**Chương 1: TỔNG QUAN**

* 1. **Đặt vấn đề**

Ngày nay internet đã và đang là nguồn kiến thức vô tận mang lại nhiều lợi ích cho con người. Sự phát triển mạnh mẽ của nó kéo theo việc những kiến thức trong ngành công nghệ thông tin tăng lên nhanh chóng làm cho việc tra cứu kiến thức cần thiết trở nên khó khăn hơn. Với các công cụ tìm kiếm hiện nay như Google, Yahoo… chỉ giúp người dùng tìm được những tài liệu có chứa từ khóa. Từ đây người dùng phải tốn thời gian và công sức vào từng tài liệu để tìm được đúng thông tin mình cần mà có khi không tìm thấy hoặc tìm thấy thông tin sai lệch. Vấn đề đặt ra là làm sao để có được một công cụ tìm kiếm theo ngữ nghĩa, hiểu được và trả lời câu hỏi của người dùng bằng ngôn ngữ tự nhiên một cách thân thiện. Đặc biệt có thể tìm kiếm bằng tiếng Việt, nhu cầu mà hầu như rất ít công cụ hỗ trợ và kết quả còn hạn chế [1].

Dùng Ontology là một giải pháp biểu diễn tri thức và chia sẻ thông tin mà cả hệ thống và con người có thể hiểu được. Ontology chứa những đặc tả rõ ràng các khái niệm về một lĩnh vực và quan hệ giữa các khái niệm đó [2]. Nó được dụng trong trí tuệ nhân tạo, công nghệ Web ngữ nghĩa (Semantic Web), các hệ thống kỹ thuật, kỹ thuật phần mềm, sinh tin học và kiến trúc thông tin như là một hình thức biểu diễn tri thức về thế giới hoặc một số lĩnh vực cụ thể [3, 4, 5].

Nhận thấy những nhu cầu trên, nhóm thực hiện đề tài:

**“Xây dựng và làm giàu Ontology tiếng Việt chuyên ngành Công nghệ thông tin”.**

Đề tài nhằm xây dựng một Ontology là nền tảng cho những ứng dụng sau này như tìm kiếm thông tin tiếng Việt, hệ thống hỏi đáp tiếng Việt cho ngành công nghệ thông tin, hỗ trợ cho web ngữ nghĩa, giúp xác định thực thể có tên trong tài liệu công nghệ thông tin tiếng Việt. Ontology này có khả năng mở rộng cấu trúc và dữ liệu để phục vụ mục đích hỏi đáp của người dùng. Ngoài ra nhóm cũng sẽ xây dựng công cụ cho phép làm giàu Ontology từ internet.

* 1. **Mục tiêu và phạm vi đề tài**
* **Mục tiêu:** Xây dựng Ontology chuyên ngành công nghệ thông tin tiếng Việt phục vụ cho việc tìm kiếm thực thể có tên và xác định quan hệ giữa chúng trong tài liệu công nghệ thông tin tiếng Việt, hỗ trợ trả lời theo ngữ nghĩa cho những câu hỏi bằng tiếng Việt về công nghệ thông tin, có thể được phát triển để sử dụng cho hệ thống tư vấn về chương trình đào tạo công nghệ thông tin.
* **Phạm vi đề tài:** Xây dựng Ontology tiếng Việt giới hạn trong lĩnh vực Công nghệ thông tin – Ontology for Vietnamese Information Technology (OVIT) nhằm lưu trữ:
* Các khái niệm trong lĩnh vực Công nghệ thông tin và quan hệ giữa chúng.
* Thông tin các công ty, trường học, tổ chức, hiệp hội, chuyên gia, các sự kiện trong ngành và quan hệ ngữ nghĩa giữa chúng.
* Các chương trình đào tạo Công nghệ thông tin.

Nguồn dữ liệu: từ ComputingOntology của nhóm nghiên cứu thuộc ACM, trang Wikipedia tiếng Việt, website Bộ thông tin và truyền thông, một số website báo điện tử, các bài báo lĩnh vực công nghệ thông tin tiếng Việt, website các trường có đào tạo ngành công nghệ thông tin trong nước, tài liệu từ internet tìm được từ công cụ tìm kiếm như Google, Yahoo.

* 1. **Kết quả dự kiến**

Kiến thức: Nắm được khái niệm, cấu trúc, mục đích, ứng dụng, cách xây dựng một Ontology. Các công cụ hỗ trợ xây dựng ontology hiện nay và sử dụng ngôn ngữ Java để xây dựng công cụ làm giàu Ontology (OVIT) bán tự động.

Phương pháp và công cụ: Xây dựng Ontology dùng công cụ Protégé, tạo công cụ làm giàu cá thể bán tự động sử dụng API của Google và Yahoo để tìm kiếm nguồn tài liệu làm giàu, dùng thuật toán SVM để phân loại tài liệu công nghệ thông tin tiếng Việt, rút trích các cá thể từ tài liệu đã phân loại cho phép người dùng kiểm tra, chỉnh sửa và lưu vào Ontology.

Dữ liệu: Dự kiến nhập bằng tay được khoảng 1000 lớp, 100 quan hệ và 100 cá thể, làm giàu cá thể bán tự động được 1000 cá thể.

* 1. **Tổng kết chương**

Trong chương này nhóm chúng em đã nêu ra mục tiêu của việc nghiên cứu và xây dựng ontology hiện nay. Các ứng dụng của nó ngày càng được quan tâm và nó đã trở thành phần “lõi” cho các ứng dụng quan tâm đến ngữ nghĩa, tri thức hơn là những dữ liệu thông thường được lưu trữ trong các hệ quản trị cơ sở dữ liệu. Từ đó nhóm đặt ra vấn đề và đề xuất thực hiện đề tài **“Xây dựng và làm giàu Ontology tiếng Việt chuyên ngành Công nghệ thông tin”** để giải quyết vấn đề đó. Đề tài được giới hạn trong phạm vi và mục tiêu đã nêu trong chương.

1. **CƠ SỞ LÝ THUYẾT**
   1. **Khảo sát các nghiên cứu có liên quan**
      1. **Thế giới**
         1. **TheComputingOntology [11]**

Được công bố năm 2005, ban đầu có tên là Ontology Project được xây dựng bởi một nhóm thuộc tổ chức ACM [10] nhằm biểu diễn kiến thức về máy tính và thông tin có quan hệ chặt chẽ theo quy tắc phục vụ cho việc nghiên cứu hay giảng dạy trong lĩnh vực liên quan tới tính toán và quản lý, xử lý thông tin. Công việc được hỗ trợ bởi tổ chức khoa học quốc gia Mỹ (*National Science Foundation*), ACM (*Association for Computing Machinery*), IEEE và Đại học Mở của Hà Lan (*Open University of the Netherlands*).

Nguồn dữ liệu gồm: Tài liệu chương trình đào tạo của các trường đại học như: Lewis University, Villanova University… và những thuật ngữ quan trọng trong các môn học về tính toán từ hệ thống phân lớp trên ACM (ACM Computing Classification System [13]). Được xây dựng dùng công cụ soạn thảo Ontology là Protégé, đến nay đã có 6 phiên bản của ComputingOntology trên trang web chính thức của nó.

*Nhận xét:* Xây dựng được một Ontology về tính toán và thông tin hỗ trợ phát triển hệ thống tư vấn về chương trình đào tạo, phát triển những chương trình học mới, kiểm tra những chương trình học đã có, làm rõ ràng các mối quan hệ giữa những môn học với nhau, phát triển những chương trình gồm nhiều ngành học, đóng góp cho việc phân lớp trong nghiên cứu. Tuy nhiên, dữ liệu của Ontology là tiếng Anh không thể phục vụ cho các nghiên cứu chuyên ngành công nghệ thông tin tiếng Việt.

* + - 1. **SwetoDBLP** [12]

Được xây dựng bởi nhóm tác giả từ khoa Khoa học máy tính của trường Đại học Georgia, Mỹ. SwetoDblp [34] là một Ontology có kích thước lớn tập trung vào dữ liệu thông tin của các bài báo về khoa học máy tính như: Tên, tác giả, nhà xuất bản… Dữ liệu chính của nó lấy từ cơ sở dữ liệu DBLP [16]. Ngoài ra, còn có 3 nguồn dữ liệu khác được dùng để tạo SwetoDblp là danh sách các trường đại học lấy từ Google được chỉnh sửa bằng tay lại cho phù hợp, danh sách các website của nhà xuất bản và danh sách các hội thảo được tạo bằng tay theo dữ liệu trong DBLP.

Dữ liệu của SwetoDblp được lưu trữ dùng định dạng RDF, sử dụng bộ từ vựng lược đồ (schema-vocabulary) có sẵn như FOAF [17] và Dublin Core [18]. Việc tạo ra và cập nhật Ontology được thực hiện dùng công cụ SAX-parser để chuyển dữ liệu dạng XML của DBLP sang RDF. Dữ liệu sẽ được cập nhật hàng tháng theo dữ liệu XML mới nhất từ DBLP và danh sách các trường đại học, nhà xuất bản và hội thảo.

SwetoDblp hiện đang được sử dụng để kiểm tra cho OptARQ, một cơ chế cho phép tối ưu hóa câu truy vấn vào Ontology dùng SPARQL [19]. Ngoài ra, Ontology này còn được dùng để tìm kiếm các bài báo và chuyên gia, phân biệt, tránh sự nhập nhằng giữa tên các nhà nghiên cứu trong danh sách mail của DBWorld [33].

*Nhận xét:* Ontology này như một thư viện điện tử với lượng thông tin lớn về các bài báo, không phục vụ cho việc tìm kiếm các khái niệm và thông tin trong ngành công nghệ thông tin.

* + 1. **Trong nước**
       1. **Ontology for Vietnamese Language (OVL) – Open version [7]**

Là một Ontology tổng quát (Universal Ontology) được thực hiện bởi Nguyễn Tuấn Đăng, Võ Hoài An, Nguyễn Trí Phúc trường Đại học Công nghệ Thông tin. Xây dựng trên phiên bản Protégé 3.4.3. Mục tiêu tác giả xây dựng Ontology này là để đóng góp cho những nghiên cứu về xử lý ngôn ngữ tiếng Việt, xây dựng tri thức mở rộng trong nhiều lĩnh vực bằng tiếng Việt.

Dữ liệu của Ontology là dữ liệu tổng quát về các lĩnh vực gồm 10 lĩnh vực chính theo các mục được lấy theo VNExpress như: Khoa học, Pháp luật, Chính trị, Kinh doanh, Thể thao, Văn hóa du lịch, Xã hội, Vi tính, Viễn thông, Ô tô xe máy. Ngoài ra còn lấy dữ liệu từ các nguồn như Wikipedia tiếng Việt, Yellow Page và nhiều website khác nhau liên quan đến các lĩnh vực trên [6].

*Nhận xét:* Kết quả tạo ra được Ontology gồm số lượng lớp là 2.543, số lượng cá thể là 10.024, với 312 ràng buộc và 87 thuộc tính thuộc nhiều lĩnh vực. Tuy nhiên, dữ liệu của Ontology mang tính phổ quát, không tập trung vào một lĩnh vực (domain) cụ thể. Ví dụ như trong ngành Công nghệ thông tin không có chứa thông tin về những khái niệm, chuyên gia hay chương trình đào tạo của ngành.

* + - 1. **Ontology khoa học công nghệ [4]**

Được thực hiện bởi Bộ môn Hệ thống thông tin của trường Đại học Bách khoa Hà Nội. Hệ thống hỗ trợ tìm kiếm dựa trên từ khóa, cấu trúc dữ liệu lưu trữ, tìm kiếm mở rộng dựa trên ngữ nghĩa và tri thức phục vụ cho việc quản lý tài liệu và thông tin trong lĩnh vực khoa học công nghệ. Nhằm giải quyết cho những yêu cầu đó tác giả đã đề xuất phương pháp xây dựng một Ontology chuyên ngành khoa học công nghệ để khai thác các suy diễn ngữ nghĩa.

Những khái niệm được xây dựng dựa trên việc khảo sát nhu cầu quản lý thông tin tại phòng KHCN thuộc Đại học Bách Khoa Hà Nội, phòng KHCN thuộc sở Khoa học Công nghệ Thành Phố Hà Nội, sở Bưu chính Viễn thông. Người bảo trì có thể là tác giả hoặc những người có quan tâm và có kiến thức về Ontology sẽ nâng cấp cập nhật thông tin khi có thay đổi.

Với việc sử dụng Ontology này hệ thống ngoài việc dùng để tra cứu các đề tài, sản phẩm công nghệ, chuyên gia, tài liệu, giải pháp, công nghệ… thì còn có thể trả lời được những câu hỏi tổng hơp phân tích như: Có những đề tài nào thuộc lĩnh vực mà người dùng quan tâm? Đề tài nào dành được sự quan tâm nhiều nhất cũng như nhận định về giá trị, khả năng ứng dụng vào thực tiễn? Tài liệu đang được xem xét có những phiên bản nào, sự đánh giá của các độc giả đối với các phiên bản của tài liệu này như thế nào? Tìm những chuyên gia đa lĩnh vực như chuyên gia vừa trong lĩnh vực CNTT vừa trong lĩnh vực Hoá sinh.

Ontology này được xây dựng dùng phần mềm soạn thảo cơ sở tri thức được viết dựa trên các API của Protégé. Cơ sở dữ liệu này chứa dữ liệu về khoảng 3000 chuyên gia, 1500 đề tài cùng với hơn 150 lĩnh vực KHCN.

*Nhận xét:* Không rút trích được khái niệm hay cá thể từ nội dung tài liệu hay bài báo khoa học. Dữ liệu của Ontology KHCN không bao trùm hết lĩnh vực công nghệ thông tin.

* 1. **Tổng quan về ontology**
     1. **Định nghĩa**

Theo wikipedia, từ “ontology” tạm dịch là “bản thể luận” có nguồn gốc từ triết học và được dùng trong nhiều lĩnh vực như khoa học máy tính, hệ thống kỹ thuật, kỹ thuật phần mềm, tin sinh học, khoa học thư viện, kiến trúc thông tin và các website ngữ nghĩa (Semantic web). Một số định nghĩa về ontology được sử dụng nhiều hiện nay gồm:

* Theo quan điểm triết học, ontology là nghiên cứu về bản chất của sinh vật, sự tồn tại hoặc những sự vật thực tế, cũng như các loại sinh vật cơ bản và các mối quan hệ của chúng (wikipedia).
* Trong khoa học máy tính, một ontology là một đặc tả rõ ràng của một sự trừu tượng hóa (An ontology is an explicit specification of a conceptualization) [20].
* Một ontology định nghĩa một tập từ vựng cho những nhà nghiên cứu sử dụng khi cần chia sẻ thông tin trong một lĩnh vực. Nó bao gồm những định nghĩa của các khái niệm cơ bản trong một lĩnh vực và mối quan hệ giữa chúng mà máy có thể hiểu được [2].

Tóm lại, Ontology gồm những khái niệm về một lĩnh vực cụ thể và các mối quan hệ giữa chúng. Một Ontology về một lĩnh vực sẽ mô tả rõ ràng những thực thể giúp con người và máy có thể hiểu và suy luận được theo ngữ nghĩa trong phạm vi lĩnh vực đó.

* + 1. **Vì sao phải xây dựng Ontology?**

Để chia sẻ kiến thức chung về kiến trúc thông tin giữa con người hoặc những tác tử phần mềm với nhau [20]. Nếu các hệ thống cùng chia sẻ chúng một Ontology bên dưới thì dữ liệu do con người nhập vào tại hệ thống này sau khi được xử lý thông qua Ontology có thể được tổng hợp, phân tích tại một hệ thống khác và cung cấp thông tin cho người sử dụng khác.

Cho phép tái sử dụng kiến thức về một lĩnh vực. Sau khi xây dựng một Ontology cho một lĩnh vực, những người khác có thể tái sử dụng và mở rộng, làm giàu thêm cho nó. Hoặc cũng có thể tích hợp những Ontology có sẵn để mô tả nhiều phần của một Ontology về một lĩnh vực lớn.

Làm rõ ràng những giả định thuộc chuyên ngành. Việc sử dụng một Ontology ở bên dưới thay vì dùng ngôn ngữ lập trình sẽ giúp dễ dàng thay đổi những giả định thuộc chuyên ngành khi kiến thức về lĩnh vực này của chúng ta thay đổi. Nếu những giả định này được viết bằng ngôn ngữ lập trình thì sẽ gây khó hiểu và khó thay đổi, sửa chữa nhất là đối với những người không phải là chuyên gia lập trình.

Có thể phân tích và suy luận kiến thức chuyên ngành vì những thuật ngữ, khái niệm cũng như các mối quan hệ giữa chúng đề được khai báo, đặc tả trong Ontology với cấu trúc có thể suy luận được theo ngữ nghĩa.

* + 1. **Thành phần của Ontology**
* **Các lớp (Classes) - Khái niệm**

Lớp là nhóm, tập hợp các đối tượng trừu tượng có thể chứa các cá thể, lớp khác hoặc cả hai. Các Ontology biến đổi tuỳ thuộc vào cấu trúc và nội dung của nó: Một lớp có thể chứa các lớp con, có thể là một lớp tổng quan (chứa tất cả mọi thứ), có thể là lớp chỉ chứa những cá thể riêng lẻ. Các lớp được sắp xếp theo cấu trúc có thứ bậc, thường là với một lớp thông dụng nhất kiểu Thing ở trên đỉnh và các lớp con rất cụ thể ở phía dưới cùng.

Lớp có thể có các ràng buộc (restrictions) cho các quan hệ của cá thể thuộc lớp đó, ví dụ như một Tác giả phải viết một hoặc nhiều tác phẩm thì một cá thể của tác giả phải có quan hệ “là tác giả của” với một hoặc nhiều cá thể của tác phẩm.

* **Các cá thể (Individuals)**

Là những đối tượng đại diện thuộc một lớp cụ thể trong domain. Mỗi cá thể có thể có các thuộc tính của lớp mà nó thể hiện và quan hệ với các cá thể khác theo ràng buộc của lớp. Những cá thể còn có thể được coi như là những trường hợp của lớp.

* **Các thuộc tính (Properties)**

Các đối tượng trong Ontology có thể được mô tả thông qua việc khai báo các thuộc tính của chúng. Mỗi một thuộc tính đều có tên và giá trị của thuộc tính đó. Các thuộc tính được sử dụng để lưu trữ các thông tin mà đối tượng có thể có. Ví dụ, đối với một cá thể của lớp người có thể có các thuộc tính: Họ\_tên, ngày\_sinh, quê\_quán, số\_cmnd…

Giá trị của một thuộc tính có thể có các kiểu dữ liệu phức tạp.

* **Các mối quan hệ (Relations)**

Là thuộc tính để mô tả mối liên hệ giữa các đối tượng trong Ontology. Một mối quan hệ là một thuộc tính có giá trị là một đối tượng nào đó trong Ontology. Một đối tượng có thể có một hoặc nhiều quan hệ trong Ontology bất kể lớp của nó có quan hệ đó hay không, quan hệ của đối tượng phải tuân theo ràng buộc của lớp chứa đối tượng đó nếu có.

* + 1. **Làm thế nào để xây dựng một Ontology?**
       1. **Phương pháp xây dựng một Ontology**

Hiện nay không có phương pháp chuẩn nào cho việc xây dựng một Ontology [1]. Khi xây dựng Ontology chúng ta nên dựa vào nhu cầu của ứng dụng sẽ sử dụng nó để thiết kế cho phù hợp.

Quá trình xây dựng một Ontology là một quá trình lặp, thường bắt đầu bằng một phiên bản thô rồi sao đó xem xét, chỉnh sửa, lọc lại Ontology phiên bản trước và thêm vào các chi tiết.

Những khái niệm trong Ontology là những đối tượng thực tế hoặc logic phản ánh thế giới thực và những quan hệ thường là những động từ trong câu mô tả khái niệm trong lĩnh vực.

Thông thường, phương pháp xây dựng Ontology gồm các bước:

*Bước 1:* ***Xác định miền và phạm vi của Ontology****.* Đây là chúng ta nên làm trước khi muốn xây dựng một Ontology. Các yêu cầu đối với Ontology trong hệ thống thường là mô tả miền quan tâm nhằm cung cấp cơ sở tri thức trong việc giải quyết những mục đích chuyên biệt. Để nhận diện chính xác những yêu cầu chúng ta cần phải trả lời một số câu hỏi như:

* Ontology cần mô tả miền nào?
* Ontology phục vụ cho mục đích chuyên biệt gì?
* Cơ sở tri thức trong Ontology sẽ trả lời những câu hỏi gì?
* Ontology nhằm vục vụ đối tượng nào?
* Ai là người sẽ xây dựng, quản trị Ontology?

Các câu trả lời có thể thay đổi ở mỗi bước lặp trong quá trình xây dựng Ontology tùy mục đích của ứng dụng hoặc có những tính năng cần bổ sung lúc đó. Trả lời các câu hỏi trên sẽ giúp giới hạn phạm vi thực sự của Ontology cần mô tả và dự trù các kỹ thuật sẽ sử dụng trong quá trình phát triển. Ví dụ như Ontology cần xây dựng có chức năng xử lý ngôn ngữ tự nhiên, ứng dụng dịch tài liệu tự động thì cần phải có kỹ thuật xác định từ đồng nghĩa.

Sau khi đã phát thảo phạm vi Ontology dựa trên việc trả lời những câu hỏi trên, chúng ta tiếp tục tinh chỉnh lại bằng cách trả lời các câu hỏi kiểm chứng khả năng (competency question):

* Ontology đã có đủ thông tin để trả lời cho các câu hỏi được quan tâm trên cơ sở tri thức hay không?
* Câu trả lời của cơ sở tri thức đã đáp ứng được mức độ, yêu cầu nào của người sử dụng?
* Các ràng buộc và quan hệ phức tạp trong miền quan tâm đã được biểu diễn hợp lý chưa?

*Bước 2:* ***Xem xét việc kế thừa các Ontology có sẵn****:* đây là một công đoạn thường hay sử dụng để giảm thiểu công sức xây dựng một Ontology. Bằng cách kế thừa các Ontology tương tự có sẵn, người xây dựng có thể thêm hoặc bớt các lớp, quan hệ giữa các lớp, thực thể.. để tinh chỉnh tùy theo mục đích của mình. Ngoài ra, việc sử dụng lại các Ontology có sẵn cũng rất quan trọng khi cần sự tương tác giữa các ứng dụng khác nhau vì các ứng dụng sẽ cần phải hiểu các lớp, thực thể, quan hệ… của nhau để thuận tiện trong việc trao đổi hoặc thống nhất thông tin.

*Bước 3:* ***Liệt kê các thuật ngữ quan trọng trong Ontology****:* Liệt kê tất cả các thuật ngữ xuất hiện trong miền quan tâm (có thể đồng nghĩa hoặc chồng nhau) như tên khái niệm, quan hệ, thuộc tính… Thông thường, các thuật ngữ là danh từ sẽ trở thành các lớp, tính từ sẽ trở thành thuộc tính, còn động từ sẽ là quan hệ giữa các lớp.

*Bước 4:* ***Xây dựng các lớp và cấu trúc lớp phân cấp****:* Định nghĩa các lớp từ một số thuật ngữ đã liệt kê trong bước 3, sau đó xây dựng cấu trúc lớp phân cấp theo quan hệ *lớp cha-lớp con.* Lớp ở vị trí càng cao trong cấu trúc này sẽ có mức độ tổng quát càng cao. Vị trí đầu tiên thuộc về ***lớp gốc***, tiếp theo là các ***lớp trung gian***, và cuối cùng là ***lớp lá***. Lớp lá là lớp không thể triển khai được nữa và chỉ được biểu hiện bằng các thực thể.



Hình 1: Cấu trúc lớp phân cấp

Thực thể của lớp con *“là-một”* thực thể của lớp cha nó.

Có nhiều hướng tiếp cận khác nhau cho vấn đề xây dựng cấu trúc lớp phân cấp như:

* Hướng xây dựng ***từ trên xuống*** *(top-down)*: bắt đầu bằng các lớp có mức độ tổng quát cao nhất, sau đó triển khai dần đến lớp lá.
* Hướng xây dựng ***từ dưới lên*** *(bottom-up)*: Ngược với hướng xây dựng cấu trúc lớp phân cấp từ trên xuống, hướng này bắt đầu bằng việc xác định các lớp được cho là cụ thể nhất, sau đó tổng quát hóa đến khi được lớp gốc.
* Cách ***kết hợp*** *(combination)*: cách này kết hợp cả hai hướng xây dựng trên. Đầu tiên chọn các lớp nổi bật nhất trong miền quan tâm, sau đó tổng quát hóa và cụ thể hóa cho đến khi được cấu trúc mong muốn.

*Bước 5:* ***Định nghĩa các thuộc tính và quan hệ cho lớp****:* các lớp tạo ra ở bước 4 chỉ mới là những tên gọi, tiếp theo chúng ta cần định nghĩa thuộc tính của lớp là các thông tin bên trong của lớp, mô tả một khía cạnh nào đó của lớp và được dùng để phân biệt với các lớp khác. Có hai loại: ***thuộc tính đơn*** *(simple property)* và ***thuộc tính phức*** *(complex property)*. Thuộc tính đơn là các giá trị đơn ví dụ: chuỗi, số,… còn thuộc tính phức có thể chứa hoặc tham khảo đến một đối tượng khác. Một lớp sẽ kế thừa toàn bộ các thuộc tính của tất cả các cha nó.

*Bước 6:* ***Định nghĩa các ràng buộc về thuộc tính và quan hệ của lớp:*** Các ràng buộc (restrictions) giới hạn giá trị mà một thuộc tính có thể nhận. Hai ràng buộc quan trọng nhất đối với một thuộc tính là ***lượng số*** *(cardinality)* và ***kiểu*** *(type)*. Ràng buộc lượng số quy định số giá trị mà một thuộc tính có thể nhận. Hai giá trị thường thấy của ràng buộc này là đơn trị *(single)* và đa trị *(multiple)*. Ràng buộc thứ hai là về kiểu, các kiểu mà một thuộc tính có thể nhận là: chuỗi, số, boolean, liệt kê và kiểu thực thể. Riêng kiểu thực thể có liên quan đến hai khái niệm gọi là: ***miền*** *(domain)* và ***khoảng*** *(range)*. Khái niệm miền được dùng để chỉ lớp (hay các lớp) mà một thuộc tính thuộc về. Trong khi đó, khoảng chính là lớp (hay các lớp) làm kiểu cho giá trị thuộc tính kiểu thực thể.



Hình 2: Ràng buộc về thuộc tính.

*Bước 7:* Đây là bước cuối cùng khép lại một vòng lặp xây dựng Ontology. Việc chúng ta cần làm ở bước này là tạo thực thể cho mỗi lớp và gán giá trị cho các thuộc tính.

* + - 1. **Ngôn ngữ để xây dựng Ontology:**

RDF: là mô hình dữ liệu cho các đối tượng và các mối quan hệ giữa chúng. Mô hình dữ liệu này dùng cú pháp của XML chỉ giúp cho thông tin được thể hiện ở dạng bộ ba theo đúng mô hình RDF chứ thông tin vẫn chưa thể hiện gì về mặt ngữ nghĩa.

RDF Schema: là một ngôn ngữ Ontology cơ bản mô tả các thuộc tính (property) và các lớp (class) của đối tượng RDF. Nó được phát triển ở tầng trên của RDF cho nên bản thân RDF-Schema cũng chính là RDF. Nó được mở rộng từ RDF và bổ sung thêm các tập từ vựng để hỗ trợ cho việc xây dựng các Ontology được dễ dàng để hình thành nên ngữ nghĩa cho thông tin, là cơ sở để xây dựng các công cụ tìm kiếm ngữ nghĩa.

OWL: là ngôn ngữ mô tả từ vựng phong phú để mô tả các thuộc tính và các lớp, các mối quan hệ giữa các lớp (như disjointness), số của giá trị (cardinality), tính tương đương (equality), định kiểu thuộc tính, đặc tính của thuộc tính (đối xứng).

* **OWL (Ontology Web Language)**

OWL là ngôn ngữ được phát triển mới nhất trong các ngôn ngữ ontology chuẩn được công nhận bởi World Wide Web Consortium (W3C) để thúc đẩy sự phát triển của các web ngữ nghĩa (Semantec Web).

OWL là ngôn ngữ Ontology khá mạnh, nó ra đời sau RDFS nên biết kế thừa những lợi thế của ngôn ngữ này đồng thời bổ sung thêm nhiều yếu tố giúp khắc phục được những hạn chế của RDFS. OWL giúp tăng thêm yếu tố logic cho thông tin và khả năng phân loại, ràng buộc kiểu cũng như lượng số tương đối mạnh.

OWL kế thừa từ DAML+OIL được phát triển bởi tổ chức W3C. Tên DAML+OIL là sự kết hợp giữa tên DAML-ONT (http://www.daml.org/2000/10/daml-ont.html) do Mỹ đề xuất và ngôn ngữ OIL (http://www.ontoknowledge.org/oil/) do Châu Âu đề xuất.

Hiện nay có ba loại OWL : OWL Lite, OWL DL (description logic), và OWL Full.

* OWL Lite: hỗ trợ cho những người dùng chủ yếu cần sự phân lớp theo thứ bậc và các ràng buộc đơn giản. Ví dụ: Trong khi nó hỗ trợ các ràng buộc về tập hợp, nó chỉ cho phép tập hợp giá trị của 0 hay 1. Điều này giúp OWL Lite dễ sử dụng và thực thi và việc cung cấp các công cụ hỗ trợ OWL Lite dễ dàng hơn so với các bản khác nhưng lại hạn chế trong việc diễn đạt.
* OWL DL (OWL Description Logic): hỗ trợ cho những người dùng cần cung cấp sự diễn đạt tối ưu và đảm bảo tất cả các kết luận là có thể dự tính được và sẽ hoàn thành trong một thời gian nhất định. OWL DL bao gồm tất cả các cấu trúc của ngôn ngữ OWL, nhưng chúng chỉ có thể được sử dụng với những hạn chế nào đó (Ví dụ: Trong khi một lớp có thể là một lớp con của rất nhiều lớp, một lớp không thể là một thể hiện của một lớp khác).

OWL mất toàn bộ tính tương thích với RDF. Thông thường, một tài liệu RDF phải được mở rộng theo một số cách và bị giới hạn theo các cách khác trước khi nó là một tài liệu OWL DL hợp lệ. Mọi tài liệu OWL DL hợp lệ là tài liệu RDF hợp lệ.

* OWL Full: sử dụng tất cả các từ vựng nền tảng (primitive) của ngôn ngữ OWL. Nó cho phép kết hợp tùy ý các từ vựng nền tảng với RDF và RDF Schema vì vậy nó tạo ra sự diễn đạt tối đa và tự do. Ví dụ, trong OWL Full, một lớp có thể được xem xét đồng thời như là một tập của các cá thể và như là một cá thể trong chính bản thân nó. OWL Full cho phép một Ontology gia cố thêm ý nghĩa của các từ vựng được định nghĩa trước (RDF hoặc OWL) và hoàn toàn tương thích với RDF. Ngôn ngữ này trở nên quá mạnh mẽ đến mức là không thể quyết định được (undecidable), ảnh hưởng đến hỗ trợ lập luận đầy đủ hoặc hỗ trợ lập luận hiệu quả.

Các phiên bản này tách biệt về các tiện ích khác nhau, OWL Lite là phiên bản dễ hiểu nhất và phức tạp nhất là OWL Full. Việc lựa chọn ngôn ngữ con nào phù hợp nhất là phụ thuộc vào nhu cầu của mỗi người.

Mối liên hệ giữa các ngôn ngữ con của OWL:

* Mọi Ontology hợp lệ dựa trên OWL Lite đều là Ontology hợp lệ trên OWL DL.
* Mọi Ontology hợp lệ dựa trên OWL DL đều là Ontology hợp lệ trên OWL Full.
* Mọi kết luận hợp lệ dựa trên OWL Lite đều là kết luận hợp lệ trên OWL DL.
* Mọi kết luận hợp lệ dựa trên OWL DL đều là kết luận hợp lệ trên OWL Full
  1. **Tổng kết chương**

Trong chương này nhóm chúng em đã trình bày một số khảo sát các nghiên cứu về ontology trong và ngoài nước. Từ đó rút ra các nhận xét về tình hình nghiên cứu và ứng dụng ontology hiện nay.

Theo xu hướng đó, việc xây dựng các ontology tiếng Việt đã bắt đầu nhận được sự quan tâm của các nhà nghiên cứu khoa học ở Việt Nam như phần khảo sát đã đề cập. Đặc biệt đối với ngành công nghệ thông tin thì lượng dữ liệu ngày càng lớn và cần được quản lý một cách có hệ thống và có ngữ nghĩa.

Ngoài ra, trong chương này đã nêu tổng quan về ontology gồm có định nghĩa, các thành phần trong ontology, ngôn ngữ xây dựng ontology và phương pháp để xây dựng một ontology.

1. **XÂY DỰNG VÀ LÀM GIÀU ONTOLOGY CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN TIẾNG VIỆT (OVIT)**

Trong chương này nhóm em xin trình bày về công cụ và quá trình xây dựng ontology chuyên ngành công nghệ thông tin tiếng Việt. Ngoài ra, chương này cũng sẽ trình bày đề xuất phương pháp của nhóm để xây dựng công cụ làm giàu ontology này.

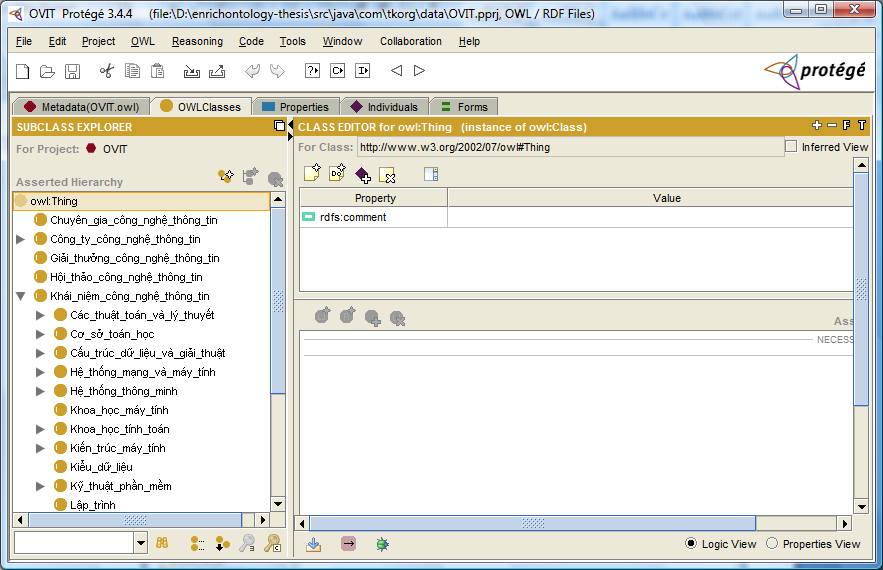
* 1. **Xây dựng OVIT**
     1. **Công cụ**

Ontology chuyên ngành công nghệ thông tin tiếng Việt (OVIT) được xây dựng dùng công cụ soạn thảo Protégé [21]. Đây là bộ phần mềm mã nguồn mở Java được nghiên cứu và phát triển từ năm 1998 bởi nhóm nghiên cứu của Mark Musen thuộc đại học Stanford, California nhằm quản lý các thông tin trong lĩnh vực sinh y học. Đây là dự án được nhận được sự quan tâm và tài trợ từ rất nhiều tổ chức, trong đó có Bộ Quốc Phòng Mỹ. Hiện nay, nó có một cộng đồng hàng nghìn người sử dụng và đã có rất nhiều miền ứng dụng khác nhau sử dụng sự hỗ trợ của công cụ này. Mã nguồn Protégé có thể được tìm thấy tại website: http://smi-protege.stanford.edu/repos/protege/owl/trunk.

Hiện tại, Protégé đã có phiên bản 4.1 hỗ trợ OWL 2. Tuy nhiên, do phiên bản này chưa có API hỗ trợ nên nhóm quyết định dùng phiên bản Protégé 3.4.4 có API hỗ trợ cho việc xây dựng công cụ làm giàu sau này. Công cụ Protégé có thể chia làm 2 loại là: Protégé-Frame và Protégé-OWL:

* Protégé-Frame cung cấp cho người dùng một giao diện chứa đầy đủ công cụ và kiến thức để hỗ trợ người dùng xây dựng và lưu trữ các ontology chuyên ngành dựa trên khung, tùy biến các hình thức nhập dữ liệu, và nhập dữ liệu tức thời.
* Protégé-OWL là một công cụ mở rộng của Protégé hỗ trợ các Web Ontology Language (OWL). Công cụ soạn thảo Protégé-OWL cho phép người dùng lưu và xem các ontology OWL và RDF, xem và chỉnh sửa các lớp, cá thể, thuộc tính, quan hệ và các ràng buộc, kiểm tra tính đúng đắn của ontology.

Theo nhóm chúng em nhận xét thì công cụ Protégé-Frame sẽ phù hợp hơn cho nhu cầu xem chỉnh sửa và nhập dữ liệu cho ontology, trong khi nếu muốn xây dựng ontology mới và có giao diện phù hợp với việc xây dựng cấu trúc cho ontology thì dùng Protégé-OWL sẽ dễ dàng hơn. Ngoài ra, Protégé-OWL cũng hỗ trợ cho ngôn ngữ OWL tốt hơn là Protégé-Frame. Vì vậy, nhóm sẽ sử dụng công cụ Protégé-OWL để xây dựng ontology. Hướng dẫn sử dụng công cụ Protégé-OWL được nêu trong phần phụ lục A.



Hình 3: Giao diện protégé 3.4.4

* Các ưu điểm của Protégé là:
* Hỗ trợ đầy đủ ba phiên bản của ngôn ngữ OWL là OWL-Full, OWL-Lite và OWL-DL.
* Nhờ sử dụng mô hình hướng đối tượng của ngôn ngữ Java, Protégé rất hiệu quả trong việc mô hình hóa các lớp, thực thể, quan hệ…
* Giao diện thiết kế trực quan có tính tương tác cao. Người sử dụng có thể định nghĩa các thành phần của Ontology trực tiếp từ các form. Nó hỗ trợ xây dựng các thành phần của một Ontology rất nhanh và hiệu quả.
* Cho phép biểu diễn trực quan Ontology dưới dạng các sơ đồ.
* Cho phép xây dựng Ontology từ nhiều nguồn khác nhau.
* Protégé tự động lưu một bản tạm của Ontology. Nếu có lỗi phát sinh trong quá trình thao tác thì Ontology cũ sẽ tự động được phục hồi.
* Cung cấp chức năng tìm kiếm lỗi, kiểm tra tính nhất quán và đầy đủ của Ontology.
* Cho phép các lớp và thuộc tính của Ontology này có thể được sử dụng trong một Namespace khác mà chỉ cần sử dụng các URL để tham khảo.
* Hỗ trợ suy luận trực tiếp trên Ontology dựa trên Interface chuẩn DL Implementation Group (DIG).
* Hỗ trợ sinh mã tự động. Protégé cho phép chuyển Ontology thành mã nguồn RDF/XML, OWL, DIG, Java, EMF Java Interfaces, Java Schema Classes.. Các mã này có thể được nhúng trực tiếp vào ứng dụng và là đầu vào cho các thao tác trên Ontology khi cần.
  + 1. **Quá trình xây dựng**
       1. **Xác định yêu cầu của ontology:** ontology lưu trữ những thông tin về công nghệ thông tin tiếng Việt đáp ứng được các nhu cầu của hệ thống như:

## Tìm kiếm thực thể có tên, xác định quan hệ giữa các thực thể

## Hỗ trợ trả lời cho hệ thống hỏi đáp về:

## Các khái niệm trong lĩnh vực công nghệ thông tin.

## Thông tin các chuyên gia trong lĩnh vực, các công ty hoạt động trong lĩnh vực công nghệ thông tin, giải thưởng, hội thảo, sự kiện, tổ chức, hiệp hội công nghệ thông tin và các trường có đào tạo công nghệ thông tin.

## Ví dụ:

## Java là gì?

## Trường nào đã đoạt giải vô địch cuộc thi Robocon năm 2009?

## Sử dụng cho hệ thống tư vấn về chương trình đào tạo công nghệ thông tin.

## **Xem xét các ontology có sẵn:** Theo khảo sát của nhóm thì có 3 Ontologies về công nghệ thông tin có thể xem xét.

## Thứ nhất là ontology về khoa học công nghệ của trường Đại học Bách Khoa Hà Nội đã khảo sát ở trên. Tuy nhiên ontology này không thể tìm được nguồn và cũng không liên lạc được với tác giả, do đó không thể kế thừa được từ ontology này.

## Thứ hai là ontology tổng quát OVL có chứa dữ liệu về công nghệ thông tin nhưng cấu trúc không phù hợp và chứa nhiều dữ liệu tổng quát thuộc nhiều lĩnh vực nên nhóm quyết định chỉ xem xét nhập một số dữ liệu chọn lọc từ ontology này chứ không sử dụng nó.

## Cuối cùng là ComputingOntology là một ontology khá đầy đủ về các khái niệm và môn học trong lĩnh vực công nghệ thông tin tiếng Anh với cấu trúc của các khái niệm lấy từ ACM. Vì vậy, nhóm sẽ xây dựng cấu trúc lớp khái niệm trong Ontology của mình theo cấu trúc lớp của Ontology này bằng cách dịch và nhập bằng tay để có thể chỉnh sửa một số chi tiết cho phù hợp với yêu cầu đối với Ontology của mình.

## Hiện nay, cũng có một số nghiên cứu đã xây dựng Ontology theo cách sử dụng một số công cụ dịch kết hợp với các chuyên gia chỉnh sửa lại như Ontology được xây dựng trong đề tài [22]. Tuy nhiên việc dịch như vậy thì cấu trúc của Ontology được xây dựng có thể không đáp ứng được yêu cầu của ứng dụng.

## **Một số thuật ngữ quan trọng trong Ontology:** Dựa vào yêu cầu đã xác định ở trên chúng ta sẽ có một số khái niệm chính trong Ontology như: Khái niệm trong lĩnh vực công nghệ thông tin, nguồn, định nghĩa, sự kiện công nghệ thông tin, công ty phần mềm, công ty phần cứng, chuyên gia công nghệ thông tin, trường đào tạo ngành công nghệ thông tin, tổ chức, hiệp hội, giải thưởng công nghệ thông tin, hợp tác đào tạo, sản xuất, trao giải thưởng, được trao giải thưởng.

## **Xây dựng cấu trúc lớp cho Ontology:** Dựa vào những thuật ngữ chính đã xác định ở trên và nguồn dữ liệu lấy từ website Wikipedia tiếng Việt nhóm chúng em đã xây dựng cấu trúc của Ontology gồm các lớp chính như sau:

## 

## *Hình 4: Các lớp chính trong Ontology*

## Các lớp chính trong Ontology:

## *Khái niệm thuộc ngành công nghệ thông tin:* tất cả các khái niệm đều được chuyển thành lớp con của lớp này, khi thêm vào những khái niệm mới sẽ là lớp con của các lớp bên dưới.

## *Tin học*

## *Công nghệ thông tin*

## *Lập trình cơ bản*

## *Hệ thống thông tin*

## *Khoa học máy tính*

## *Mạng máy tính*

## *Kỹ thuật phần mềm*

## *Kỹ thuật máy tính*

## *Khái niệm trong xã hội*

## *Bảo mật*

## *Hệ thống pháp lý*

## *Hợp đồng*

## *Kiểm soát*

## *Sở hữu trí tuệ*

## *Trách nhiệm nghề nghiệp*

## *Đạo đức nghề nghiệp*

## *Tác động của thay đổi công nghệ*

## *Lịch sử máy tính*

## *Thiết bị máy móc ban đầu*

## *Hệ thống phần mềm phần cứng*

## *Phần cứng không thuộc hệ thống*

## *Phần mềm không thuộc hệ thống*

*Sự kiện công nghệ thông tin:* gồm nhiều lớp con là những năm có xảy ra sự kiện, mỗi sự kiện là một cá thể của lớp năm.

*Công ty hoạt động trong ngành công nghệ thông tin*

*Công ty phần mềm*

*Công ty phần cứng*

*Công ty dịch vụ*

*Chi nhánh*

*Trường đào tạo công nghệ thông tin*

*Trung tâm dạy nghề CNTT*

*Trung cấp*

*Cao đẳng*

*Đại học*

*Tổ chức/hiệp hội công nghệ thông tin*

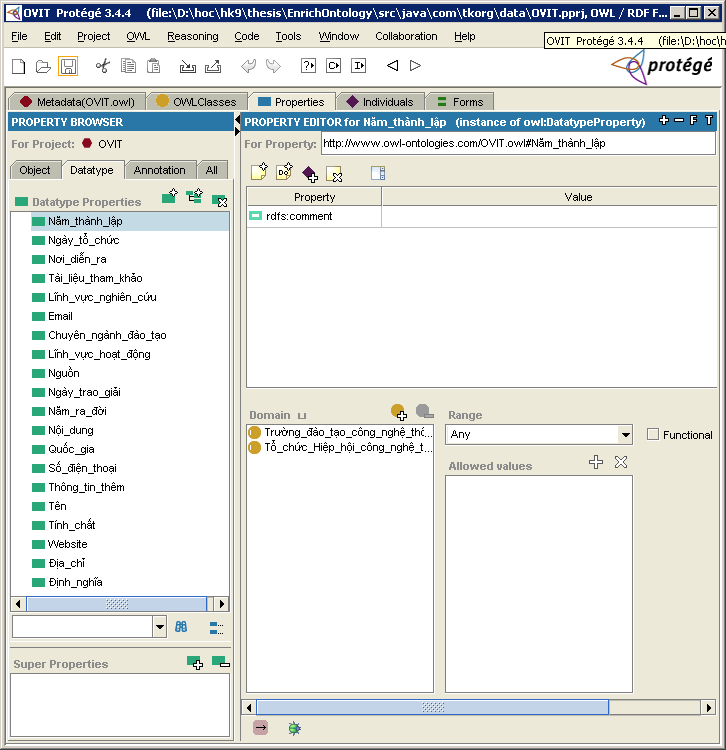
*Giải thưởng công nghệ thông tin:* Tên của mỗi giải thưởng là một lớp, mỗi lần giải thưởng được trao tạo ra một cá thể của giải thưởng đó.

*Chuyên gia công nghệ thông tin*

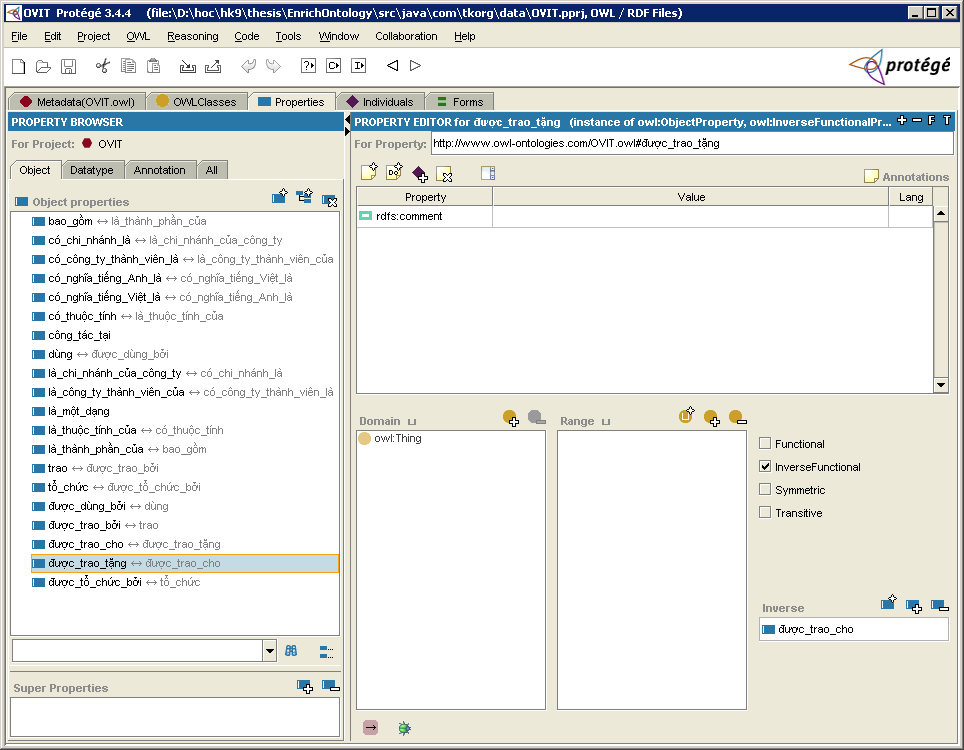
*Hội thảo công nghệ thông tin:* Giống như lớp *Giải thưởng công nghệ thông tin* ở trên, tên của mỗi hội thảo là một lớp, mỗi lần hội thảo được tổ chức tạo ra một cá thể của hội thảo đó.

Ngoài ra, nhóm đã nhập dữ liệu bổ sung thêm các khái niệm cho Ontology bằng cách dịch và nhập bằng tay các lớp từ ComputingOntology vào như là các lớp con của lớp *Khái niệm thuộc ngành công nghệ thông tin*. Như vậy cấu trúc của các lớp khái niệm sẽ theo cấu trúc của ComputingOntology.

* + - 1. **Định nghĩa các thuộc tính và quan hệ cho lớp:**



Hình 5: Các thuộc tính trong Ontology



Hình 6: Các quan hệ trong Ontology

Khái niệm thuộc ngành công nghệ thông tin:

Thuộc tính:

Định nghĩa: là nội dung định nghĩa làm rõ cho khái niệm.

Nguồn: là đường dẫn tới trang web chứa định nghĩa.

Sự kiện công nghệ thông tin:

Thuộc tính:

Nguồn: Là đường dẫn đến bài báo chi tiết về sự kiện

Nội dung: nội dung sự kiện tóm tắt

Quan hệ: mỗi cá thể này có quan hệ tùy theo nội dung của sự kiện. Ví dụ như sự kiện do công ty tổ chức thì cá thể sự kiện này sẽ có quan hệ “được tổ chức bởi” với cá thể của công ty tương ứng, và ngược lại, cá thể của công ty sẽ có quan hệ “tổ chức” với cá thể sự kiện. Hai quan hệ “được tổ chức bởi” và “tổ chức” là hai quan hệ nghịch đảo với nhau.

Công ty hoạt động trong ngành công nghệ thông tin:

Thuộc tính:

Tên: Tên công ty

Địa chỉ

Số điện thoại

Lĩnh vực hoạt động

Quốc gia: Hoạt động ở nước nào

Website: trang web chính thức của công ty

Thông tin thêm: Một số thông tin của công ty ngoài những thông tin trên.

Quan hệ:

* Công ty phần cứng, Công ty phần mềm, Công ty dịch vụ: có quan hệ “có công ty thành viên là” đối với các tập đoàn có nhiều công ty con, và quan hệ nghịch đảo của nó là “là công ty thành viên của”
* Công ty phần cứng, Công ty phần mềm, Công ty dịch vụ: có quan hệ “có chi nhánh là” với một hoặc nhiều cá thể của lớp Chi nhánh
* Chi nhánh: có quan hệ “là chi nhánh của công ty” với cá thể công ty, là quan hệ nghịch đảo của quan hệ “có chi nhánh là”.
* có quan hệ “tổ chức” với cá thể của Sự kiện công nghệ thông tin, Hội thảo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ “được tổ chức bởi”.
* có quan hệ “được trao tặng” với một hay nhiều cá thể của lớp Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo với quan hệ “được trao cho”.

Trường đào tạo công nghệ thông tin:

Thuộc tính:

Tên

Địa chỉ

Năm thành lập

Số điện thoại

Website

Chuyên ngành đào tạo

Thông tin thêm

Quan hệ:

* có quan hệ “tổ chức” với cá thể của Sự kiện công nghệ thông tin, Hội thảo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ “được tổ chức bởi”.
* có quan hệ “được trao tặng” với một hay nhiều cá thể của lớp Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo với quan hệ “được trao cho”.

Hướng phát triển: Sẽ thêm vào những thông tin về Chương trình đào tạo, Môn học để bổ sung dữ liệu đáp ứng được yêu cầu của ứng dụng trả lời cho những câu hỏi tư vấn về các chương trình đào tạo công nghệ thông tin.

Tổ chức/hiệp hội công nghệ thông tin:

Thuộc tính:

Tên

Địa chỉ

Quốc gia

Năm thành lập

Website

Thông tin thêm

Quan hệ:

* có quan hệ “trao” với 1 hoặc nhiều cá thể của Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ “được trao bởi”.
* có quan hệ “tổ chức” với cá thể của Sự kiện công nghệ thông tin, Hội thảo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ “được tổ chức bởi”.

Giải thưởng công nghệ thông tin:

Thuộc tính:

Năm ra đời

Ngày trao giải

Thông tin thêm

Quan hệ:

* có quan hệ “được trao bởi” với cá thể của lớp Tổ chức/hiệp hội công nghệ thông tin, trao tặng là quan hệ nghịch đảo với quan hệ “trao”.
* có quan hệ “được trao cho” với cá thể của các lớp Chuyên gia, Công ty hoặc Trường đào tạo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ “được trao tặng”.

Chuyên gia công nghệ thông tin:

Thuộc tính:

Tên

Email

Quốc gia

Lĩnh vực nghiên cứu

Tài liệu tham khảo: những bài báo, sách, công trình nghiên cứu có sự tham gia của chuyên gia.

Quan hệ:

* có quan hệ “công tác tại” với cá thể của các lớp Trường đào tạo công nghệ thông tin, Công ty hoạt động trong ngành công nghệ thông tin hay Tổ chức/hiệp hội công nghệ thông tin.
* có quan hệ “được trao tặng” với cá thể của lớp Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo với quan hệ “được trao cho”.

Hội thảo công nghệ thông tin:

Thuộc tính:

Nơi diễn ra

Ngày tổ chức

Quan hệ:

* có quan hệ “được tổ chức bởi” với cá thể của lớp Tổ chức/hiệp hội công nghệ thông tin, Trường đào tạo công nghệ thông tin, Công ty hoạt động trong ngành công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ “tổ chức”.
  + - 1. Định nghĩa về thuộc tính và quan hệ của lớp:

**Các ràng buộc về thuộc tính:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Lượng số** | **Kiểu** | **Miền** |
| Định nghĩa | Đa trị | String | Khái niệm trong ngành công nghệ thông tin |
| Nguồn | Đa trị | String | Khái niệm thuộc ngành công nghệ thông tin  Sự kiện công nghệ thông tin |
| Nội dung | 1 | String | Sự kiện công nghệ thông tin |
| Tên | Đa trị | String | Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin  Tổ chức/hiệp hội công nghệ thông tin  Chuyên gia công nghệ thông tin |
| Địa chỉ | Đa trị | String | Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin  Tổ chức/hiệp hội công nghệ thông tin |
| Số điện thoại | Đa trị | String | Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin |
| Lĩnh vực hoạt động | Đa trị | String | Công ty hoạt động trong ngành công nghệ thông tin |
| Quốc gia | Đa trị | String | Công ty hoạt động trong ngành công nghệ thông tin  Tổ chức/hiệp hội công nghệ thông tin |
| Website | Đa trị | String | Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin  Tổ chức/hiệp hội công nghệ thông tin |
| Thông tin thêm | 1 | String | Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin  Tổ chức/hiệp hội công nghệ thông tin  Giải thưởng công nghệ thông tin |
| Năm thành lập | 1 | Date | Trường đào tạo công nghệ thông tin  Tổ chức/hiệp hội công nghệ thông tin |
| Chuyên ngành đào tạo | Đa trị | String | Trường đào tạo công nghệ thông tin |
| Năm ra đời | 1 | Date | Giải thưởng công nghệ thông tin |
| Ngày trao giải | 1 | Date | Giải thưởng công nghệ thông tin |
| Email | Đa trị | String | Chuyên gia công nghệ thông tin |
| Lĩnh vực nghiên cứu | Đa trị | String | Chuyên gia công nghệ thông tin |
| Tài liệu tham khảo | Đa trị | String | Chuyên gia công nghệ thông tin |
| Nơi diễn ra | Đa trị | String | Hội thảo công nghệ thông tin |
| Ngày tổ chức | Đa trị | Date | Hội thảo gia công nghệ thông tin |

**Các ràng buộc về quan hệ:**

* + - 1. Nhập các cá thể vào Ontology: Nhóm tập hợp nguồn dữ liệu từ Wikipedia, tin tức trên các trang báo điện tử như: tuoitre.vn, vnexpress.net, vietnamnet… và thông tin từ trang web của Bộ thông tin truyền thông [30].

**Kết quả xây dựng và nhập dữ liệu cho Ontology:**

Xây dựng cấu trúc ontology – OVIT với 980 lớp, trong đó có gần 950 lớp là các khái niệm công nghệ thông tin lấy từ nhiều nguồn trên internet chủ yếu là Wikipedia và ComputingOntology. Nhập được 50 cá thể, 19 thuộc tính của các lớp và 20 quan hệ.

* 1. **Phương pháp làm giàu OVIT:**

Trước tiên chúng ta cần phân tích một chút về việc làm giàu Ontology. Bản chất của việc làm giàu Ontology là bổ sung dữ liệu cho Ontology, làm cho nó chứa nhiều thông tin hơn. Do đó các ứng dụng sử dụng nó sẽ “thông minh” hơn trong việc “hiểu” và trả lời những kiến thức liên quan.

Vì sao cần phải xây dựng công cụ làm giàu Ontology tự động và bán tự động? Đó là vì kiến thức của một lĩnh vực rất lớn, lượng thông tin cần lưu trữ của một Ontology chuyên ngành để có thể sử dụng được cho các ứng dụng cũng phải thật phong phú và từ nhiều nguồn khác nhau. Với phương pháp nhập thủ công bởi con người thì sẽ tốn rất nhiều thời gian và chi phí. Vì vậy chúng ta cần có một công cụ để thu thập dữ liệu và nhập thông tin vào Ontology một cách tự động. Tuy nhiên, nếu làm tự động từ bước từ thập dữ liệu đến việc rút trích và lưu trữ thông tin vào Ontology thì độ chính xác không cao, do đó nhóm chúng em sẽ xây dựng công cụ làm giàu Ontology bán tự động. Có nghĩa là từ việc thu thập dữ liệu đến rút trích ra thông tin để lưu trữ sẽ thực hiện tự động, sau đó sẽ cho người dùng kiểm tra và xem xét lại kết quả trước khi lưu trữ vào Ontology. Như vậy sẽ giúp loại bỏ bớt một số kết quả sai, giảm bớt lượng thông tin rác trong Ontology.

Vì cấu trúc của mỗi Ontology khác nhau nên công cụ làm giàu này chỉ phục vụ cho việc làm giàu Ontology về công nghệ thông tin tiếng Việt (OVIT). Tuy nhiên, việc xây dựng công cụ làm giàu cho các Ontology khác có thể thực hiện tương tự.

Hình 7: Mô hình phương pháp làm giàu Ontology

Việc làm giàu Ontology có thể chia làm các phần nhỏ như: làm giàu các lớp, các cá thể, các quan hệ trong Ontology. Ở mức giới hạn của đề tài tốt nghiệp, nhóm chúng em chỉ dừng ở việc xây dựng công cụ làm giàu cá thể. Hướng phát triển sau này của nhóm là cải tiến giai đoạn rút trích để làm giàu các lớp và các quan hệ trong Ontology.

* + 1. **Nguồn dữ liệu**

Nhóm chúng em sẽ làm giàu Ontology OVIT từ nguồn dữ liệu là internet. Cụ thể sẽ đưa các từ khóa, chính là các khái niệm trong Ontology lên internet và dùng Google, Yahoo API để tìm kiếm những tài liệu có liên quan làm nguồn dữ liệu đầu vào cho công cụ làm giàu Ontology.

* + 1. **Phân loại**

Sau khi dùng Google và Yahoo API để tìm kiếm các tài liệu có liên quan đến từ khóa trong Ontology ta tiến hành phân loại các tài liệu đó. Phân loại văn bản là một tiến trình đưa các văn bản chưa biết chủ đề vào các lớp văn bản đã biết (tương ứng với các chủ đề hay lĩnh vực khác nhau). Mỗi lĩnh vực được xác định bởi một số tài liệu mẫu của lĩnh vực đó. Để thực hiện quá trình phân loại, các phương pháp huấn luyện được sử dụng để xây dựng bộ phân loại từ các tài liệu mẫu, sau đó dùng bộ phân loại này để dự đoán lớp của những tài liệu mới (chưa biết chủ đề). Trong quá trình phân loại, các văn bản được biểu diễn dưới dạng vector với các thành phần (chiều) của vector này là các đặc trưng của lớp.

Cụ thể ở bài toán phân loại này các tài liệu được phân thành 2 lớp là lớp tài liệu về công nghệ thông tin và lớp tài liệu không thuộc công nghệ thông tin. Ở đây nhóm chúng em sử dụng thuật toán SVM để phân loại văn bản.

SVM (Support Vector Machine) là một phương pháp học có giám sát bằng cách phân tích dữ liệu và nhận ra các mẫu được sử dụng để phân lớp và phân tích hồi quy (wikipedia). Phương pháp này do Vapnik cùng nhóm nghiên cứu của ông đề nghị ở phòng thí nghiệm AT&T Bell vào năm 1992 [25].

* + 1. **Các thuật toán và công cụ hỗ trợ phân loại tài liệu**
* **TF\*IDF:**

Đây là một phương pháp để đánh giá các thuật ngữ trong một tài liệu, là một cách định trọng số từ thông dụng. Ở đây ta sẽ dùng trọng số này để xác định các đặc trưng cho thuật toán phân loại SVM.

**Tần suất từ (term frequency - TF):** Trọng số từ là tần suất xuất hiện của từ đó trong tài liệu. Cách định trọng số này nói rằng một từ là quan trọng đối với một tài liệu nếu nó xuất hiện nhiều lần trong tài liệu đó.

**TFIDF:** Trọng số từ là tích của tần suất từ TF và tần suất tài liệu nghịch đảo của từ đó và được xác định bằng công thức

IDF = log(N / DF) + 1

trong đó:

N là kích thước của tập tài liệu huấn luyện;

DF là tần suất tài liệu: là số tài liệu mà một từ xuất hiện trong đó.

Trọng số TFIDF kết hợp thêm giá trị tần suất tài liệu DF vào trọng số TF. Khi một từ xuất hiện trong càng ít tài liệu (tương ứng với giá trị DF nhỏ) thì khả năng phân biệt các tài liệu dựa trên từ đó càng cao. Các từ được dùng để biểu diễn các tài liệu cũng thường được gọi là các đặc trưng. Để nâng cao tốc độ và độ chính xác phân loại, tại bước tiền xử lý văn bản, ta loại bỏ các từ không có ý nghĩa cho phân loại văn bản.

Tại bước này, chúng ta gặp phải một bài toán nữa đó là tách từ tiếng Việt. Như chúng ta đã biết, nếu như đối với tài liệu tiếng Anh mỗi từ sẽ mang một nghĩa của riêng nó do vậy việc tách từ không mấy khó khăn và hiện nay cũng có nhiều công cụ hỗ trợ tốt cho việc này ví dụ như Gate. Đối với tiếng Việt thì mỗi từ (2 từ cách nhau bằng 1 khoảng trắng) nếu đi với những từ khác nhau sẽ có nghĩa khác nhau, có khi 2 hoặc 3 từ mới tạo thành nghĩa. Sau một thời gian khảo sát và tìm kiếm, nhóm quyết định sử dụng công cụ hỗ trợ tách từ tiếng Việt **vnTokenizer** do nhóm của tác giả Lê Hồng Phương xây dựng được nêu trong bài báo [23].

* **Công cụ tách từ vnTokenizer:**

Đây là công cụ tách từ tự đông cho tiếng Việt được viết bằng ngôn ngữ Java và độc lập nền. Phiên bản cũ nhất hiện giờ còn được công bố trên website chính thức của tác giả [24] là phiên bản vnTokenizer 2.0 được xây dựng vào năm 2005 khi đó nó mới là một ứng dụng đơn với giao diện đơn giản. Và phiên bản nhóm sử dụng là phiên bản mới nhất hiện giờ được công bố chính thức trên website vào ngày 4/8/2010, phiên bản vnTokenizer 4.1.1c.

Công cụ này được xây dựng sử dụng kết hợp từ điển (từ điển tiếng Việt được lấy từ đề tài VLSP [28]) và giải thuật ngram, trong đó mô hình ngram được huấn luyện sử dụng treebank tiếng Việt (70,000 câu đã được tách từ). Treebank là kho ngữ liệu câu được chú giải ngữ pháp.

Với độ chính xác xấp xỉ 97% (theo thống kê của tác giả trên website) là kết quả rất cao so với công cụ tách từ hiện nay, nhóm chúng em sẽ sử dụng công cụ này cho công đoạn tiền xử lý tài liệu để rút ra các đặc trưng.

Sau khi tách từ, chúng ta tiến hành loại bỏ những hư từ trong tiếng Việt vì không những các từ này không có ý nghĩa gì đối với việc phân lớp mà nó còn có thể gây nhiễu cho việc tìm các đặc trưng. Danh sách hư từ tham khảo từ website [29] và từ đề tài “**Nghiên Cứu Phân Loại Văn Bản Tiếng Việt**” của Trịnh Quốc Sơn. Danh sách tổng hợp các hư từ được liệt kê trong bảng phụ lục B.

* **Công cụ LibSVM:**

Đây là một thư viện đơn giản, dễ sử dụng và hiệu quả đối với việc phân lớp bằng SVM. Thư viện này được tạo ra bởi hai tác giả Chih-Chung Chan và Chih-Jen Lin. Mục tiêu của nó là để giúp người dùng từ các lĩnh vực khác nhau dễ dàng sử dụng như một công cụ. LIBSVM cung cấp một giao diện đơn giản mà người sử dụng có thể dễ dàng liên kết nó với các chương trình riêng của họ. Phiên bản hiện tại của của Libsvm là 3.0 được công bố vào tháng 9 năm 2010 [26].

Để có thể tiến hành sử dụng công cụ trên, ta phải xây dựng một tập tin huấn luyện và một tập tin để test. Hai tập tin này đều có định dạng giống nhau và được trình bày như bên dưới:

|  |
| --- |
| <label 1> <index1>:<value1> <index2>:<value2> ...  <label 2> <index1>:<value3> <index3>:<value4> ... |

Trong đó:

* <label> là giá trị đích của tập huấn luyện. Đối với việc phân lớp, nó là một số nguyên xác định một lớp.
* <index> là một số nguyên bắt đầu từ 1. Cụ thể trong bài toán phân loại, nó đại diện cho các đặt trưng.
* <value> là một số thực. Giá trị này thể hiện mức độ liên quan của đặc trưng đối với một phân loại nằm trong khoảng [-1,1]. Nếu là đặc trưng nhị phân thì lúc huấn luyện giá trị này sẽ là 1.

Sau khi có được tập tin huấn luyện đúng định dạng, nhiệm vụ của libsvm là sẽ huấn luyện dựa trên tập tin định dạng và cho kết quả trả về là một tập tin *train\_model* có đuôi là *.model*. Tập tin này là mô hình xây dựng dựa trên việc huấn luyện. Từ đó, ta chỉ việc sử dụng lại mô hình này để dự đoán các dữ liệu kiểm thử. Quá trình đưa dữ liệu kiểm thử cũng giống như huấn luyện, vẫn phải xây dựng tập tin kiểm thử theo định dạng như trên.

* 1. **Rút trích**
  2. **Lưu trữ**

1. **HIỆN THỰC CÔNG CỤ LÀM GIÀU ONTOLOGY**
   1. **Kiến trúc chương trình**

Hình 8: Kiến trúc chương trình làm giàu Ontology (OVIT)

Chương trình gồm có hai quá trình lớn là quá trình *tìm kiếm đặc trưng và tạo file TRAIN cho SVM* và quá trình *làm giàu Ontology*. Trong hai quá trình, quá trình thứ nhất chỉ cần làm một lần, còn quá trình thứ hai cần làm nhiều lần. Bây giờ ta sẽ xét từng quá trình:

* Quá trình *tìm kiếm đặc trưng và tạo file train cho SVM*:
  + Giai đoạn tạo *tập huấn luyện*: đầu tiên ta cần phải tìm kiếm *tập huấn luyện*. *Tập huấn luyện* được chia làm hai phần: *phần thuộc công nghệ thông tin* và *phần không thuộc công nghệ thông tin*. Trong mỗi phần sẽ chứa danh sách các file text. Ta cần tìm các trang báo liên quan theo từng phần. Trong quá trình tìm kiếm, ta đã tìm được hơn một trăm trang báo cho mỗi phần.
  + Giai đoạn *tách từ và loại bỏ hư từ*: sau khi có được *tập huấn luyện*, ta load nội dung các file trong *tập huấn luyện* lên máy. Tiếp theo, ta sử dụng tokenizer để tách từ theo ngôn ngữ việt. Sau đó, ta sẽ loại bỏ các hư từ trong các nội dung đó (đã được tách từ). Cuối cùng, các nội dung này (đã được tách từ và loại bỏ hư từ) sẽ được lưu xuống tương ứng với tên các file.
  + Giai đoạn *tìm đặc trưng tài liệu công nghệ thông tin*: giai đoạn này ta có thể chia làm hai phân đoạn nhỏ.
    - Phân đoạn 1: ta sẽ tính TFIDF trong phần thuộc công nghệ thông tin của *tập huấn luyện.* Tiếp theo, ta chọn ra những từ có giá trị TFIDF lớn nhất làm đặc trưng (ở đây ta chọn 20 từ).
    - Phân đoạn 2: ta tính tần số xuất hiện của từng đặc trưng đó theo từng file thuộc công nghệ thông tin. Tiếp theo, ta lưu thành từng dòng với ký hiệu mỗi file là “1” và lưu lại thành chuỗi String gọi là chuỗi *cntt*. Tiếp đó, ta sẽ tính tần số xuất hiện của từng đặc trưng đó theo từng file không thuộc công nghệ thông tin của *tập huấn luyện*. Sau đó, ta cũng lưu thành dòng với ký hiệu mỗi file là “-1” và lưu lại thành chuỗi String gọi là chuỗi *không cnnt*. Sau cùng, ta ghép chuỗi *cnnt* và chuỗi *không cntt* lại rồi lưu xuống file gọi làfile *tfidf\_features.txt*.
  + Giai đoạn *SVM\_Train*: đầu vào của quá trình này là file *tfidf\_features.txt* và đầu ra là file *train\_model.txt*. File này sẽ được sử dụng ở bên dưới.
* Quá trình *làm giàu Ontology*:
  + Đầu tiên, chương trình bắt đầu từ *OVIT*: chương trình sẽ load hệ thống ontology đã tạo sẵn. Sau đó, người dùng sẽ chọn những từ khóa (là tên những khái niệm trong hệ thống Ontology) để đưa vào phần *tìm kiếm tài liệu liên quan*.
  + Trong phần *tìm kiếm tài liệu liên quan*: những từ khóa được chọn trong phần *OVIT* sẽ được đưa vào trong API của google hay yahoo để tìm kiếm những link liên quan đến chúng trên mạng. Qua giai đoạn này ta có được các link của các file html, pdf có liên quan đến những từ khóa. Bây giờ sang giai đoạn *HTML và PDF files*.
  + Giai đoạn *HTML và PDF files*: ở giai đoạn này nội dung của các link trên sẽ được tải xuống và lưu xuống máy dưới dạng file text (nếu một link dẫn đến một trang pdf thì chương trình sẽ down xuống dưới dạng pdf và chuyển nó sang dạng text). Giai đoạn này cần sử dụng gói htmlparser. Tất cả các file text có được sẽ được chứa trong folder *download\_files*.
  + Giai đoạn *tách từ và loại bỏ hư từ*: ta sẽ sử dụng API Tokenizer trong giai đoạn này. Đầu tiên, ta load nội dung các file trong folder *download\_files* lên máy. Tiếp theo, ta sử dụng tokenizer để tách từ theo ngôn ngữ việt. Sau đó, ta sẽ loại bỏ các hư từ trong các nội dung đó (đã được tách từ). Cuối cùng, các nội dung này (đã được tách từ và loại bỏ hư từ) sẽ được lưu xuống tương ứng với tên các file.
  + Giai đoạn *tính TFIDF theo đặc trưng*: ở giai đoạn này, ta sẽ sử dụng đặc trưng có được để tính TFIDF của từng file (là những file đã tách từ và loại bỏ hư từ). Tất cả các kết quả của chúng sẽ được lưu xuống một file text tên *tfidf\_download\_files.txt*.
  + Giai đoạn *phân loại bằng SVM*: trong giai đoạn này, ta sẽ dùng SVM để phân loại văn bản, kết quả của chúng là ta có thể lấy ra được nhưng file có liên quan đến các từ khóa. Đầu vào của giai đoạn *phân loại bằng SVM* là file *tfidf\_download\_files.txt* và *SVM\_Train*. Còn đầu ra của chúng là một file text *result.txt* cho chúng ta biết file nào được chọn, để từ đó chúng ta sẽ lưu lại các file được chọn này (*tài liệu công nghệ thông tin*).
  + Giai đoạn *rút trích cá thể*: đầu tiên ta sẽ sử dụng nhiều bài báo để có thể rút ra được các luật, ví dụ như “thì, mà, là”. Sau đó, ta sẽ dùng các luật này vào trong *tài liệu công nghệ thông tin* để rút ra được những định nghĩa và những thuộc tính liên quan đến các từ khóa.
  + Giai đoạn *cập nhật vào Ontology*: sau khi rút ra được những định nghĩa và những thuộc tính trên, ta mới cập nhật vào Ontology theo từ khóa.
  + Quá trình trên sẽ lập đi lặp lại càng nhiều lần thì hệ thống Ontology sẽ được làm giàu càng nhiều.

**Tài liệu tham khảo**

[1] **Lương Quý Tịnh Hà.** *Xây dựng công cụ tìm kiếm tài liệu học tập bằng các truy vấn ngôn ngữ tự nhiên trên kho học liệu mở tiếng Việt.*

[2] **Natalya F. Noy and Deborah L. McGuinness.** *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA, 94305

[3] **Lê Thành Nhân, Võ Trung Hùng, Cao Xuân Tuấn, Hoàng Thị Mỹ Lệ.** *MATHIS – Hệ thống hỗ trợ tạo chú thích và tìm kiếm tài liệu khoa học*. Tạp chí khoa học và công nghệ, Đại học Đà Nẵng - Số 4(39).2010

[4**] Trần Đình Khang, Vũ Tuyết Trinh, Đỗ Đức Thành, Đỗ Thị Ngọc Quỳnh**. *Một phương pháp tìm kiếm dựa trên Ontology phục vụ quản lý thông tin khoa học công nghệ.* Bộ môn Hệ thống Thông tin, Trường Đại Học Bách Khoa Hà Nội.

[5] **Phạm Thị Mỹ Phượng, Từ Thị Ngọc Thanh.** *Tìm kiếm ngữ nghĩa ứng dụng trên lĩnh vực eDoc.*

[6]*Tài liệu hướng dẫn phiên bản mã nguồn mở OVL – Open 1.0*

[7] **Nguyen Phi Minh Tri, Nguyen Tuan Dang.** *Building a Universal Ontology for Vietnamese Language.* Faculty of Computer Science, University of Information Technology.

[8] **Sean Bechhofer, Ian Horrocks, Daniele Turi.** *The OWL Instance Store: System Description*. Information Management Group, School of Computer Science,The University of Manchester

[9] **Bijan Parsia and Evren Sirin.** *Pellet: An OWL DL Reasoner.* MINDSWAP Research Group, University of Maryland, College Park, MD

[20] **Thomas R.Gruber.** *Toward Principles for the Design of Ontologies Used for Knowledge Sharing.* Stanford Knowledge Systems Laboratory, 701 Welch Road, Building C Palo Alto, CA 94304, [gruber@ksl.stanford.edu](mailto:gruber@ksl.stanford.edu).

[22] **Nhóm nghiên cứu của thầy Đỗ Phúc.** *Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lãnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt.*

[23] **L. H. Phuong, N. T.M. Huyen, R. Azim, H. T. Vinh.** *A hybrid approach to word segmentation of Vietnamese texts.* Proceedings of the 2nd International Conference on Language and Automata Theory and Applications, LATA 2008, Springer LNCS 5196, Tarragona, Spain, 2008.

[25] **B. E. Boser, I. M. Guyon, and V. N. Vapnik.** *A training algorithm for optimal margin classifiers.* In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144-152. Pittsburgh, PA, 1992.

[27] **Nguyễn Linh Giang, Nguyễn Mạnh Hiển .** *Phân loại văn bản tiếng Việt với bộ phân loại vectơ hỗ trợ SVM.*

[34] **Boanerges Aleman-Meza, Farshad Hakimpour, I. Budak Arpinar**. *SwetoDblp Ontology of Computer Science Publications.* LSDIS Lab, Computer Science Department, University of Georgia, Athens, GA.

**Website tham khảo:**

[10] [http://www.acm.org/](http://www.google.com/url?q=http%3A%2F%2Fwww.acm.org%2F&sa=D&sntz=1&usg=AFQjCNHRmO8pE9Aelibj1B2wMEZTwNJQeQ)

[11] <http://what.csc.villanova.edu/twiki/bin/view/Main/TheComputingOntology>

[12] <http://knoesis.wright.edu/library/ontologies/swetodblp/>

[13][http://www.acm.org/education/curricula-recommendations](http://www.google.com/url?q=http%3A%2F%2Fwww.acm.org%2Feducation%2Fcurricula-recommendations&sa=D&sntz=1&usg=AFQjCNExo5scqZGuscEVQvRcSj4VSbpJGQ) [2001 -- 2005 curriculum recommendations]

[16] [http://dblp.uni-trier.de/](http://www.google.com/url?q=http%3A%2F%2Fdblp.uni-trier.de%2F&sa=D&sntz=1&usg=AFQjCNEH92QzwG53nqoBMlHR8bkn8qQcKA)

[17] [http://xmlns.com/foaf/spec/](http://www.google.com/url?q=http%3A%2F%2Fxmlns.com%2Ffoaf%2Fspec%2F&sa=D&sntz=1&usg=AFQjCNHowKlBMu-6T-JqCFpsUX0XLD-Ysg)

[18] [http://dublincore.org/](http://www.google.com/url?q=http%3A%2F%2Fdublincore.org%2F&sa=D&sntz=1&usg=AFQjCNHc99lxIzXFMIljRA80fBMutr0soA)

[19] <http://www.w3.org/TR/rdf-sparql-query/>

[21] <http://protege.stanford.edu/>

[24] http://www.loria.fr/∼lehong/tools/vnTokenizer.php

[26] <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

[28] <http://vlsp.vietlp.org:8080/demo/?page=resources>

[29] <http://www.xulyngonngu.com/sharing/?p=145>

[30] <http://mic.gov.vn/Trang/default.aspx>

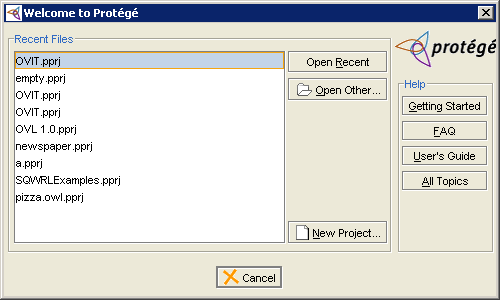
[33] <http://www.cs.wisc.edu/dbworld/>

**Phụ lục A: Hướng dẫn sử dụng Protégé**

Trong phần hướng dẫn sử dụng này sử dụng chương trình Protégé 3.4.4 với giao diện Protégé-OWL. Chúng ta sẽ cùng tìm hiểu cách để:

* Tạo và mở một ontology
* Lưu một ontology
* Tạo lớp và ràng buộc
* Tạo các thuộc tính và quan hệ
* Tạo cá thể

1. Tạo và mở một ontology:
2. Hiểu khái niệm Project trong Protégé: Khi vừa khởi động Protégé lên chúng ta sẽ thấy cửa sổ "Welcome to Protégé" hiện ra đâu tiên như hình sau:



Hình: Giao diện “Welcome to Protégé”

Khi dùng Protégé để tạo và chỉnh sửa ontology chúng ta sẽ tạo ra ít nhất là 2 file:

* File project (có đuôi là .pprj): Lưu trữ thông tin liên quan đến việc tùy biến giao diện hoặc tùy chọn của trình soạn thảo mà bạn cài đặt. Nếu không có file này chúng ta vẫn có thể tạo một file project khác cho một ontology từ file nguồn.
* File nguồn (có đuôi là .owl, .rdf hoặc .rdfs): đây là file chứa dữ liệu thật sự của ontology, nó chứa các lớp, cá thể và thuộc tính được định nghĩa.

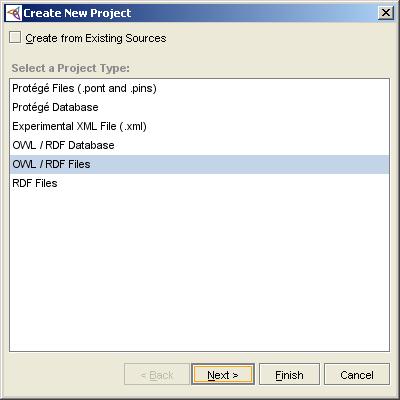
1. Mở một ontology có sẵn: có thể mở từ file project hoặc file nguồn.

Nếu có sẵn trên màn hình welcome thì nhấp đúp lên file ontology, hoặc chọn ontology muốn mở rồi bấm nút **Open Recent**.

Nếu không có sẵn trên màn hình welcome thì chọn nút **Open Other** và chọn ontology muốn mở.

1. Tạo một ontology mới:

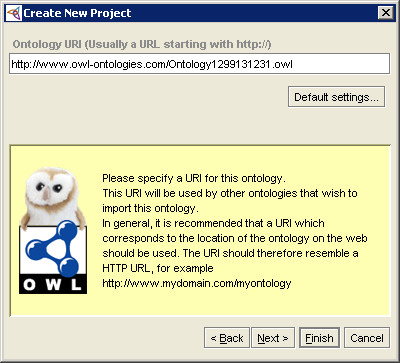
Từ giao diện welcome, chọn **New Project** hiển thị hộp thoại tạo project mới như hình sau:



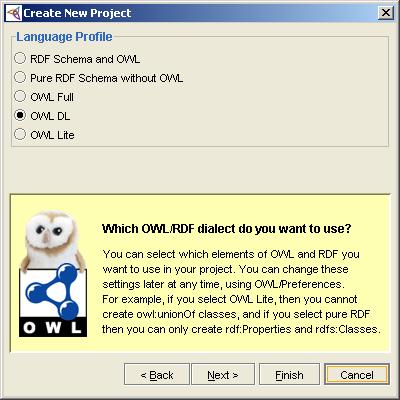
Hình: Hộp thoại tạo project mới

Chọn **OWL/RDF Files** và bấm **Next** **>**

Màn hình xuất hiện cho bạn xác nhận một URI cho ontology của mình. Thông thường, URI sẽ biểu diễn nơi mà chúng ta công bố ontology, tuy nhiên cũng không bắt buộc phải tuân theo. Việc thiết lập một URI duy nhất cho ontology sẽ giúp phòng những vấn đề sau này nếu ta nhập thêm những ontology khác.

  
Hình : Hộp thoại đặt URI cho ontology mới

Bấm **Next >**. Một hộp thoại xuất hiện cho phép chọn ngôn ngữ muốn dùng



Hình: Hộp thoại chọn ngôn ngữ xây dựng ontology

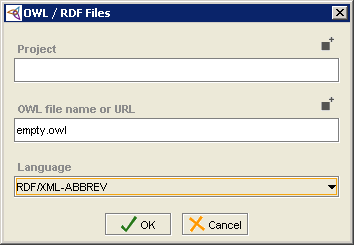
Bấm Finish để tạo ontology mới. Một hộp thoại xuất hiện để chúng ta có thể chọn cách hiển thị. Logic View là giao diện phù hợp hơn cho người dùng đã quen thuộc vì nó không được trực quan lắm, còn với người mới bắt đầu thì nên chọn Properties View vì nó có giao diện đơn giản hơn.

  
Hình: hộp thoại chọn cách hiển thị

Bấn Finish và một project mới được tạo sẵn sang để bạn có thể nhập và chỉnh sửa một ontology.

Nếu lúc mở Protégé lên mà hộp thoại welcome không xuất hiện thì ta chọn File | New Project

1. Lưu một ontology: bấm bút **Save** trên trình soạn thảo hoặc vào **File** chọn **Save Project.** Một hộp thoại xuất hiện để ta nhập tên ontology và tên project vào nếu như ta một project mới.

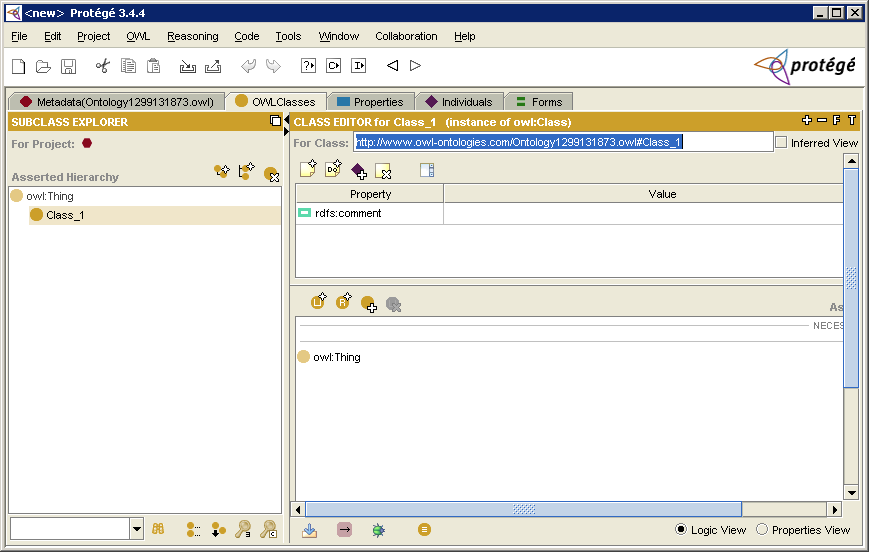


Hình : hộp thoại lưu ontology.

Thông thường tên project giống với tên của file OWL.

1. Tạo lớp:

Tại giao diện của Protégé ta chọn tab **OWL Classes,** mặc định mỗi ontology mới tạo có lớp cha là Thing. Để tạo một lớp mới ta chọn nút **Create subclass ** sẽ tạo ra lớp con tên Class\_1 như hình, ta có thể đổi tên ở textbox bên phải



Hình: tạo lớp mới trong ontology

Để tạo lớp con của lớp Class\_1 ta chọn nó rồi làm tương tự, hoặc nhấp chuột phải lên nó chọn **Create subclass.**

Để tạo lớp ngang hàng với một lớp ta chọn nó rồi chọn nút **Create Sibling Class **, hoặc nhấp chuột phải chọn **Create Sibling Class.**

Để tạo ra một cấu trúc cây của gồn nhiều lớp có lớp cha là Class\_1 ta nhấp phải vào nó chọn **Create subclasses**, sau đó một hộp thoại sẽ xuất hiện để ta nhập vào cấu trúc lớp với mỗi lớp là 1 dòng và lớp con thụt vào so với lớp cha. Ví dụ ta nhập:

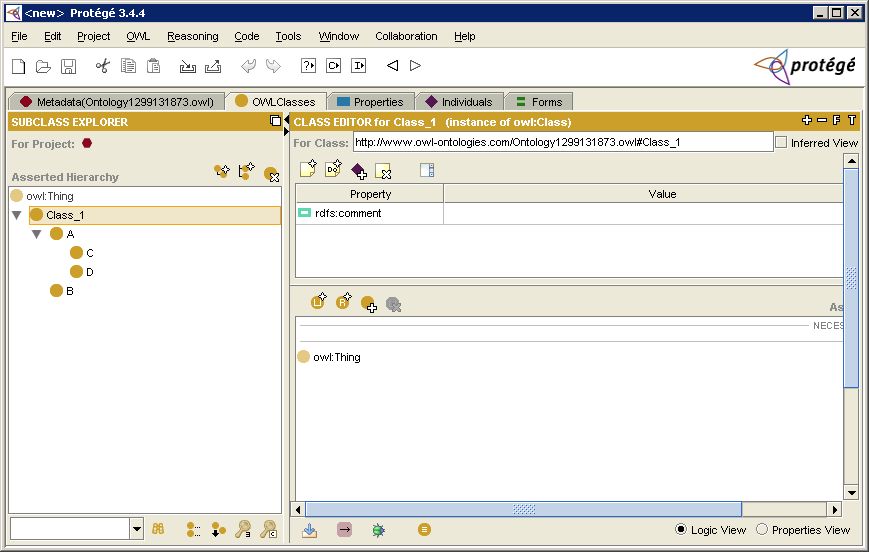
A

C

D

B

Ta sẽ tạo ra cấu trúc cây trong Class\_1 như sau:



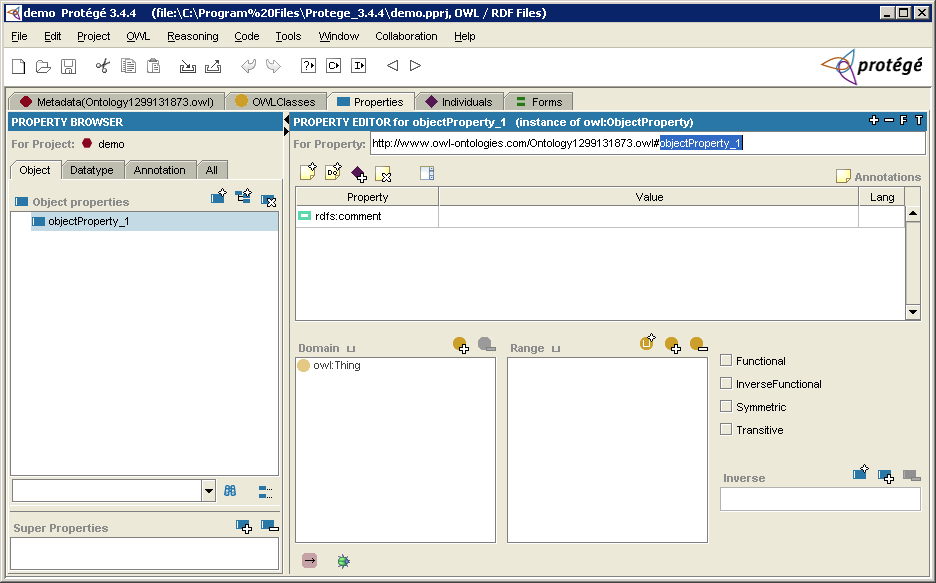
Hình: Tạo nhiều lớp trong ontology

Xóa một lớp ta chọn **Delete Class** . Nếu xóa lớp cha thì tất cả lớp con của nó đều bị xóa.

Ta có thể tạo các ràng buộc cho lớp dùng các nút có sẵn như **Create new expression** , **Create restriction**  và **Add Named Class**  để thêm lớp cha cho lớp đang chọn. Hoặc xóa đi các ràng buộc đã thêm dùng **Delete selected row .**

1. Tạo các thuộc tính và quan hệ:

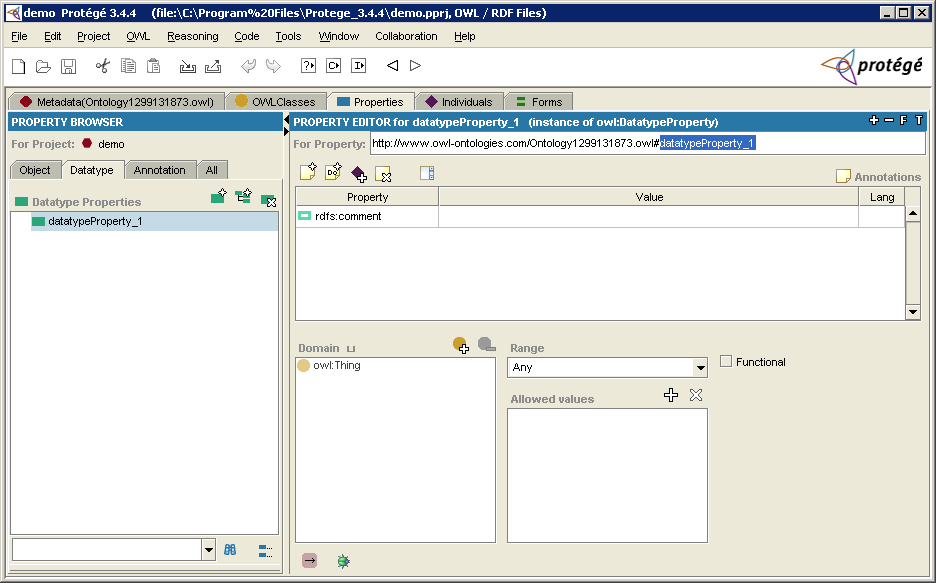
Tại giao diện Protégé ta chọn tab **Properties**. Trong đó ta chọn tab **Object** để thêm và chỉnh sửa các quan hệ trong ontology, chọn tab **Datatype** để thêm và chỉnh sửa các thuộc tính trong ontology.



Hình : Tạo quan hệ mới

Để tạo quan hệ mới ta cũng chọn nút **Create object property** , và đổi tên quan hệ ở textbox bên tay phải như tạo lớp. Đối với quan hệ ta chú ý đến **Domain** và **Range** có thể được chỉnh sửa và thêm ở bên phải. Và một số tính chất của quan hệ như: Functional, InverseFunctional, Symmetric, Transitive, ngoài ra ta có thể thêm một vào quan hệ nghịch đảo của một quan hệ bằng cách thêm tại textbox Inverse.

Để tạo thuộc tính mới ta chọn tab Datatype và cũng chọn nút **Create Datatype property** theo hình ở dưới

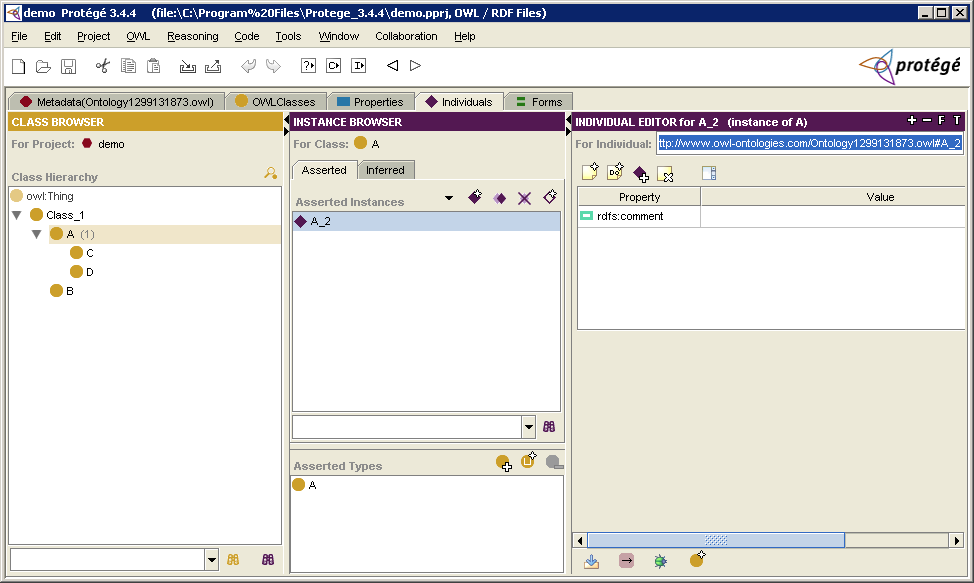


Hình : Tạo thuộc tính mới.

Đối với thuộc tính ta cũng có thể thay đổi **Domain** và **Range** cho nó bên phải màn hình. **Domain** sẽ xác định nó thuộc lớp nào, và **Range** sẽ xác định loại dữ liệu của nó. Ngoài ra, ta cũng có thể thêm vào một vài ràng buộc đơn giản cho nó như giới hạn một số giá trị được cho phép cho thuộc tính này.

1. Tạo cá thể

Để tạo cá thể ta chọn tab **Individuals,** chọn lớp mình muốn tạo cá thể rồi chọn nút **Create instance** ta sẽ tạo được cá thể như hình dưới. Các chức năng khác tương tự như trên.



Hình: Tạo cá thể mới

**Phụ lục B: Danh sách các hư từ**

|  |  |  |  |
| --- | --- | --- | --- |
| bao giờ | chúng mình | nhất định | thật ra |
| á | chúng nó | nhất loạt | thật vậy |
| à | chung qui | nhất luật | thẩy |
| ạ | chung quy | nhất mực | thế |
| á à | chung quy lại | nhất nhất | thế à |
| a ha | chúng ta | nhất quyết | thế đấy |
| a lô | chúng tao | nhất sinh | thế là |
| à ơi | chúng tôi | nhất tâm | thế mà |
| ạ ơi | có | nhất tề | thế nào |
| ai | cô | nhất thiết | thế nên |
| ái | cơ | nhau | thế nhưng |
| ai ai | cô ấy | nhé | thế ra |
| ái chà | có chăng là | nhỉ | thế thì |
| ái dà | cơ chừng | nhiên hậu | thếch |
| ai nấy | cỡ chừng | nhiệt liệt | theo |
| alô | có dễ | nhiều | thì |
| amen | cơ hồ | nhỡ ra | thì ra |
| áng | cổ lai | nhón nhén | thi thoảng |
| anh | cơ mà | như | thình lình |
| anh ta | cô mình | như chơi | thỉnh thoảng |
| ào | có vẻ | như không | thoắt |
| ắt | cóc khô | như quả | thoạt |
| ắt hẳn | coi bộ | như thể | thoạt nhiên |
| ắt là | coi mòi | như tuồng | thốc |
| âu là | con | như vậy | thộc |
| ầu ơ | còn | nhưng | thốc tháo |
| ấy | cơn | những | thôi |
| ba | công nhiên | những | thôi thì |
| bài | cứ | những ai | thỏm |
| bản | cu cậu | nhưng mà | thốt |
| bạn | cụ thể | nhung nhăng | thọt |
| bằng | cứ việc | những như | thốt nhiên |
| bằng ấy | của | nhược bằng | thửa |
| bằng không | của bạn | nó | thuần |
| bằng nấy | của mày | nọ | thục mạng |
| bao giờ | của tôi | nớ | thực ra |
| bao gồm | cực kì | nóc | thực vậy |
| bao lâu | cực kỳ | nổi | thúng thắng |
| bao nả | cực lực | nữa | thuộc |
| bao nhiêu | cùng | nức nở | thương ôi |
| bập bà bập bõm | cũng | ồ | tiện thể |
| bập bõm | cùng cực | ơ | tiếp đến |
| bất | cùng nhau | ớ | tiếp đó |
| bất chấp | cũng như | ờ | tiếp theo |
| bất chợt | cũng vậy | ở | tiếp tục |
| bất cứ | cũng vậy thôi | ô hay | tít mù |
| bắt đầu từ | cùng với | ơ hay | tớ |
| bất đồ | cuộc | ô hô | tỏ ra |
| bất giác | cuối | ô kê | tò te |
| bất kể | cuối cùng | ô kìa | tỏ vẻ |
| bất kì | cuốn | ơ kìa | toà |
| bất kỳ | dạ | oái | tốc tả |
| bất luận | đã | oai oái | toé khói |
| bất nhược | đại để | ôi | toẹt |
| bất quá | đại loại | ối | tôi |
| bất thình lình | đại nhân | ơi | tối ư |
| bất tử | đại phàm | ôi chao | tông tốc |
| bấy | dần dà | ối dào | tọt |
| bảy | dần dần | ối giời | tột |
| bây bẩy | đang | ối giời ơi | trái |
| bay biến | đáng lẽ | ôi thôi | tràn cung mây |
| bấy chầy | đáng lí | ối trời | trên |
| bây chừ | đáng lý | ôkê | trển |
| bấy chừ | đằng sau | ổng | trệt |
| bây giờ | đằng trước | phải | trếu tráo |
| bấy giờ | đành đạch | phải chăng | trệu trạo |
| bấy lâu | đánh đùng | phải chi | trời đất ơi |
| bấy lâu nay | dào | phăn phắt | trời ơi |
| bấy nay | đáo để | phắt | trong |
| bây nhiêu | dẫu | phè | trỏng |
| bấy nhiêu | đâu | phỉ phui | trong khi |
| bèn | dầu sao | pho | trong lúc |
| bên | dẫu sao | phóc | trừ khi |
| bển | đây | phốc | trừ phi |
| bên dưới | để | phỏng | trước |
| bên phải | dễ sợ | phỏng như | trước đây |
| bên trái | dễ thường | phót | trước đó |
| bên trên | đều | phương chi | trước khi |
| béng | do | phụt | trước kia |
| bệt | đó | phứt | trước lúc |
| bị | dở chừng | quá | trước nay |
| biết bao | do đó | quả | trước tiên |
| biết bao nhiêu | do vậy | quá chừng | từ |
| biết chừng nào | do vì | quá độ | tự |
| biết đâu | đồng thời | quá đỗi | tù tì |
| biết đâu chừng | dù | quả đúng | tự vì |
| biết đâu đấy | dữ | quả là | tuần tự |
| biết mấy | đủ | quá lắm | tức |
| bớ | dù cho | qua quít | tức khắc |
| bộ | dù là | qua quýt | tức là |
| bỏ mẹ | dù rằng | quá sá | tức thì |
| bởi | dù thế | quả tang | tức tốc |
| bởi nhưng | được | quả thật | từng |
| bội phần | dưới | quá thể | tuốt luốt |
| bởi thế | duy | quả tình | tuốt tuồn tuột |
| bởi vậy | gì | quá trời | tuốt tuột |
| bởi vì | giữa | quá ư | tựu trung |
| bốn | gồm | quả vậy | tuy |
| bông | hai | quá xá | tuy là |
| bỗng | hầu hết | quý hồ | tuy nhiên |
| bỗng chốc | Hay | quyển | tuy rằng |
| bỗng đâu | hãy | quyết | tuy thế |
| bỗng dưng | hiện nay | quyết nhiên | tuy vậy |
| bỗng không | họ | ra | tuyệt nhiên |
| bỗng nhiên | hoặc | ra phết | ư |
| bức | hoàn toàn | ra trò | ừ |
| cả | hỡi | răng | ử |
| cả thảy | hồi nãy | rằng | ứ hự |
| cả thể | hơn | rằng là | ứ ừ |
| các | ít | ráo | ủa |
| cái | kế tiếp | ráo trọi | úi |
| cái gì | khi | rất | úi chà |
| căn | khoảng | rất chi là | úi dào |
| cần | khoảng chừng | rất đỗi | và |
| căn cắt | không | rất mực | vả chăng |
| càng | là | rày | vả lại |
| cật lực | lại | rén | vài |
| cật sức | làm | ren rén | vẫn |
| cây | lẫn | rích | vẫn là |
| cha | lên | riêng | vạn nhất |
| cha chả | liên tiếp | riệt | vân vân |
| chắc | liên tục | riu ríu | vâng |
| chậc | lúc | rồi | văng tê |
| chắc hẳn | lúc ấy | rón rén | vào lúc |
| chầm chập | lúc trước | rốt cục | vậy |
| chăn chắn | luôn | rốt cuộc | vậy là |
| chăng | luôn luôn | rứa | vậy mà |
| chẳng | mà | rút cục | vậy nên |
| chẳng lẽ | mặc dù | sa sả | vậy thế |
| chẳng những | mặc kệ | sạch | vậy thì |
| chẳng nữa | mãi | sẵn sàng | vậy thôi |
| chẳng phải | mãi mãi | sao | về |
| chành chạnh | mặt khác | sắp | về mặt |
| chao ôi | mày | sất | về phía |
| chết nỗi | mầy | sau | veo |
| chết thật | mi | sáu | vèo |
| chết tiệt | mỗi | sau chót | veo veo |
| chỉ | mọi | sau cùng | vì |
| chị | một | sau cuối | ví bằng |
| chí chết | mười | sau đó | vì chưng |
| chỉ do | năm | sẽ | ví dù |
| chỉ là | này | sì | ví dụ |
| chỉ tại | nãy | số | ví phỏng |
| chỉ vì | nấy | sở dĩ | vị tất |
| chiếc | nè | số là | vì thế |
| chín | nên | song le | ví thử |
| chỉn | nền | sốt sột | vì vậy |
| chính | nên chi | sự | vở |
| chính anh | nếu | suýt | vô hình trung |
| chính chị | nếu như | tà tà | vô kể |
| chính là | ngăn ngắt | tại | vô luận |
| chính thị | ngay | tại vì | vô vàn |
| chính tôi | ngay cả | tám | với |
| cho | ngày càng | tấm | với lại |
| chớ | ngay khi | tăm tắp | vốn dĩ |
| chớ chi | ngay lập tức | tấn | vừa |
| cho đến | ngay lúc | tanh | vừa mới |
| cho đến khi | ngày ngày | tao | vung tán tàn |
| cho là | ngay từ | tắp | vung tàn tán |
| cho nên | ngay tức khắc | tắp lự | vung thiên địa |
| cho rằng | ngày xưa | tất cả | vụt |
| cho tới | ngày xửa | tất tần tật | xa xả |
| cho tới khi | nghe chừng | tất tật | xăm xăm |
| choa | nghe đâu | tất thảy | xăm xắm |
| chốc chốc | nghen | tênh | xăm xúi |
| chợt | nghĩa là | thà | xềnh xệch |
| chú | nghiễm nhiên | tha hồ | xệp |
| chứ | nghỉm | thà là | xiết bao |
| chu cha | ngõ hầu | thà rằng | xoẳn |
| chứ lị | ngộ nhỡ | thái quá | xoành xoạch |
| chú mày | ngoài | thậm | xoét |
| chú mình | ngoải | thậm chí | xoẹt |
| chưa | ngôi | than ôi | xon xón |
| chui cha | ngọn | thanh | xuất kì bất ý |
| chủn | ngọt | thành ra | xuất kỳ bất ý |
| chùn chùn | ngươi | thành thử | xuể |
| chùn chũn | nhân dịp | thảo hèn | xuống |
| chúng | nhân tiện | thảo nào | ý |
| chung cục | nhất | thật là | ý chừng |
| chúng mày | nhất đán | thật lực | ý da |