# LỜI CẢM ƠN

Con xin gửi lời cảm ơn sâu sắc nhất đến bố mẹ. Trong thời gian vừa qua, gia đình đã gặp những khó khăn, thử thách tưởng chừng không thể vừa qua. Ba mẹ đã gánh vác, lo liệu những điều khó khăn nhất, và tiếp thêm cho con sức mạnh tinh thần để hoàn thành luận văn này. Xin dành tặng luận văn này cho người mẹ thân yêu đã qua đời và người bố đáng kính đã hết lòng chăm sóc mẹ cũng như gia đình trong những lúc ngặt nghèo nhất.

Tôi xin gửi lời cảm ơn chân thành đến PGS TS Lê Hoài Bắc và PGS TS Ryutaro Ichise. Cảm ơn các thầy đã tận tình gợi mở, hướng dẫn các vấn đề liên quan đến luận văn. Tôi cũng xin gửi lời cảm ơn đến các thầy cô Khoa Công nghệ Thông tin, Trường ĐH Khoa học Tự nhiên đã hết lòng hướng dẫn tôi trong quá trình học tập cao học của trường. Xin cảm ơn các thầy cô trong Ban chủ nhiệm Khoa đã tạo điều kiện để tôi thực hiện chuyến thực tập tại Viện nghiên cứu Tin học Quốc gia Nhật Bản và cám ơn Viện đã cung cấp một môi trường vô cùng thuận lợi cho việc nghiên cứu đề tài luận văn của mình.

Tp Hồ Chí Minh, tháng 4/2009

# Mục lục

# Danh mục các ký hiệu, các chữ viết tắt

| | |
|---|---|
| DTD | Document Type Definition |
| EM | Expectation Maximization |
| ENB | Ehanced Naïve Bayes |
| LCS | Least Common Superconcept |
| Malfom | Machine Learning Framework for Ontology Matching |
| MalfomUI | Machine Learning Framework for Ontology Matching with User Interaction |
| OAEI | Ontology Alignment Evaluation Initiative |
| OWL | Ontology Web Language |
| RDF | Resource Description Framework |
| RDFS | RDF Schema |
| SGML | Standard General Markup Language |
| SVM | Support Vector Machine |
| W3C | World Wide Web Consortium |
| XML | eXtensible Markup Language |

# Danh mục các bảng

# Danh mục các hình vẽ, đồ thị

# MỞ ĐẦU

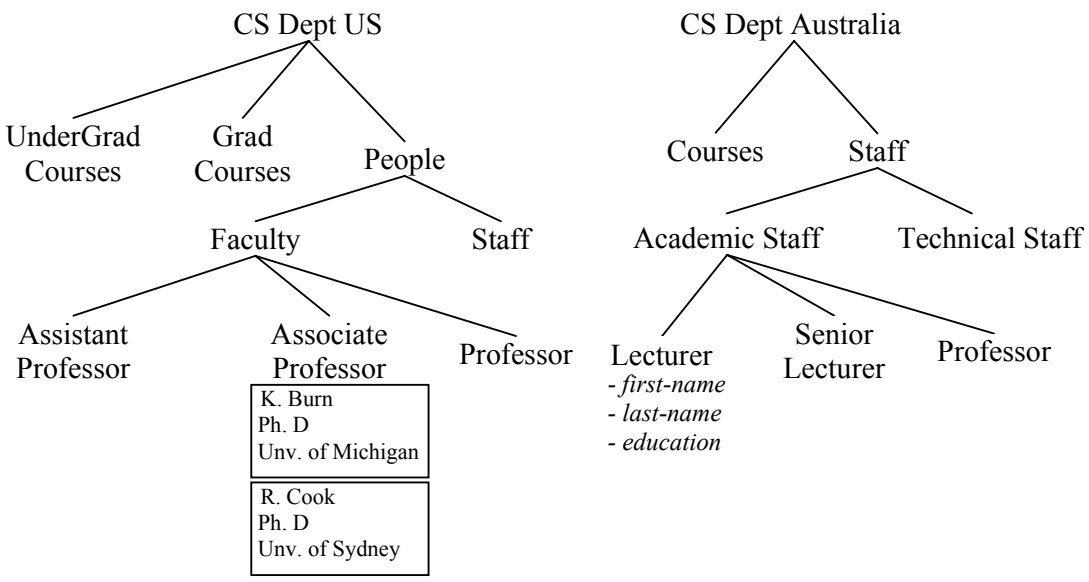
Sự tiến bộ vượt bậc của công nghệ trong thời gian gần đây đã tạo ra một lượng dữ liệu trực tuyến khổng lồ trên Internet. Lượng dữ liệu khổng lồ này biến Internet trở thành nguồn thông tin chủ yếu của con người. Người ta dựa vào Internet để tìm kiếm thông tin cho hầu hết các hoạt động của mình như việc làm, mua sắm, giải trí, du lịch… Tuy nhiên, chính nguồn dữ liệu khổng lồ này lại đang làm tiêu tốn nhiều công sức của con người để tìm kiếm thông tin thích hợp. Con người đang bị chìm ngập trong khối dữ liệu do chính mình tạo ra. Chính vì thế, người ta đang nghĩ đến việc sử dụng máy tính làm công cụ tự động thu thập thông tin trên Internet để phục vụ cho nhu cầu của mình. Tuy nhiên hầu hết các tài liệu trên Internet hiện giờ được lưu trữ theo định dạng mà chỉ có con người mới hiểu được nội dung. Do đó, ta cần một chuẩn web mới cho phép máy tính có thể hiểu và xử lý được dữ liệu trên Internet. Web ngữ nghĩa (*Semantic Web*) hay *Web 2.0* là một định nghĩa về thông tin mà máy tính có thể hiểu được, nhờ đó nó có thể thực hiện nhiều hơn những công việc buồn tẻ, chẳng hạn như tìm kiếm, chia sẻ và kết hợp thông tin trên web. Các công nghệ ngữ nghĩa đang thu hút sự quan tâm đáng kể [17]. Theo Gartner[1], các kỹ thuật ngữ nghĩa nằm trong danh sách mười công nghệ đột phá hàng đầu trong giai đoạn 2008 –2012.

Ontology là phương tiện cung cấp ngữ nghĩa cho dữ liệu trong môi trường web mới. Các ontology cho phép người sử dụng tổ chức thông tin theo các phân loại khái niệm, cùng với các thuộc tính của nó và mô tả các mối liên hệ giữa các khái niệm. Khi dữ liệu được đánh dấu với các ontology, các chương trình tìm kiếm tự động có thể hiểu được ngữ nghĩa của dữ liệu và do đó có thể định vị và thu thập dữ liệu một cách thông minh cho nhiều nhiệm vụ khác nhau. Chúng ta xét một ví dụ sau trích từ [7] để hiểu được viễn cảnh về web ngữ nghĩa.

[1] http://www.gartner.com/it/page.jsp?id=681107

*Ví dụ:* Giả sử giáo sư Henry ở Đại học Washington (Mỹ) muốn tìm hiểu thêm về một người đã gặp tại một hội nghị. Ông chỉ biết rằng tên của người này là Cook, dạy tại khoa "Computer Science" của một trường đại học gần đây nhưng ông không biết đó là trường nào. Giáo sư cũng biết rằng người ấy vừa mới từ Úc đến Mỹ và ông ta là một "associate professor" tại trường đại học đang công tác.

Trên web ngày nay chúng ta sẽ gặp rắc rối khi muốn tìm người này. Thông tin trên không chứa trong một trang web duy nhất, do đó tìm kiếm bằng từ khoá sẽ không hiệu quả. Ngược lại, trên web ngữ nghĩa, ta có thể nhanh chóng tìm câu trả lời. Một dịch vụ thư mục đánh dấu sẽ giúp chương trình tìm kiếm tìm ra những khoa "Computer Science" ở xung quanh trường. Những khoa này cũng có dữ liệu đánh dấu bằng cách dùng một ontology tương tự như trong Hình 0.1. Ở đây dữ liệu được tổ chức thành một cấu trúc phân loại bao gồm các "course", "people" và "professor". "Professor" có các thuộc tính như "name", "degree", "degree-grating institution". Những dữ liệu được đánh dấu như thế làm cho chương trình tìm kiếm dễ dàng tìm ra một "professor" với tên "Cook". Sau đó bằng cách kiểm tra tại thuộc tính "granting institution", chương trình tìm kiếm nhanh chóng tìm thấy khoa CS của trường đại học tại Úc. Ở đây, chương trình tìm kiếm tự động biết được rằng dữ liệu đã được đánh dấu dùng một ontology riêng của các đại học Úc, ví dụ như trong Hình 1.b và có nhiều thực thể có tên Cook. Tuy nhiên, biết được rằng "associate professor" tương đương với "senior lecturer", chương trình có thể lựa chọn nhánh đúng trong cấu trúc phân loại của khoa và mở trang chủ cũ của của người ta cần tìm hiểu.




*Hình 0.1. Ví dụ về các ontology của khoa "Computer Science" [7]*

Một trong những thách thức quan trọng đối với việc xây dựng web ngữ nghĩa là tìm các ánh xạ ngữ nghĩa giữa các ontology. Do bản chất không tập trung của sự phát triển web ngữ nghĩa, có một số lượng bùng nổ các ontology. Phần nhiều trong những ontology này cùng biểu diễn một lĩnh vực nhưng với các tên gọi khác nhau, hoặc các lĩnh vực khác nhau nhưng có sự chồng lấp về tên gọi. Để tích hợp dữ liệu từ những ontology không đồng nhất, chúng ta phải biết được những tương ứng ngữ nghĩa giữa các thành phần của chúng. Ví dụ, trong kịch bản về người quen tại hội nghị ở trên, để tìm đúng người, chương trình máy tính phải biết rằng "associate professor" ở Mỹ tương ứng với "senior lecture" ở Áo. Những tương ứng về ngữ nghĩa giúp liên kết các ontology lại với nhau và cũng giúp cho web ngữ nghĩa thật sự có "ngữ nghĩa". Việc đánh dấu thủ công các kết nối ngữ nghĩa thường tốn nhiều thời gian, chi phí, chứa nhiều lỗi và không khả thi khi số lượng ontology bùng nổ trên môi trường web. Do đó, việc phát triển các công cụ trợ giúp việc so khớp ontology có ý nghĩa quyết định cho sự thành công của Web ngữ nghĩa. [7]

Luận văn này giới thiệu một hệ thống học linh hoạt, sử dụng nhiều chiến lược học để xử lý tương tác người dùng cho bài toán so khớp ontology. Cấu trúc của luận văn được tổ chức như sau:

- Chương 1 giới thiệu một số kiến thức cơ bản về ontology, các thành phần của ontolgy, cách mã hoá ontology dựa trên ngôn ngữ web cùng với các ví dụ.
- Chương 2 trình bày một cách hình thức về bài toán so khớp ontology cùng ứng dụng, các kỹ thuật cơ bản và chiến lược so khớp ontology. Phần này cũng giới thiệu thông tin về tổ chức đánh giá các hệ thống ontology và quan trọng là vấn đề tương tác người dùng trong ontology. Đây là bài toán mà luận văn đề xuất ra mô hình để giải quyết.
- Chương 3 trình bày một số kiến thức cơ sở về học máy bao gồm hai loại học có giám sát và học bán giám sát. Phần cuối của chương giới thiệu về một số nghiên cứu về học máy trong bài toán so khớp ontology. Đây là những công trình có liên quan đến nghiên cứu của luận văn.
- Chương 4 trình bày về hệ thống học máy được đề xuất. Hệ thống này sử dụng nhiều chiến lược học khác nhau để đáp ứng với nhiều môi trường người dùng thực tế. Phần đầu của chương trình bày về việc xây dựng vector tương tự cho bài toán so khớp. Phần thứ hai mô tả chi tiết hệ thống học máy.
- Chương 5 trình bày về các thử nghiệm của hệ thống cùng kết quả và những nhận xét, thảo luận trên kết quả đạt được. Phần cuối của chương trình bày kết luận và hướng phát triển của luận văn.

# Chương 1 ONTOLOGY

## 1.1 Định nghĩa

Trong khoa học máy tính và thông tin, ontology được định nghĩa là một biểu diễn hình thức cho tập hợp các khái niệm thuộc một lĩnh vực nào đó và quan hệ giữa những khái niệm này. Nói cụ thể hơn, ontology cung cấp một bộ từ vựng chung dùng để mô tả một lĩnh vực – nghĩa là một loại đối tượng hay khái niệm hiện hữu, cùng với các thuộc tính và quan hệ giữa chúng – và lời đặc tả cho nghĩa của những từ trong bộ từ vựng. Dựa vào độ chính xác của đặc tả này, khái niệm ontology bao gồm một số mô hình dữ liệu hay mô hình khái niệm, ví dụ, các bảng phân loại (*classifications)*, từ điển chuyên đề *(thesauri)*, lược đồ cơ sở dữ liệu *(database schemas)*, lý thuyết được tiên đề hoá đầy đủ *(fully axiomatized theories)*, v.v… Ontology có khuynh hướng xuất hiện ở mọi nơi. Ontology được sử dụng trong các lĩnh vực trí tuệ nhân tạo, web ngữ nghĩa, kỹ thuật phần mềm, sinh-y tin học, khoa học thư viện và kiến trúc thông tin như là một dạng biểu diễn tri thức về thế giới hay một phần của nó. Ontology là một giải pháp đơn giản nhưng hiệu quả cho nhiều ứng dụng như tích hợp thông tin, các hệ thống ngang hàng, thương mại điện tử, các dịch vụ web ngữ nghĩa, các mạng xã hội, v.v… Chúng thực sự là những phương tiện thiết thực để khái niệm hoá những thứ cần được biểu diễn theo định dạng của máy tính.

## 1.2 Các thành phần của ontology

Các ontology hiện nay đều có nhiều điểm tương tự về mặt cấu trúc, bất kể ngôn ngữ được dùng để biểu diễn. Hầu hết các ontology đều mô tả các đối tượng (thể hiện), lớp (khái niệm), thuộc tính và các quan hệ.

### 1.2.1 Cá thể

*Cá thể (hay thể hiện)* là thành phần cơ bản, "mức nền" của một ontology. Các cá thể trong một ontology có thể bao gồm các đối tượng rời rạc như con

người, con thú, xe, nguyên tử, hành tinh, trang web, cũng như các đối tượng trừu tượng như con số và từ (mặc dù có một vài khác biệt về ý kiến liệu các con số và từ là lớp hay là đối tượng). Nói đúng ra, một ontology không cần chứa bất cứ cá thể nào, nhưng một trong những mục đích chung của ontology là cung cấp một phương tiện để phân loại các đối tượng, ngay cả khi các đối tượng này không phải là một phần rõ ràng của ontology.

### 1.2.2 Lớp

*Lớp – khái niệm* – có thể được định nghĩa theo cách bên ngoài hay bên trong. Theo định nghĩa bên ngoài, chúng là những nhóm, bộ hoặc tập hợp các đối tượng. Theo định nghĩa bên trong, chúng là các đối tượng trừu tượng được định nghĩa bởi giá trị của các mặt ràng buộc khiến chúng phải là thành viên của một lớp khác. Lớp có thể phân loại các cá thể, các lớp khác, hay một tổ hợp của cả hai. Một số ví dụ của lớp:

- *Person*, lớp của tất cả con người, hay các đối tượng trừu tượng có thể được mô tả bởi các tiêu chuẩn làm một con người.
- *Vehicle*, lớp của tất cả xe cộ, hay các đối tượng trừu tượng có thể được mô tả bởi các tiêu chuẩn làm một chiếc xe.
- *Car*, lớp của tất cả xe hơi, hay các đối tượng trừu tượng có thể được mô tả bởi các tiêu chuẩn làm một chiếc xe hơi.
- *Class*, biểu diễn lớp tất cả các lớp, hay các đối tượng trừu tượng có thể được mô tả bởi các tiêu chuẩn để làm một lớp.
- *Thing*, biểu diễn lớp tất cả mọi thứ, hay các đối tượng trừu tượng có thể được mô tả bởi các tiêu chuẩn để làm một thứ gì đó (và không phải không-là-gì cả).

Một lớp có thể gộp nhiều lớp hoặc được gộp vào lớp khác; một lớp xếp gộp vào lớp khác được gọi là *lớp con* (hay *kiểu con*) của lớp gộp (hay *kiểu cha*). Ví dụ, *Vechicle* gộp *Car*, bởi vì bất cứ thứ gì là thành viên của lớp sau cũng đều là thành viên của lớp trước. Quan hệ xếp gộp được dùng để tạo nên một cấu trúc phân cấp các lớp, thông thường có một lớp tổng quát lớn nhất chẳng hạn

*Anything* nằm ở trên cùng và những lớp rất cụ thể như *2002 Ford Explorer* nằm ở dưới cùng. Hệ quả cực kỳ quan trọng của quan hệ xếp gộp là tính kế thừa của các thuộc tính từ lớp cha đến lớp con. Do vậy, bất cứ thứ gì hiển nhiên đúng với một lớp cha cũng hiển nhiên đúng với các lớp con của nó. Trong một số ontology, một lớp chỉ được cho phép có một lớp cha, nhưng trong hầu hết các ontology, các lớp được cho phép có một số lượng lớp cha bất kỳ và trong trường hợp sau tất cả các thuộc tính hiển nhiên của từng lớp cha được kế thừa bởi lớp con. Do đó một lớp cụ thể của lớp thú (*HouseCat*) có thể là một con của lớp Cat và cũng là một con của lớp *Pet*.

### 1.2.3 Thuộc tính

Các đối tượng trong một ontology có thể được mô tả bằng cách liên hệ chúng với những thứ khác, thường là các mặt hay bộ phận. Những thứ được liên hệ này thường được gọi là *thuộc tính*, mặc dù chúng có thể là những thứ độc lập. Một thuộc tính có thể là một lớp hay một cá thể. Kiểu của đối tượng và kiểu của thuộc tính xác định kiểu của quan hệ giữa chúng. Một quan hệ giữa một đối tượng và một thuộc tính biểu diễn một sự kiện đặc thù cho đối tượng mà nó có liên hệ. Ví dụ đối tượng *Ford Explorer* có các thuộc tính như:

- <có tên> Ford Explorer
- <có bộ phận> *door* (với số lượng tối thiểu và tối đa: 4)
- <có một trong các bộ phận> *{4.0L engine, 4.6L engine}*
- <có bộ phận> *6-speed transmission*

Giá trị thuộc tính có thể thuộc kiểu dữ liệu phức; trong ví dụ này, động cơ liên hệ chỉ có thể là một trong số các dạng con của động cơ, chứ không phải là một cái đơn lẻ.

Các ontology chỉ mang đầy đủ ý nghĩa nếu các khái niệm có liên hệ với các khái niệm khác (các khái niệm đều có thuộc tính). Nếu không rơi vào trường hợp này, thì hoặc ta sẽ có một *phân loại* (nếu các quan hệ bao hàm tồn tại giữa các

khái niệm) hoặc một *từ điển có kiểm soát*. Những thứ này đều hữu ích nhưng không được xem là ontology.

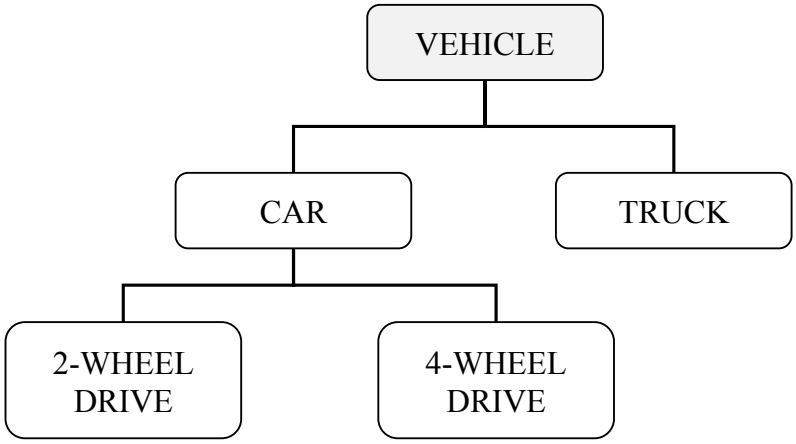
### 1.2.4 Quan hệ

Quan hệ giữa các đối tượng trong một ontology cho biết các đối tượng liên hệ với đối tượng khác như thế nào. Thông thường một quan hệ là của một loại (hay lớp) cụ thể nào đó chỉ rõ trong ngữ cảnh nào đối tượng được liên hệ với đối tượng khác trong ontology. Ví dụ trong ontology chứa khái niệm *Ford Explorer* và khái niệm *Ford Bronco* có thể được liên hệ bởi một quan hệ loại <được định nghĩa là một con của>. Phát biểu đầy đủ của sự kiện như sau:

- Ford Explorer *được định nghĩa là một con của* : Ford Bronco

Điều này cho ta biết *Explorer* là mô hình thay thế cho *Bronco*. Ví dụ này cũng minh họa rằng quan hệ có cách phát biểu trực tiếp. Phát biểu ngược biểu diễn cùng một sự kiện nhưng bằng một ngữ nghịch đảo trong ngôn ngữ tự nhiên.

Phần lớn sức mạnh của ontolgy nằm ở khả năng diễn đạt quan hệ. Tập hợp các quan hệ cùng nhau mô tả ngữ nghĩa của domain. Tập các dạng quan hệ được sử dụng (lớp quan hệ) và cây phân loại thứ bậc của chúng thể hiện sức mạnh diễn đạt của ngôn ngữ dùng để biểu diễn ontology.




*Hình 1.1. Một ontology biểu diễn quan hệ của xe cộ*

*superclass-of*, hay ngược lại, ‘là dạng con của’ – *is-a-subtype-of* – hay ‘là lớp con của’ – *is-a-subclass-of*). Nó định nghĩa đối tượng nào được phân loại bởi lớp nào.

Ví dụ, ta đã thấy lớp *Ford Explorer* là lớp con của *4-Wheel Drive Car* và lớp *4-Wheel Drive Car* lại là lớp con của *Car*.

Sự xuất hiện của quan hệ 'là lớp con của' tạo ra một cấu trúc phân cấp thứ bậc; dạng cấu trúc cây này (hay tổng quát hơn, là tập có thứ tự từng phần) mô tả rõ ràng cách thức các đối tượng liên hệ với nhau. Trong cấu trúc này, mỗi đối tượng là 'con' của một 'lớp cha' (Một số ngôn ngữ giới hạn quan hệ là lớp con của trong phạm vi một cha cho mọi nút, nhưng đa số thì không như thế).

Một dạng quan hệ phổ biến khác là quan hệ *meronymy*, gọi là 'bộ phận của', biểu diễn làm thế nào các đối tượng kết hợp với nhau để tạo nên đối tượng tổng hợp. Ví dụ, nếu ta mở rộng ontology trong ví dụ để chứa thêm một số khái niệm như *Steering Wheel* (vô lăng), ta sẽ nói rằng "Vô lăng được định nghĩa là một bộ phận của *Ford Explorer*" vì vô lăng luôn luôn là một trong những bộ phận của xe *Ford Explorer*. Nếu đưa quan hệ *meronymy* vào ontology này, ta sẽ thấy rằng cấu trúc cây đơn giản và nhẹ nhàng trước đó sẽ nhanh chóng trở nên phức tạp và cực kỳ khó hiểu. Điều này không khó lý giải; một lớp nào đó được mô tả rằng luôn luôn có một thành viên là bộ phận của một thành viên thuộc lớp khác thì lớp này cũng có thể có một thành viên là bộ phận của lớp thứ ba. Kết quả là các lớp có thể là bộ phận của nhiều hơn một lớp. Cấu trúc này được gọi là đồ thị chu trình có hướng.

Ngoài những quan hệ chuẩn như 'là lớp con của' và 'được định nghĩa là bộ phận của', ontology thường chứa thêm một số dạng quan hệ làm trau chuốt hơn ngữ nghĩa mà chúng mô hình hóa. Ontology thường phân biệt các nhóm quan hệ khác nhau. Ví dụ nhóm các quan hệ về:

- Quan hệ giữa các lớp
- Quan hệ giữa các thực thể
- Quan hệ giữa một thực thể và một lớp
- Quan hệ giữa một đối tượng đơn và một tập hợp
- Quan hệ giữa các tập hợp.

Các dạng quan hệ đôi khi đặc thù chuyên ngành và do đó chỉ sử dụng để lưu trữ các dạng sự kiện đặc thù hoặc trả lời cho những loại câu hỏi cụ thể. Nếu định nghĩa của dạng quan hệ được chứa trong một ontology thì ontology này định ra ngôn ngữ định nghĩa ontology cho chính nó. Một ví dụ về ontology định nghĩa các dạng quan hệ của chính nó và phân biệt các nhóm quan hệ khác nhau là ontology Gellish.

Ví dụ, trong lĩnh vực xe ô tô, ta cần quan hệ 'được sản xuất tại' để cho biết xe được lắp ráp tại chỗ nào. Như vậy, Ford Explorer được sản xuất tại Louisville. Ontology có thể cũng biết được Louisville 'tọa lạc tại' Kentucky và Kentucky 'được định nghĩa là' một bang và 'là bộ phận của' Hoa Kỳ. Phần mềm sử dụng ontology này sẽ có thể trả lời một câu hỏi như 'những xe hơi nào được sản xuất tại Hoa Kỳ?"

## 1.3 Mã hoá các ontology

Các ontology thường được mã hoá bằng những ngôn ngữ ontology. Ngôn ngữ ontology là ngôn ngữ hình thức cho phép mã hoá tri thức về một lĩnh vực cụ thể và thường cũng bao gồm luôn các luật suy diễn hỗ trợ việc xử lý tri thức đó. Có nhiều ngôn ngữ ontology được sử dụng. Các ngôn ngữ này có thể thuộc về nhóm các ngôn ngữ ontology "truyền thống" được sử dụng rộng rãi trong cộng đồng nghiên cứu ontology hoặc những ngôn ngữ ontology "dựa trên web" (Ontology Web Language – OWL) xuất hiện trong ngữ cảnh của Internet và được khuyến cáo sử dụng bởi W3C (World Wide Web Consortium) [5]. Trong nghiên cứu này, luận văn tập trung xử lý trên các ngôn ngữ mã hoá dựa trên web. Nhóm này bao gồm XML, RDF và RDFS.

*XML* viết tắt của eXtensible Markup Language được suy dẫn từ SGML (Standard General Markup Language). Nó được phát triển bởi XML Working Group thuộc W3C và sắp tới sẽ trở thành một ngôn ngữ chuẩn. Là một ngôn ngữ cho World Wide Web, những ưu điểm chính của nó là: dễ dàng phân tích, cú pháp được định nghĩa tốt và con người có thể đọc được. Có khá nhiều phần mềm phân tích và thao tác với XML. Nó cho phép người dùng định nghĩa các nhãn và

thuộc tính riêng của mình, định nghĩa cấu trúc, rút trích dữ liệu từ các tài liệu và phát triển các ứng dụng để kiểm tra tính hợp lệ về mặt cấu trúc của một tài liệu.

Khi dùng XML làm cơ sở cho một ngôn ngữ đặc tả ontology, các ưu điểm chính của nó là:

- Định nghĩa đặc tả cú pháp chung bằng DTD (*Document Type Definition)*.
- Con người có thể đọc được dữ liệu mã hoá bằng XML dễ dàng.
- Có thể được sử dụng để biểu diễn tri thức phân tán giữa một số trang web vì nó có thể được nhúng trong các trang web.

XML cũng có vài điểm bất lợi ảnh hưởng đến đặc tả ontology:

- Nó được định nghĩa để cho phép sự thiếu cấu trúc của thông tin bên trong các nhãn XML. Điều này làm cho việc tìm kiếm các thành phần của một ontology bên trong một tài liệu trở nên khó khăn.
- Các công cụ chuẩn đang có sẵn là để phân tích cú pháp và thao tác trên các tài liệu XML nhưng không thể thực hiện việc suy diễn. Cần phải tạo ra công cụ suy diễn với ngôn ngữ dựa trên XML.

Bản thân XML không có những đặc trưng đặc biệt nào để đặc tả các ontology, nó chỉ đưa ra một phương thức đơn giản nhưng mạnh mẽ để mô tả cấu trúc cho một ngôn ngữ đặc tả ontology. Bên cạnh đó, nó có thể được sử dụng để quán xuyến các yêu cầu trao đổi, khai thác điều kiện liên lạc thuận lợi của WWW.

*RDF* viết tắt của Resource Description Framework. Nó được phát triển bởi W3C để tạo siêu dữ liệu mô tả các tài nguyên Web. Có một sự quan hệ mạnh mẽ giữa RDF và XML. Trên thực tế, chúng được định nghĩa bổ sung cho nhau: một trong những mục đích của RDF là khả thi hoá việc đặc tả ngữ nghĩa cho dữ liệu dựa trên XML theo một cách thức chuẩn hoá và có thể xử lý tổng quát được. Mục tiêu của RDF là định nghĩa một cơ chế mô tả các tài nguyên mà không cần đưa ra giả định nào về lĩnh vực ứng dụng hoặc cấu trúc cụ thể của một tài liệu chứa thông tin. Mô hình dữ liệu của RDF (dựa trên các mạng ngữ nghĩa) bao gồm ba loại: các tài nguyên (các đối tượng), các thực thể có thể được tham chiếu

tới bởi một địa chỉ trong WWW; các thuộc tính (các vị từ), định nghĩa các khía cạnh, đặc điểm cụ thể, các thuộc tính hay các quan hệ dùng để mô tả một tài nguyên; và một phát biểu (đối tượng) gán một giá trị cho một thuộc tính trong một tài nguyên cụ thể.

*RDFS* (RDF Schema)[3] là một ngôn ngữ khai báo dùng để định nghĩa lược đồ RDF. Đây là một ngôn ngữ biểu diễn tri thức có thể mở rộng, cung cấp các thành phần cơ bản để mô tả các ontology. Mô hình dữ liệu RDFS cung cấp cơ chế để định nghĩa những mối quan hệ giữa các thuộc tính và tài nguyên. Các lớp lõi là *class*, *resource* và *property*; các cấu trúc phân cấp và các ràng buộc kiểu có thể được định nghĩa (các thuộc tính lõi là *type*, *subclassOf*, *subPropertyOf*, *seeAlso* và *isDefinedBy*). Một số ràng buộc cũng có thể được định nghĩa.

Bảng 2.1 trình bày một đoạn mã hoá ontology với RDFS. Hai thành phần RDFS chính được dùng trong đoạn mã hoá này:

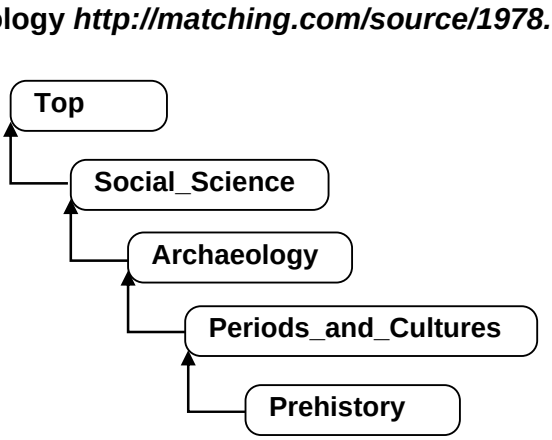
- *rdf:Class* cho phép định nghĩa một tài nguyên như là một lớp hoặc khái niệm cho các tài nguyên khác.
- *rdf:subClassOf* cho phép định nghĩa cấu trúc phân cấp giữa các lớp.

*Bảng 1.1. Một đoạn mã hoá ontology bằng RDFS*

```
<?xml version="1.0"?>
<!DOCTYPE owl [...]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" ...>
<owl:Ontology rdf:about="http://matching.com/source/1978.owl"/>
<owl:Class
rdf:about="http://matching.com/source/1978.owl#Archaeology">
  <rdfs:subClassOf>
    <owl:Class
rdf:about="http://matching.com/source/1978.owl#Social_Sciences">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class
rdf:about="http://matching.com/source/1978.owl#Periods_and_Cultures">
  <rdfs:subClassOf>
```

```
    <owl:Class
rdf:about="http://matching.com/source/1978.owl#Archaeology">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://matching.com/source/1978.owl#Prehistory">
  <rdfs:subClassOf>
    <owl:Class
rdf:about="http://matching.com/source/1978.owl#Periods_and_Cultures">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://matching.com/source/1978.owl#Science">
  <rdfs:subClassOf>
    <owl:Class rdf:about="http://matching.com/source/1978.owl#Top">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class
rdf:about="http://matching.com/source/1978.owl#Social_Sciences">
  <rdfs:subClassOf>
    <owl:Class
rdf:about="http://matching.com/source/1978.owl#Science">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://matching.com/source/1978.owl#Top">
</owl:Class>
</rdf:RDF>
```

Đoạn mã RDFS trong Bảng 1.1 ở trên khai báo cho một ontology có tên "http://matching.com/source/1978.owl". Các thành phần *rdf:Class* được dùng để khai báo các lớp. Ví dụ lớp đầu tiên được khai báo là "Archaeology " là con của lớp "Social_Science" (định nghĩa bằng thành phần *rdf:subClassOf*). Lớp gốc trong ontology là "Top", là lớp trên cùng và không có lớp cha nào. Toàn bộ cấu trúc của ontology trên được minh hoạ trong Hình 1.2.




*Hình 1.2. Cấu trúc ontology tương ứng với đoạn mã ví dụ*

## 1.4 Tóm tắt

Chương này đã giới thiệu một số kiến thức cơ bản về ontology và các vấn đề liên quan. Trong nền tảng Web ngữ nghĩa mới, ontology đóng một vai trò quan trọng do đây là phương tiện giúp cung cấp ngữ nghĩa cho các trang web. Do đó các nghiên cứu về ontology cần thiết được hoàn thiện để phục vụ cho nhu cầu của chuẩn web mới và nó đang thu sự quan tâm rộng lớn từ giới nghiên cứu. Kỹ nghệ ontology là lĩnh vực mới trong khoa học máy tính và khoa học thông tin, nghiên cứu các phương pháp và phương pháp luận để xây dựng các ontology. Mục tiêu của nó nhằm làm rõ nghĩa các tri thức chứa đựng trong một lĩnh vực cụ thể. Kỹ nghệ ontology đưa ra một phương hướng nhắm tới việc giải quyết các vấn đề hoạt động tương tác xuất hiện bởi các rào cản ngữ nghĩa, các rào cản liên quan đến các định nghĩa của các thuật ngữ hay các khái niệm. Kỹ nghệ ontology là một tập hợp các nhiệm vụ liên quan đến việc phát triển các ontology cho một lĩnh vực cụ thể. Bài toán so khớp ontology là một trong những nhiệm vụ như thế. Chương 2 kế tiếp sẽ trình bày về bài toán so khớp ontology và những kỹ thuật có liên quan.

# Chương 2 BÀI TOÁN SO KHỚP ONTOLOGY

Như đã trình bày trong phần trên, ontology thực sự là một giải pháp hữu hiệu trong việc tổ chức và chia sẻ thông tin trong kỷ nguyên mới. Tuy nhiên, trong các hệ thống mở và tiến hoá, ví dụ như web, những nhóm khác nhau nói chung thường dùng những ontology khác nhau. Do đó chỉ sử dụng ontology, cũng giống như việc chỉ sử dụng XML, không làm giảm sự hỗn tạp của thông tin: nó làm nảy sinh ra một sự hỗn tạp mới ở mức cao hơn. Sự không đồng nhất giữa các ontology có thể xảy ra do một hoặc nhiều nguyên nhân sau đây:

- Các nhóm làm việc ở các quốc gia khác nhau nên sử dụng các ngôn ngữ khác nhau.
- Các nhóm sử dụng những thuật ngữ khác nhau trong cùng một ngôn ngữ để biểu diễn các khái niệm.
- Các nhóm sử dụng các mô hình ontology khác nhau theo mục đích, lĩnh vực chuyên môn của mình,…

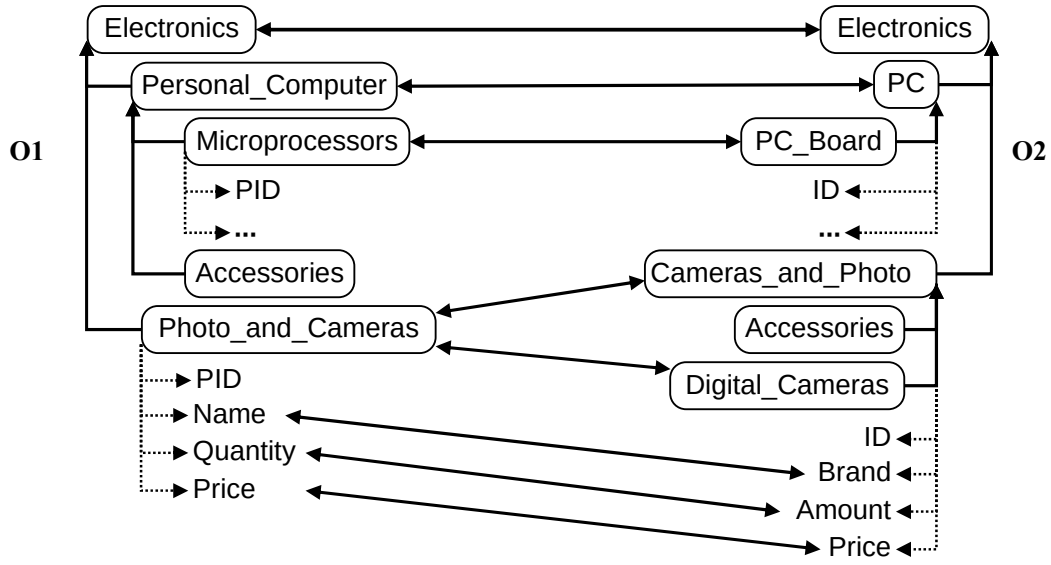
So khớp ontology là một cách tiếp cận hợp lý cho vấn đề hỗn tạp về mặt ngữ nghĩa. Thao tác so khớp nhận hai ontology, chứa một tập các thực thể rời rạc (ví dụ các lớp, các thuộc tính, các bảng, các phần tử XML,…), làm đầu vào và xác định kết quả là các quan hệ (ví dụ, quan hệ tương đương, quan hệ gộp) thoả giữa những thực thể này. Những ánh xạ này có thể được dùng cho nhiều mục đích khác nhau. Phần đầu trong chương này sẽ giới thiệu một ví dụ minh hoạ cho bài toán so khớp ontology. Phần kế tiếp giới thiệu phát biểu hình thức của một bài toán so khớp ontology cùng một số kỹ thuật xử lý cơ bản. Phần tiếp theo giới thiệu một số kiến thức tổng quan về kỹ thuật xử lý và chiến lược so khớp. Trong hai phần cuối, một số vấn đề liên quan đến bài toán so khớp đặc biệt vấn đề tương tác người dùng sẽ được trình bày.

## 2.1 Bài toán Ví dụ

Để làm ví dụ mở đầu cho bài toán so khớp ontology, giả sử xét hai lược đồ XML đơn giản như trong Hình 2.1, đây là một dạng cụ thể của ontology, mỗi phần tử được gán nhãn đại diện cho một khái niệm (hay lớp).

Giả sử một công ty thương mại điện tử cần thu mua một công ty khác. Về mặt kĩ thuật, hai công ty này cần hợp nhất cơ sở dữ liệu của các bên. Tài liệu của cả hai công ty đều lưu dưới dạng lược đồ XML, gọi tuần tự là O1 và O2. Bước đầu tiên để hợp nhất là xác định các ứng viên để trộn lại hoặc có quan hệ thứ bậc trong lược đồ hợp nhất. Bước này liên quan đến quá trình so khớp. Ví dụ, các phần tử có nhãn *Price* trong O1 và O2 là những ứng viên cần được trộn lại, trong khi đó, phần tử có nhãn *Digital_Cameras* trong O2 vào cần được sắp xếp vào nhóm có nhãn *Photo_and_Cameras* trong O1. Khi xác định được mối quan hệ tương ứng giữa hai lược đồ, bước kế tiếp cần phát sinh, chẳng hạn như, các câu truy vấn tự động dịch các thực thể dữ liệu của hai lược đồ sang lược đồ hợp nhất.

Nhìn vào ví dụ trên, dễ dàng nhận ra bài toán so khớp ontology không phải là bài toán với lời giải tầm thường. Trong khi các phần tử có nhãn như *Electronics* và *Price* được chia sẻ ở cả O1 và O2, hoặc cặp phần tử có nhãn *Personal_Computer* và *PC* có thể được nhận biết trùng khớp một cách trực quan dễ dàng, việc xác định trùng khớp giữa các cặp phần tử có nhãn *Name* và *Brand*, *Quantity* và *Amount* hay *Microprocessors* và *PC_Board* là không đơn giản. Nó có thể cần một chút kiến thức về ngữ nghĩa. Việc so khớp có thể còn gặp khó khăn do sự khác biệt về mặt cấu trúc, ví dụ trong trường hợp cả phần tử có nhãn *Cameras_and_Photo* và phần tử con *Digital_Cameras* trong O2 đều cùng có thể ánh xạ vào phần tử có nhãn *Photo_and_Cameras* trong O1. Cuối cùng, ta cũng để ý đến trường hợp phần tử có nhãn *Accessories* cùng xuất hiện trong O1 và O2 nhưng trong ngữ cảnh này rõ ràng không phải là một cặp ứng viên để trộn lại. Như vậy, việc so khớp ontology đòi hỏi nhiều kỹ thuật xử lý hơn việc chỉ so sánh chuỗi thông thường. Phần tiếp theo sẽ đưa ra một phát biểu hình thức cho bài toán so khớp ontology.




*Hình 2.1. Hai lược đồ XML đơn giản cần so khớp*

## 2.2 Phát biểu Bài toán

**Định nghĩa 3.1 (Tương ứng - *Correspondence*) [9]:**

*Cho hai ontology O và O', một **tương ứng** là bộ năm ⟨id, $e_1$, $e_2$, n, r⟩, trong đó*

- *id là định danh đơn nhất của tương ứng đang xét;*
- *$e_1$ và $e_2$ lần lượt là thực thể (ví dụ, bảng, phần tử XML, tính chất, lớp, khái niệm…) của O và O';*
- *r là quan hệ (ví dụ, tương đương (=), tổng quát hơn (⊇), rời nhau (⊥),… ) giữa $e_1$ và $e_2$.*
- *n là độ tin cậy theo một cấu trúc toán học nào đó (thông thường trong đoạn [0,1]);*

*Tương ứng ⟨id, $e_1$, $e_2$, n, r⟩ khẳng định mối quan hệ r giữa hai thực thể ontology $e_1$ và $e_2$ với độ tin cậy n. Độ tin cậy càng cao, quan hệ càng có khả năng xảy ra.*

Như trong ví dụ ở phần trước, theo một số thuật toán so khớp dựa trên phân tích cấu trúc và ngôn ngữ học, độ tin cậy (để quan hệ tương đương xảy ra) giữa

các thực thể có nhãn *Photo_and_Cameras* trong *O1* và *Cameras_and_Photo* trong *O2* có thể là 0.67. Giả sử thuật toán so khớp sử dụng ngưỡng 0.55 để xác định phép so khớp, tức là thuật toán xem mọi cặp thực thể có độ tin cậy lớn hơn 0.55 là tương ứng đúng. Như thế, thuật toán so khớp sẽ trả về cho người dùng tương ứng sau: ⟨$id_{3,3}$, *Photo_and_Cameras*, *Cameras_and_Photo*, 0.67, =⟩. Quan hệ giữa cặp thực thể giống nhau cũng có thể được xác định theo một cách khác, ví dụ quan hệ tương đương chính xác giữa hai thực thể (không cần phải tính độ tin cậy). Do đó, kết quả trả về cho người dùng trong trường hợp này là ⟨$id_{3,3}$, *Photo_and_Cameras*, *Cameras_and_Photo*, n/a, =⟩

**Định nghĩa 3.2 (So khớp – Alignment) [9]:**

*Cho hai ontology O và O', một* ***so khớp*** *A giữa O và O' là:*

- *Một tập hợp các tương ứng giữa O và O'.*
- *Một lực lượng nào đó: 1-1, 1-*,…*
- *Một số siêu dữ liệu bổ sung nào đó (ví dụ ngày tháng, thuộc tính,…)*

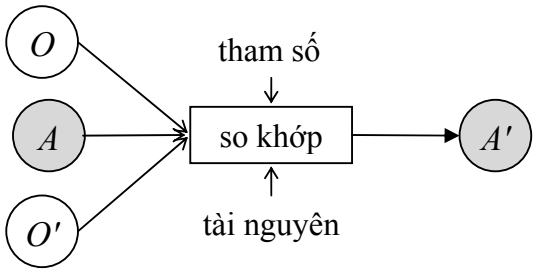
Thao tác so khớp xác định phép so khớp *A'* cho cặp ontology *O* và *O'*, mỗi ontology là một tập các thực thể rời rạc, chẳng hạn như lớp, thuộc tính, hay thực thể. Hình 2.2 minh hoạ cho quá trình so khớp tổng quát. Ở đây, một số tham số mở rộng định nghĩa quá trình so khớp là: (*i*) việc sử dụng *so khớp đầu vào A*, là cái sẽ được hoàn chỉnh trong quá trình thực hiện so khớp, tham số đầu vào này sẽ được trình bày thêm trong phần tương tác người dùng; (*ii*) các *tham số* so khớp, ví dụ, trọng số, ngưỡng; và (*iii*) *tài nguyên* bên ngoài mà quá trình so khớp sử dụng đến, ví dụ tri thức phổ biến và từ điển chuyên ngành. Định nghĩa hình thức của một bài toán so khớp như sau:

**Định nghĩa 3.3 (Quá trình so khớp) [9]:**

*Quá trình so khớp có thể xem như một hàm f mà nó nhận vào một cặp ontology cần so khớp O và O', một so khớp đầu vào, một tập các tham số p và*

*một một tập tài nguyên và tri thức r, trả ra một so khớp A' giữa các ontology này:*

$$A' = f(O, O', A, p, r)$$




*Hình 2.2. Sơ đồ biểu diễn quá trình so khớp ontology [9]*

## 2.3 Ứng dụng của So khớp ontology

So khớp ontology là một tác vụ quan trọng trong các ứng dụng truyền thống, ví dụ phát triển ontology, tích hợp ontology, tích hợp dữ liệu, tích hợp các lược đồ và cất giữ dữ liệu (*data warehouse*). Thông thường, các ứng dụng này được đặc trưng hoá bởi các mô hình có cấu trúc hỗn tạp mà chúng sẽ được phân tích và so khớp hoặc bằng tay hay bán tự động vào thời điểm thiết kế. Trong những ứng dụng như thế, so khớp là điều kiện tiên quyết để chạy hệ thống thực tế.

Hiện đang có một số ứng dụng nổi lên có thể bởi tính động của chúng, ví dụ chia sẻ thông tin ngang hàng, tích hợp dịch vụ web, giao tiếp đa tác nhân, trả lời truy vấn và duyệt web ngữ nghĩa. Các ứng dụng như thế, ngược với những ứng dụng truyền thống, đòi hỏi thao tác so khớp ngay trong lúc thực thi và có ưu điểm là mô hình có tính khái niệm rõ ràng hơn.

## 2.4 Các kỹ thuật Cơ bản

Mục đích của so khớp là tìm mối quan hệ giữa những thực thể được biểu diễn trong các ontology khác nhau. Những mối quan hệ này thường là quan hệ tương đương nghĩa được khám phá qua độ đo tương tự giữa các thực thể trong các ontology.

**Định nghĩa 3.4 (Độ tương tự) [9]**. *Một độ tương tự* $\sigma : o \times o \rightarrow \mathrm{R}$ *là ánh xạ từ một cặp thực thể thành một số thực biểu diễn sự tương tự giữa hai đối tượng sao cho*

$$\forall x, y \in o,\ \sigma(x, y) \geq 0 \qquad \textit{(dương)}$$
$$\forall x \in o,\ \forall y, z \in o,\ \sigma(x, x) \geq \sigma(y, z) \qquad \textit{(tính tối đại)}$$
$$\forall x, y \in o,\ \sigma(x, y) = \sigma(y, x) \qquad \textit{(đối xứng)}$$

Dựa vào độ tương tự được sử dụng, các kỹ thuật có thể được chia thành bốn cách tiếp cận: tên gọi, khái niệm, mở rộng và ngữ nghĩa. Có một điểm lưu ý là các kỹ thuật này không thể dùng một cách đơn lẻ, mà mỗi cái trong chúng phải tận dụng các kết quả được cung cấp bởi những kỹ thuật khác. Một phần của nghệ thuật so khớp ontology nằm ở chỗ lựa chọn và kết hợp các phương pháp này theo cách thích hợp nhất. Các mục dưới đây giới thiệu một số nội dung cơ bản của các kỹ thuật này.

### 2.4.1 Các kỹ thuật dựa trên tên

Một số phương pháp dựa trên tên so sánh các chuỗi. Chúng có thể được áp dụng đối với tên, nhãn hay các chú thích của thực thể để tìm những cặp tương tự. Nó có thể dùng để so sánh tên lớp và/hay các URI (Uniform Resource Identifier). Khó khăn chính trong việc so sánh các thực thể ontology dựa trên cơ sở nhãn của chúng là do sự tồn tại của từ đồng nghĩa và từ đồng âm. Các từ đồng nghĩa là các từ khác nhau dùng để đặt tên cho cùng một thực thể. Ví dụ, “Article” và “Paper” là các từ đồng nghĩa trong một số ngữ cảnh nào đó. Các từ đa nghĩa là các từ dùng để đặt tên cho các thực thể khác nhau. Ví dụ, “peer” là một danh từ có nghĩa là “người ngang hàng” cũng có một nghĩa khác là “người quý tộc”. Việc một từ có thể có nhiều nghĩa còn được gọi là tính đa nghĩa. Kết quả là, không thể không thể suy luận chắc chắn là hai thực thể tương tự nhau nếu chúng có cùng tên hay chúng khác nhau bởi vì chúng có tên khác nhau.

Có hai loại phương pháp chính để so sánh các tên dựa vào việc chúng chỉ quan tâm đến chuỗi ký tự hay chúng dùng một số tri thức ngôn ngữ để hiểu những chuỗi này.

*Các phương pháp dựa trên chuỗi* tận dụng cấu trúc của chuỗi (là một chuỗi các ký tự). Ví dụ, trong tiếng Anh, dựa vào *tiền tố* hệ thống có thể xác định sự tương tự của "net" và "network" cũng như "book" với "textbook" nếu dựa vào *hậu tố*. Tuy nhiên, cách này có thể xác định sai sự tương tự của những từ như "hot" và "hotel" hoặc bỏ qua các cặp quan hệ như "book" và "volume". Một số độ đo tương tự tiêu biểu cho loại kỹ thuật này là:

- Độ tương tự dựa trên *edit distance* (như khoảng cách Hamming, khoảng cách Levenshtein)
- Độ tương tự chuỗi con
- Độ tương tự n-gram
- Độ đo Jaro và độ đo Jaro-Winkler
- Độ tương tự cosine
- TFIDF (*Term frequency-Inverse Document frequency*)
- Độ tương tự dựa trên khoảng cách đường đi

Các phương pháp so sánh chuỗi là hữu ích nếu người ta dùng các chuỗi rất tương tự nhau để biểu thị những khái niệm giống nhau. Nếu các từ đồng nghĩa với các cấu trúc khác nhau được dùng, việc này đưa đến độ tương tự thấp. Lựa chọn các cặp chuỗi với độ tương tự thấp có thể đưa đến các kết quả sai bởi vì hai chuỗi có thể rất tương tự nhau nhưng dùng biểu diễn những khái niệm khá khác biệt. Các độ đo này thường dùng để phát hiện hai chuỗi rất tương tự có được dùng hay không. Nếu không, việc so khớp phải dùng các nguồn thông tin đáng tin cậy hơn. Một số gói phần mềm để tính toán khoảng cách chuỗi là: Simetrics, SecondString, Alignment API, SimPack.

*Các phương pháp dựa trên ngôn ngữ* dùng các kỹ thuật xử lý ngôn ngữ tự nhiên để giúp rút trích các từ ngữ có ý nghĩa từ văn bản. So sánh những từ ngữ này và quan hệ của chúng có thể giúp đánh giá độ tương tự giữa các thực thể

ontology mà chúng đặt tên hoặc chú thích. Mặc dù những phương pháp này dựa trên ngôn ngữ, chúng ta có thể phân biệt chúng dựa là chỉ dựa trên thuật toán hay dùng thêm các tài nguyên bên ngoài như các từ điển. Một số độ tương tự được dùng trong các phương pháp này là:

- Độ tương tự đồng nghĩa
- Độ tương tự *cosynonymy*
- Độ tương tự ngữ nghĩa Resnik
- Độ tương tự lý thuyết thông tin
- Độ chồng lấp của chú thích

### 2.4.2 Các kỹ thuật dựa trên cấu trúc

Cấu trúc của các thực thể có thể được dùng trong các ontology được so khớp, bên cạnh việc so sánh tên hay định danh của chúng. Sự so sánh này có thể được chia thành so sánh cấu trúc bên trong một thực thể, nghĩa là ngoài tên và nhãn là các thuộc thuộc tính hay, trong trường hợp của OWL ontology, các thuộc tính mà sẽ nhận giá trị trong một kiểu dữ liệu, hoặc so sánh thực thể với các thực thể khác mà chúng có quan hệ. Loại đầu tiên là cấu trúc nội bộ và loại thứ hai được gọi là cấu trúc quan hệ.

*Các phương pháp dựa trên cấu trúc nội bộ* dựa vào cấu trúc bên trong các thực thể và dùng các tiêu chuẩn như tập các thuộc tính, miền giá trị, tính hữu hạn hay vô hạn và tính bắc cầu hay đối xứng của các thuộc tính để tính toán độ tương tự giữa chúng. Một số độ đo được dùng trong các phương pháp này là:

- Khoảng cách kích thước tương đối
- Độ tương tự bội số

*Các phương pháp dựa trên cấu trúc quan hệ* sử dụng tập các quan hệ mà thực thể có với các quan hệ khác. Trong phương pháp này, một ontology có thể được xem là một đồ thị với các đỉnh được gán nhãn bởi các tên quan hệ (nói theo toán học, đây là đồ thị của các đa quan hệ của ontology). Việc tìm các tương ứng giữa các phần tử của các đồ thị như thế tương đương với việc giải một dạng của

bài toán đẳng cấu đồ thị. Cụ thể là nó có thể được liên hệ với việc tìm một đồ thị con chung tối đại. Một số độ đo của phương pháp này bao gồm:

- Độ không tương tự topology cấu trúc trong các cấu trúc phân cấp
- Độ tương tự Wu & Palmer
- Độ tương tự cotopic hướng lên

### 2.4.3 Các kỹ thuật mở rộng

Việc có sẵn các biểu diễn cá thể (hay thể hiện) là cơ hội rất tốt cho các hệ thống so khớp . Khi hai ontology có chung một tập các cá thể, sự so khớp có thể trở nên dễ dàng hơn. Ví dụ, nếu hai lớp có chính xác cùng một tập các cá thể, thì có thể có một giả định mạnh rằng những lớp này biểu diễn cho một so khớp đúng. Ngay cả khi các lớp không có chung tập cá thể, những phương pháp này cũng cho phép đặt quá trình so khớp trên những chỉ số xác thực không dễ gì thay đổi. Ví dụ "title" của "Book" không có lý do gì để thay đổi. Nên nếu "title" của "Book" là khác nhau, thì hầu như chắc chắn chúng không phải là như nhau. Khi đó, việc so khớp một lần nữa có thể dựa trên việc so sánh cá thể. Do đó các phương pháp mở rộng được chia thành ba loại: những phương pháp áp dụng với các ontology có các tập thể hiện chung, những phương pháp áp dụng kỹ thuật nhận diện thể hiện trước khi dùng những kỹ thuật mở rộng và những phương pháp không cần việc nhận diện.

*Các phương pháp so sánh mở rộng chung* đơn giản thực hiện kiểm tra phần giao của các thể hiện giữa hai tập. Một số độ đo của phương pháp này là:

- Khoảng cách Hamming
- Độ tương tự Jaccard

*Các kỹ thuật nhận diện thể hiện* cố gắng nhận diện thể hiện nào từ một tập là tương ứng với thể hiện khác từ tập khác, nếu giữa hai tập không tồn tại một tập thể hiện con chung. Phương pháp này hữu ích khi biết rằng các thể hiện là như nhau. Ví dụ, phương pháp này hoạt động được khi tích hợp cơ sở dữ liệu nhân sự

của cùng công ty, nhưng không áp dụng được với những công ty khác nhau hay cơ sở dữ liệu của các sự kiện mà chúng không có quan hệ nào cả.

*Các phương pháp so sánh mở rộng rời nhau* sử dụng các kỹ thuật xấp xỉ để so sánh các mở rộng lớp khi không thể trực tiếp suy luận ra một tập dữ liệu chung giữa hai ontology. Các phương pháp này có thể dựa trên độ đo thống kê về các đặc trưng của các thành viên lớp, dựa trên độ tương tự được tính giữa các thể hiện hay các lớp hoặc dựa vào việc so khớp giữa các tập thực thể. Một số độ đo của các phương pháp này là:

- Khoảng cách Hausdorff
- Độ tương tự dựa trên trùng khớp

### 2.4.4 Các kỹ thuật dựa trên ngữ nghĩa

Đặc điểm chính của những phương pháp ngữ nghĩa là dùng các ngữ nghĩa theo lý thuyết mô hình để đánh giá các kết quả. Do đó chúng là các phương pháp suy diễn. Dĩ nhiên, các phương pháp suy diễn nếu chỉ đơn thuần hoạt động một mình sẽ không có nhiều hiệu quả đối với nhiệm vụ suy diễn cơ bản như so khớp ontology. Do đó chúng cần một bước tiền xử lý cung cấp các điểm neo, ví dụ các thực thể được khai báo là tương đương (dựa vào việc nhận diện theo tên hay do người dùng nhập vào đối với thể hiện). Các phương pháp ngữ nghĩa đóng vai trò như bộ khuếch đại cho những so khớp hạt giống này. Các phương pháp này dựa trên việc sử dụng tài nguyên hình thức đang có để khởi tạo một so khớp mẫu mà chúng có thể được xem xét sâu hơn. Các kỹ thuật này bao gồm *các kỹ thuật dựa trên các ontology bên ngoài* và *các kỹ thuật suy diễn*.

## 2.5 Các Chiến lược So khớp

Các kỹ thuật cơ bản được giới thiệu ở phần trên là các khối cơ bản mà dựa trên đó người ta xây dựng lời giải so khớp. Khi độ tương tự giữa các thực thể ontology đã sẵn sàng, phần còn lại là tính so khớp. Việc này liên quan nhiều hơn đến các giải pháp toàn cục. Cụ thể, việc xây dựng một hệ thống so khớp hoạt động thường gồm các mặt sau:

- Tổng hợp các kết quả của các phương pháp cơ bản để tính độ tương tự phức giữa các thực thể và tổ chức tổ hợp các độ tương tự hay các thuật toán so khớp khác nhau.
- Phát triển một chiến lược để tính những độ tương tự này bất chấp các chu trình và sự phi tuyến trong các ràng buộc bao trùm các độ tương tự.
- Học từ dữ liệu phương pháp tốt nhất và các tham số tốt nhất để so khớp.
- Sử dụng các phương pháp xác suất để kết hợp các chương trình so khớp hay để suy diễn ra các tương ứng còn thiếu.
- Đưa người dùng vào quá trình so khớp.
- Rút trích các so khớp từ các độ tương tự kết quả: thực vậy, các so khớp với các đặc điểm khác nhau có thể được rút ra từ cùng một độ tương tự.

Luận văn này tập trung vào các khía cạnh học máy và tương tác người dùng trong so khớp ontology. Nội dung về tương tác người dùng sẽ được giới thiệu ở phần sau và nội dung về học máy sẽ được trình bày trong Chương 3.

## 2.6 Ontology Alignment Evaluation Initiative

Bài toán so khớp ontology đã và đang nhận được sự quan tâm rộng rãi trong thời gian gần đây. Số lượng các phương pháp và hệ thống hiện có ngày càng gia tăng. Điều đó đặt ra yêu cầu thiết lập một sự thống nhất trong việc đánh giá các hệ thống. *Ontology Alignment Evaluation Innitiative* (OAEI)[2] là một sáng kiến mang tính hợp tác quốc tế nhằm thúc đẩy cho sự thống nhất này.

Mục tiêu của OAEI bao gồm:

- Đánh giá điểm mạnh và điểm yếu của các hệ thống sắp xếp/so khớp.
- So sánh hiệu quả của các kỹ thuật.
- Gia tăng tính cộng đồng giữa những nhà phát triển thuật toán.
- Cải tiến các kỹ thuật đánh giá.
- Giúp cải tiến các nghiên cứu về bài toán so khớp ontology.

[2] http://oaei.ontologymatching.org/

Các mục tiêu trên được thực hiện thông qua việc đánh giá thực nghiệm về hiệu quả của các phương pháp so khớp. AOEI tổ chức các cuộc thi hàng năm và công bố các bộ test cùng kết quả của cuộc thi phục vụ cho việc phân tích sâu hơn.

Hai sự kiện đầu tiên được tổ chức vào năm 2004: *(i)* hội nghị *Information Interpretation and Integration Conference (I3CON)* tổ chức tại hội thảo *NIST Performance Metrics for Intelligent Systems (PerMIS)* và *(ii) Ontology Alignment Contest* tổ chức tại hội thảo *Evaluation of Ontology-based Tools (EON)* tại hội nghị hàng năm *International Semantic Web Conference (ISWC)*. Sau đó, các cuộc thi AOEI riêng biệt diễn ra năm 2005 tại hội thảo về *Integrating Ontologies* diễn ra chung với hội nghị *International Conference on Knowledge (K-Cap)*, năm 2006 tại hội thảo *Ontology Matching* đầu tiên diễn ra chung với ISWC và năm 2007 tại hội thảo *Ontology Matching* lần thứ hai diễn ra chung với ISWC + ASWC. Cuối cùng, vào năm 2008, các kết quả OAEI được trình bày tại hội thảo *Ontology Matching* lần thứ ba đồng diễn ra với ISWC tại Karlsruhe, Đức [4].

Các cuộc thi hàng năm có khuynh hướng đa dạng với nhiều loại test case nhấn mạnh vào các mặt khác nhau của việc so khớp ontology. Cuộc thi OAEI 2008 bao gồm bốn vòng thi tập hợp tám tập dữ liệu và các phương pháp đánh giá khác nhau:

1. Vòng thi so sánh: Vòng thi này sử dụng tập dữ liệu *benchmark2008* để nhận diện những lĩnh vực mạnh và yếu của từng thuật toán so khớp.
2. Các ontology có ý nghĩa: vòng thi này có hai tập dữ liệu
    - *anatomy*: tập dữ liệu thế giới thực này dùng để so khớp Giải phẫu Chuột Trưởng thành (2744 lớp) và Từ điển NCI (3304 lớp) mô tả giải phẫu con người.
    - *fao*: mô tả các ontology trên mạng từ các lĩnh vực liên quan đến nghề cá được điều hành bởi Tổ chức Lương thực và Nông nghiệp của Mỹ (FAO).
3. Các thư mục và từ điển chuyên đề:

- *directory*: nhiệm vụ trong thế giới thực bao gồm việc so khớp các thư mục trên các website (như của *open directory* và *Yahoo*). Có hơn bốn ngàn test cơ sở.
- *mldirectory*: nhiệm vụ trong thế giới thực bao gồm việc so khớp các thư mục web (như của *Dmoz, Licos* và *Yahoo*) trong các ngôn ngữ khác nhau (tiếng Anh và tiếng Nhật). Đây là các thư mục chuyên môn với khoảng một ngàn loại.
- *library*: hai từ điển chuyên đề SKOS về các sách phải được sắp xếp dùng những quan hệ từ bộ từ vựng *SKOS Mapping*. Các mẫu kết quả sẽ được đánh giá bởi các chuyên gia trong lĩnh vực.

4. Hội thảo chung: Những người tham dự sẽ được yêu cầu tự do tìm hiểu một tập hợp các ontology dùng để tổ chức hội nghị (các nhà nghiên cứu có thể nắm rõ lĩnh vực) ứng với hai nhiệm vụ có thể (nhiệm vụ chung hay ứng dụng cụ thể). Các kết quả sẽ được đánh giá sau một phần bằng tay và một phần bằng các phương pháp khai thác dữ liệu và lập luận logic. Cũng có một phần đánh giá dựa trên các ánh xạ tham chiếu trong một phần nhỏ dữ liệu.

## 2.7 Vấn đề Tương tác Người dùng trong So khớp Ontology

Trong một nghiên cứu tổng quan gần đây, Shvaiko và Euzenat [17] đưa ra mười thách thức đối với các hệ thống so khớp ontology. Một trong những thách thức này là đưa sự tương tác với người dùng vào trong hệ thống. Bởi vì hiệu quả cuối cùng của hệ thống phụ thuộc vào sự hài lòng của người dùng cuối, tương tác của người dùng hiển nhiên là nhân tố quan trọng ảnh hưởng đến thành công của một hệ thống. Shvaiko và Euzenat [17] quan sát được rằng việc so khớp ontology tự động trong các ứng dụng truyền thống thường không cho ra những kết quả có chất lượng.

Vòng thi *directory*, cung cấp bởi cuộc thi OAEI 2008 [4] là một ví dụ tiêu biểu cho trường hợp này. Đây là một nhiệm vụ so khớp giữa ba thư mục Interrnet thực tế, Google, Yahoo và Looksmart. Những hệ thống tham dự vào vòng thi này

đều cho kết quả khá xấu so với các vòng thi khác của cuộc thi OAEI 2008. Cụ thể, recall trung bình của các hệ thống là 0.30, độ chính xác trung bình là 0.59 và f-measure trung bình là 0.39. Các hệ thống đều có recall thấp, đặc biệt ASMOV (0.12) và RiMOM (0.17). Ví dụ này cho thấy trường hợp mà kết quả so khớp của hệ thống không thoả mãn với mong muốn của người dùng. Do đó đối với các ứng dụng truyền thống, so khớp bán tự động là một cách để cải thiện tính hiệu lực của kết quả. Cho đến bây giờ, chỉ có một số ít nghiên cứu về việc làm sao đưa người dùng vào quá trình so khớp ontology. Hầu hết những nỗ lực này đều dành cho việc tương tác vào so khớp trong lúc thiết kế.

Tuy nhiên một số nghiên cứu gần đây chỉ tập trung vào khía cạnh công thái học trong việc trau chuốt lại các so khớp, dành cho việc thiết kế các so khớp này bằng tay hoặc dành cho việc kiểm tra và sửa lỗi các so khớp. Nghiên cứu trong [10] đề xuất một sự trực quan hoá bằng đồ hoạ cho các so khớp dựa trên nghiên cứu về nhận thức. Đến lượt mình, các nghiên cứu trong [15] đã cung cấp một môi trường để thiết kế thủ công các so khớp phức tạp thông qua việc sử dụng hình vẽ kết nối cho phép nhanh chóng giảm tầm quan trọng của các mặt không có liên quan của ontology được so khớp trong khi vẫn giữ các kết nối giữa những thực thể có liên quan. Dòng nghiên cứu này vẫn tiếp tục được củng cố và những kết quả đạt được ở đây có thể được gắn liền vào hệ thống quản lý so khớp. Với sự phát triển của những hướng tiếp cận tương tác, các vấn đề về tính sử dụng sẽ có đòi hỏi cao hơn. Vấn đề này nói chung bao gồm khả năng mở rộng của việc trực quan hoá và các giao diện người dùng tốt hơn, với hy vọng chúng sẽ đem lại lợi ích lớn về hiệu quả; và thậm chí lợi ích còn lớn hơn từ các thuật toán so khớp chính xác hơn.

Còn một hướng đáng quan tâm nữa liên quan đến việc liên hệ người dùng với hệ thống: dựa trên người dùng hệ thống để học từ họ những gì hữu ích cho các so khớp đang xem xét. Nó có thể được khai thác hoặc ở mức độ của chương trình so khớp bằng cách điều chỉnh các tham số của nó hoặc cung cấp các so khớp đầu vào mới (từng phần), hoặc ở mức độ kết quả bằng các thử nghiệm với các trọng

số tin cậy để cải thiện kết quả đưa ra bởi người dùng. Một hướng khác cũng rất hứa hẹn trong khía cạnh này được gọi là "so khớp ngầm", nghĩa là, bằng cách đóng góp theo kiểu cầu may vào việc cải thiện các so khớp hiện có. Ví dụ, trong một hệ thống ngang hàng có ngữ nghĩa, nếu người dùng đưa ra một truy vấn và ở đó không có so khớp nào trong hệ thống đưa đến câu trả lời, người dùng này có thể sẵn sàng giúp hệ thống bằng cách cung cấp một vài ánh xạ cần để trả lời truy vấn. Những ánh xạ này có thể được tập hợp bởi hệ thống và theo thời gian hệ thống sẽ thu nhận được đủ tri thức về các ánh xạ hữu ích. Ví dụ được thảo luận ở trên có thể cũng được xem là một phần của sự tương tác phổ biến trong môi trường cộng tác. Vấn đề ở đây là thiết kế các mô hình tương tác gây ít gánh nặng cho người dùng trong quá trình so khớp ở cả giai đoạn thiết kế và giai đoạn thực thi. Trong giai đoạn thiết kế, sự tương tác nên tự nhiên và đầy đủ; trong giai đoạn thực thi, nó nên hạn chế trong tác vụ người dùng.

Mục tiêu của luận văn này theo hướng khai thác tương tác của người dùng nhằm nâng cao độ chính xác của hệ thống so khớp đồng thời tạo nên sự thuận lợi cho người dùng trong quá trình tương tác. Với mục tiêu trên, luận văn xem xét tương tác người dùng theo hai hướng: sử dụng tập so khớp mẫu (*pre-alignment*) và phản hồi người dùng (*user feedback*). Tập so khớp mẫu là một tập các tương ứng với giá trị mong muốn được cung cấp từ người dùng. Tập này thường được cung cấp cho hệ thống trước khi quá trình so khớp được thực thi. Trong khi đó, phản hồi người dùng là quá trình xảy ra đồng thời với thao tác so khớp. Phản hồi người dùng là một tính năng thường thấy trong các hệ thống truy vấn thông tin hiện tại. Trong các ứng dụng với phản hồi người dùng, hệ thống sẽ lặp lại việc lựa chọn một số tương ứng và cho phép người dùng đánh giá xem các tương ứng đó là tương ứng đúng, các cặp thực thể trùng khớp, hay tương ứng sai, các cặp thực thể không trùng khớp. Những thông tin này sau đó được dùng để thực hiện việc so khớp.

Để rút gọn phần công việc của người dùng, luận văn giả định những tương ứng do người dùng cung cấp qua tập so khớp mẫu hay qua thao tác phản hồi là

những tương ứng đơn giản với độ tin cậy nhận giá trị 1 hoặc 0; nghĩa là, các tương ứng do người dùng cung cấp hoặc là của một cặp thực thể trùng khớp hoặc không trùng khớp giữa hai ontology. Trong cả hai trường hợp trên, hệ thống sẽ nhận từ người dùng một tập các tương ứng đã biết sẵn giá trị và dựa vào thông tin này để xác định giá trị cho những tương ứng còn lại mà cụ thể là xác định những cặp thực thể trùng khớp giữa hai ontology. Với thông tin này, tiếp cận học máy là một tiếp cận hợp lý cho bài toán vì các phương pháp học máy có thể dựa vào các dữ liệu đã gán nhãn, các tương ứng đã biết giá trị, để dự đoán cho dữ liệu chưa gán nhãn, toàn bộ tương ứng giữa hai ontology. Các chương tiếp theo sẽ trình bày các kiến thức về học máy và mô hình học máy áp dụng cho bài toán so khớp ontology với tương tác người dùng cùng với kết quả thực nghiệm của mô hình.

## 2.8 Tóm tắt

Chương 2 đã trình bày bài toán so khớp ontology và một số vấn đề có liên quan, đặc biệt là vấn đề tương tác người dùng trong hệ thống so khớp ontology. Một hướng tiếp cận đầy hứa hẹn để xử lý các thông tin nhận được từ tương tác người dùng là sử dụng các mô hình học máy. Các mô hình học máy đã được sử dụng thành công trong nhiều bài toán trong ngành khoa học máy tính đặc biệt đối với bài toán có liên quan đến bài toán so khớp ontology như truy vấn thông tin. Chương 3 kế tiếp sẽ giới thiệu các kiến thức cần thiết về các mô hình học máy và việc áp dụng các mô hình này vào giải quyết bài toán so khớp ontology.

# Chương 3 HỌC MÁY VÀ SO KHỚP ONTOLOGY

## 3.1 Các phương pháp học máy

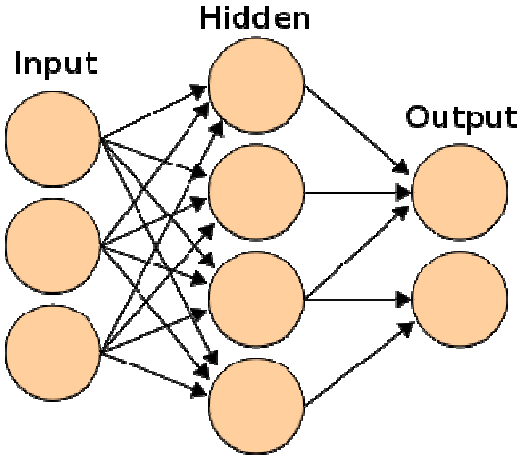
Học máy là một lĩnh vực thuộc ngành trí tuệ nhân tạo liên quan đến việc thiết kế và phát triển các thuật toán cho phép máy tính cải thiện hiệu quả qua thời gian dựa trên dữ liệu. Tuỳ thuộc vào tính chất của dữ liệu huấn luyện, các thuật toán máy học được chia thành ba nhóm. Nhóm thứ nhất là nhóm các thuật toán học có giám sát (supervised learning), huấn luyện trên tập mẫu được gán nhãn, thường được sử dụng trong các bài toán phân lớp hoặc nội suy. Nhóm thứ hai là các thuật toán học không giám sát (unsupervised learning), sử dụng các thuật toán gom cụm để khai thác các cấu trúc vốn có trong dữ liệu chưa gán nhãn. Nhóm các phương pháp học bán giám sát (semi-supervised learning), sử dụng cả các mẫu gán nhãn và chưa gán nhãn trong quá trình gán nhãn. Các thuật toán này quan tâm đến các tập dữ liệu mà tập mẫu gán nhãn chỉ chiếm một phần nhỏ (từ một đến vài mẫu trong mỗi lớp), trong đó a) không đạt được đủ số mẫu cần thiết để đạt được độ tin cậy cao và b) không cho phép tích hợp các thông tin *biết trước* vào trong quá trình học. Những tiểu mục dưới đây sẽ tóm lược một số kiến thức cơ bản về hai loại học có giám sát và bán giám sát.

### 3.1.1 Học có giám sát

Trong các thuật toán học có giám sát, dữ liệu huấn luyện bao gồm các cặp đối tượng đầu vào (thường là các vector) và kết xuất mong muốn tương ứng. Các kết xuất có thể là một giá trị liên tục hoặc có thể dự đoán nhãn lớp của đối tượng đầu vào. Nhiệm vụ của chương trình học có giám sát là dự đoán giá trị của hàm cho bất kỳ đối tượng đầu vào sau khi nhìn qua một số mẫu huấn luyện (các cặp đầu vào và kết xuất mục tiêu). Để đạt điều này, chương trình học phải tổng quát hoá từ những dữ liệu cho trước và đưa ta đến những tình huống chưa thấy theo một cách thức "hợp lý". Các chương trình phân loại được dùng rộng rãi là Mạng Nơ-

ron Nhân tạo, Support Vector Machine, k-láng giềng gần nhất, Naïve Bayes, Mô hình Hỗn hợp Gauss.

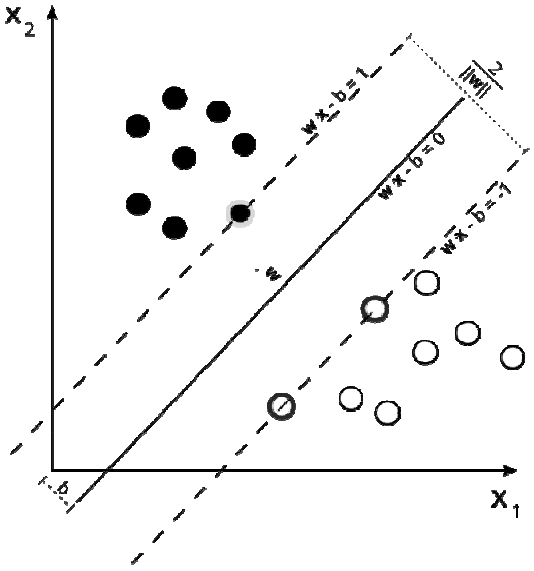
*Mạng Nơ-ron Nhân tạo* là một mô hình toán học hay mô hình tính toán dựa trên mạng nơ-ron sinh học. Nó bao gồm một nhóm các nơ-ron nhân tạo nối với nhau và xử lý thông tin dùng một cách tiếp cận kết nối để tính toán. Hình 3.1 mô tả một mạng nơ-ron lan truyền tiến nhiều lớp, loại mạng nơ-ron phổ biến nhất được sử dụng trong các bài toán phân lớp. Mạng nơ-ron này nhận tín hiệu đầu vào, các giá trị của vector đầu vào qua các nút ở lớp nhập, lan truyền các tín hiệu qua nút ẩn và cho kết xuất của mạng qua các nút xuất. Quá trình trên được gọi là lan truyền tiến và dùng trong pha phân lớp đối tượng. Quá trình học được thực hiện bằng sự lan truyền ngược sai số để điều chỉnh trọng số kết nối giữa node thuộc các lớp. Mạng nơ-ron nhân tạo có ưu điểm có thể giúp xác định các hàm số và các siêu phẳng phân biệt phức tạp nhưng mô hình của nó là một hộp đen đối với người sử dụng, ý nghĩa của các tham số trong mô hình lại không thể dễ dàng hiểu thấu đáo được.




*Hình 3.1. Mạng lan truyền tiến nhiều lớp*

Các *Support Vector Machine (SVM)* là một họ các thuật toán học có giám sát liên hệ với nhau. Xem dữ liệu đầu vào như hai tập vector trong không gian n-chiều, một *Support Vector Machine* sẽ xây dựng một siêu phẳng phân biệt trong không gian đó sao cho nó tối đa hoá biên lề giữa hai tập dữ liệu. Để tính lề, hai

siêu phẳng song song được xây dựng, mỗi cái nằm ở một phía của siêu phẳng phân biệt và chúng được đẩy về phía hai tập dữ liệu (xem ví dụ minh hoạ trong Hình 3.2). Một cách trực quan, một phân biệt tốt thu được bởi siêu phẳng có khoảng cách lớn nhất đến các điểm lân cận của cả hai lớp, vì nói chung lề càng lớn thì sai số tổng quát hoá của bộ phân lớp càng tốt hơn. SVM ban đầu là một thuật toán phân lớp tuyến tính, nhưng nhờ việc áp dụng của các hàm *kernel*, thuật toán có thể giúp tìm ra các siêu phẳng phân biệt phi tuyến trong không gian đặc trưng biến đổi. Và đây chính là điểm nổi bật của các SVM.




*Hình 3.2. Siêu phẳng lề tối đại và các biên cho một SVM được huấn luyện với các mẫu từ hai lớp*

*Thuật toán phân lớp Naïve Bayes* là một thuật toán phân lớp xác suất đơn giản dựa trên việc áp dụng định lý Bayes với giả định độc lập mạnh. Dựa vào bản chất rõ ràng của mô hình xác suất, các bộ phân lớp Naïve Bayes có thể được huấn luyện rất hiệu quả trong một môi trường học có giám sát. Trong nhiều ứng dụng thực tế, việc ước lượng tham số cho các mô hình Naïve Bayes sử dụng phương pháp khả suất tối đại; nói cách khác, người ta có thể dùng mô hình Naïve Bayes mà không cần tin vào xác suất Bayes hay dùng bất kỳ phương pháp Bayes nào. Dù thiết kế chất phác và các giả định quá đơn giản, các bộ phân lớp Naïve Bayes thường hoạt động tốt hơn mong đợi trong nhiều tình huống thế giới thực

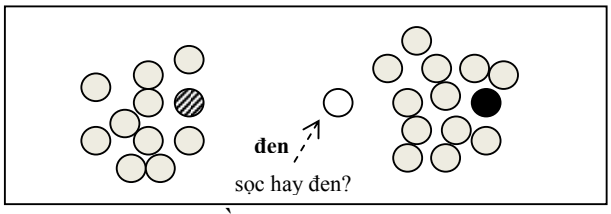
phức tạp. Một ưu điểm của thuật toán phân lớp Naïve Bayes là nó đòi hỏi ít dữ liệu huấn luyện để ước lượng các tham số cần thiết để phân lớp.

*Thuật toán k-láng giềng gần nhất* là thuật toán đơn giản nhất trong số các thuật toán học máy. Một đối tượng được phân loại bằng một cuộc bỏ phiếu đa số giữa các láng giềng của nó và đối tượng sẽ được gán lớp có nhiều có nhiều đối tượng chung nhất trong số $k$ láng giềng gần nhất. $k$ là một số nguyên dương, thường là số nhỏ. Nếu $k = 1$, đối tượng sẽ đơn giản được gán cho lớp của láng giềng gần nó nhất. Trong bài toán phân loại hai lớp, việc chọn $k$ là số lẻ sẽ hữu ích giúp tránh được trường hợp số phiếu bầu bằng nhau.

### 3.1.2 Học bán giám sát

Về cơ bản, các thuật toán học bán giám sát sử dụng các mẫu dữ liệu chưa gán nhãn để làm giàu cho tập huấn luyện bằng cách từ từ gán nhãn cho chúng dựa vào ước lượng từ tập mẫu gán nhãn ban đầu. Hình 3.3 minh hoạ một ví dụ trực quan cho phương pháp học bán giám sát. Bởi vì chúng ta chỉ có một mẫu đen và một mẫu sọc biểu diễn cho hai lớp, khó mà quyết định mẫu trắng chưa gán nhãn sẽ thuộc lớp nào. Nhưng với sự hiện diện của các mẫu xám chưa biết, mẫu trắng có thể được phân vào lớp đen với độ chính xác cao hơn. Một số phương pháp học bán giám sát tiêu biểu là: EM (Expectation Maximization) với mô hình sinh hỗn hợp, tự huấn luyện, huấn luyện cộng tác, *transductive support vector machine* và các phương pháp đồ thị. Các nghiên cứu tổng quan về các phương pháp này được giới thiệu trong [23], [16]. Các thuật toán học bán giám sát dựa trên giả định phân phối của dữ liệu chưa biết và phân phối của dữ liệu đã biết là như nhau hoặc giả định nhất quán (consistent assumption), các điểm dữ liệu ở gần nhau trong không gian metric hoặc có cấu trúc gần giống nhau sẽ có cùng nhãn.

*Mô hình sinh* có lẽ là phương pháp học bán giám sát sớm nhất. Nó giả định phân phối của các điểm dữ liệu thuộc các phân lớp là phân phối hỗn hợp đồng nhất, ví dụ tuân theo mô hình hỗn hợp Gauss. Với số lượng lớn dữ liệu chưa gán nhãn, các thành phần hỗn hợp có thể được xác định; sau đó một cách lý tưởng




*Hình 3.3. Ví dụ về trường hợp học bán giám sát.*

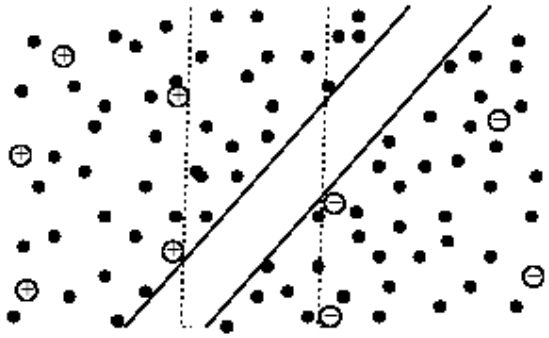
chúng ta chỉ cần một mẫu gán nhãn trên mỗi thành phần cũng đủ xác định phân phối hỗn hợp.

*Tự huấn luyện* là kỹ thuật được sử dụng được dùng nhiều nhất cho học bán giám sát. Trong tự huấn luyện một bộ phân lớp đầu tiên được huấn luyện với một lượng nhỏ dữ liệu gán nhãn. Bộ phân lớp sau đó được dùng để phân loại các dữ liệu chưa gán nhãn. Thông thường những điểm được gán nhãn với độ tin cậy cao nhất cùng với những nhãn dự đoán của nó sẽ được thêm vào tập huấn luyện. Bộ phân lớp được huấn luyện lại và thủ tục trên lặp lại. Lưu ý rằng bộ phân lớp dùng dự đoán của chính nó để dạy lại nó. Mô hình sinh và thuật toán EM có thể xem là một trường hợp đặc biệt của tự huấn luyện mềm. Người ta có thể nghĩ rằng một lỗi phân lớp có thể tăng cường thêm chính nó. Một số thuật toán cố gắng loại lỗi này bằng cách "không học" những điểm chưa gán nhãn nếu độ tin cậy dự đoán xuống dưới một ngưỡng nào đó.

*Huấn luyện cộng tác* giả định rằng (i) các đặc trưng có thể được chia thành hai tập; (ii) mỗi tập đặc trưng phụ là đủ để huấn luyện một bộ phân lớp tốt; (iii) hai tập là độc lập có điều kiện cho trước phân lớp. Đầu tiên hai bộ phân lớp độc lập được huấn luyện với dữ liệu gán nhãn, trên hai tập đặc trưng phụ tương ứng. Mỗi bộ phân lớp sau đó sẽ phân lớp dữ liệu chưa gán nhãn và "dạy" bộ phân lớp kia với một vài mẫu chưa gán nhãn (vùng với nhãn dự đoán của nó) mà chúng cảm thấy tin cậy nhất. Mỗi bộ phân lớp được huấn luyện với mẫu huấn luyện bổ sung cho bởi bộ phân lớp kia và quá trình lặp lại.

*Transductive support vector machine* (TSVM) là một mở rộng của support vector machine chuẩn với dữ liệu chưa gán nhãn. Trong một SVM chuẩn chỉ có dữ liệu gán nhãn được dùng và mục tiêu là tìm một biên tuyến tính có lề tối đại

trong không gian. Trong TSVM dữ liệu chưa gán nhãn cũng được dùng. Mục tiêu là tìm một gán nhãn của các dữ liệu chưa gán nhãn, sao cho tồn tại một biên tuyến tính có lề tối đại trên cả dữ liệu gán nhãn ban đầu và dữ liệu chưa gán nhãn. Biên quyết định có sai số tổng quát hoá nhỏ nhất giới hạn trên dữ liệu chưa gán nhãn. Hình 3.4 minh hoạ trực quan cho trường hợp TSVM, dữ liệu chưa gán nhãn hướng dẫn biên tuyến tính ra xa khỏi vùng có mật độ dữ liệu dày. Chỉ với dữ liệu gán nhãn, biên lề tối đại là đường chấm chấm. Với thêm các dữ liệu chưa gán nhãn (các điểm đen), biên lề tối đại là đường thẳng màu đen.



*Hình 3.4. Một ví dụ về Tranductive SVM*

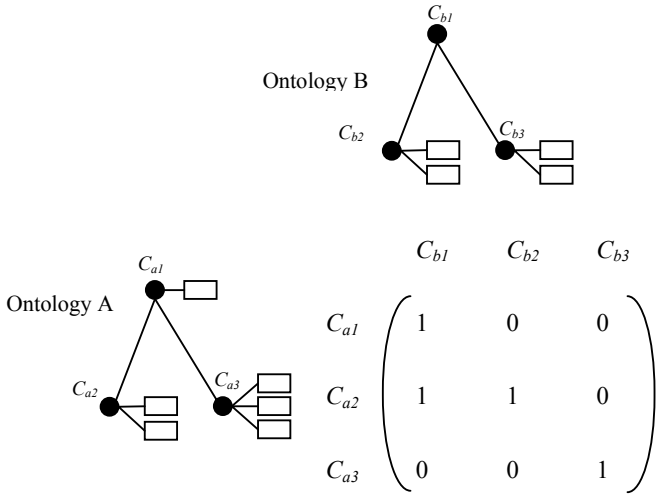
## 3.2 Học máy trong So khớp Ontology

Những kỹ thuật học máy rút trích được tri thức tự động từ dữ liệu. Do đó, những kỹ thuật này có ý nghĩa khi chúng ta cần giải quyết các bài toán mà lượng dữ liệu nhiều đến mức tràn ngập, không cho phép xử lý bằng tay và các hệ thống tự động cũng chưa đưa ra được kết quả cao, chẳng hạn như trong bài toán so khớp ontology [17]. Tiểu mục đầu tiên trong phần này trình bày biểu diễn bài toán so khớp ontology như một bài toán học máy có thể được giải quyết trong một mô hình học tổng quát Cách biểu diễn và mô hình học này được giới thiệu trong [11]. Tiểu mục tiếp theo giới thiệu các công trình liên quan đến việc nghiên cứu học máy trong bài toán so khớp ontology cùng với vấn đề được giải quyết trong luận văn này.

### 3.2.1 Bài toán So khớp Ontology như là một Bài toán học máy

Trong nghiên cứu này, luận văn quan tâm đến bài toán so khớp ontology với khái niệm tương ứng đơn giản, nghĩa là quan hệ giữa hai khái niệm được định nghĩa là quan hệ tương đương với độ tin cậy nhận giá trị 0 hoặc 1. Để giải quyết bài toán so khớp ontology, hệ thống tổ hợp các khái niệm giữa những ontology khác nhau. Trong trường hợp này, vấn đề là xác định giá trị của những cặp tổ hợp này. Nói cách khác, bài toán so khớp ontology bao gồm việc định nghĩa giá trị của các cặp khái niệm trong một ma trận cặp khái niệm, như trình bày trong Hình 3.5. Các dòng của ma trận biểu diễn các khái niệm của Ontology A, đó là $C_{a1}$, $C_{a2}$ và $C_{a3}$ và các cột của ma trận biểu diễn các khái niệm của Ontology B: $C_{b1}$, $C_{b2}$ và $C_{b3}$. Giá trị của ma trận biểu diễn giá trị của ánh xạ. Giá trị 1 khi hai khái niệm có thể được ánh xạ và giá trị 0 khi hai khái niệm không thể được ánh xạ. Ví dụ, giá trị ở dòng thứ hai và cột thứ ba của ma trận biểu diễn giá trị của ánh xạ đối cho $C_{a2}$ của Ontology A và $C_{b3}$ của Ontology B. Ánh xạ cụ thể này là không hợp lệ bởi vì giá trị trong ma trận là 0.

Câu hỏi tiếp theo là cần thông tin gì để suy ra được ma trận. Như đã trình bày trong Chương 2, kỹ thuật cơ bản để xác định được ánh xạ giữa hai cặp khái niệm của hai ontology là sử dụng các độ đo tương tự. Chúng ta có thể sử dụng một độ đo khái niệm, ví dụ độ tương tự dựa trên tên, sử dụng so sánh chuỗi, hoặc các độ đo khác. Tuy nhiên, một độ đo tương tự duy nhất là không đủ để xây dựng được ma trận bởi tính đa dạng của các ontology. Ví dụ, xét trường hợp khái niệm "bank" giữa hai ontology. Các khái niệm trên dường như là một cặp tương ứng nếu dùng độ đo tương tự dựa trên chuỗi. Tuy nhiên, khi một khái niệm trong một ontology có khái niệm cha là "finance" và một khái niệm trong ontology kia có khái niệm cha là "construction", hai khái niệm này không phải là một tương ứng đúng vì chúng diễn tả những khái niệm khác nhau. Trong trường hợp như thế, một độ đo tương tự khác của các khái niệm. Do đó, hệ thống cần dùng nhiều độ đo tương tự để xác định các ánh xạ đúng.




*Hình 3.5. Biểu diễn ma trận của bài toán so khớp ontology [11]*

Như vậy để xác định giá trị cho ma trận so khớp, đầu tiên cần định nghĩa một vector tương tự sử dụng nhiều độ đo tương tự. Kết quả là ta có thể xây dựng được một bảng biểu diễn cho bài toán này như trình bày trong Bảng 3.1. Cột *ID* trong bảng đại diện cho một cặp khái niệm: *Class* biểu diễn giá trị của tương ứng và các cột ở giữa biểu diễn độ tương tự giữa các khái niệm. Ví dụ, dòng đầu tiên của bảng biểu diễn tương ứng cho $C_{a1}$ và $C_{b1}$ có giá trị tương tự 0.75 cho độ đo tương tự 1. Khi biết một số ánh xạ, ví dụ $C_{a1} \Leftrightarrow C_{b1}$ và $C_{a1} \Leftrightarrow C_{b2}$, hệ thống có thể dùng những ánh xạ này để xác định độ quan trọng của các độ đo tương tự. Sau đó, hệ thống có thể quyết định giá trị ánh xạ cho những cặp chưa biết ví dụ $C_{a5} \Leftrightarrow C_{b7}$ bằng cách dùng độ quan trọng của các độ đo tương tự. Bảng ví dụ 3.1 này tương tự như bài toán trong một hệ thống học máy có giám sát. Do đó, bài toán so khớp ontology có thể được chuyển thành một bài toán học máy.

*Bảng 3.1. Biểu diễn dạng bảng của bài toán so khớp ontology*

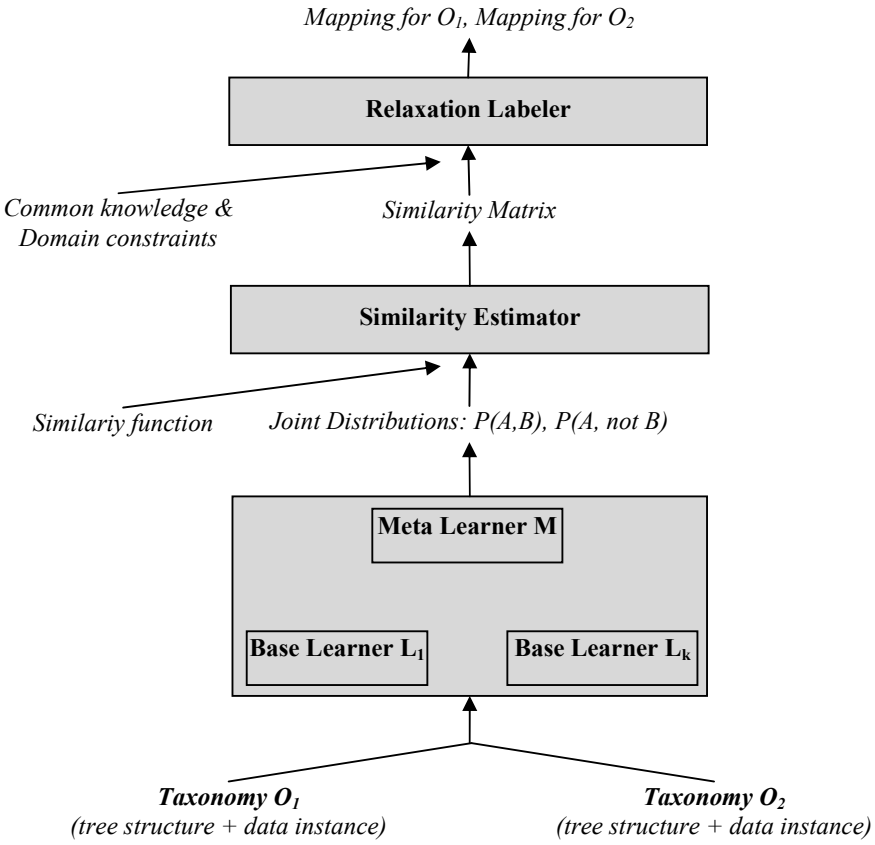
| ID | Độ tương tự 1 | Độ tương tự 2 | … | Độ tương tự n | Lớp |
|---|---|---|---|---|---|
| $C_{a1} \Leftrightarrow C_{b1}$ | 0.75 | 0.4 | … | 0.38 | 1 (Dương) |
| $C_{a1} \Leftrightarrow C_{b2}$ | 0.52 | 0.7 | … | 0.42 | 0 (Âm) |
| … | … | … | … | … | … |
| $C_{a5} \Leftrightarrow C_{b7}$ | 0.38 | 0.6 | … | 0.25 | ? |
| … | … | … | … | … | … |

### 3.2.2 Các nghiên cứu có liên quan

Ngoài mô hình học tổng quát từ [11] như được trình bày ở trên, cách tiếp cận học máy cũng được giới thiệu trong một vài công trình có liên quan đến bài toán so khớp ontology. Agrawal và Srikant [1] giới thiệu mô hình ENB (Enhanced Naïve Bayes) cho bài toán tích hợp các catalog hàng hoá. ENB là thuật toán cải tiến của thuật toán học cơ sở Naïve Bayes, trong đó các tác giả sử dụng các thông tin bổ sung về quan hệ giữa lớp đề hỗ trợ cho việc phân lớp các thể hiện vào các lớp của catalog. Kết quả phân tích và thử nghiệm cho thấy mô hình học cải tiến giúp cải thiện đáng kể độ chính xác của việc tích hợp dữ liệu.

Wang và cộng sự [19] giới thiệu hệ thống cũng giới thiệu một hệ thống so khớp ontology trong đó sử dụng nội dung của các thể hiện để xây dựng độ đo tương tự giữa các khái niệm. Tiếp đó, sử dụng nhân lực để gán nhãn bằng tay cho các cặp khái niệm chọn lọc, họ xây dựng một tập dữ liệu huấn luyện mẫu và sử dụng phương pháp Markov Random Field để làm bộ học phân lớp cho bài toán so khớp các bộ chỉ mục thư viện tại Thư viện Quốc gia Hà Lan. Trong hệ thống này, các tác giả sử dụng thông tin là các trường siêu dữ liệu mô tả cho các đối tượng sách và đa phương tiện làm cơ sở để tính độ đo tương tự. Thông tin này được dùng riêng trong trường hợp của tác giả nhưng có thể dễ dàng tích hợp vào các hệ thống học máy tổng quát như [11], các thông tin này có sẵn trong một số bài toán so khớp khác.

Doan và cộng sự [7] giới thiệu GLUE là hệ thống so khớp ontology trong đó sử dụng kỹ thuật học trong một số bước để xây dựng độ tương tự giữa các khái niệm. GLUE cũng sử dụng nhiều bộ học bao gồm các bộ học trên các loại dữ liệu khác nhau và một bộ siêu học để lựa chọn đặc trưng tương tự cho các bước so khớp tiếp theo. Hình 3.6 mô tả kiến trúc tổng quát của GLUE.




*Hình 3.6. Kiến trúc của GLUE [7]*

Jeong và cộng sự [14] giới thiệu một mô hình học cho bài toán tổng quát cho bài toán so khớp các lược đồ XML. Mô hình này cũng tương tự như mô hình được giới thiệu trong [11] bao gồm việc xây dựng vector tương tự nhiều đặc trưng và áp dụng các chiến lược học khác nhau. Các tác giả cũng thử nghiệm các phương pháp học khác nhau trên hệ thống bao gồm học cả học có giám sát và bán giám sát.

Các thuật toán học máy có giám sát cần sử dụng một tập dữ liệu đã được gán nhãn để huấn luyện mô hình, việc này thường gây tốn kém vì chi phí nhân công cho việc gán nhãn cao. Hơn nữa, do đặc thù đa dạng của các môi trường ứng dụng so khớp ontology thực tế, hệ thống học cần sử dụng một tập dữ liệu huấn luyện riêng nhận từ người dùng cuối cho từng bài toán. Do đó, việc giới hạn kích thước tập huấn luyện là cần thiết để bảo đảm sự hài lòng của người dùng. Những

người dùng cuối thường không sẵn lòng để gán nhãn hàng ngàn mẫu dữ liệu khác nhau như yêu cầu của các hệ thống học máy. Trong trường hợp số mẫu huấn luyện được giới hạn đến mức ít nhất, hệ thống sử dụng phương pháp học bán giám sát kết hợp với học chủ động để giải quyết vấn đề số mẫu huấn luyện ít hơn nhiều so với số mẫu cần dự đoán.

APPEL [8] cũng là một hệ thống học máy tương tự như [11], nhưng hệ thống này đòi hỏi việc sử dụng các ontology khác cũng như yêu cầu người dùng thẩm định là một số cặp so khớp hạt giống được phát sinh tự động trước sử dụng chúng làm tập huấn luyện cho mô hình. Hệ thống này có thể đáp ứng về mặt hiệu quả đối với chương trình nhưng gây khó khăn đối với những người dùng không chuyên do phải cung cấp một số tham số chuyên môn như độ tin cậy của tương ứng.

Có một điểm lưu ý khi sử dụng phương pháp học bán giám sát là cần thiết lập một môi trường thích hợp để sử dụng. Qua thử nghiệm, Jeong và cộng sự [14] nhận thấy các thuật toán học bán giám sát không thực sự cho kết quả cải thiện đáng kể so với các thuật toán học có giám sát. Điều này có thể lý giải do môi trường thử nghiệm không thật sự thích hợp với các thuật toán học bán giám sát, cụ thể số mẫu gán nhãn không thực sự vượt trội so với số mẫu gán nhãn (190 mẫu chưa gán nhãn trên 60 mẫu gán nhãn).

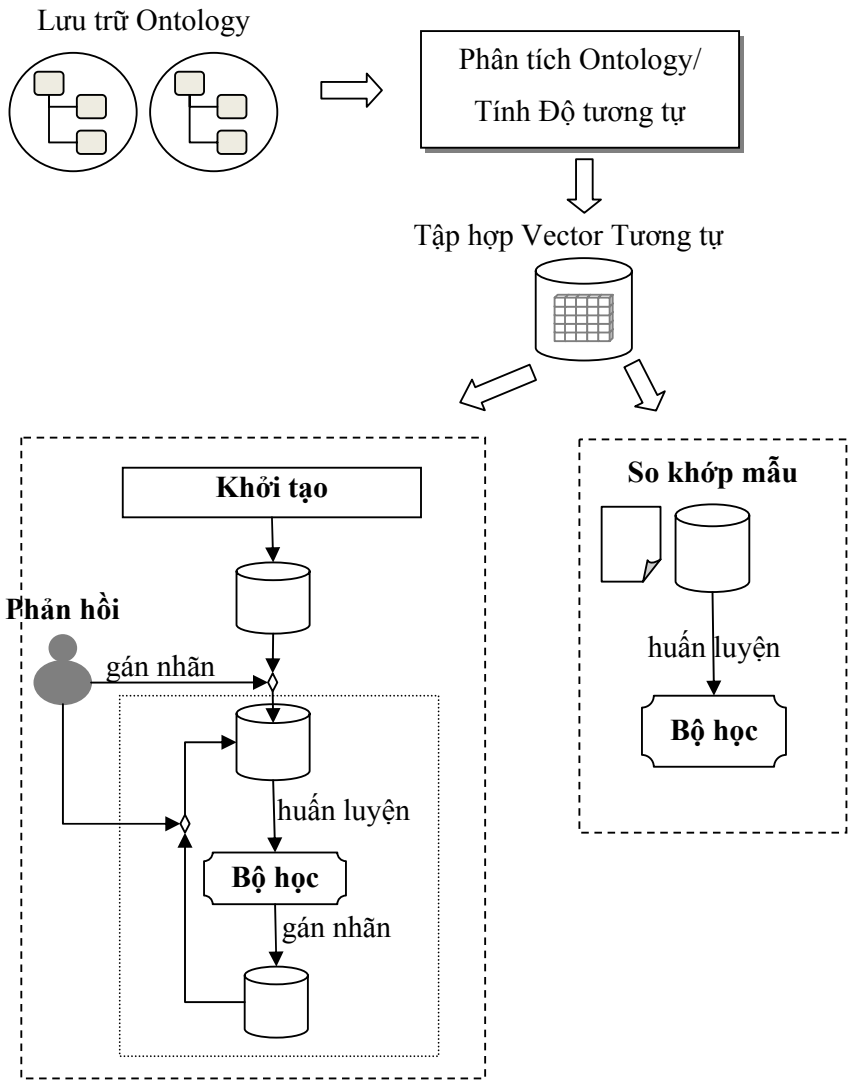
Ngoài ra, việc mẫu chưa gán nhãn có thể là giảm hiệu quả học trong các thuật toán học bán giám sát cũng được ghi nhận trong [6]. Tian và cộng sự [18] xem xét hiện tượng này qua việc khảo sát hiệu quả của các thuật toán học trong các điều kiện phân phối xác suất của các tập dữ liệu có gán nhãn (L) và tập dữ liệu chưa gán nhãn (U). Với tình huống giả định về phân phối dữ liệu thoả, tức là $P_L = P_U$, dữ liệu chưa gán nhãn giúp nâng cao hiệu quả học của các các học bán giám sát. Trong trường hợp $P_L \neq P_U$, việc thay đổi của hiệu quả là không đoán trước. Tuy nhiên, ngược với những ghi nhận trên, Zhou và cộng sự [22] đề xuất một mô hình học cộng tác trong bài toán truy vấn ảnh với phản hồi người dùng.

Thử nghiệm cho thấy mô hình được đề xuất cho hiệu quả cao hơn các mô hình học có giám sát do ảnh hưởng của kích thước tập huấn luyện nhỏ.

Với những thông tin trên, luận văn đề xuất mở rộng mô hình học tổng quát trong [11] thành một hệ thống học linh hoạt trong đó bổ sung phương pháp học bán giám sát kết hợp học chủ động vào mô hình để xử lý cho trường hợp phản hồi người dùng.

# Chương 4 HỆ THỐNG HỌC LINH HOẠT VỚI TƯƠNG TÁC NGƯỜI DÙNG CHO BÀI TOÁN SO KHỚP ONTOLOGY

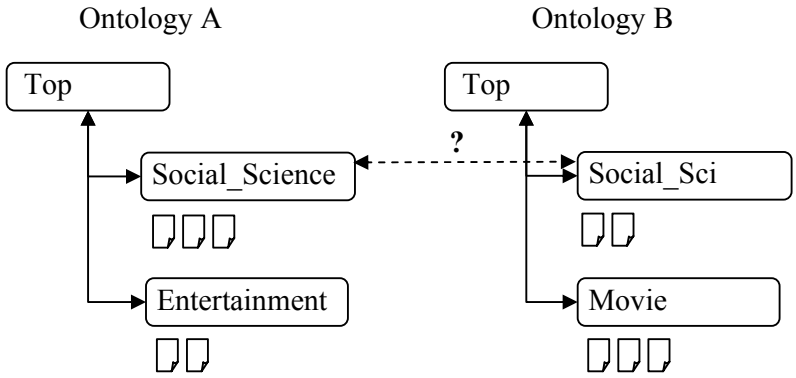
Trong nghiên cứu này, luận văn đề xuất sử dụng một hệ thống học máy linh hoạt với tương tác người dùng cho bài toán so khớp ontology. Hệ thống này là sự mở rộng từ hệ thống học máy tổng quát đã được giới thiệu bởi [11]. Hệ thống sử dụng nhiều chiến lược học, học có giám sát và học bán giám sát kết hợp với học chủ động, để xử lý các tình huống khác nhau trong tương tác người dùng. Hình 4.1 mô tả luồng công việc tổng quát của hệ thống. Hệ thống bắt đầu bằng việc đọc và phân tích cấu trúc của các ontology từ lưu trữ dữ liệu. Tiếp đến hệ thống phát sinh tất cả các cặp khái niệm giữa hai ontology và tính toán độ đo tương tự của các cặp khái niệm trên. Sau đó, dựa trên yêu cầu tương tác của người dùng, hệ thống lựa chọn một phương pháp học thích hợp với môi trường: chọn một phương pháp học có giám sát nếu nó được cung cấp một tập so khớp mẫu với nhiều dữ liệu đã được gán nhãn, hoặc một phương pháp học bán giám sát nếu người dùng muốn tương tác với hệ thống trong quá trình huấn luyện. Các vector tương tự rút trích từ dữ liệu gán nhãn cung cấp bởi người dùng được sử dụng để huấn luyện cho mô hình học. Mô hình sau khi huấn luyện sẽ dùng để dự đoán giá trị so khớp cho tất cả các cặp khái niệm giữa hai ontology. Các mục sau sẽ giới thiệu về vector tương tự được áp dụng trong mô hình và mô tả chi tiết về các cách tiếp cận học máy đã sử dụng.




*Hình 4.1. Hệ thống học tổng quát cho bài toán so khớp ontology với tương tác người dùng*

## 4.1 Xây dựng Vector Tương tự

Như đã được nhận xét trong [9], chỉ sử dụng một loại độ đo tương tự là không đủ tốt cho việc so khớp ontology. Ở đây, chúng tôi sử dụng các độ đo tương tự được đề xuất trong [11] để xây dựng vector tương tự cho các cặp khái niệm. Các độ đo này thuộc ba loại: độ tương tự khái niệm (concept similarity), độ tương tự phân cấp khái niệm (concept hierarchy similarity) và độ tương tự cấu trúc (structure




*Hình 4.2. Hai ontology cần so khớp trong đó ta cần xác định độ tương tự giữa các khái niệm*

similarity). Các độ đo này được tính từ các thông tin tương ứng: thông tin khái niệm (concept information), thông tin phân cấp khái niệm (concept hierarchy information) và thông tin cấu trúc (structure). Ví dụ, giả sử cần tính độ đo tương tự cho hai khái niệm "Social_Science" và "Social_Sci" của hai ontology A và B như trình bày trong Hình 4.2. Thông tin khái niệm chính là nhãn của bản thân khái niệm. Thông tin phân cấp khái niệm là chuỗi các nhãn theo đường đi từ khái niệm gốc đến khái niệm đang xét. Thông tin cấu trúc là nhãn của khái niệm cha khái niệm đang xét. Bảng 4.1 trình bày những thông tin được dùng để tính độ tương tự giữa hai khái niệm trên. Vector tương tự giữa hai khái niệm được xây dựng bằng cách kết hợp độ tương tự giữa các cặp thông tin. Phần kế tiếp thảo luận về cách đo độ tương tự giữa các thông tin này. Do các thông tin được cấu tạo từ các từ, luận văn bắt đầu bằng cách định nghĩa độ tương tự giữa các từ trong nhãn làm cơ sở để tính toán.

*Bảng 4.1. Các thông tin được dùng để tính độ tương tự giữa hai ontology*

| | Ontology A | Ontology B |
|---|---|---|
| Concept information | Social_Science | Social_Sci |
| Concept hierarchy information | Top / Social_Science | Top / Social_Sci |
| Structure information | Top | Top |

### 4.1.1 Độ tương tự của Từ

#### *4.1.1.1 Độ tương tự dựa trên chuỗi*

Để tính toán độ tương tự của khái niệm, hệ thống sử dụng bốn độ tương tự dựa trên chuỗi và bốn độ tương tự dựa trên tri thức làm cơ sở. Các độ tương tự này được dùng để tính toán cho từng cặp từ. Các độ đo tương tự dựa trên chuỗi ký tự là:

- Tiền tố
- Hậu tố
- Edit distance
- N-gram

Độ tương tự tiền tố để tính toán sự tương tự giữa các từ tiền tố như "Eng" và "England". Độ tương tự hậu tố dùng để tính toán sự tương tự giữa các hậu tố như "phone" và "telephone". Với hai độ đo trên, giá trị tương tự bằng một nếu một trong hai từ là tiền tố (hoặc hậu tố) của từ kia và bằng không trong trường hợp ngược lại.

Trong lý thuyết thông tin, *edit distance* giữa hai chuỗi là số phép toán cần thiết để biến đổi một trong hai chuỗi thành chuỗi kia. Do đó, *edit distance* có thể được dùng để tính toán sự tương tự giữa hai chuỗi. Cụ thể, chúng tôi sử dụng *Levenshtein distance*, là số thao tác tối thiểu dùng để biến đổi một chuỗi thành chuỗi kia với ba phép toán thêm, xoá và thay thế một ký tự. Xét ví dụ, *edit distance* của hai từ "kitten" và "sitting" là 3, bởi vì cần 3 phép tính sau để thực hiện biến đổi một chuỗi thành chuỗi kia và không có cách biến đổi nào khác dùng ít hơn 3 phép biến đổi:

- thay 's' vào 'k': "kitten" → "sitten"
- thay 'i' và 'e': "sitten" → "sittin"
- thêm 'g' vào cuối chuỗi: "sittin" → "sitting"

*Edit distance* tính theo khoảng cách Levenshtein có thể được tính bằng một thuật toán quy hoạch động. Sau đó, độ tương tự giữa hai từ bằng *edit distance* được tính theo công thức sau:

$$sim(s,t) = 1 - \frac{d(s,t)}{max(l(s),l(t))}$$

với *s, t* : là cặp từ cần tính độ tương tự

*d(s,t)* : là khoảng cách *edit distance* giữa 2 từ

*l(s), l(t)* : độ dài của hai từ

Đối với n-gram, từ được chia thành các chuỗi con có n ký tự và độ tương tự được tính bằng số các chuỗi con giống nhau giữa hai tập chuỗi con. Ví dụ, độ tương tự giữa "word" và "ward" được tính như sau: Từ đầu tiên, "word" được chia thành "wo, or, rd" đối với trường hợp 2-gram và từ thứ hai "ward" được chia thành "wa, ar, rd". Kết quả, chuỗi "rd" là chuỗi chung của hai tập trên. Trong hệ thống này, chúng tôi sử dụng 3-gram để tính độ tương tự. Công thức tính độ tương tự đối với n-gram:

$$sim(s,t) = \frac{2 \times |gs \cap gt|}{|gs| + |gt|}$$

với *s, t* : là cặp từ cần tính độ tương tự

*gs, gt* : là tập các n-gram tương ứng với những từ trên

#### *4.1.1.2 Độ tương tự dựa trên Wordnet*

Độ tương tự dựa trên tri thức cũng được sử dụng để tính toán cho từ. Hệ thống dùng WordNet[3] là nguồn tri thức để tính toán độ tương tự.

WordNet là một cơ sở dữ liệu từ vựng được cho phép sử dụng online, cung cấp một kho chứa lớn các mục từ vựng tiếng Anh. WordNet được thiết kế để thiết lập mối liên kết giữa bốn loại từ loại (Parts of Speech – POS) – danh từ, động từ, tính từ và trạng từ. Đơn vị nhỏ nhất trong một WordNet là một tập đồng nghĩa biểu diễn một nghĩa cụ thể nào đó của một từ. Nó bao gồm từ, giải thích của từ đó và các từ đồng nghĩa. Nghĩa cụ thể của một từ trong một loại từ loại được gọi là một *sense*. Mỗi *sense* của một từ nằm trong một tập đồng nghĩa khác nhau. Các tập đồng nghĩa tương đương với các nghĩa, là cấu trúc chứa các tập từ đồng nghĩa. Mỗi tập đồng nghĩa có một chú giải định nghĩa khái niệm mà nó biểu

[3] http://wordnet.princeton.edu/

diễn. Ví dụ các từ "night", "nighttime" và "dark" tạo thành một tập đồng nghĩa có chú thích như sau: thời gian sau khi mặt trời lặn và trước khi mặt trời mọc khi trời bên ngoài tối.

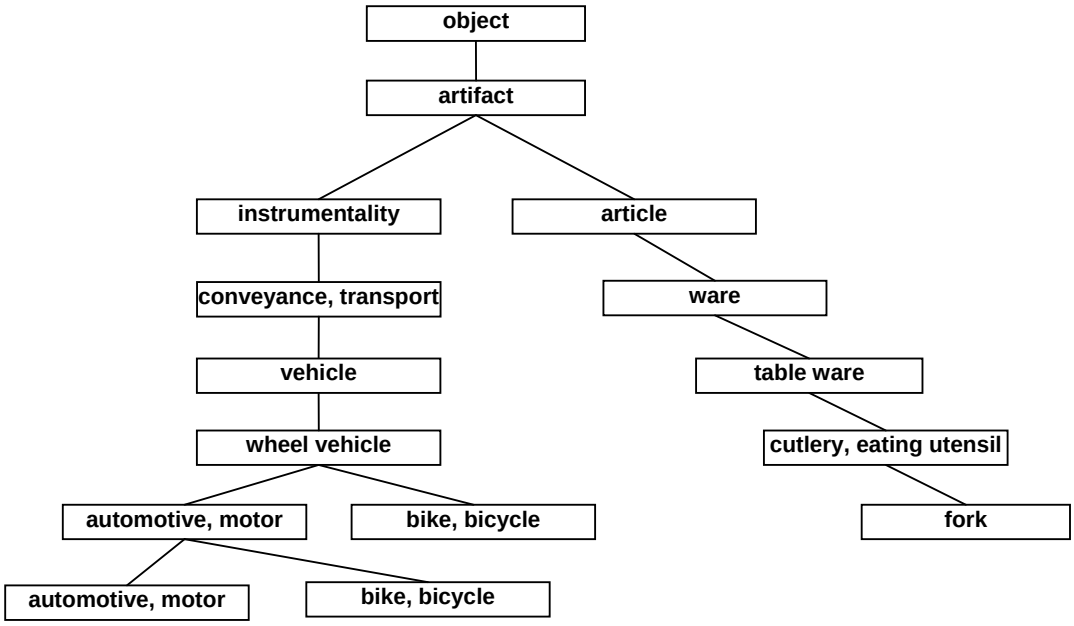
Các tập đồng nghĩa được liên kết với nhau qua mối quan hệ ngữ nghĩa rõ. Một số trong những quan hệ này (*hypernym*, *hyponym* dành cho danh từ và *hypernym* và *troponym* dành cho động từ) tạo thành các cấu trúc phân cấp *là-một-loại-con* (*holonymy*) và *là-một-phần* (*meronymy* dành cho danh từ). Ví dụ, "tree" là một loại "plant", "tree" là một loại con của "plant" và "plant" là một loại cha của "tree". Tương tự, "trunk" là một phần của "tree" và chúng ta có "trunk" là *meronym* của "tree" và "tree" là một *holynum* của "trunk". Với một từ và một loại từ loại, nếu có nhiều hơn một nghĩa, WordNet tổ chức chúng theo thứ tự tần số sử dụng giảm dần.

Trong WordNet, mỗi phần trong các từ cùng loại (danh từ/động từ…) được tổ chức thành các phân loại trong đó mỗi nút là một tập các từ đồng nghĩa (tập đồng nghĩa) biểu diễn một nghĩa nào đó. Nếu một từ có nhiều hơn một nghĩa, nó có thể xuất hiện trong nhiều tập đồng nghĩa tại các vị trí khác nhau trong phân loại. Để đo độ tương tự ngữ nghĩa giữa hai tập đồng nghĩa, chúng tôi sử dụng quan hệ là-một-loại-con/là-một-loại-cha. Do hạn chế của cấu trúc phân cấp là một loại, chúng tôi chỉ làm việc trên hai loại từ "danh từ-danh từ" và "động từ-động từ". Hai độ đo dựa trên WordNet được sử dụng trong hệ thống là:

- Tập đồng nghĩa
- Wu & Palmer

Độ đo tương tự tập đồng nghĩa dựa trên độ dài đường đi giữa các tập đồng nghĩa. Một cách đơn giản để tính độ tương tự ngữ nghĩa giữa hai tập đồng nghĩa là xem phân loại như một đồ thị vô hướng và đo khoảng cách giữa chúng trong WordNet. Quãng đường giữa hai nút càng ngắn thì chúng càng tương tự nhau. Lưu ý rằng độ dài đường đi được đo trong các nút/đỉnh thay vì trong các cạnh/cung. Độ dài đường đi giữa hai thành viên của cùng một tập đồng nghĩa là 1 (các quan hệ đồng nghĩa).

Hình 4.3 trình bày một ví dụ về phân loại là một loại con trong WordNet dùng để tính toán độ tương tự theo chiều dài. Trong hình này, chúng ta thấy rằng chiều dài giữa "car" và "auto" là 1, "car" và "truck" là 3, "car" và "bicycle" là 4, "car" và "fork" là 12.




*Hình 4.3. Một phép phân loại trong Wordnet*

Độ đo tương tự giữa hai tập đồng nghĩa dựa vào độ dài đường đi là

$$sim(s,t) = \frac{1}{distance(s,t)}$$

trong đó *distance(s,t)* là độ dài đường đi ngắn nhất giữa *s* và *t* bằng cách đếm các node trên đường đi

Độ đo tương tự Wu & Palmer [21] dùng độ sâu và khái niệm cha chung tối tiểu (least common superconcept - LCS) của các từ. Một cha chung của hai tập đồng nghĩa được gọi là một *sub-sumer*. *Cha chung tối tiểu* (LCS) của hai tập đồng nghĩa là nút cha mà không có nút con nào cũng là cha chung của hai tập đồng nghĩa đó. Nói cách khác, LCS của hai tập đồng nghĩa là nút cha chung gần nhất của hai tập đồng nghĩa. Trở lại với ví dụ trên, LCS của {auto, car…} và

{truck} là {automotive, motor vehicle} bởi vì {automotive, motor vehicle} gần hơn tập cha chung {wheeled vehicle}. Độ tương tự được tính theo công thức sau:

$$sim(s,t) = \frac{2 \times depth(LCS)}{depth(s) + depth(t)}$$

với *s, t* :là nhãn của cặp từ cần tính độ tương tự

*depth(w)* : là độ sâu của từ *w* tính từ nút gốc đến *w*

*LCS* : là khái niệm cha chung tối thiểu của *s* và *t*

Độ dài đường đi cung cấp một cách đơn giản để tính khoảng cách quan hệ giữa hai nghĩa từ. Có một số vấn đề cần được giải quyết:

- Có thể có hai tập đồng nghĩa trong cùng một loại từ loại không có tập cha chung nào. Bởi vì tất cả các nút trên cùng khác nhau của từng phần trong phân loại từ loại không liên kết với nhau, một đường đi không phải lúc nào cũng có thể tìm thấy giữa hai tập đồng nghĩa. Nhưng nếu một node gốc duy nhất được sử dụng, thì một đường đi sẽ luôn tồn tại giữa bất kỳ hai tập đồng nghĩa danh từ/động từ nào.
- Lưu ý rằng đa kế thừa được cho phép trong WordNet; một số tập đồng nghĩa thuộc về nhiều hơn một phân loại. Do đó nếu có nhiều hơn một đường đi giữa hai tập đồng nghĩa, đường đi ngắn nhất sẽ được chọn.
- Độ đo này chỉ so sánh các nghĩa từ có cùng loại từ loại. Điều đó có nghĩa là hệ thống không so sánh một danh từ và một động từ bởi vì chúng được đặt ở các phân loại khác nhau. Hệ thống chỉ xem xét các từ là danh từ, động từ và tính từ tương ứng với nhau. Khi xem xét một từ, đầu tiên hệ thống kiểm tra xem nó là danh từ hay không và nếu đúng chúng ta sẽ xem nó như danh từ và các động từ và tính từ của nó sẽ được loại bỏ. Nếu nó không là danh từ, hệ thống sẽ kiểm tra xem nó có là động từ hay không…

Với hai độ đo dựa trên WordNet, hệ thống sử dụng thư viện WordNet 2.1 for Windows[4] và WordNet.Net[5] để tính độ tương tự của các từ.

[4] http://wordnet.princeton.edu/
[5] http://opensource.ebswift.com/WordNet.Net/

### 4.1.2 Độ tương tự của Danh sách Từ

Phần này mở rộng các độ đo tương tự của từ đã được giới thiệu trong phần trước. Độ đo tương tự của tự được thiết kế cho các từ và những độ đo này không áp dụng được cho một danh sách từ ví dụ như "Food_Wine". Một danh sách từ như thế có thể được dùng làm một nhãn khái niệm. Nếu tách rời những từ này bằng dấu gạch ngang hay gạch dưới sẽ thu được một danh sách từ. hệ thống định nghĩa hai loại độ tương tự cho danh sách từ: *maximum word similarity* và *word edit distance*.

Trước tiên là phần giải thích về *maximum word similarity*. Khi sử dụng tổ hợp các từ trong cả hai danh sách, chúng ta có thể tính độ tương tự cho mỗi cặp từ theo các độ đo tương tự của từ như ở phần trên. Giá trị lớn nhất của độ tương tự từ của các cặp từ được sử dụng làm độ đo tương tự từ. Do có sáu độ tương tự của từ được sử dụng như đã trình bày trong phần trên, hệ thống có thể thu được sáu giá trị *maximum word similarity* bằng cách sử dụng các độ tương tự từ khác nhau.

Độ đo tương tự thứ hai, *word edit distance*, được dẫn xuất từ *edit distance*. Trong định nghĩa của *edit distance*, độ được tự được tính cho các chuỗi ký tự. Phương pháp này được mở rộng bằng cách xem các từ trong danh sách từ như là các ký tự. Ví dụ, giả sử có hai danh sách từ, "Pyramid" và "Pyramid, Theory", độ tương tự của hai danh sách này khá rõ ràng. Nếu xem một từ là một ký tự, hệ thống có thể tính *edit distance* cho các danh sách từ này. Trong trường hợp này, "Pyramid" là như nhau trong cả hai danh sách từ, do đó có thể tính *word edit distance* bằng một. Ngược lại, nếu xét "Top" và "Pyramid, Theory", *word edit distance* của chúng là hai. Tóm lại, với cách tính này, có thể tính độ tương tự dựa vào *word edit distance*. Tuy nhiên, có một vấn đề đối với danh sách các từ tương tự. Ví dụ, khi xét "Social, Science" và "Social, Sci" thì độ tương tự sẽ được quyết định là bao nhiêu? Vấn đề nằm ở việc tính toán độ tương tự giữa "Science" và "Sci": nghĩa là, phải quyết định xem liệu hai từ này có như nhau hay không. Nếu quyết định rằng hai từ là như nhau, *word edit distance* bằng không và ngược

lại, *word edit distance* bằng một. Để tính toán sự tương tự của các từ, hệ thống dùng các độ đo tương tự của từ với một ngưỡng cụ thể nào đó. Ví dụ, nếu dùng tiền tố là độ đo tương tự của từ, có thể xem hai từ là như nhau khi tính *word edit distance*. Ngược lại, nếu ta dùng *tập đồng nghĩa* làm độ đo tương tự của từ, không thể xem hai từ là như nhau vì "sci" không nằm trong Wordnet. Với diễn giải ở trên, có thể định nghĩa *word edit distance* ứng với sáu độ đo tương tự của từ. Kết quả là, hệ thống thu được 12 độ đo tương tự cho các các danh sách từ, bao gồm sáu *maximum word similarity* và sáu *word edit distance similarity*. Trong các thông tin được sử dụng để tính toán độ tương tự, *thông tin khái niệm* là nhãn của chính khái niệm đang xét và *thông tin cấu trúc* là nhãn của khái niệm cha khái niệm đang xét. Do đó, với 12 độ đo tương tự dành cho danh sách từ, độ tương tự cho các *thông tin khái niệm* và *thông tin cấu trúc* có thể được tính toán.

### 4.1.3 Độ tương tự của Phân cấp Khái niệm

Phần này thảo luận về độ tương tự đối với phân cấp khái niệm trong ontology. Như đã giới thiệu trong phần định nghĩa, các ontology được tổ chức theo cấu trúc phân cấp. Để sử dụng thông tin về cấu trúc phân cấp khái niệm này, luận văn sử dụng các độ đo tương tự của cấu trúc phân cấp khái niệm. Độ đo tương tự của cấu trúc phân cấp khái niệm được tính cho đường đi từ khái niệm gốc đến khái niệm đang xét. Giả sử cần tính độ tương tự cho hai đường đi "Top / Social_Science" trong Ontology A và "Top / Social_Sci" trong Ontology B như trong Hình 4.2. Để tính độ tương tự, trước tiên tách đường đi thành một danh sách các khái niệm, như trong cột ở giữa của Bảng 4.2. Sau đó độ tương tự có thể được tính bằng *edit distance* nếu xem mỗi khái niệm là một thành phần. Ví dụ, khái niệm "Top" là như nhau trong cả hai ontology, nhưng khái niệm thứ hai là khác nhau. Sau đó, có thể tính *edit distance* cho đường đi. Tuy nhiên, làm sao có thể quyết định được liệu khái niệm có như nhau hay không? Để tính toán điều này, chúng ta tách khái niệm ra thành các danh sách từ và tính độ tương tự dùng độ tương tự cho danh sách từ như thực hiện đối với thông tin khái niệm và thông tin cấu trúc. Trong trường hợp này, nếu "Social, Science" và "Social, Sci" được

xem là các khái niệm tương đương với một độ đo tương tự cho danh sách từ nào đó, *edit distance* bằng không; ngược lại nếu hai khái niệm không được xem là tương đương, *edit distance* bằng một. Nói cách khác, edit distance được tính với danh sách bên tay phải trong Bảng 4.2. Kết quả là, có thể tính độ tương tự cho phân cấp khái niệm bằng cách dùng *edit distance* cho đường đi. Bởi vì có thể dùng bất cứ độ đo tương tự cho danh sách từ nào để quyết định sự tương đương cho các khái niệm, hệ thống thu được mười hai độ đo tương tự cho phân cấp khái niệm.

*Bảng 4.2. Ví dụ về phân cấp khái niệm dùng để tính độ tương tự phân cấp khái niệm*

| | Đường đi | Danh sách đường đi | Danh sách từ |
|---|---|---|---|
| Ontology A | Top / Social_Science | {Top, Social_Science} | {Top} {Social, Science} |
| Ontology B | Top / Social_Sci | {Top, Social_Sci} | {Top} {Social, Sci} |

Như vậy, với các độ đo tương tự được sử dụng, một vector tương tự ba mươi sáu giá trị cho mỗi cặp khái niệm giữa hai ontology được xây dựng.

## 4.2 Hệ thống Học Linh hoạt cho So khớp Ontology

Sau khi tính toán các độ tương tự, dữ liệu nhận được từ tương tác người dùng, ví dụ tập so khớp mẫu, được dùng để huấn luyện hệ thống học. Như trên đã trình bày, trong tập so khớp mẫu này, người dùng sẽ gán giá trị trùng khớp hoặc không trùng khớp cho một số cặp khái niệm giữa hai ontology. Hệ thống xem các cặp khái niệm trùng khớp là các mẫu dương và các cặp khái niệm không trùng khớp là các mẫu âm. Bài toán so khớp ontology được đưa về bài toán phân loại hai lớp trùng khớp hoặc không trùng khớp cho dữ liệu là tất cả các cặp khái niệm giữa hai ontology. Hệ thống sẽ học các luật phân lớp từ dữ liệu nhận tương tác người dùng và sau đó dùng những luật phân lớp này để dự đoán giá trị so khớp cho tất cả các cặp còn lại, nghĩa là quyết định xem những cặp nào là trùng khớp và những cặp nào không trùng khớp.

Trong đề xuất này, thay vì dùng một thuật toán học cố định, hệ thống lựa chọn chiến lược học dựa trên môi trường sử dụng thực tế. Nếu được cung cấp một tập so khớp mẫu với nhiều dữ liệu gán nhãn, hệ thống sử dụng một phương pháp học có giám sát để huấn luyện mô hình. Trong trường hợp người dùng muốn tương tác với hệ thống trong quá trình so khớp bằng quá trình phản hồi để tối thiểu hoá chi phí gán nhãn bằng tay, hệ thống sử dụng phương pháp học bán giám sát kết hợp với học chủ động để giải quyết vấn đề kích thước tập mẫu huấn luyện nhỏ. Hệ thống sử dụng cùng một bộ học cơ sở trong cả hai trường hợp. Những nhóm mục dưới đây sẽ giới thiệu về bộ học cơ sở này và kế tiếp giới thiệu về thuật toán học bán giám sát và học chủ động được dùng để xử lý quá trình phản hồi của người dùng.

### 4.2.1 Bộ học cơ sở

Hệ thống sử dụng cách tiếp cận xác suất làm bộ học cơ sở trong cả hai trường hợp. Cho một cặp khái niệm với vector tương tự *v*, quyết định phân loại cặp khái niệm là một mẫu dương (một khái niệm trùng khớp) hay một mẫu âm (một cặp khái niệm không trùng khớp) được thực hiện bằng cách xác định lớp $C_i$ sao cho xác suất $P(C_i|v)$ là lớn nhất. Theo luật Bayes, xác suất này được tính theo công thức:

$$P(C_i|v) = \frac{P(C_i).P(v|C_i)}{P(v)}$$

Trong đó, $P(C_i|v)$ là xác suất mẫu được dự đoán vào phân lớp $C_i$ khi biết được giá trị vector *v*; $P(C_i)$ xác suất tiên nghiệm của lớp $C_i$; $P(v/C_i)$ là xác suất phân bố của vector tương tự *v* trong phân lớp $C_i$. Mẫu số $P(v)$ là xác suất của vector *v,* được xem như nhân tử chuẩn hoá và bằng nhau với mọi $P(C_i|v)$ do đó có thể bỏ qua và chỉ cần so sánh tử số. Trong trường hợp *v* là vector tương tự *m*-chiều với giá trị thực cho mỗi chiều, $P(v|C_i)$ được giả định tuân theo phân phối Gauss:

$$P(v \mid C_i) \sim N(v|\mu,) = \frac{1}{(2\pi)^{m/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(v-\mu)^{\mathrm{T}}\Sigma^{-1}(v-\mu)\right\}$$

trong đó vector $m$-chiều $\mu$ được gọi là trung bình và ma trận $m \times m$ $\Sigma$ được gọi là hiệp phương sai và $|\Sigma|$ là định thức của $\Sigma$. Các giá trị này là tham số của mô hình ứng với mỗi phân lớp.

Mô hình được huấn luyện bằng cách tìm ước lượng khả suất tối đại cho các tham số cho mỗi lớp và dùng các tham số này để tính xác suất phân lớp. Trong trường hợp học có giám sát, tập so khớp mẫu được dùng để tính toán các tham số này một lần trước khi sử dụng. Đối với trường hợp học bán giám sát kết hợp với học chủ động, quá trình học diễn ra phức tạp hơn và được mô tả ở phần tiếp theo.

### 4.2.2 Học Bán giám sát và Học chủ động với Phản hồi Người dùng

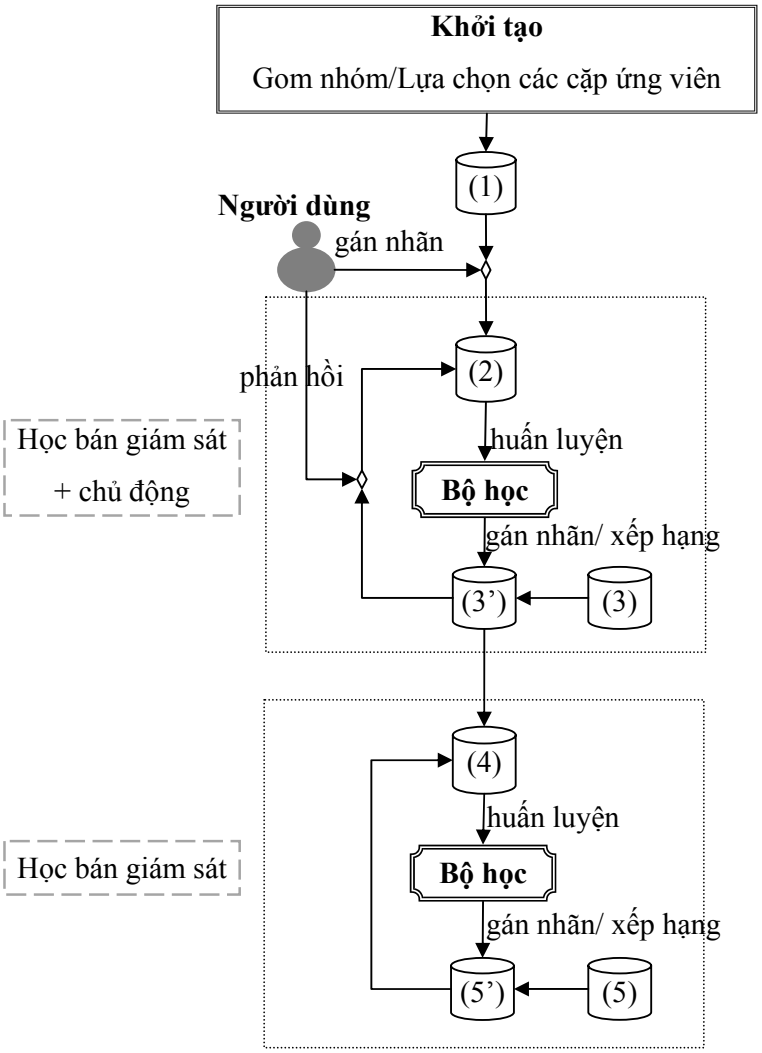
Trong trường hợp sử dụng phản hồi người dùng, hệ thống áp dụng thuật toán học bán giám sát kết hợp học chủ động. Phương pháp học bán giám sát ở đây là tự huấn luyện (self-training) với bộ học cơ sở tương tự như trình bày ở phần trên.

Mô hình học bán giám sát kết hợp học chủ động được trình bày chi tiết trong Hình 4.4 . Trong bước khởi tạo, hệ thống lựa chọn một cặp khái niệm ứng viên (1) và yêu cầu người dùng gán nhãn bằng tay. Để các cặp dữ liệu được chọn có độ tin cậy cao, trước tiên tất cả các mẫu dữ liệu được gom nhóm lại và những cặp khái niệm nằm gần các tâm cụm sẽ được chọn làm ứng viên. Dữ liệu sau khi gán nhãn bằng tay (2) được dùng làm tập huấn luyện nhân để huấn luyện cho bộ phân lớp. Hệ thống phát sinh một tập dữ liệu con ngẫu nhiên từ các dữ liệu chưa gán nhãn (3) và sử dụng bộ phân lớp này để dự đoán và gán nhãn cho các dữ liệu mới (3’). Một số mẫu ứng viên với độ tin cậy dự đoán thấp sẽ được đưa cho người dùng để phản hồi để kiểm chứng lại việc gán nhãn của hệ thống. Những dữ liệu này sau khi phản hồi sẽ bổ sung vào tập huấn luyện (2) cho các vòng huấn luyện kế tiếp.

Quá trình phản hồi được lặp trong vài vòng lặp, sau đó hệ thống chuyển sang quá trình học bán giám sát. Trong quá trình này, hệ thống cũng sử dụng cơ chế học tương tự trên nhưng không yêu cầu phản hồi từ người dùng. Các tập dữ liệu (4), (5) và (5’) đóng vai trò như các tập (2), (3), (3’) tương ứng. Trong quá trình

này sau mỗi vòng lặp, một số mẫu dữ liệu được dự đoán với độ tin cậy cao từ (5') sẽ được bổ sung vào (4) là tập huấn luyện mới cho lần lặp sau.

Một trong những yêu cầu đối với các hệ thống sử dụng phản hồi người dùng là vấn đề thời gian, người dùng sẽ không hài lòng nếu phải đợi quá lâu sau mỗi vòng lặp. Bởi vì mô hình xác suất có thể được huấn luyện và sử dụng để dự đoán nhanh chóng trong mỗi lần lặp, nó có thể đáp ứng được yêu cầu về mặt thời gian này khi được sử dụng làm bộ học cơ sở trong trường hợp sử dụng phản hồi từ người dùng.




*Hình 4.4. Mô hình học bán giám sát kết hợp học chủ động với tương tác người dùng*

# Chương 5 THỬ NGHIỆM VÀ ĐÁNH GIÁ

Chương 5 trình bày thử nghiệm so khớp ontology nhằm đánh giá hiệu quả của hệ thống học linh hoạt được đề xuất. Trong [11], Ichise đã đề xuất một framework học tổng quát cho bài toán so khớp ontology và gọi framework là Malfom (Machine Learning framework for Ontology Matching). Mô hình học được đề xuất trong nghiên cứu này cũng là một mô hình tương tự với Malfom nhưng nhấn mạnh đến yếu tố tương tác người dùng bằng cách linh hoạt sử dụng nhiều phương thức học khác nhau tuỳ vào môi trường sử dụng. Do đó luận văn gọi hệ thống này là MalfomUI (Machine Learning framework for Ontology Matching with User Interaction).

Luận văn thực hiện việc thử nghiệm đánh giá hệ thống trong hai tình huống: đánh giá hiệu quả phương pháp học có giám sát khi hệ thống được cung cấp một tập so khớp mẫu (pre-alignment) và đánh giá hiệu quả phương pháp học bán giám sát kết hợp học chủ động trong trường hợp sử dụng phản hồi người dùng. Mục ngay sau đây sẽ giới thiệu về môi trường chung để thực hiện thử nghiệm, tập dữ liệu kiểm tra dùng để đánh giá hệ thống. Tiếp theo là các phần mô tả và kết quả của hệ thống qua hai tình huống. Các mục cuối dành để phân tích, thảo luận trên kết quả đạt được và nêu ra kết luận cùng hướng phát triển của luận văn.

## 5.1 Môi trường Thử nghiệm Chung

### 5.1.1 Dữ liệu Thử nghiệm

Trong nghiên cứu này, luận văn sử dụng tập dữ liệu kiểm tra được sử dụng trong phần thi so khớp *directory* từ cuộc thi OAEI 2008 để thử nghiệm cho hệ thống học máy dùng cho so khớp ontology [4]. Test case *directory* có mục tiêu đưa ra một nhiệm vụ đầy thách thức cho các hệ thống so khớp trong lĩnh vực các thư mục lớn. Tập dữ liệu dùng trong phần thi này được xây dựng từ các thư mục web Google, Yahoo và Looksmart. Nó được biểu diễn dưới dạng phân loại trong đó

các node của các thư mục web được mô hình hoá bởi các lớp và quan hệ phân lớp kết nối các node này được mô hình hoá thành quan hệ *rdfs:subClassOf*.

Tập dữ liệu bao gồm 4639 nhiệm vụ so khớp. Các ontology biểu diễn các thư mục web chỉ chứa một loại quan hệ lớp con. Mỗi nhiệm vụ so khớp gồm một cặp ontology, trong đó chúng ta cần so khớp các node được tạo thành từ đường đi từ nút gốc nghĩa đến chính nó. Đi kèm với tập dữ liệu là các tương ứng tham chiếu cho tất cả các nhiệm vụ, tức là ta có thể biết được trong những cặp ontology này những cặp nào là trùng khớp và cặp nào không trùng khớp. So khớp của ontology được tạo nên dựa trên các thuật ngữ và nguyên tắc mô hình hoá mập mờ, do đó các nhiệm vụ so khớp trong tập dữ liệu này liên hệ chặt chẽ với các lỗi mô hình hoá và thuật ngữ thường gặp trong thế giới thực. Cũng vì lý do đó, các chương trình so khớp tham gia vào vòng thi này đều cho các kết quả khá thấp (xem Hình 5.1 trích từ [4] trình bày kết quả so khớp chi tiết của 7 hệ thống tham gia vào vòng thi này để biết thêm chi tiết).

Tập dữ liệu này chứa 4639 thư mục được đánh số từ 1 đến 4639, đánh số của các thư mục được xáo trộn ngẫu nhiên so với số hiệu của mẫu dữ liệu. Mỗi thư mục chứa một cặp ontology được biểu diễn bằng ngôn ngữ OWL/RDF trong hai tập tin *source.owl* và *target.owl*. Hai ontology này đều có tên chung và là số hiệu của mẫu dữ liệu. Tập tương ứng tham chiếu bao gồm: danh sách các tổ hợp khái niệm giữa hai ontology, danh sách các mẫu dương (các cặp tương ứng đúng) và danh sách các mẫu âm (các cặp không tương ứng đúng). Danh sách các tổ hợp khái niệm giữa hai ontology được cho theo định dạng:

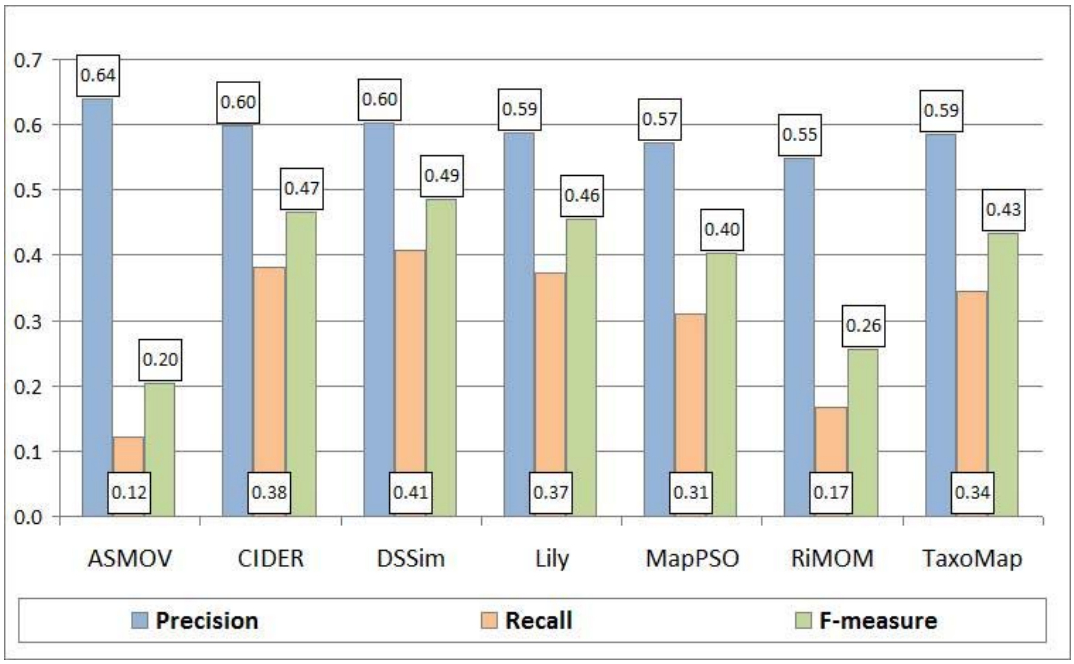
data****c$$$-->[Node1]#[Node2]

trong đó ****: là số hiệu của tập dữ liệu

$$$: số hiệu của tổ hợp

*Node1*, *Node2*: tên của 2 khái niệm thuộc ontology nguồn (source.owl) và ontology đích (target.owl).

Danh sách các mẫu dương và danh sách các mẫu âm liệt kê số hiệu của các tương ứng. Ví dụ, phần tử đầu tiên trong danh sách các mẫu âm là




*Hình 5.1. Biểu đồ kết quả của các hệ thống so khớp tham dự vòng thi directory cuộc thi OAEI 2008 [4]*

"data0001c023" và trong danh sách tổ hợp khái niệm có "data0001c023--> Celebrities#Architecture", điều đó có nghĩa là cặp "Celebrities" trong ontology nguồn trong dữ liệu 1 và "Architecture" trong ontology đích trong dữ liệu 1 là một mẫu ánh xạ âm (không phải là một cặp khái niệm tương ứng).

Trong 4693 tương ứng được cung cấp, có 2265 cặp là các tương ứng đúng, các mẫu dương và 2374 cặp là tương ứng sai, các mẫu âm. Tuy nhiên, do có một số sai sót trong định dạng dữ liệu, chúng tôi chỉ sử dụng 4487 cặp khái niệm, trong đó có 2160 mẫu dương và 2327 mẫu âm.

### 5.1.2 Độ đo Đánh giá

Trong những thử nghiệm này, luận văn sử dụng các độ đo đánh giá vốn bắt nguồn từ lĩnh vực truy vấn thông tin và được dùng để đánh giá các hệ thống so khớp ontology là: *precision*, *recall* và *f-measure*. Bên cạnh đó, luận văn sử dụng thêm *accuracy* trong phần đánh giá hiệu quả của thuật toán học. Precision và recall được tính dựa trên việc so sánh so khớp kết quả $A$ với một so khớp tham chiếu $R$. Precision đo tỷ lệ các tương ứng được tìm thấy đúng trên tổng số tương

ứng được trả về. Recall đo tỷ lệ các tương ứng được tìm thấy đúng trong tổng số tương ứng được mong đợi của tập tập tham chiếu.

Cụ thể, gọi $A$ là tập so khớp kết quả với $A^+$ là tập các cặp khái niệm kết quả trùng khớp và $A^-$ là tập các cặp khái niệm kết quả không trùng khớp. $R$ là tập so khớp tham chiếu với $R^+$ là tập các cặp khái niệm tham chiếu trùng khớp và $R^-$ là tập các cặp khái niệm tham chiếu không trùng khớp. Precision P(A,R) và recall R(A,R) được tính toán theo công thức sau:

$$P(A,R) = \frac{|A^+ \cap R^+|}{|A^+|}$$

$$R(A,R) = \frac{|A^+ \cap R^+|}{|R^+|}$$

*Accuracy* là tỷ lệ các cặp khái niệm được phân loại đúng trên tổng số các cặp khái niệm.

$$A(A,R) = \frac{|A \cap R|}{|R|}$$

Mặc dù precsion và recall là những độ đo được dùng rộng rãi và phổ biến nhất, nhưng chúng lại gây khó khăn khi phải đánh giá các hệ thống vì hai độ đo trên lại không tăng/giảm tương ứng với nhau. Hệ thống có recall cao có thể có precision thấp và ngược lại. Hơn nữa, việc so sánh mà chỉ dựa trên một mình precision và recall không phải là một ý hay. Với mục tiêu này, độ đo F-measure được sử dụng để đánh giá tổng quát các hệ thống. F-measure là trung bình điều hoà có trọng số của *precision* và *recall* và có công thức:

$$M_\alpha(A,R) = \frac{P(A,R) \times R(A,R)}{(1-\alpha) \times P(A,R) + \alpha \times R(A,R)}$$

trong đó $\alpha$ là một tham số có giá trị nằm giữa 0 và 1. Nếu $\alpha = 1$, F-measure bằng với precision và nếu $\alpha = 0$, F-measure bằng với recall. Giữa đoạn đó, giá trị $\alpha$ càng cao, độ quan trọng của precision càng cao so với recall. Luận văn sử dụng giá trị thường được dùng là $\alpha = 0.5$, nghĩa là

$$M_{0.5}(A,R) = \frac{2 \times P(A,R) \times R(A,R)}{P(A,R) + R(A,R)}$$

## 5.2 Thử nghiệm 1 (Học có giám sát)

Trong thử nghiệm này, luận văn đánh giá hiệu quả của phương pháp học có giám sát khi xử lý với tập so khớp mẫu của người dùng. Thử nghiệm này được thực hiện với các mục đích:

- Đánh giá ảnh hưởng của kích thước của tập dữ liệu huấn luyện lên hiệu quả của hệ thống học có giám sát.
- Sử dụng kết quả của phương pháp học có giám sát làm cơ sở để so sánh kết quả của phương pháp học bán giám sát kết hợp học chủ động được thực hiện trong thử nghiệm sau.

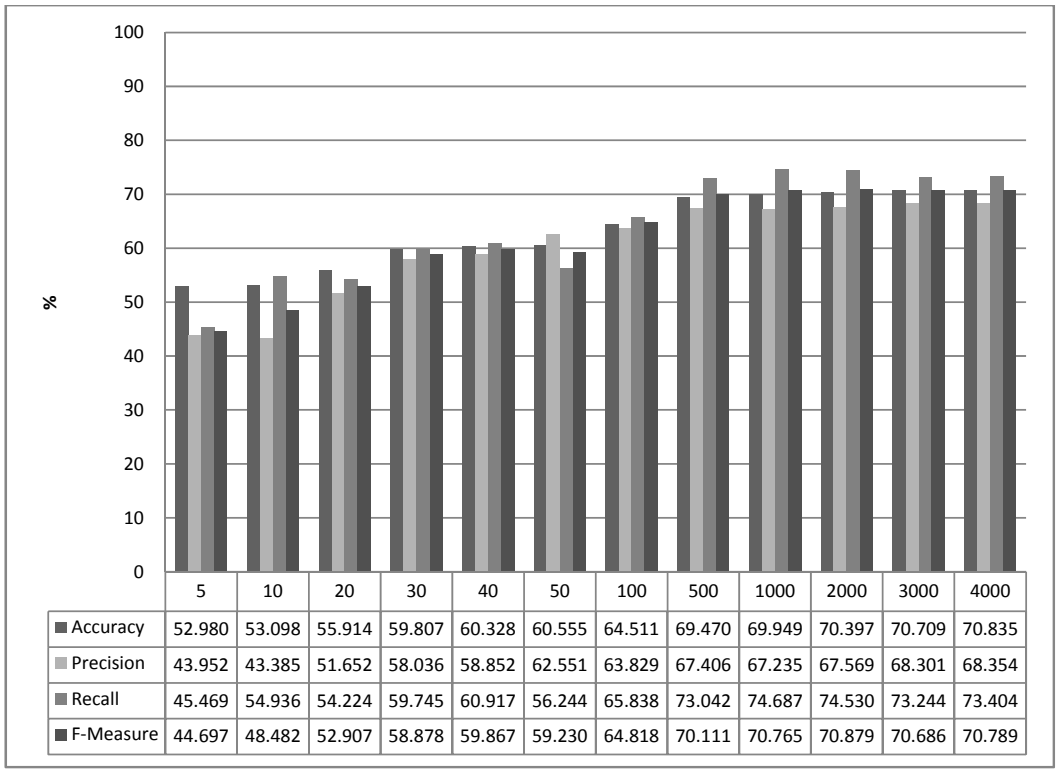
Hầu hết các phương pháp học máy đều sử dụng phép kiểm tra *k*-fold (thường là 5-fold hay 10-fold) để đánh giá. Trong kiểm tra *k*-fold, tập dữ liệu gán nhãn được chia làm *k* phần, trong đó *k*-1 phần sử dụng để huấn luyện và phần còn lại được sử dụng để kiểm tra. Điều này có nghĩa là phần lớn tập dữ liệu gán nhãn được dùng để huấn luyện. Thử nghiệm trong nghiên cứu này có khác biệt so với những thử nghiệm trên. Luận văn cũng chia tập dữ liệu thành 10 tập, nhưng thay vì dùng toàn bộ 9 tập làm tập huấn luyện, một tập con với một kích thước nào đó được trích ra để dùng làm tập huấn luyện. Thao tác này giả lập tương tác người dùng trong đó tập so khớp mẫu có thể được cung cấp với kích thước bất kỳ. Các tập này được luân chuyển 10 lần để mỗi tập đóng vai trò như nhau trong thử nghiệm, trong mỗi lần như thế, thử nghiệm được lặp lại 10 lần và tính hiệu quả trung bình.

Hình 5.2 trình bày kết quả của thực nghiệm này, ảnh hưởng của kích thước tập huấn luyện đến hiệu quả của phương pháp học có giám sát. Trục hoành biểu diễn kích thước tập huấn luyện và trục tung biểu diễn tỷ lệ phần trăm của các độ đo accuracy, precision, recall và f-measure. Về cơ bản, các chỉ số đều tăng khi kích thước tập huấn luyện tăng. Khi kích thước nhỏ (dưới 500 mẫu), sự gia tăng là rõ ràng dù có một số điểm hiệu quả hơi giảm khi kích thước tăng (ở giữa kích thước 40 và 50 mẫu). Điều này có thể bị gây ra do sự phân bố trong dữ liệu. Từ kích thước 500 mẫu trở đi, hiệu quả của hệ thống trở nên ổn định và không có sự

gia tăng đáng kể nào khi kích thước tăng. Với kích thước tập huấn luyện là 500 mẫu, hệ thống cho hiệu quả gần như tương tự với kết quả khi sử dụng tập huấn luyện có kích thước 4000 mẫu (tám lần lớn hơn).

Để so sánh với các phương pháp khác, luận văn chọn kết quả của hệ thống khi kích thước của tập huấn luyện là 100 và 500 mẫu. So sánh được trình bày trong Bảng 5.1 và Hình 5.3. Bảng này trình bày kết quả của bảy hệ thống tham gia trong vòng thi *directory* của cuộc thi OAEI 2008 [4] và kết quả của hệ thống MalfomUI với kích thước tập huấn luyện là 100 và 500 mẫu, MalfomUI-100 và MalfomUI-500. Có một điểm lưu ý là các phương pháp so sánh không phải là các thuật toán học máy, do đó điều kiện thực nghiệm là không như nhau dù cùng sử dụng một tập dữ liệu. Với điều kiện đó, kết quả của MalfomUI so với các hệ thống khác giúp nhận thức được vai trò của tương tác người dùng đối với hiệu quả của hệ thống so khớp.

Luận văn chọn kích thước tập huấn luyện là 100 và 500 mẫu để so sánh bởi vì chúng là những kích thước lý tưởng có thể thoả mãn cả người dùng và hệ thống học. Chi phí để gán nhãn dữ liệu không quá đắt và hiệu quả là chấp nhận được. Malfom-100 có precision 64%, hầu như bằng với các hệ thống khác, nhưng đạt được recall 66, một cải thiện đáng kể so với các hệ thống so sánh. Recall của bảy hệ thống tham gia so sánh tốt nhất là 41% và xấu nhất là 12%. Malfom-500 đạt được tốt hơn tất cả các hệ thống khác ở cả hai chỉ số. Nó đạt được precision 67% và recall 73%. Trong cả hai thử nghiệm, hệ thống đều đạt được f-measure cao hơn các hệ thống khác, lần lượt là 65% và 70%.




| | 5 | 10 | 20 | 30 | 40 | 50 | 100 | 500 | 1000 | 2000 | 3000 | 4000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 52.980 | 53.098 | 55.914 | 59.807 | 60.328 | 60.555 | 64.511 | 69.470 | 69.949 | 70.397 | 70.709 | 70.835 |
| Precision | 43.952 | 43.385 | 51.652 | 58.036 | 58.852 | 62.551 | 63.829 | 67.406 | 67.235 | 67.569 | 68.301 | 68.354 |
| Recall | 45.469 | 54.936 | 54.224 | 59.745 | 60.917 | 56.244 | 65.838 | 73.042 | 74.687 | 74.530 | 73.244 | 73.404 |
| F-Measure | 44.697 | 48.482 | 52.907 | 58.878 | 59.867 | 59.230 | 64.818 | 70.111 | 70.765 | 70.879 | 70.686 | 70.789 |

*Hình 5.2. Đồ thị biểu diễn tác dụng của kích thước tập huấn luyện lên hiệu quả của thuật toán học có giám sát*

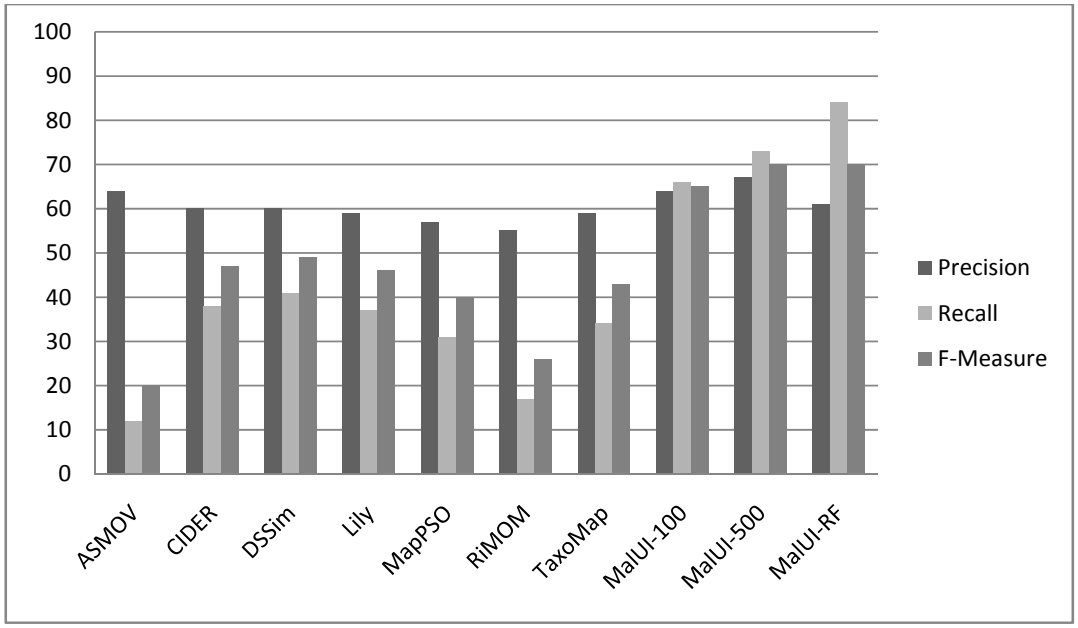
*Bảng 5.1. Bảng kết quả so sánh giữa hệ thống được đề xuất và các hệ thống khác*

| | ASMOV | CIDER | DSSim | Lily | MapPSO | RiMOM | TaxoMap | MalUI-100 | MalUI-500 | MalUI-RF |
|---|---|---|---|---|---|---|---|---|---|---|
| Precision | 64 | 60 | 60 | 59 | 57 | 55 | 59 | 64 | 67 | 61 |
| Recall | 12 | 38 | 41 | 37 | 31 | 17 | 34 | 66 | 73 | 84 |
| F-Measure | 20 | 47 | 49 | 46 | 40 | 26 | 43 | 65 | 70 | 70 |

## 5.3 Thử nghiệm 2 (Học bán giám sát kết hợp học chủ động)

Kế tiếp luận văn đánh giá phương pháp học bán giám sát kết hợp học chủ động với quá trình phản hồi người dùng. Trong thuật toán này, hệ thống chia dữ liệu thành hai cụm và chọn *seed* cặp khái niệm gần tâm mỗi cụm nhất. Người dùng được yêu cầu gán nhãn chúng và chúng được sử dụng để huấn luyện mô hình. Trong *n* lần lặp tiếp theo, người dùng sẽ gán nhãn thêm N mẫu từ các cặp ứng viên do hệ thống học đưa ra. Sau đó, hệ thống sẽ bắt đầu quá trình học bán giám

sát với thuật toán tự học dựa trên bộ học cơ sở. Do cần đáp ứng yêu cầu thoả mãn của người dùng, luận văn sử dụng những tham số có giá trị nhỏ, cụ thể là *seed* = 10, $N = 4$ và $n = 2$. Trong thử nghiệm này, giá trị từ tập tham chiếu sẽ được dùng để cung cấp nhãn cho dữ liệu.




*Hình 5.3. Biểu đồ so sánh hiệu quả của hệ thống được đề xuất và các hệ thống khác*

Bảng 5.2 cũng trình bày kết quả của thử nghiệm thứ hai, đánh giá hệ thống với phản hồi người dùng, trong cột MalfomUI-RF. Trong kịch bản của học bán giám sát kết hợp học chủ động với phản hồi người dùng, hệ thống đạt được precision 61%, recall 84% và f-measure 70%. Lưu ý đến rằng người dùng chỉ cần gán nhãn một ít dữ liệu, tổng cộng 28 cặp dữ liệu, để đạt được kết quả như trên. Bảng 6.2 so sánh kết quả của kịch bản học bán giám sát với kết quả phương pháp học có giám sát với một số kích thước tập huấn luyện xấp xỉ, cụ thể là 30, 40, 50 và 100 mẫu. Phương pháp học bán giám sát được được hiệu quả tốt hơn so với phương pháp học có giám sát trong các trường hợp so sánh, ngoại trừ trường hợp tập so khớp mẫu có 100 mẫu cho có precision tốt hơn nhưng sử dụng phản hồi người dùng lại có recall tốt hơn hẳn và do đó f-measure cao hơn. Học bán giám sát kết hợp với học chủ động trong trường hợp người dùng phản hồi cũng cho kết

quả f-measure ngang ngửa với kết quả của hệ thống khi sử dụng tập dữ liệu huấn luyện lớn (kích thước dữ liệu 500 mẫu) dù có khác biệt về precision và recall của các phương pháp.

*Bảng 5.2. Kết quả so sánh của hệ thống học máy MalfomUI với các điều kiện khác nhau*

| | MalUI-30 | MalUI-40 | MalUI-50 | MalUI-100 | MalUI-RF |
|---|---|---|---|---|---|
| Precision | 58% | 59% | 63% | 64% | 61% |
| Recall | 60% | 61% | 56% | 66% | 84% |
| F-Measure | 59% | 60% | 59% | 65% | 70% |

## 5.4 Thảo luận

Sự biến thiên của kết quả so khớp trong Hình 5.2 cho thấy rằng hệ thống không cần một tập huấn luyện với kích thước rất lớn mới đạt được một kết quả tốt. Với tập huấn luyện 500 mẫu, hệ thống đạt được kết quả xấp xỉ khi sử dụng tập huấn luyện 4000 mẫu. Điều này cho thấy rằng trong một ứng dụng so khớp thực tế, có thể tiết kiệm đáng kể chi phí gán nhãn bằng các thiết kế và sử dụng mô hình học máy một cách cẩn thận.

Đặc biệt, hệ thống đạt được hiệu quả đáng kể trong khi chỉ dùng ít dữ liệu gán nhãn khi được sử dụng với phản hồi người dùng. Tất cả các độ đo đánh giá đều tốt hơn so với các hệ thống khác cũng như với phương pháp học có giám sát với những kích thước kích thước tương đương. Kết quả này chứng tỏ hiệu quả của học bán giám sát và học chủ động trong trường hợp kích thước tập huấn luyện nhỏ. Có thể giải thích lý cho của việc tốt hơn này: phương pháp học này có thể chọn những dữ liệu với phân phối tin cậy hơn từ dữ liệu chưa gán nhãn để người dùng gán nhãn chúng và sau đó đưa chúng vào thuật toán học. Trong trường hợp học có giám sát, hệ thống chỉ có một tập dữ liệu với được rút ra ngẫu nhiên, Kết quả này khác so với kết luận được rút ra từ [14] bởi vì luận văn đã thiết lập một môi trường sử dụng thích hợp đối với phương pháp học bán giám

sát. Môi trường này cũng thuận lợi với người dùng hệ thống và do đó hệ thống học linh hoạt được đề xuất trong luận văn cũng cho thấy khả năng áp dụng thực tế.

Như đã đề cập ở trên, hệ thống học được đề xuất là một sự mở rộng của hệ thống học tổng quát cho bài toán so khớp ontology Malfom được đề xuất trong [11]. Là một hệ thống học tổng quát, Malfom có ưu điểm là tách quá trình xây dựng vector tương tự ra khỏi quá trình học, nhờ đó hệ thống này cho phép hoặc bổ sung thêm những độ đo tương tự hoặc có thể thay thế thay thế thuật toán học hiện tại (Malfom được thử nghiệm với SVM trong [11]) bằng một thuật toán học hiệu quả hơn. Sự khác biệt hệ thống được đề xuất với hệ thống trong [11] là luận văn quan tâm đến vai trò của tương tác người dùng đối với hiệu quả hệ thống, thể hiện ở hai điểm:

- Luận văn nghiên cứu chi tiết ảnh hưởng của tập so khớp mẫu đối với hiệu quả của hệ thống, do đó có thể giúp người thiết kế xác định một kích thước hợp lý của tập so khớp mẫu sao cho vừa thoả mãn yêu cầu của người dùng về chi phí gán nhãn vừa đảm bảo hiệu quả hệ thống.
- Luận văn bổ sung vào hệ thống chiến lược học bán giám sát kết hợp với học chủ động để giải quyết vấn đề tập mẫu huấn luyện ít và tận dụng phản hồi người dùng trong quá tình so khớp. Qua đó, giúp tiết kiệm thêm chi phí người dùng phải bỏ ra trong quá trình tương tác với hệ thống.

Cũng như các hệ thống học máy, hệ thống từ [11] giả định tồn tại một tập dữ liệu gán nhãn lớn, mà điều đó rất tốn chi phí trên thực tế. Hệ thống học linh hoạt tích hợp nhiều chiến lược học khác nhau để tương thích với môi trường người dùng thực tế. Kết quả thực nghiệm đã cho thấy sự tích hợp này có thể giải quyết các yêu cầu của người dùng và có thể giúp cắt giảm chi phí nhân công cần thiết để hỗ trợ cho hệ thống.

## 5.5 Kết luận và Hướng phát triển

Luận văn này đề xuất xây dựng một hệ thống học linh hoạt, sử dụng nhiều chiến lược học khác nhau, để đáp ứng được nhiều điều kiện khác nhau của tương tác

người dùng cho bài toán so khớp ontology. Hệ thống học linh hoạt sử dụng phương pháp học có giám sát và phương pháp học bán giám sát kết hợp học chủ động. Kết quả thử nghiệm cho thấy các phương pháp học máy, nếu được sử dụng trong điều kiện thích hợp có thể giúp các hệ thống so khớp thực tế cải thiện hiệu quả thông qua sự tương tác với người dùng. Đặc biệt, với việc kết hợp phương pháp học bán giám sát và học chủ động, luận văn đã đưa ra một cách thức để tích hợp phản hồi người dùng, vốn là một quá trình được sử dụng rộng rãi trong các hệ thống truy vấn thông tin, vào trong quá trình so khớp. Qua đó, giúp hệ thống có sự quan tâm thích đáng đến mong muốn của người dùng và cải thiện được hiệu quả của hệ thống theo yêu cầu chủ quan của người dùng. Các kết quả thu được khích lệ những nghiên cứu tiếp theo hoàn thiện các chức năng của hệ thống, với mong muốn xây dựng một hệ thống so khớp đầy đủ, thực tế.

Tương tự Malfom, hệ thống được đề xuất cũng có ưu điểm là cho phép tích hợp thêm các độ đo đặc trưng. Hệ thống thử nghiệm sử dụng ba loại độ đo tương tự: tương tự từ vựng, tương tự cấu trúc và tương tự dựa vào từ điển. Một loại độ đo tương tự khác có thể  bổ sung vào hệ thống là độ tương tự dựa trên thể hiện [9]. Mặc dù đến bây giờ độ đo này vẫn chưa nhận được một sự quan tâm đầy đủ [13],  một vài nghiên cứu gần đây cho thấy độ đo tương tự dựa trên thể hiện là một độ đo có triển vọng. Bên cạnh một số độ đo truyền thống [9] đã được thử nghiệm và cho kết quả khả quan trong [13], một số độ đo mới đã được đề xuất cho bài toán này như: κ-statistic [12], Jensen-Shannon divergence [20] hay Wang và cộng sự đề xuất xây dựng độ tương tự dựa trên nội dung thể hiện trong [19]…

Do đó, hướng phát triển tiếp của luận văn là đưa vào hệ thống độ đo dựa trên thể hiện trong trường hợp độ tương tự ngữ nghĩa gặp khó khăn. Một hướng phát triển nữa của hệ thống là ở việc sử dụng thuật toán học bán giám sát. Hệ thống đề xuất sử dụng mô hình tự huấn luyện trong các thử nghiệm. Một mô hình khác có ý tưởng tương tự nhưng cải tiến hơn là mô hình học cộng tác trong đó sử dụng hai bộ học cơ sở. Hai bộ học này có thể cùng một thuật toán học nhưng dựa trên hai tập thuộc tính khác nhau như trong [2] hoặc hai thuật toán học dựa trên một

tập thuộc tính như trong [22]. Hướng phát triển này có triển vọng vì trong so khớp ontology, có thể sử dụng nhiều độ đo tương tự đồng thời, ví dụ độ tương tự dựa vào khái niệm và độ tương tự dựa vào thể hiện.

Với những nhận xét trên, luận văn nhận thấy khả năng tiếp tục phát triển của các hướng nghiên cứu trong việc áp dụng học máy vào trong bài toán so khớp ontology nhằm hoàn thiện một hệ thống so khớp thật sự, có khả năng áp dụng trong thực tế.

# TÀI LIỆU THAM KHẢO


[1] Agrawal, R., Srikant, R. (2001), On integrating catalogs. In: *Proceedings of the Tenth International World Wide Web Conference (WWW-10)*, pp. 603–612

[2] Blum, A. and Mitchell, T. (1998), Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*. Madison, WI, pp. 92-100.

[3] Brickley, D., Guha, R.V. (1999), *Resource Description Framework (RDF) Schema Specification*. W3C Proposed Recommendation.

[4] Caracciolo, C. Euzenat, J., Hollink, L., Ichise, R., Isaac, A., Malaisé, V. Meilicke, C., Pane, J., Shvaiko, P., Stuckenschmidt, H., Šváb-Zamazal, O., and Svátek, V. (2008), Results of the Ontology Alignment Evaluation Initiative 2008. In *Proceeding of The Third International Workshop on Ontology Matching*.

[5] Corcho, O., Gomez-Perez, A. (1998), A Roadmap to Ontology Specification Languages, In *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, pages 80-96.

[6] Cozman, F. G., & Cohen, I. (2002). Unlabeled data can degrade classification performance of generative classifiers. *Int. Conf. of the Florida Artificial Intelligence Research Society* (pp. 327–331). Pensacola, Florida.

[7] Doan, A., Madhavan, J., Dhamankar, R., *et al.* (2003), Learning to match ontologies on the Semantic Web. *VLDB Journal: Very Large Data Bases, 12(4):303–319*.

[8] Ehrig, M., Staab, S. Sure, Y. (2005), Bootstrapping Ontology Alignment Methods with APFELi. In: *Proceedings of the 4th International Semantic Web Conference (ISWC)*, pp. 186-200.

[9] Euzenat, J., and Shvaiko, P. (2007), *Ontology Matching*. Springer.

[10] Falconer, S., Storey, M. (2007), A cognitive support framework for ontology mapping. In *Proceedings of ISWC/ASWC*.

[11] Ichise, R. (2008), Machine Learning Approach for Ontology Mapping Using Multiple Concept Similarity Measures. In: *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science*, pp. 340–346

[12] Ichise, R., Takeda, H., Honiden, S., (2003), Integrating multiple internet directories by instance-based learning. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pp. 22–28.

[13] Isaac, A., van derMeij, L., Schlobach, S.,Wang, S. (2007), An empirical study of instance-basedontology matching. In: *Proceedings of the 6th International Semantic Web Conference (ISWC)*, pp. 253-266.

[14] Jeong, B., Lee, D., Cho, H., Lee, J. (2008), A novel method for measuring semantic similarity for XML schema matching. *Expert Systems with Applications.*

[15] Mocan, A., Cimpian, E., Kerrigan, M. (2006), Formal model for ontology mapping creation. In: *Proceedings of International Semantic Web Conference*.

[16] Seeger. M. (2000), *Learning with labeled and unlabeled data*. Technical report, University of Edinburgh.

[17] Shvaiko, P., Euzenat, J. (2008), Ten Challenges for Ontology Matching. In: *Proceedings of The 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, pp. 1164-1182

[18] Tian, Q.; Yu, J.; Xue, Q.; Sebe, N. (2004), A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval, *Multimedia and Expo, 2004. ICME apos;04. 2004 IEEE International Conference on Volume 2, Issue ,* pp. 1019 - 1022

[19] Wang, S., Englebienne, G., Schlobach, S. (2008), Learning Concept Mappings from Instance Similarity. In: *Proceedings of the 7th International Semantic Web Conference (ISWC)*, pp. 339-355.

[20] Wartena, C., Brussee, R. (2008), Instanced-Based Mapping between Thesauri and Folksonomies. In: *Proceedings of the 7th International Semantic Web Conference (ISWC)*, pp. 356-370.

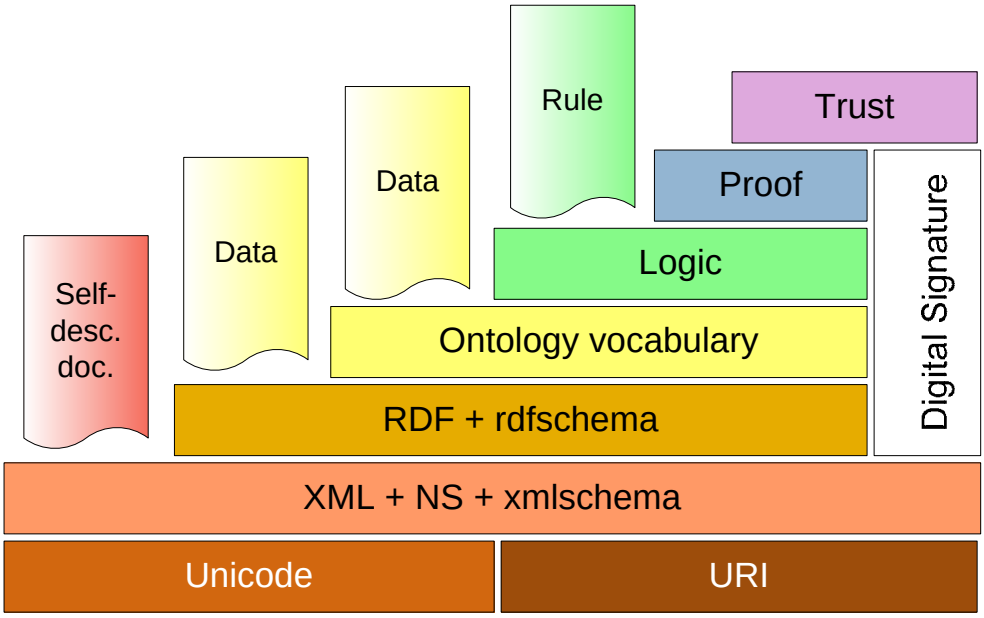
[21] Wu, Z., and Palmer, M. (1994), Verb semantics and lexical selection. In *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, NewMexico State University, Las Cruces, New Mexico.

[22] Zhou, Z.H., Chen, K.J., Dai, H.B. (2006), Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Trans. on Information Systems 24(2)*, pp. 219–244.

[23] Zhu, X. (2006), *Semi-supervised learning literature survey*. Computer Science Technical Report 1530, University of Wisconsin – Madison.

# PHỤ LỤC A

Ontology là một thành phần cấu tạo nên kiến trúc nền tảng của Web ngữ nghĩa. Phụ lục này sẽ trình bày sơ lược về kiến trúc của Web ngữ nghĩa. Hình 2 biểu diễn của mô hình kiến trúc của Web ngữ nghĩa, mô hình này do Tim Berners-Lee, cha đẻ của Web đề xuất[6]. Trong mô hình này, Web ngữ nghĩa bao gồm 7 lớp ngôn ngữ được chồng lên nhau, các lớp này được sử dụng để bảo đảm độ an toàn và giá trị thông tin được biểu diễn ở mức tốt nhất.




*Hình 2. Kiến trúc Semantic Web*

Vai trò của các lớp trong Web ngữ nghĩa được tóm lược như sau:

- Lớp *Unicode và URI*: nhằm bảo đảm việc sử dụng tập kí tự quốc tế và cung cấp phương tiện nhằm định danh các đối tượng trong Web ngữ nghĩa. URI - *Uniform Resource Identifier*, URI là một định danh Web ví dụ các chuỗi bắt đầu bằng "http" hay "ftp". Bất kỳ ai cũng có thể tạo một URI, và có quyền sở hữu chúng vì vậy chúng đã hình thành nên một công nghệ nền tảng lý tưởng để thông qua đó xây dựng một hệ thống mạng toàn

[6] http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html

cầu. Khi sử dụng URI, chúng ta có thể dùng cùng một cách đặt tên đơn giản để đề cập đến các tài nguyên dưới các giao thức khác nhau: HTTP, FPT, GOPHER, EMAIL,.... Một dạng thức quen thuộc của URI là URL - *Uniform Resource Locator*. Một URL là một địa chỉ cho phép chúng ta thăm một trang Web, như: http://www.w3.org/Addressing/... Mặc dù thường được đề cập đến như URL, nhưng URI cũng được đề cập đến như các khái niệm trong Web ngữ nghĩa để chỉ các tài nguyên. URI là nền tảng của Web ngữ nghĩa. Trong khi mọi thành phần khác của Web gần như có thể được thay thế nhưng URI thì không. URI liên hệ các thành phần của Web lại với nhau.

- Lớp *XML* cùng với các định nghĩa về *namespace* và *schema* (*lược đồ*) bảo đảm rằng chúng ta có thể tích hợp các định nghĩa Web ngữ nghĩa với các chuẩn dựa trên XML khác. XML là một mở rộng của ngôn ngữ đánh dấu cho các các cấu trúc tài liệu bất kỳ, trái với HTML, là một loại ngôn ngữ đánh dấu chỉ dành cho các loại tài liệu siêu liên kết. Một tài liệu XML bao gồm một tập các thẻ đóng và thẻ mở được lồng vào nhau, ở đó mỗi một thẻ có một cặp các thuộc tính và giá trị. Phần cốt yếu của tài liệu XML là bộ từ vựng của các thẻ và sự kết hợp được cho phép thì không cố định, nhưng có thể được xác định thông qua mỗi ứng dụng XML. XML cho phép người dùng thêm cấu trúc tùy ý cho các tài liệu của họ nhưng không đề cập gì đến ý nghĩa của các cấu trúc. Tên các tag không cung cấp ý nghĩa. Web ngữ nghĩa chỉ dùng XML cho mục đích cú pháp.
- Lớp *RDF và RDFS (RDF Schema)* giúp ta có thể tạo các phát biểu (*statement)* để mô tả các đối tượng với những từ vựng và định nghĩa của URI, và các đối tượng này có thể được tham chiếu đến bởi những từ vựng và định nghĩa của URI ở trên. Đây cũng là lớp mà chúng ta có thể gán các kiểu (*type*) cho các tài nguyên và liên kết và cũng là lớp quan trọng nhất trong kiến trúc Web ngữ nghĩa.

- Lớp *Ontology*: hỗ trợ sự tiến hóa của từ vựng vì nó có thể định nghĩa mối liên hệ giữa các khái niệm khác nhau. Đây là lớp trọng tâm trong nghiên cứu của luận văn.
- Lớp *Digital Signature*: được dùng để xác định chủ thể của tài liệu, ví dụ tác giả của một tài liệu hay một lời tuyên bố…
- Các lớp *Logic, Proof, Trust***:** đang trong giai đoạn nghiên cứu và các thể hiện của các ứng dụng giản đơn đang được xây dựng. Lớp *Logic* cho phép viết ra các luật (*rule*) trong khi lớp *Proof* thi hành các luật và cùng với lớp *Trust* đánh giá nhằm quyết định ứng dụng nên hay không nên tin tưởng/chấp nhận (trust) chứng cớ (proof).

# PHỤ LỤC B

Phụ lục này giới thiệu thông tin về một số hệ thống so khớp ontology phổ biến và có tham dự vào cuộc thi OAEI 2008. Đây là các hệ thống so khớp tự động và được phát triển từ các phòng nghiên cứu, các trường đại học ở khắp nơi trên thế giới. Các hệ thống bao gồm:

- Anchor-Flood: thuật toán so khớp được phát triển bởi phòng nghiên cứu Knowledge Data Engineering thuộc Đại học Công nghệ Toyohashi[7], Nhật Bản. Thuật toán Anchor-Flood được thiết kế chủ yếu nhắm vào việc so khớp một cách hiệu quả các ontology có kích thước lớn hay giữa một ontology có kích thước lớn và một ontology có kích thước nhỏ hơn. Thuật toán không so sánh mọi cặp thực thể giữa hai ontology mà thay vào đó sử dụng một quá trình so khớp cục bộ trên một tập các khối nhỏ khái niệm. Nhờ chiến lược này thuật toán rút ngắn đáng kể thời gian thực thi trên các ontology lớn. Đặc biệt, đây là thuật toán có thời gian thực thi nhanh nhất trong vòng thi *anatomy* tại cuộc thi OAEI 2008.
- AROMA: được phát triển bởi Trung tâm Nghiên cứu INRIA Grenoble - Rhône-Alpes[8], thuộc INRIA, Học viện nghiên cứu Khoa học Máy tính và Điều khiển Quốc gia Pháp. AROMA là một phương pháp so khớp lai, mở rộng và bất đối xứng được thiết kế để tìm các mối liên hệ (tương đường và xếp gộp) giữa các thực thể từ hai phân loại văn bản (thư mục web hay ontology OWL). Phương pháp này dùng mô hình luật kết hợp và một độ đo sự quan tâm thống kê được dùng trong ngữ cảnh này. AROMA dựa trên giả định sau: Một thực thể A sẽ cụ thể hơn hay tương đương với một thực thể B nếu từ vựng dùng để mô tả A, các con cháu của nó, và các thể hiện của nó có khuynh hướng được bao hàm bởi B.

[7] http://www.kde.ics.tut.ac.jp/news.en.html
[8] http://www.inrialpes.fr/05247154/1/fiche___pagelibre/&RH=ACCUEIL_EN?RF=1143810810877

- ASMOV (Automated Semantic Mapping of Ontologies with Validation) là một công cụ so khớp được phát triển bởi công ty INFOTECH Soft[9], Mỹ, sử dụng các tri thức ngữ nghĩa được chứa đựng trong các cặp ontology để rút trích các tương ứng giữa các thực thể của chúng.
- CIDER: hệ thống so khớp được phát triển bởi nhóm nghiên cứu về các Hệ thống Thông tin Phân tán – Distributed Information Systems (SID)[10], đại học Zaragoza, Tây Ban Nha. CIDER so sánh mỗi cặp tên bằng cách rút ra ngữ cảnh ontology dựa trên một độ sâu nào đó và sau đó kết hợp các kỹ thuật so khớp ontology nguyên tố khác nhau (ví dụ, khoảng cách ngữ ngĩa và mô hình không gian vector).
- DSSim: được phát triển tại Viện Phương tiện Tri thức (Knowledge Media Institute – KDI)[11], Anh. DSSim sử dụng thuyết Dempster-Shafer để mô hình hoá và lập luận với các giá trị không chính xác trong các so khớp. Hệ thống này dùng phương pháp xác suất để kết hợp nhiều độ đo khác nhau như độ đo dựa trên cấu trúc, ngữ nghĩa, và từ vựng để tìm các so khớp.
- SPIDER: cũng được phát triển tại KDI, Anh. SPIDER sử dụng hai hệ thống con rời CIDER và Scarlet. Điểm đặc biệt của hệ thống này là bên cạnh các ánh xạ tương đương trong so khớp kết quả, nó còn cung cấp nhiều loại ánh xạ không tương tư khác nhau ví dụ quan hệ xếp gộp, quan hệ rời nhau, và quan hệ theo tên.
- GeRoMe: được phát triển bởi Informatik 5, Đại học RWTH Aachen[12], Đức. Đây là một chức năng trong bộ phần mềm GeRoMeSuite, một hệ thống  quản lý mô hình tổng quát cung cấp một số chức năng quản lý các loại mô hình dữ liệu phức tạp ví dụ như tích hợp các lược đồ, định nghĩa và thực thi các ánh xạ lược đồ, biến đổi mô hình, và so khớp. GeRoMe

[9] http://support.infotechsoft.com
[10] http://sid.cps.unizar.es
[11] http://kmi.open.ac.uk
[12] http://www.dbis.rwth-aachen.de

thực hiện chức năng so khớp các lược đồ XML dựa vào một siêu mô hình tổng quát.

- Lily: được phát triển tại Trường Khoa học và Kỹ nghệ Máy tính, Đại học Đông Nam, Trung Quốc. Lily sử dụng một các chiến lược so khớp lai ghép để thực hiện nhiệm vụ so khớp. Lily gồm bốn chức năng chính: so khớp ontology tổng quát, so khớp ontology kích thước lớn, so khớp ontology ngữ nghĩa và dò lỗi so khớp.
- MapPSO: được phát triển tại Khoa Kỹ nghệ Thông tin và Khoa học Máy tính[13], Đại học Trento, Ý. MapPSO là một công cụ so khớp ontology Alignment, sử dụng Discrete Particle Swarm Optimisation. Xem bài toán so khớp như là một bài toán tối ưu hoán, MapPSO dùng một bầy đàn để tìm kiếm so khớp tối ưu. Thuật toán là song song và thích ứng một cách tự nhiên với các cấu trúc song song.
- RiMOM (Risk Minimization based Ontology Mapping): được phát triển tại Nhóm Kỹ nghệ Tri thức, Đại học Thanh Hoa[14], Trung Quốc. RiMOM là một công cụ so khớp ontology bằng cách kết hợp nhiều chiến lược khác nhau như so khớp dựa trên khái niệm, so khớp dựa trên cấu trúc,… nhằm tìm kết quả so khớp tối ưu.
- SAMBO và SAMBOdtf: là hai hệ thống so khớp ontology được phát triển bởi các nhà nghiên cứu thuộc Khoa Khoa học Máy tính và Thông tin (IDA)[15], đại học Linköping, Thuỵ Điển. Đây là các hệ thống so khớp ontology với mục tiêu tập trung vào các ontology sinh-y học. Các hệ thống trợ giúp người dùng trong việc so khớp và trộn các ontology bằng cách đưa ra các so khớp đề nghị được tính toán sẵn bởi hệ thống. Hệ thống này được cài đặt một số chiến lược so khớp bao gồm các thuật toán so khớp ngôn ngữ, các chiến lược dựa trên cấu trúc và các thuật toán dựa trên học máy.

[13] http://www.dit.unitn.it/
[14] http://keg.cs.tsinghua.edu.cn/english.htm
[15] http://www.ida.liu.se/department/index.en.shtml

- TaxoMap: được phát triển tại Phòng nghiên cứu Khoa học Máy tính (Laboratoire de Recherche en Informatique – LRI)[16], Đại học Paris-Sub 11, Pháp. TaxoMap là một công cụ so khớp nhắm tời việc khám phá các tương ứng phong phú giữa các khái niệm. Nó thực hiện một so khớp hướng từ một ontology nguồn đến một ontology đích và sử dụng các nhãn cùng các mô tả lớp con. Ba loại tương ứng được tính toán là: tương đương, lớp con, và xấp xỉ.

[16] http://www.lri.fr/presentation_en.php