (TBT) của tỉnh Long An*" là đề tài được tổ chức bởi Trường Đại học Kinh tế Công nghiệp Long An, thuộc lĩnh vực Kỹ thuật và Công nghệ, có 3 mục tiêu:

- Mục tiêu 1: Xây dựng cổng thông tin điện tử TBT tỉnh Long An quản lý trực tuyến và tập trung các thông tin về hàng rào kỹ thuật trong thương mại tỉnh Long An (gọi tắt là cổng thông tin TBT Long An).
- Mục tiêu 2: Thiết kế và xây dựng cơ sở dữ liệu TBT Long An
- Mục tiêu 3: Nghiên cứu và xây dựng công cụ hỏi đáp thông minh TBT Long An.

Các công cụ hỏi đáp ở Mục tiêu 3 được chia ra làm các thành phần nhỏ hơn và được cấu tạo từ các thành phần nhỏ hơn đó mà có tính chất rời rạc. Các thành phần rời rạc này có mối quan hệ hữu cơ

(*) Trường ĐH KTCN Long An

với nhau và có thể thiết kế và phát triển riêng biệt. Một trong các thành phần quan trọng là cần phải xây dựng hệ thống tìm kiếm (tìm kiếm theo ngữ nghĩa). Hệ tìm kiếm ngữ nghĩa này là một trong các nhiệm vụ quan trọng của Mục tiêu 3 và có thể phát triển qua các phiên bản khác nhau.

Một hệ thống tìm kiếm ngữ nghĩa thường được xây dựng dựa trên một miền và ngôn ngữ cụ thể. Cấu trúc tổng quát bên trong của hệ thống tìm kiếm ngữ nghĩa thường được tạo thành từ 2 thành phần chính: Phân lớp câu hỏi; Cơ sở dữ liệu tri thức (ngữ nghĩa).

Để xây dựng hệ thống tìm kiếm ngữ nghĩa ở đây thì cần phải thực hiện được 3 công việc chính:

- **Công việc 1: Phân tích và thiết kế cấu trúc dữ liệu** để chuẩn bị dữ liệu cho việc xây dựng Cơ sở dữ liệu tri thức (mạng ngữ nghĩa). Dữ liệu thô ban đầu cần phải xử lý và tổ chức lại một cách có hệ thống để trở thành dữ liệu vào và có thể sử dụng được. Trong bài báo có trình bày thiết kế cây Taxonomy và cấu trúc Ontology cho Cơ sở dữ liệu tri thức.
- **Công việc 2: Xây dựng kiến trúc cơ bản của một máy tìm kiếm** để làm cơ sở cho việc thiết kế kiến trúc chung cho chương trình chuyên dụng của Máy tìm kiếm ngữ nghĩa.
- **Công việc 3: Thiết kế thuật toán cho việc phân lớp câu hỏi** thực hiện việc áp dụng thuật toán *SVM (Support Vector Machines)* vào phân lớp câu hỏi, đây là thành phần quan trọng của Máy tìm kiếm ngữ nghĩa.

Phần còn lại bài báo như sau: Phần 2, Xây dựng mạng ngữ nghĩa để trình bày nội dung của Công việc 1; Phần 3, Thiết kế kiến trúc mạng ngữ nghĩa để thực hiện nội dung của Công việc 2; Phần 4, Phân lớp câu truy vấn để thực hiện việc triển khai nội dung Công việc 3; Phần 5, Kết qua, đánh giá và kết luận.

2. Xây dựng Mạng dữ liệu ngữ nghĩa

2.1 Phân tích dữ liệu đầu vào

Xét một bảng theo khung HS, bảng có cấu trúc theo danh mục phân loại như sau:

Bảng 1: Danh mục bảng phân loại HS

Chỉ mục	Ký hiệu	Tiếng Việt	Tiếng Anh
0	01	Động vật sống	Animals; Live
1	01.01	Ngựa, lừa, la sống.	Horses, asses, mules and hinnies; live
2	01.02	Động vật sống họ trâu bò.	Bovine animals; live
...	....	....	...
Tổng số dòng -1	97.06.00.00	Đồ cổ có tuổi trên 100 năm.	Antiques of an age exceeding one hundred years.

Cấu trúc bảng phân loại ICS (và một số khung/bảng phân loại khác) cũng có cấu trúc tương tự như bảng phân loại HS ở trên, nên những phân tích và thiết kế đều sẽ được áp dụng tương tự với nhau.

Cấu trúc của chỉ số phân loại HS được trình bày cụ thể trong [1]. Xét một mã HS cụ thể 1001.11.00, thì có: chỉ số quốc tế là 1001.11; chỉ số riêng của quốc gia là 00. Mã HS quốc tế gồm 6 chữ số. Hai chữ số đầu tiên chỉ định Chương HS. Hai chữ số tiếp thứ hai chỉ định Nhóm HS. Hai chữ số thứ

ba chỉ phân nhóm HS. Xét chỉ số HS quốc tế 1001.11 thì: chương 10 (Ngũ cốc); nhóm 01 (Lúa mì và meslin); phân nhóm 11 (Lúa mì Durum).

Tóm lại, với mã HS 1001.11.00 thì có nghĩa là: Thuộc phần II, các Sản phẩm thực vật; Chương 10, Ngũ cốc; nhóm 01, Lúa mì và meslin; Phân nhóm 11, Lúa mì Durum; Phân nhóm phụ riêng quốc gia 00, hạt giống.

Như vậy, cấu trúc của tanh mục bảng phân loại HS được gom nhóm lại theo các phần như hình sau.

Bảng 2: Bảng phân loại HS được phân thành các Phần

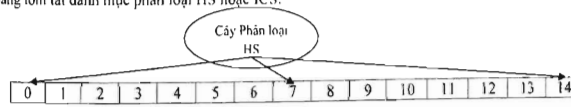
Chỉ mục	Phần	Khoảng dãy ký hiệu	Tên tiếng Việt	Tên tiếng Anh
0	I	01-05	Động vật và các sản phẩm từ động vật	Animal & Animal Products
1	II	06-15	Các sản phẩm thực vật	Vegetable Products
2	III	16-24	Các sản phẩm thực phẩm	Foodstuffs
3	IV	25-27	Các sản phẩm khoáng sản	Mineral Products
4	...	....	....	...
14	XV	90-97	Các sản phẩm còn lại khác	Miscellaneous

Xét một dữ liệu văn bản TBT (Đối tượng thông báo 1) trong Đối tượng thông báo 1 cần phải định nghĩa lại sao cho con người và chương trình máy tính làm việc với nhau hiệu quả hơn. Dữ liệu được định nghĩa lại ngoài việc chứa thông tin (văn bản, hình ảnh, ...) mà còn phải có chứa các liên kết. Các liên kết này chứa nhiều loại liên kết khác nhau như: Đến tài nguyên khác; Nhiều loại quan hệ được định nghĩa thêm; ... Các đặc điểm này sẽ làm cho dữ liệu có chứa thông tin nội dung được đa dạng hơn, chi tiết hơn và đầy đủ hơn. Các thông tin trong dữ liệu nhờ vào các mối liên kết mà quan hệ chặt chẽ với nhau. Sự chặt chẽ này hỗ trợ cho việc tìm kiếm thông tin mạnh mẽ và hiệu quả hơn.

2.2 Thiết kế cấu trúc dữ liệu

Xây dựng cây cấu trúc Taxonomy[1]

Danh mục bảng phân loại HS sẽ được xây dựng thành một cây Phân loại HS. Cây này có một nút sẽ nắm giữ một danh sách T có chứa các nút đỉnh. Danh sách T sẽ có 15 phần tử tương ứng như Bảng tóm tắt danh mục phân loại HS hoặc ICS.



Hình 1: Cây Phân loại HS

Như vậy, biểu thức truy cập phần tử của Cây Phân loại HS là: T[i] với i = 0 ... 14. Mỗi một T[i] chứa một nút đỉnh Top có cấu trúc dữ liệu như sau và sẽ có một giá trị cụ thể:

Ký hiệu
Tiếng Việt
Tiếng Anh
Giá trị đánh giá
Danh sách L, có chứa các nút Nội dung

Nút đinh

01-05
Động vật và các sản phẩm từ động vật
Animal & Animal Products
5
L[i] với i = 0 .. 4

Nút đinh đầu tiên T[0]

Mỗi một L[i] chứa một nút Nội dung có cấu trúc dữ liệu và có một giá trị cụ thể:

Giá trị đánh giá
Vị trí
Danh sách C, có chứa các nút con là Nút nội dung

Nút nội dung

5
0
C[]

Nút nội dung đầu tiên của một nút đinh T[i]

Mỗi một C[i] nằm trong một nút Nội dung thì có thể chứa các nút Nội dung khác và mang ý nghĩa là con của nó:

Giá trị đánh giá
Vị trí
Danh sách C, có chứa các nút con là Nút nội dung

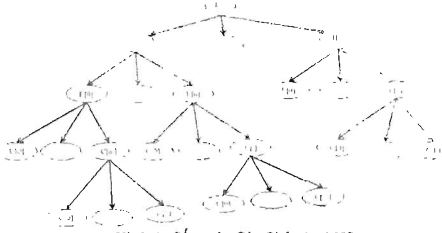
Nút nội dung cha

7
1
C[]

Nút nội dung con

Hình 2: Tập hợp cấu trúc và tính giá trị các nút

Tóm lại, hình ảnh của cấu trúc Cây Phân loại HS sẽ như sau:



Hình 1: Cấu trúc Cây Phân loại HS

Xây dựng Ontology cho mạng ngữ nghĩa

Ontology: có nhiều định nghĩa về Ontology, ở đây sử dụng định nghĩa như ở dưới đây.

Một Ontology: Là một mô hình dữ liệu biểu diễn một lĩnh vực và được sử dụng để suy luận về các đối tượng trong lĩnh vực đó và mối quan hệ giữa chúng; Cung cấp bộ từ vựng các thuộc tính, ràng buộc. Một Ontology mô tả:

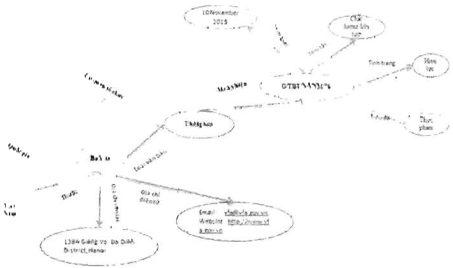
- Các cá thể, Individuals: Các đối tượng cơ bản, nền tảng
- Các lớp, Classes: Các tập hợp, hay kiểu của các đối tượng

- Các thuộc tính, Properties: Thuộc tính, tính năng, đặc điểm, tính cách, hay các thông số mà các đối tượng có và có thể đem ra chia sẻ.
- Các mối liên hệ, Relations: Các con đường (liên kết) mà các đối tượng có thể liên hệ tới một đối tượng khác.

Bộ từ vựng Ontology được xây dựng trên cơ sở tầng của RDF và RDFS [10], cung cấp khả năng biểu diễn ngữ nghĩa và có khả năng hỗ trợ lập luận.

Thiết kế cấu trúc dữ liệu (Ontology):

Xét Đối tượng thông báo 1, nếu xem Mã ký hiệu là một định danh thì có thể xây dựng được cấu trúc sau:



Hình trên mô tả về 1 dữ liệu TBT có ngữ nghĩa, và chứa thông tin của một văn bản TBT của "*Sữa và các sản phẩm sữa chế biến*" do Bộ Y tế của nước Việt Nam ban hành. Dữ liệu có cấu trúc như một đồ thị có hướng mang trọng số, mỗi đỉnh trong đồ thị mô tả thông tin hoặc chính dữ liệu ngữ nghĩa khác. Các cạnh của đồ thị thể hiện một kiểu liên kết (thuộc tính của dữ liệu).

Mỗi tài nguyên (dữ liệu ngữ nghĩa) trong mạng ngữ nghĩa là một đối tượng. Các đối tượng đều có: Tên gọi; Thuộc tính; Giá trị của thuộc tính; Mối liên kết;.... Trước tiên cần phải xây dựng từng đơn vị dữ liệu ngữ nghĩa (đối tượng), sau đó xây dựng mạng liên kết lại các đối tượng với nhau (đối tượng có thể lồng vào nhau được), gọi là mạng ngữ nghĩa. Mạng này sẽ được chia sẻ rộng khắp cho các hệ thống khác sử dụng lại, nên cần phải xây dựng với quy cách thống nhất. Ontology sẽ được sử dụng để mô tả dữ liệu (đối tượng/tài nguyên mạng) cho mạng ngữ nghĩa.

Cấu trúc chung cho một Ontology dữ liệu như sau:

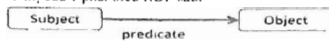
- **Lớp (classes):** Văn bản; Quốc gia; Cơ quan/tổ chức; ....
- **Cá thể (individuals):** Văn bản G/TBT/N/VNM/78; Quốc gia Việt Nam; cơ quan (Bộ Y tế);....
- **Thuộc tính (Attributes):** một thuộc tính thuộc Ontology có 2 phần: Tên thuộc tính; Giá trị tương ứng.

  Ví dụ cá thể có tên là Văn bản G/TBT/N/VNM/78 có các thuộc tính: Mã số (G/TBT/N/VNM/78); Hiệu lực (có); Thời gian (10 November 2015); Tiêu đề (Thực phẩm); ...
- **Quan hệ (Relation):** một quan hệ được hình thành khi một giá trị của một thuộc tính nào đó nằm ở trong là một cá thể khác. Có nhiều mối quan hệ: Xếp gộp (subsumption); Xem là một cây phân cấp; Lớp cha (is_superclass_of); Là (is_a); Lớp con (is_subclass_of), ...

RDF (Resource Description Framework): là mô hình được W3C đề xuất là mở rộng của công nghệ XML. RDF sử dụng một mô hình trừu tượng để phân rã thông tin thành những phần con, bao gồm các phần chính sau:

- Statement (phát biểu hay mệnh đề)
- Các nguồn tài nguyên subject (chủ ngữ) và object (tân ngữ, bổ ngữ)
- Predicate (vị ngữ)

Xét 1 cấu trúc đồ thị của 1 phát biểu RDF sau:

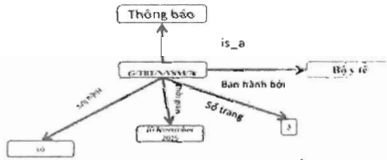


Hình 3: Cấu trúc 1 phát biểu chuẩn RDF

Statement (phát biểu, mệnh đề).

- Luật 1: Kiến thức (hoặc thông tin) được diễn giải là 1 danh sách các statement; Mỗi statement có dạng Subject-Predicate_Object, và thứ tự này không bao giờ được thay đổi (cố định).
- Luật 2: Tên của 1 tài nguyên phải có tính toàn cầu/cục và được nhận diện bởi Uniform Resource Identifier (URI) "ở đây sẽ xây dựng một bộ quy tắc để định nghĩa lại".

Ta có một ví dụ như sau về tài liệu có mã là G/TBT/N/VNM/78



Hình 4: Ví dụ dữ liệu theo cấu trúc RDF

Xây dựng được một bảng sau:

Bảng 3: Bảng thông tin văn bản G/TBT/N/VNM/78 theo RDF

Nút đầu	Cạnh	Nút cuối
G/TBT/N/VNM/78	is_a	Thông báo
G/TBT/N/VNM/78	Ban hành bởi	Bộ Y tế
G/TBT/N/VNM/78	Số trang	"3"
G/TBT/N/VNM/78	Thời gian	"10 November 2015"
G/TBT/N/VNM/78	Hiệu lực	"Có"

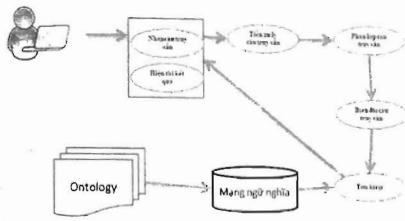
Văn bản G/TBT/N/VNM/78 có các mối quan hệ: Là một thông báo; Được ban hành bởi Bộ Y tế; Có số trang là "3"; Có hiệu lực là "có"; Có thời gian phát hành là "10 November 2015".

3. Thiết kế kiến trúc của máy tìm kiếm ngữ nghĩa

3.1. Kiến trúc chung

Máy tìm kiếm ngữ nghĩa cho hệ hỏi đáp thông minh TBT Long An có 2 thành phần chính: Giao diện người dùng có 2 chức năng chính (giao diện truy vấn, hiển thị kết quả); Kiến trúc bên trong (phần lõi của máy tìm kiếm) có 3 phần chính (phân tích câu hỏi, tìm kiếm kết quả, xử lý dữ liệu trên mạng ngữ nghĩa).

Máy tìm kiếm có cấu trúc chính:



Hình 5: Máy tìm kiếm ngữ nghĩa

3.2 Giao diện người dùng

Giao diện người dùng GUI được thiết kế tương tự như máy tìm kiếm thông thường, và đạt được các yêu cầu chung như: Đơn giản; Dễ nhìn; Dễ dùng. Ngoài ra máy tìm kiếm ngữ nghĩa sẽ cần có thêm các chức năng:

- Lựa chọn tìm kiếm theo miền, lĩnh vực
- Phải có từ gợi ý, ví dụ khi người dùng nhập vào "quy" thì thông thường gợi ý đơn giản là "hoạch", hoặc có các từ về (thời gian, nơi chốn, cái gì, ….).

3.3. Kiến trúc bên trong

Như đã trình bày ở mục trên, các thành phần của kiến trúc bên trong gồm có 4 phần: Tiền xử lý câu hỏi; Phân lớp câu hỏi; Biến đổi dạng câu hỏi; Mạng ngữ nghĩa.

Các bước để xử lý câu truy vấn của người dùng nhập vào như sau:

B1: Người dùng nhập câu truy vấn Q ở giao diện chức năng.

B2: Câu Q sẽ được một module Tiền xử lý để phân tích về ngôn ngữ: Tiếng Việt (có dấu và không dấu); Sai lỗi chính tả; Các thuật ngữ chuyên môn;… Kết quả sẽ cho ra một câu Q' đã được làm mịn lại để dễ dàng cho Bộ phân tích câu hỏi thực hiện ở bước kế tiếp.

B3: Câu Q' được đưa vào bộ phân tích để xác định xem Q' thuộc miền nào và lĩnh vực nào trong miền đó.

B4: Câu Q' được biến đổi về dạng chuẩn (biểu thức logic để tiến hành tìm kiếm).

B5: Q' được tìm kiếm trên mạng ngữ nghĩa của máy tìm kiếm.

B6: Hiển thị kết quả đưa ra.

Máy tìm kiếm hỗ trợ cho Hệ hỏi đáp thông minh TBT Long An sẽ xử lý câu hỏi của người dùng qua 6 bước chính như trên. Tùy theo module Tiền xử lý mà máy tìm kiếm sẽ có các phiên bản khác nhau. Kế tiếp là chi tiết của 5 phần của kiến trúc bên trong:

Tiền xử lý câu hỏi: Module sẽ được phát triển riêng và qua các phiên bản khác nhau, trong đề tài sẽ dùng lại kết qua của chương trình máy tính đã được phát triển riêng.

Phân lớp câu hỏi: Bước này có nhiệm vụ phân loại theo lĩnh vực và các chủ đề. Quá trình này có thể phân chia làm 2 bước chính:

1) *Phân lớp xác định miền câu hỏi*: Ban đầu từ CSDL của máy tìm kiếm, mạng ngữ nghĩa xây dựng được từ các Ontology mà các Ontology được xây dựng theo từng miền lĩnh vực riêng biệt. Câu hỏi sau khi xác định được miền cụ thể thì không gian tìm sẽ giảm xuống rất nhiều. CSDL TBT có thể được chia ra các miền theo khung phân loại HS/ICS như: Nông nghiệp; Công nghiệp chế biến; ....

Ví dụ câu hỏi "Tiêu chuẩn xuất khẩu trái thanh long ruột đỏ sang thị trường Hoa Kỳ ?" được phân về miền "Nông nghiệp". Ngoài ra do đặc thù tài liệu TBT được phân cấp theo quốc gia/khu vực địa lý hành chính, nên có thể thêm miền địa lý cho câu hỏi, ở đây câu hỏi sẽ được phân về miền quốc gia (Hoa Kỳ). Như vậy để phù hợp thì miền CSDL được phân cấp thành 2 miền ngang cấp nhau là: Miền khu vực địa lý và Miền lĩnh vực. Với ví dụ câu hỏi trên thì không gian tìm kiếm sẽ giảm rất nhiều.

2) *Phân loại câu hỏi trong một miền cụ thể:* cấu trúc bên trong của một Ontology được phân cấp rất lớn do các Ontology lồng vào nhau. Ontology như vậy sẽ là một cây có cấu trúc phân tầng, mỗi lớp trong Ontology sẽ thuộc một tầng xác định. Bước này giảm giới hạn tối đa của không gian tìm kiếm.

Tổng thể thì sự khác nhau giữa phân loại miền tìm kiếm và nội miền chỉ ở không gian tìm kiếm. Phân loại miền câu hỏi có không gian là toàn bộ tập Ontology của mạng, còn phân loại nội miền chỉ ở trong một miền cụ thể.

Hiện nay có 2 các hướng tiếp cận để phân loại câu hỏi: Hướng biểu thức chính qui (regular expression); Hướng xác suất. Cụ thể cho phương pháp phân loại câu hỏi thì sẽ được trình bày ở phần sau của đề tài.

Biến đổi dạng câu hỏi: câu hỏi sau khi làm mịn thì sẽ được đưa về dạng chuẩn đã định trước (biểu thức logic), từ ngôn ngữ tự nhiên sang ngôn ngữ logic (Question Logic Language - QLL).

Mỗi câu thuộc QLL chứa các thuật ngữ term (biến, thủ tục, biểu thức, ...) của ngôn ngữ Proloc. Theo [6] thì có một số quy tắc biến đổi S_1 (dạng ngôn ngữ tự nhiên) sang S_2 (dạng QLL) như sau:

1) Một danh từ đơn (đơn vị từ) trong S_1 sẽ tạo 1 vị từ đơn trong S_2. Ví dụ có "G/TBT/N/VNM/78" là mã ký hiệu" thì S_2 chứa vị từ mã_ký_hiệu ("G/TBT/N/VNM/78").
2) Cụm danh từ trong S_1 sẽ tạo vị từ phức trong S_2 với tham số bằng số lượng từ đơn chứa trong cụm danh từ của S_1 + 1. Ví dụ câu hỏi "Số lượng thông báo của Việt Nam?" thì S_2 chứa thông_báo (Việt Nam, x).
3) Một động từ đơn trong S_1 sẽ tạo 1 vị từ phức với 1 hoặc nhiều tham số, tham số đầu tiên là chủ ngữ của S_1, tham số thứ 2 là vị ngữ chính trong S_1, tham số thứ 3 là vị ngữ phụ trong S_1,... Ví dụ có S_1 "Bộ y tế ban hành thông báo "G/TBT/N/VNM/78" thì S_2 chứa vị từ ban_hành (Bộ Y tế, "G/TBT/N/VNM/78")

4) Một giới từ trong S_1 sẽ tạo 1 vị từ phức với 2 tham số là 2 từ được nối bởi giới từ đó. Ví dụ S_1 "Trái thanh long nằm trên bàn" thì S_2 chứa nằm_trên(trái thanh long, bàn)
5) Một tính từ định tính tạo nên 1 vị từ. Ví dụ S_1 "Cộng đồng chung Châu Âu" thì S_2 chứa Châu_Âu(x)
6) Một tính từ định lượng tạo nên 1 cặp vị từ. Ví dụ S_1 "Tập HS lớn bao nhiêu" thì S_2 chứa kích_thước(Tập HS, x).

Với 6 quy tắc trên thì một câu hỏi bất kỳ đều được chuyển sang dạng QLL, câu hỏi chuẩn QLL sẽ có hiệu quả về suy luận và tối ưu xử lý.

Tìm kiếm câu trả lời: câu hỏi ở dạng QLL sẽ được xử lý theo các phép toán thực hiện trên dữ liệu là (Ontology) mạng ngữ nghĩa đã xây dựng từ trước.

Từ S1 chuẩn QLL thì hệ thống cần tìm ra câu trả lời cho S1. Ở đây bài báo đề nghị thuật toán:

Thuật toán Tìm Câu Trả Lời

Procedure **Tìm_Câu_Trả_Lời**

Input: Câu truy vấn Q; Mạng ngữ nghĩa O; Điều kiện so khớp S; Điều kiện tương đồng A

Output: Danh sách R chứa câu trả tìm được

B1: t ← true *// bắt đầu tìm kiếm*

B2: khi t = true thì lặp:

B2.1: o ← lấy_ra(O)

B2.2: nếu o tồn tại:

B.2.2.1: K ← so_khớp(o, Q, S)

B.2.2.2: nếu K đúng thì:

To ← Lấy_thuộc_tính(o)

B.2.2.3: Xác_định(t)

B3: Đánh_giá(R)

Bkt: Kết thúc

Đầu vào của thuật toán gồm có: Điều kiện so khớp S; Điều kiện tương đồng A. Hai tham số S, A này đã được xác định từ CSDL đã xây dựng sẵn (một số giá trị tham số sẽ được giả định sẵn). Giá trị của S và A luôn luôn sẽ được lưu vết lại trong quá trình chạy chương trình, từ đó hệ thống có được tập điều kiện đã chạy và cho ra kết quả. Sau này hệ thống có thể lựa chọn lại các điều kiện mà đã chạy cho ra kết quả lần trước.

Thuật toán Tìm_Câu_Trả_Lời sẽ lần lượt duyệt qua từng đối tượng Ontology o trong CSDL O nhờ vào cờ hiệu t, sau đó sẽ so khớp o và câu truy vấn Q với tập điều kiện S nhờ vào hàm so_khớp( ), nếu xác định o phù hợp thì lấy tập thuộc tính To trong o ra. Tập thuộc tính To sẽ được hàm

Đánh_giá_tương_đồng(.) xác định các thuộc tính có thỏa điều kiện để đưa vào tập kết quả R. Đến đây đã xét xong một đối tượng Ontology o, kể đến thuật toán xác định cờ hiệu t nhờ vào hàm Xác_định(.). Thuật toán sẽ dừng khi cờ hiệu t có trị Fasle, và công việc cuối cùng của thuật toán là rà soát lại tập kết quả R theo tiêu chí đã thiết lập sẵn trong hàm Đánh_giá(.).

Mạng ngữ nghĩa: là CSDL tri thức cho Hệ thống tìm ngữ nghĩa, các tập Ontology trong mạng sẽ được xây dựng dần dần theo thời gian. Dữ liệu TBT sau khi nhận được thì cần có quy trình và cơ chế để đưa vào CSDL. Dữ liệu sẽ được đưa vào quy trình theo như đã trình bày trong .

4. Phân lớp câu truy vấn

4.1. Giới thiệu chung phân lớp câu truy vấn

Giới thiệu

Một hệ tìm kiếm ngữ nghĩa thường quy định một số bước cố định trong quá trình thực thi, trong đó bước đầu tiên là xử lý câu truy vấn Q để xác định: Cái gì? (what); Thời gian? (when): .. Để thực hiện bước đầu tiên này thì hệ thống cần sử dụng một số thông tin đặc trưng của Q để xác định kiểu của câu trả lời. Kiểu của câu trả lời phải xoay quanh vấn đề trọng tâm của câu hỏi, từ đó đưa ra thông tin như mong muốn của người dùng.

Trong hệ tìm kiếm ngữ nghĩa thì đối tượng cần tìm chính là các dữ liệu trong nút mạng ngữ nghĩa, vì vậy phân lớp câu hỏi chính là việc phân lớp có ngữ nghĩa cho câu hỏi, đảm nhận vai trò: Giảm không gian tìm kiếm; Nâng cao độ chính xác câu trả lời. Ngoài ra đối với mạng ngữ nghĩa lớn (đa tầng, đa lớp) thì việc phân lớp còn phải thực hiện việc: Xác định miền Ontology; Xác định lĩnh vực trong miền đang xét.

Các phương pháp phân lớp câu hỏi

Như đã trình bày ở phần trên, bài báo tập trung vào hướng tiếp cận xác suất để phân lớp câu hỏi. Hướng tiếp cận này có 2 hướng chính: Hướng học máy (machine learning); Hướng mô hình hóa ngôn ngữ (language modeling). Các thuật toán thuộc các tiếp cận này sẽ tính toán xác suất phân lớp cho câu hỏi dựa trên những đặc trưng/mối quan hệ của các từ trong câu truy vấn đưa vào.

Hướng học máy sử dụng các thuật toán và kỹ thuật cho phép máy tính có thể học được. Hướng này được nhiều nhà nghiên cứu phát triển các thuật toán khác nhau, ở đây để tài tập trung vào thuật toán Support Vector Machine (SVM). Support Vector Machine (SVM) lần đầu tiên được đề xuất bởi Vapnik trong những năm 1960, sau đó thuật toán này liên tục được cải tiến và áp dụng vào nhiều lĩnh vực khác nhau.

Support Vector Machine (SVM) có ý tưởng chính là: Chuyển tập mẫu từ không gian biểu diễn Rn của chúng sang một không gian Rd có số chiều lớn hơn; Trong không gian Rd, tìm một siêu phẳng tối ưu để phân hoạch tập mẫu này dựa trên phân lớp của chúng, cũng có nghĩa là tìm ra miền phân bố của từng lớp trong không gian Rn để từ đó xác định được phân lớp của 1 mẫu cần nhận dạng; Siêu phẳng là một mặt hình học f(x) trong không gian N chiều, với $x \in RN$.

Các ưu điểm chính của SVM.

- Rất hiệu quả để giải quyết bài toán dữ liệu có số chiều lớn (ảnh của dữ liệu biểu diễn gene, protein, tế bào, …).
- Giải quyết vấn đề overfitting (không phù hợp) rất tốt (dữ liệu có nhiễu và tách dời nhóm hoặc dữ liệu huấn luyện quá ít).
- Là phương pháp phân lớp nhanh.

- Có hiệu suất tổng hợp tốt và hiệu suất tính toán cao.

Các ứng dụng của SVM:

- Nhận dạng: Tiếng nói; Ảnh; Chữ viết tay (hiệu quả từ mạng nơron trở lên), ...
- Phân loại văn bản, khai mỏ dữ liệu văn bản.
- Phân tích dữ liệu theo thời gian.
- Phân tích dữ liệu gene, nhận dạng bệnh, công nghệ bào chế thuốc.
- Phân tích dữ liệu marketing.
- ....

SVM ban đầu được thiết kế để giải quyết bài toán phân lớp nhị phân (số lớp là 2), hiện nay nó được đánh giá là một trong các thuật toán có hiệu quả rất tốt trong việc phân lớp văn bản [8].

4.2.Thuật toán

SVM

Cho tập dữ liệu cần học $D = \{(x_i, y_i), \text{với } i = 1, .., n\}$ với $x_i \in R_m$ và $y_i \in \{0, 1\}$ là một số nguyên xác định x_i dương hay âm. Một tài liệu x_i được gọi là dữ liệu dương nếu nó thuộc lớp c_i, là âm nếu nó không thuộc c_i. Bộ phận lớp tuyến tính được xác định bằng siêu phẳng

$\{x : f(x) = W^T + w_0 = 0\}$

Trong đó $W \in R^m$ và $w_0 \in R$ là tham số của mô hình. Hàm phân lớp nhị phân $h: R^m \rightarrow \{0, 1\}$, thu được bằng cách xác định dấu của $f(x)$

Ta xét mỗi dữ liệu là một điểm trong mặt phẳng, dữ liệu học là tách rời tuyến tính, nếu tồn tại một siêu phẳng sao cho hàm phân lớp phù hợp với tất cả các nhãn, $y_i f(x_i) > 0$ với mọi $i = 1, ..., n$.

Rosenblatt đã đưa ra một thuật toán xác định siêu phẳng:

Thuật toán Rosenblatt

Procedure Rosenblatt

B1. $W \leftarrow 0$

B2 $W_0 \leftarrow 0$

B3. Lặp

B3.1: $c \leftarrow 0$

B3.2: for $i \leftarrow i, ..., n$

Do $s \leftarrow sign(y_i(W^T x_i + W_0))$

Nếu $s < 0$ thì

$W \leftarrow W + y_i x_i$

$W_0 \leftarrow W_0 + y_i x_i$

$c \leftarrow c + 1$

B4. Điều kiện lặp $c = 0$

B5. Return(W, W_0)

B6. Kết thúc.

Đánh_giá_tương_đồng(.) xác định các thuộc tính có thỏa điều kiện để đưa vào tập kết quả R. Đến đây đã xét xong một đối tượng Ontology o, kế đến thuật toán xác định cờ hiệu t nhờ vào hàm Xác_định(.). Thuật toán sẽ dừng khi cờ hiệu t có trị Fasle, và công việc cuối cùng của thuật toán là rà soát lại tập kết quả R theo tiêu chí đã thiết lập sẵn trong hàm Đánh_giá(.).

Mạng ngữ nghĩa: là CSDL tri thức cho Hệ thống tìm ngữ nghĩa, các tập Ontology trong mạng sẽ được xây dựng dần dần theo thời gian. Dữ liệu TBT sau khi nhận được thì cần có quy trình và cơ chế để đưa vào CSDL. Dữ liệu sẽ được đưa vào quy trình theo như đã trình bày trong .

4. Phân lớp câu truy vấn

4.1. Giới thiệu chung phân lớp câu truy vấn

Giới thiệu

Một hệ tìm kiếm ngữ nghĩa thường quy định một số bước cố định trong quá trình thực thi, trong đó bước đầu tiên là xử lý câu truy vấn Q để xác định: Cái gì? (what); Thời gian? (when);... Để thực hiện bước đầu tiên này thì hệ thống cần sử dụng một số thông tin đặc trưng của Q để xác định kiểu của câu trả lời. Kiểu của câu trả lời phải xoay quanh vấn đề trọng tâm của câu hỏi, từ đó đưa ra thông tin như mong muốn của người dùng.

Trong hệ tìm kiếm ngữ nghĩa thì đối tượng cần tìm chính là các dữ liệu trong nút mạng ngữ nghĩa, vì vậy phân lớp câu hỏi chính là việc phân lớp có ngữ nghĩa cho câu hỏi, đảm nhận vai trò: Giảm không gian tìm kiếm; Nâng cao độ chính xác câu trả lời. Ngoài ra đối với mạng ngữ nghĩa lớn (đa tầng, đa lớp) thì việc phân lớp còn phải thực hiện việc: Xác định miền Ontology; Xác định lĩnh vực trong miền đang xét.

Các phương pháp phân lớp câu hỏi

Như đã trình bày ở phần trên, bài báo tập trung vào hướng tiếp cận xác suất để phân lớp câu hỏi. Hướng tiếp cận này có 2 hướng chính: Hướng học máy (machine learning); Hướng mô hình hóa ngôn ngữ (language modeling). Các thuật toán thuộc các tiếp cận này sẽ tính toán xác suất phân lớp cho câu hỏi dựa trên những đặc trưng/mối quan hệ của các từ trong câu truy vấn đưa vào.

Hướng học máy sử dụng các thuật toán và kỹ thuật cho phép máy tính có thể học được. Hướng này được nhiều nhà nghiên cứu phát triển các thuật toán khác nhau, ở đây đề tài tập trung vào thuật toán Support Vector Machine (SVM). Support Vector Machine (SVM) lần đầu tiên được đề xuất bởi Vapnik trong những năm 1960, sau đó thuật toán này liên tục được cải tiến và áp dụng vào nhiều lĩnh vực khác nhau.

Support Vector Machine (SVM) có ý tưởng chính là: Chuyển tập mẫu từ không gian biểu diễn Rn của chúng sang một không gian Rd có số chiều lớn hơn; Trong không gian Rd, tìm một siêu phẳng tối ưu để phân hoạch tập mẫu này dựa trên phân lớp của chúng, cũng có nghĩa là tìm ra miền phân bố của từng lớp trong không gian Rn để từ đó xác định được phân lớp của 1 mẫu cần nhận dạng; Siêu phẳng là một mặt hình học f(x) trong không gian N chiều, với $x \in RN$.

Các ưu điểm chính của SVM

- Rất hiệu quả để giải quyết bài toán dữ liệu có số chiều lớn (ảnh của dữ liệu biểu diễn gene, protein, tế bào, ...).
- Giải quyết vấn đề overfitting (không phù hợp) rất tốt (dữ liệu có nhiễu và tách dời nhóm hoặc dữ liệu huấn luyện quá ít).
- Là phương pháp phân lớp nhanh.

- Có hiệu suất tổng hợp tốt và hiệu suất tính toán cao.

. Các ứng dụng của SVM:

- Nhận dạng: Tiếng nói; Ảnh, Chữ viết tay (hiệu quả từ mạng nơron trở lên), ...
- Phân loại văn bản, khai mở dữ liệu văn bản.
- Phân tích dữ liệu theo thời gian.
- Phân tích dữ liệu gene, nhận dạng bệnh, công nghệ bào chế thuốc.
- Phân tích dữ liệu marketing.
- ....

SVM ban đầu được thiết kế để giải quyết bài toán phân lớp nhị phân (số lớp là 2), hiện nay nó được đánh giá là một trong các thuật toán có hiệu quả rất tốt trong việc phân lớp văn bản [8].

4.2.Thuật toán

SVM

Cho tập dữ liệu cần học D ={(x_i, y_i), với i = 1,..., n} với $x_i \in R_m$ và $y_i \in \{0, 1\}$ là một số nguyên xác định x, dương hay âm. Một tài liệu x_i được gọi là dữ liệu dương nếu nó thuộc lớp c_i, là âm nếu nó không thuộc c_i. Bộ phận lớp tuyến tính được xác định bằng siêu phẳng

$\{x : f(x) = W^T + w_0 = 0\}$

Trong đó $W \in R^m$ và $w_0 \in R$ là tham số của mô hình. Hàm phân lớp nhị phân $h: R^m \rightarrow \{0, 1\}$, thu được bằng cách xác định dấu của f(x)

Ta xét mỗi dữ liệu là một điểm trong mặt phẳng, dữ liệu học là tách rời tuyến tính, nếu tồn tại một siêu phẳng sao cho hàm phân lớp phù hợp với tất cả các nhãn, $y_i f(x_i) > 0$ với mọi i = 1, ..., n.

Rosenblatt đã đưa ra một thuật toán xác định siêu phẳng:

Thuật toán Rosenblatt

```
Procedure Rosenblatt
B1. W ← 0
B2. Wo ← 0
B3. Lặp
    B3.1: e ← 0
    B3.2: for i ← i, ..., n
            Do s ← sign(y_i(W^T x_i + Wo))
               Nếu s < 0 thì
                  W ← W + y_i x_i
                  Wo ← Wo + y_i x_i
                  e ← e + 1
B4. Điều kiện lặp e = 0
B5. Return(W, Wo)
B6. Kết thúc.
```

Điều kiện để D tách rời tuyến tính là số dữ liệu học n = |D| ≤ m + 1. Điều kiện này thường đúng với bài toán phân lớp văn bản vì số lượng từ mục thường sẽ ≥ số lượng dữ liệu học. Xét thuật toán *Rosenhlatt* trên thì độ phức tạp của quá trình xác định siêu phẳng sẽ tăng theo số chiều không gian m.

Phân lớp đa lớp với SVM

Việc phân lớp câu hỏi yêu cầu phải phân lớp đa lớp, do đó SVM cơ bản (nhị phân) sẽ được chuyển thành đa lớp. Một trong phương pháp cải tiến đã được trình bày trong [9]. Ý tưởng chính của [9]:

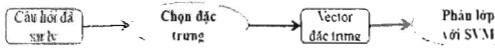
- Gia sử tập dữ liệu mẫu $(x_1, y_1), \ldots, (x_m, y_m)$, là một vector n chiều và $y_i \in Y$ là nhãn lớp được gán cho vector x_i
- Chia tập Y thành m tập lớp con có cấu trúc như sau $z_i = \{y_i, Y - y_i\}$.
- Áp dụng SVM phân lớp nhị phân cơ bản với m tập z_i để xây dựng siêu phẳng cho phân lớp này

Bộ phân lớp kết hợp của m bộ phân lớp trên được gọi là bộ phân lớp đa lớp mở rộng với SVM

Áp dụng SVM vào phân lớp câu hỏi

Để thực hiện phân lớp thì sẽ có 2 bước chính: Thiết kế mô hình phân cấp (taxonomy) cho tập câu hỏi; Xây dựng tập dữ liệu mẫu đã gán nhãn cho từng lớp câu hoi. Xét bước 1 thì nhận thấy miền ứng dụng câu hỏi sẽ tạo nên độ phức tạp (phân cấp) của cây. Xét bước 2 thì việc trích chọn đặc trưng là công việc quan trọng bậc nhất và phụ thuộc vào đặc điểm từng ngôn ngữ mà có sự khác nhau. Do đó, đề tài sẽ nhận lấy kết quả từ chương trình chuyên dụng về xử lý ngôn ngữ tự nhiên khác.

Sau khi xây dựng tập lớp câu hỏi và tập dữ liệu thì bước kế tiếp chính là tiến hành "học". Để quy trình hóa việc học thì có mô hình như sau:



Hình 1: Sơ đồ phân lớp với SVM

Thuật toán để phân lớp câu hỏi với SVM

Procedure Phân_Lớp_Với_SVM

Input: câu truy vấn Q, CSDL tri thức S, tập đặc trưng vào C

Output: tập kết quả R

B1: T ← Tạo_Cây(Q, S)

B2: L ← Gán_Nhãn(T, S)

B3: C ← Lấy_Đặc_Trưng(L, C)

B4: R ← MSVM(V, S)

Bkt: Kết thúc.

Các hàm Tạo_Cây (Q, S), Gán_Nhãn (T, S), Lấy_Đặc_Trưng (L, C), MSVM (V, S) được lập trình theo các thuật toán chi tiết sẽ được thiết kế riêng. Xét cụ thể việc lập trình liên quan đến Cây, thì có các kỹ thuật truyền thống để nghị như sau:

- Lưu trữ cấu trúc cây thì có thể dùng danh sách liên kết (có thể áp dụng thêm bảng băm Hash Table).

- Duyệt cây thì có thể dùng cách tổng quát (Duyệt tiền thứ tự, Duyệt trung thứ tự, Duyệt hậu thứ tự), hoặc duyệt theo các mức.
- Đối với các thao tác sắp xếp dữ liệu trên bộ nhớ trong thì thông thường sẽ dùng thuật toán Quick Sort (có đệ quy hoặc khử đệ quy)
- Phương pháp Đệ quy tuy rằng thể hiện nhiều sức mạnh và ưu điểm trong giải quyết các bài toán, nhưng có nhiều trường hợp thì cần phải khử đệ quy. Đối với các hàm có áp dụng thuật toán đệ quy để lập trình thì có thể khử đệ quy bằng các kỹ thuật kinh điển như dùng: Vòng lặp; Thủ tục dạng đệ quy đuôi; Cấu trúc dữ liệu kho dầy (stack); …

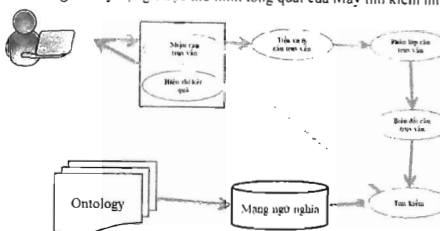
5. Kết quả, đánh giá và kết luận

5.1. Kết quả

Trong bài báo này đã thực hiện việc phân tích các dữ liệu đầu vào và đã thiết kế được cấu trúc dữ liệu của cây phân loại theo một khung cho trước (HS. ICS, …) dựa trên mô hình Taxonomy. Các nút trên cây phân loại có một số thuộc tính, trong đó thuộc tính giá trị nút là quan trọng nhất. Giá trị của nút có được là do thiết kế từ trước (theo đề nghị của chuyên gia, hoặc theo các công thức quy định sẵn) như Hình 3 Cấu trúc Cây Phân loại HS.

Tiếp đến, bài báo xây dựng mạng ngữ nghĩa cho hệ tìm kiếm từ thành phần nền là các Ontology. Một Ontology sẽ có kiểu theo như Hình 4, và các tài nguyên dữ liệu sẽ chuyển dạng chuẩn RDF như Hình 6. Trong CSDL tri thức của mạng ngữ nghĩa thì có nhiều dữ liệu được lưu trữ thành dạng bang như Bảng 3 để có ưu thế trong việc truy cập và xử lý.

Bài báo cũng đã xây dựng được mô hình tổng quát của Máy tìm kiếm như sau:



Tổng quát lại thì khi người dùng nhập vào câu hỏi đến khi kết quả đưa ra là theo qui trình 6 bước chung như sau:

B1: Người dùng nhập câu truy vấn Q ở giao diện chức năng.

B2: Câu Q sẽ được một module Tiền xử lý để phân tích về ngôn ngữ: Tiếng việt (có dấu và không dấu); Sai lỗi chính tả; Các thuật ngữ chuyên môn; … Kết quả sẽ cho ra một câu Q' đã được làm mịn lại để dễ dàng cho Bộ phân tích câu hỏi thực hiện ở bước kế tiếp.

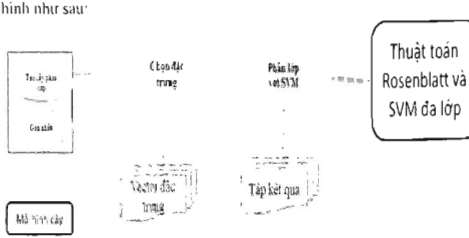
B3: Câu Q' được đưa vào bộ phân tích để xác định xem Q' thuộc miền nào và lĩnh vực nào trong miền đó.

B4: Câu Q' được biến đổi về dạng chuẩn (biểu thức logic để tiến hành tìm kiếm).

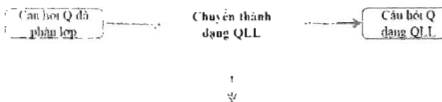
B5: Q' được tìm kiếm trên mạng ngữ nghĩa của máy tìm kiếm.

B6: Hiển thị kết quả đưa ra.

Trong quy trình ở trên thì các công việc nhận câu truy vấn, hiển thị kết qua, tiền xử lý câu truy vấn là công việc của chương trình khác thực hiện. Tiếp đó là đến việc phân lớp câu truy vấn thì đề tài áp dụng Hướng học máy (machine learning) và cụ thể là thuật toán SVM đa lớp. Việc phân lớp thì bài báo đề nghị mô hình như sau:



Tiếp đến để thực hiện là biến đổi câu truy vấn về dạng QLL, theo như mô hình sau:



Bước cuối cùng mà Hệ tìm kiếm cần phải thực hiện là thực thuật toán **Tìm_Câu_Trả_Lời** đã được trình bày ở mục trên. Đầu vào của thuật toán gồm có 4 thành phần: Câu truy vấn Q; Mạng ngữ nghĩa O; Điều kiện so khớp S; Điều kiện tương đồng A. Hai thành phần Câu truy vấn Q: Mạng ngữ nghĩa O thì đề tài đã đề nghị mô hình thiết kế chi tiết, còn lại 2 thành phần Điều kiện so khớp S; Điều kiện tương đồng A được xây dựng với sự hỗ trợ của chuyên gia và chương trình phần mềm chuyên dụng.

5.3. Thảo luận

Mục tiêu 3 "*Nghiên cứu và xây dựng công cụ hỏi đáp thông minh TBT Long An*" là mục tiêu quan trọng thuộc loại ưu tiên hàng đầu của. Để thực hiện mục tiêu này thì có nhiều giải pháp để thực hiện, và công việc tìm giải pháp tối ưu cho nó là điều rất khó khăn và đặc biệt thú vị. Bài báo mong muốn đóng góp một giải pháp hiệu quả cho mục tiêu 3, và việc lựa chọn thuật toán SVM là vì kết quả của. Trong thì Zhang và cộng sự cho ra kết quả sau khi áp dụng 5 loại thuật toán khác nhau:

Bảng 4: Độ chính xác trên các phân lớp con với 5 giải thuật

Thuật toán	1000	2000	3000	4000	5000
Láng giềng gần nhất	57.4%	62.8%	65.2%	67.2%	68.4%
Naïve Bayes	48.8%	52.8%	56.6%	56.2%	58.4%
Cây quyết định	67.0%	70.0%	73.6%	75.4%	77.0%
SNoW	42.2%	66.2%	69 0%	66.6%	74.0%
SVM	68.0%	75.0%	77.2%	77.4%	80.2%

Kết quả trên có được là khi chọn vector đặc trưng là *Túi từ. Túi từ.* biểu diễn văn bản/câu hỏi độc lập với ngôn ngữ và ngữ pháp. Mỗi văn bản/câu hỏi được biểu diễn bằng tập các từ, các từ này không sắp thứ tự.

Từ các thực nghiệm của Zhang và cộng sự thì thấy rằng: SVM mang lại độ chính xác khá cao so với các phương pháp còn lại; Độ chính xác sẽ tỉ lệ thuận với độ lớn của dữ liệu học; Phương pháp Cây quyết định có thể cho thấy rằng độ chính xác không kém nhiều với SVM. Tuy nhiên, việc cần làm tiếp theo là triển khai lập trình theo như đã thiết kế và đánh giá lại kết quả chạy của chương trình thực tế.

Sử dụng kỹ thuật Cây quyết định vào việc xây dựng Hệ tìm kiếm ở trên thì có thể xem là một giải pháp thay thế tốt và được xem là một phiên bản để so sánh với giải pháp sử dụng kỹ thuật SVM.

Tài liệu tham khảo

Tiếng Việt

[1]. Lê Đình Tuấn, (2017), *Nghiên cứu và xây dựng Hệ hỏi đáp thông minh cho thông tin về Hàng rào Kỹ thuật trong Thương mại (TBT) của tỉnh Long An*", Tạp chí Kinh tế - Công nghiệpTrường Đại học Kinh tế Công nghiệp Long An.

[2]. Nguyễn Minh Đế (2017), *Nghiên cứu và xây dựng quy trình và thuật toán để phân loại tài liệu TBT,* Tạp chí Kinh tế - Công nghiệp Trường Đại học Kinh tếCông Nghiệp Long An

[3]. Trần Cao Đệ, Phạm Nguyên Khang, *Phân loại văn bản với máy học vector hỗ trợ và cây quyết định"*. Trường Đại học Cần Thơ.

[4]. Trần Thị Thu Thảo, Vũ Thị Chinh, *Xây dựng hệ thống phân loại tài liệu tiếng Việt.* Khoa Công Nghệ Thông Tin, Trường Đại học Lạc Hồng.

Tiếng Anh

[5]. Customs Cooperation Council (WCO) (1983), *Công ước Quốc tế về Hệ thống hài hòa mô tả và mã hàng hóa (Công ước HS)"*. Brussels.

[6]. Clocksin W. F. and Mellish C.S. (1981), *Programming in Prolog,* Springer-Verlag.

[7]. Maria Vargas-Vera, Enrico Motta, John Domingue (2003), *AQUA. An Ontology-Driven Question Answering Syste..* New Directions in Question Answering:53-57.

[8]. Soumen Chakrabarti (2003). *"Mining the Web: discoveringknowledge from hypertext data"*. Morgan Kaufmann Publishers.

[9]. Liu Yi, Zheng Y F (2005) "*One-against-all multi-Class SVM classification using reliability measures"*. Proceedings of the 2005 International Joint Conference on Neural Networks Montreal, Canada.

[10]. Swick, Ralph (1997-12-11), Resource Description Framework (RDF)", W3C. Archived from the original on February 14, 1998. Retrieved 2015-11-24.

[11]. T. Mitchell (1997), *Machine Learning,* McGraw Hill, New York.

[12]. Zhang, D. and Lee, W.S (2003), *Question Classification using Support Vector Machines,* In Proceedings of SIGIR 2003.

[13]. Zhang, D. and Lee, W.S (2003). *Question Classification using Support Vector Machines,* In Proceedings of SIGIR 2003.

Ngày nhận: 06/7/2019

Ngày duyệt đăng:12/12/2019