ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Vũ Bội Hằng

PHÁT HIỆN QUAN HỆ NGỮ NGHĨA NGUYÊN NHÂN-KẾT QUẢ TỪ CÁC VĂN BẢN

LUẬN VĂN THẠC SỸ

Hà Nội – 2005

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Vũ Bội Hằng

PHÁT HIỆN QUAN HỆ NGỮ NGHĨA NGUYÊN NHÂN-KẾT QUẢ TỪ CÁC VĂN BẢN

Ngành: Công nghệ thông tin.
Mã số: 1.01.10

LUẬN VĂN THẠC SỸ

NGƯỜI HƯỚNG DẪN KHOA HỌC:
PGS.TS HÀ QUANG THỤY

Hà Nội - 2005

Những lời đầu tiên

Với những dòng chữ đầu tiên này, tôi xin dành để gửi lời cảm ơn chân thành và sâu sắc nhất tới thầy giáo, tiến sỹ Hà Quang Thụy - người đã tận tình hướng dẫn, chỉ bảo và tạo cho tôi những điều kiện tốt nhất từ khi bắt đầu cho tới khi hoàn thành công việc của mình.

Đồng thời, xin trân trọng gửi lời cảm ơn tới tập thể các thầy giáo-Bộ môn Các hệ thống thông tin-trường Đại học Công nghệ-Đại học Quốc gia Hà Nội đã tạo cho tôi một môi trường làm việc đầy đủ và thuận tiện.

Xin cảm ơn tất cả những người thân yêu trong gia đình tôi cùng toàn thể bạn bè, những người đã luôn mỉm cười và động viên tôi mỗi khi vấp phải những khó khăn, bế tắc.

Cuối cùng, xin chân thành cảm ơn Thạc sỹ Nguyễn Phương Thái (Bộ môn Khoa học máy tính-trường đại học Công nghệ- Đại học Quốc gia Hà Nội), nghiên cứu sinh Vũ Hải Long (University of Illinois at Urbana Champaign- United State), anh Đỗ Mạnh Hùng (công ty Elcom), những người đã đem đến cho tôi những lời khuyên vô cùng bổ ích để giúp tháo gỡ những khó khăn, vướng mắc trong quá trình làm luận văn.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

MỞ ĐẦU

World Wide Web là một kho thông tin khổng lồ với những tiềm năng không giới hạn. Có rất nhiều tiềm năng của World Wide Web mà cho đến nay vẫn chưa được khai thác một cách hiệu quả. Các văn bản Web được làm ra với mục đích ban đầu là dành cho con người đọc. Nhưng với số lượng khổng lồ của các trang Web trên Internet, một người có dành cả đời mình cũng sẽ không bao giờ đọc hết tất cả những trang Web này để thu được đầy đủ các tri thức cần thiết. Nhận thức được vấn đề này, có rất nhiều hướng nghiên cứu đã hình thành, thu hút nhiều nhóm nhà khoa học trên thế giới, nhằm mục đích sử dụng máy tính để hỗ trợ con người trong việc thu thập thông tin và tổng hợp tri thức từ các trang Web trên Internet. Ví dụ như việc áp dụng các kỹ thuật Data Mining để khai thác thông tin từ các văn bản Web, công nghệ Agent trong kinh doanh trực tuyến… Tuy nhiên trong thời gian vừa qua, những hướng nghiên cứu này chủ yếu mới chỉ tập trung vào việc khai thác thông tin dựa trên các từ vựng đơn lẻ hoặc dựa trên một số cấu trúc cố định của trang Web. Thật là khó khăn để máy tính có thể truy cập và tổng hợp các thông tin trong các văn bản về phương diện ngữ nghĩa. Gần đây, một số hướng nghiên cứu mới đã được mở ra nhằm mục đích khai thác khả năng kết hợp nội dung trang Web với các thông tin ngữ nghĩa, để tạo ra Semantic Web. Semantic Web không phải là một loại Web mới tách biệt mà là sự nâng cấp của Web hiện tại (thế hệ Web thứ ba), ở đó các thông tin ngữ nghĩa được xác định tốt hơn và được kết hợp vào cùng với trang Web. Như vậy, việc đọc và hiểu các trang Web không chỉ thi hành được bởi con người mà còn có thể được thi hành bởi máy tính.

Semantic Web ra đời đòi hỏi một loạt các công nghệ kèm theo nó. Một trong số những công nghệ quan trọng nhất đối với Semantic Web là Ontology. Thành phần cơ bản của Ontology là một tập hợp các đối tượng (hay còn gọi là các khái niệm) với các thuộc tính của các đối tượng và tập hợp các mối quan hệ giữa các đối tượng đó. Việc xây dựng Ontology trong một miền ứng dụng là quá trình tổng hợp tri thức trong miền ứng dụng đó. Công việc này đòi hỏi những người xây dựng ontology phải có những hiểu biết và tri thức nhất định để tìm ra đầy đủ đối tượng, thuộc tính và quan hệ.

Xuất phát từ nhu cầu nghiên cứu các phương pháp hỗ trợ trong việc xây dựng các Ontology cho Semantic Web, luận văn trình bày một phương pháp phát hiện mối quan hệ ngữ nghĩa nguyên nhân-kết quả dựa trên ý tưởng nghiên cứu của bài toán Semantic Role (CoNLL Share Task 2004 [31]) và thuật toán khai phá quan hệ nguyên nhân-kết quả mà Corina Roxana Girju đã tiến hành (Luận án Tiến sỹ 2002 [11]). Kết quả tìm được của thuật toán chính là những thông tin cần thiết hỗ trợ trong việc phát hiện các đối tượng mới và mối quan hệ về mặt ngữ nghĩa nguyên nhân-kết quả của các đối tượng này trong quá trình xây dựng Ontology.

Ngoài phần giới thiệu, kết luận và các phụ lục. Luận văn được chia thành 3 chương chính:

Chương 1 - Tổng quan về Semantic Web. Giới thiệu một cách tổng quan những nhu cầu dẫn đến sự ra đời của thế hệ Web thứ ba (Semantic Web). Những khái niệm cơ bản và những công nghệ thiết yếu để phát triển Semantic Web cũng được trình bày trong chương này.

Chương 2 – Quan hệ nguyên nhân-kết quả và thuật toán phát hiện quan hệ nguyên nhân-kết quả. Chương này đi sâu vào phân tích cấu trúc quan hệ ngữ nghĩa nguyên nhân-kết quả trong ngôn ngữ của con người và cấu trúc thể hiện của nó trong văn bản. Thông qua đó luận văn trình bày một thuật toán nhằm phát hiện quan hệ nguyên nhân-kết quả từ tập các văn bản dựa vào tần suất xuất hiện của các cặp danh từ trong những câu chứa động từ chỉ nguyên nhân.

Chương 3 – Kết quả cài đặt thử nghiệm thuật toán. Chương này trình bày các kết quả thực nghiệm về thuật toán phát hiện quan hệ nguyên nhân - kết từ các văn bản. Chương trình cài đặt thử nghiệm cho thuật toán được viết trên ngôn ngữ Java. Thông qua các nhận xét về giá trị các độ đo đánh giá, kết quả thực hiện chương trình là khả quan.

Phần Kết luận trình bày tổng hợp các kết quả thực hiện luận văn và phương hướng nghiên cứu tiếp theo về các nội dung của luận văn.

Mặc dù đã có một môi trường làm việc tương đối đầy đủ và thuận tiện, nhưng luận văn chắc hẳn sẽ không tránh khỏi có nhiều sai sót. Rất mong được sự đóng góp ý kiến, nhận xét để tôi có thể hoàn thiện được kết quả làm việc của mình.

CHƯƠNG 1 - TỔNG QUAN VỀ SEMANTIC WEB

1.1. Giới thiệu

Internet ra đời và đã mau chóng trở thành một kho thông tin khổng lồ. Hiện nay, trên Internet có hàng tỉ các trang Web được hàng trăm triệu người trên khắp thế giới sử dụng [18,20,24]. Tuy nhiên, khi lượng thông tin trên Internet ngày càng tăng thì cũng đồng nghĩa với việc tìm kiếm, khai thác, tổ chức, truy cập và duy trì thông tin ngày càng trở nên khó khăn hơn đối với người sử dụng.

Chúng ta xem xét một ví dụ. Trong một trường hợp tìm kiếm trên Internet, người sử dụng muốn tìm kiếm trang chủ của *Mr và Mrs. Cook*. Tất cả những thông tin mà người sử dụng có thể nhớ được là tên họ của hai người này là *Cook*, cả hai người đó cùng làm việc cho một ông chủ, là một người có liên quan tới một tổ chức có tên là "*ARPA-123-4567"*. Đây chắc chắn là những thông tin hữu ích để tìm ra trang chủ của những người này, theo một cơ sở tri thức có cấu trúc hợp lý chứa đựng tất cả các nhân tố có liên quan. Có vẻ như điều đó đã đủ những thông tin để tìm ra trang chủ của họ bằng cách tìm kiếm trên World Wide Web. Nhưng khi tìm kiếm, lại xảy ra các tình trạng sau:

- Sử dụng danh mục Web có sẵn, người sử dụng có thể tìm ra trang chủ của ARPA nhưng ở đó có hàng trăm người "thầu phụ" và các "nhóm nghiên cứu" đang làm việc cho chi nhánh *"123-4567"*
- Nếu tìm kiếm theo từ khoá *"Cook"* thì kết quả sẽ trả lại hàng nghìn trang Web nói về *"Nấu ăn"*.

- Nếu tìm kiếm một trong hai cụm từ *"ARPA "* và *"123-4567"* thì có hàng trăm kết quả trả về. Còn nếu tìm kiếm cho cả ba từ khoá trên thì sẽ trả về kết quả rỗng.

Vậy thì giải quyết trường hợp này như thế nào?

Tình trạng trên là khá phổ biến đối với nhiều trường hợp tìm kiếm trên World Wide Web [18,19]. Vấn đề chính ở đây là do dữ liệu Web có quá ít sự tổ chức ngữ nghĩa. Khi mà Web càng ngày càng được mở rộng thì việc thiếu tổ chức ngữ nghĩa như vậy sẽ làm cho việc tìm kiếm thông tin càng ngày càng khó, thậm chí nếu có thêm cả những kỹ nghệ xử lý ngôn ngữ tự nhiên, cơ chế đánh chỉ mục…

Tóm lại, hiện nay vẫn chưa có một cách tìm kiếm hiệu quả nào trên WWW [18,19] để trả lời câu truy vấn có dạng như :

Find webpage for all x,y and e such that

X is a person, y is a person, z is a person

Where

lastName (x,"Cook") and

lastName (y, "Cook") and

employee (z,x) and

employee (z,y) and

married (x,y) and

involvedIn (z, "ARPA 123-4567")

- ⇨ Sự thiếu khả năng hiểu khung cảnh của các từ và các mối quan hệ giữa các thuật ngữ tìm kiếm giải thích tại sao trong nhiều trường hợp máy tìm kiếm lại trả về kết quả tìm kiếm sai trong khi lại không tìm thấy những tài liệu mong muốn [18,19,20,24].
- ⇨ Nếu các máy tìm kiếm có thể hiểu được nội dung ngữ nghĩa của các từ, hoặc hơn thế nữa, nó có thể hiểu được cả mối quan hệ về mặt ngữ nghĩa giữa các từ đó thì độ chính xác tìm kiếm sẽ được cải thiện rất nhiều [19,24].
- ⇨ Đây chính là một trong những nguyên nhân dẫn đên sự ra đời của thế hệ Web thứ ba: *Semantic Web[24]*.

1.2. *Khái niệm Semantic Web*

Tim Berners-Lee (người phát minh ra Web) đưa ra định nghĩa Semantic Web như sau:

"Bước đầu tiên là đặt dữ liệu trên Web theo một định dạng mà máy tính có thể hiểu được, hoặc chuyển thành định dạng mà máy tính có thể hiểu được. Điều này tạo ra một loại Web gọi là Semantic Web - là một Web dữ liệu mà có thể được xử lý được trực tiếp hoặc gián tiếp bằng máy tính." [24]

Semantic Web không phải là một Web riêng biệt mà nó chỉ là một sự mở rộng của Web hiện tại, mà ở đó có các thông tin về ngữ nghĩa nhiều hơn, làm cho máy tính và con người có thể phối hợp làm việc tốt hơn [19,24].

Semantic Web không phải chỉ dành cho World Wide Web. Nó kèm theo một tập hợp các công nghệ mà cũng có thể làm việc trên intranet của nội bộ các công ty, doanh nghiệp…[20,24]

1.3. Các ứng dụng của Sematic Web

Semantic Search engine. Cải thiện tìm kiếm là một trong rất nhiều những lợi ích tiềm năng của Sematic Web. Hầu hết các cơ chế tìm kiếm hiện nay trên World Wide Web thường là một trong ba cách tiếp cận sau:

+ Đánh chỉ mục cho các từ khoá [1,4,16].

+ Phân mục bằng tay [11,16] .

+ Sử dụng các cơ chế đặc biệt để thu thập các thông tin ngữ nghĩa từ các trang Web (nhưng rất bị hạn chế) [2,14,16].

Mỗi cách tiếp cận trên đều có nhược điểm. Đánh chỉ mục các từ khoá thì chỉ liên kết với các từ vựng mà không hiểu được ngữ nghĩa của chúng nên có thể gây ra sự nhầm lẫn (như trong ví dụ ở phần giới thiệu chương). Trong khi đó, việc phân mục bằng tay đòi hỏi phải tiêu tốn rất nhiều nhân công và thời gian. Còn việc sử dụng một số cơ chế đặc biệt để thu thập thông tin ngữ nghĩa thì lại rất bị hạn chế do các trang Web mang rất ít thông tin ngữ nghĩa hoặc còn phải phụ thuộc vào cách bố trí theo một số cấu trúc nhất định của các trang Web.

Không có một cách tiếp cận nào trong số những cách tiếp cận ở trên (trừ cách tiếp cận cuối cùng nếu xét trong một miền ứng dụng cụ thể) cho phép suy luận được mối quan hệ của các trang Web (ngoại trừ mối quan hệ giữa các

link). Vì vậy mà các truy vấn theo kiểu như trong ví dụ ở phần giới thiệu là không thể thực hiện được.

=> Giải pháp cho vấn đề này chính là Semantic Web.

Thay vì cố gắng để thu thập các tri thức từ các trang HTML hiện tại, chúng ta hãy kết gán trực tiếp các thông tin ngữ nghĩa cho các trang HTML, làm cho nó trở thành đơn giản để máy tính có thể tự xử lý các thông tin về mặt ngữ nghĩa mà không cần tới sự hỗ trợ của con người [6,19,20].

Agent Internet [19,24]: Các Agent Internet, là các chương trình tự trị mà tương tác với Internet, cũng có thể có hiệu quả hơn nhiều nếu chúng được hoạt động trên môi trường Sematic Web. Để thực hiện một mục đích nào đó, một Agent Internet có thể yêu cầu phải hiểu các trang Web để thi hành các dịch vụ Web. Về mặt lý thuyết, một agent như thế có thể thực hiện việc bán hàng, tham gia trong một cuộc bán đấu giá hoặc xếp lịch cho một kỳ nghỉ…Ví dụ: một Agent có thể được yêu cầu đặt chỗ cho một chuyến du lịch ở Jamaica, và Agent sẽ đặt vé máy bay, tìm một xe car để thuê và đặt một phòng ở khách sạn. Tất cả phải dựa trên giá cả rẻ nhất hiện có và phù hợp với nhu cầu. Mặc dù đã tồn tại những Agent có thể thực hiện được một vài nhiệm vụ như vậy, nhưng chúng được xây dựng để hoạt động trên chỉ một tập hữu hạn các trang Web biết trước và phải phụ thuộc nhiều vào cấu trúc cố định của các trang Web này. Vì vậy, sẽ tốt hơn rất nhiều nếu như với bất kỳ một trang Web, các Agent có thể xem xét ngữ nghĩa của các trang Web thay vì xem xét cấu trúc bố trí cố định của trang Web này.

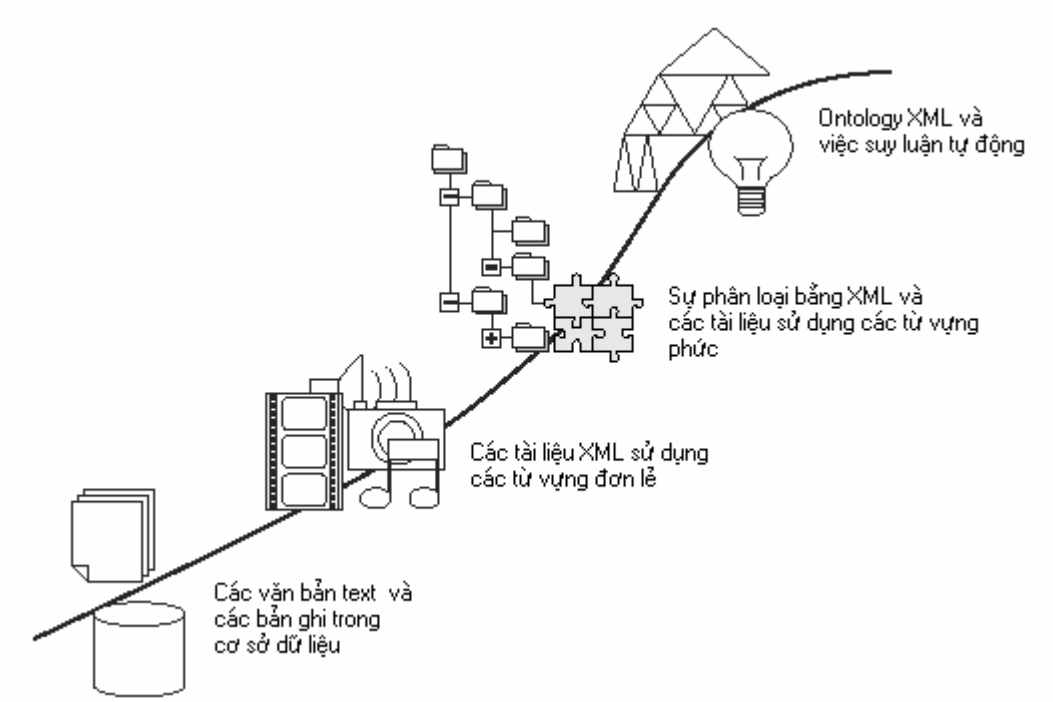
Stovepipe system [24]: stovepipe system là một hệ thống mà ở đó thì tất cả các thành phần đều là các mạch điện tử làm việc với nhau. Vì vậy, các

thông tin chỉ là các dòng trong các stovepipe mà không thể được chia sẻ bởi một hệ thống khác hoặc một tổ chức khác mà cần những thông tin đó. Phân tích các hệ thống stovepipe là cần thiết ở tất cả các tầng kiến trúc thông tin doanh nghiệp. Công nghệ Semantic Web là hiệu quả nhất để phân tích các hệ thống CSDL stovepipe.

1.4. Các công nghệ cần thiết cho Semantic Web

*C*ách để làm cho dữ liệu có thể xử lý được bằng máy tính là làm cho dữ liệu "***thông minh hơn"*** ("smarter").

Hình vẽ sau thể hiện các cấp độ trạng thái phát triển của "dữ liệu thông minh" ("smart data") [24].



Hình 1: Các giai đoạn phát triển của "smart data"

Các văn bản Text và các cơ sở dữ liệu (tiền XML). Hầu hết dữ liệu là độc quyền trong các ứng dụng. Ở đây khái niệm "smart" là khái niệm của ứng dụng chứ không phải của dữ liệu.

Các tài liệu XML sử dụng các từ vựng đơn lẻ. Dữ liệu độc lập với ứng dụng trong một phạm vi ứng dụng cụ thể. Dữ liệu bây giờ thì đủ thông minh để chuyển đổi giữa các ứng dụng trong phạm vi đó. Ví dụ: các chuẩn XML trong: công nghiệp y tế, công nghiệp bảo hiểm…

Sự phân loại bằng XML và các tài liệu với các từ vựng phức. Dữ liệu có thể được kết hợp từ nhiều miền khác nhau và được phân lớp một cách chính xác trong một bảng phân cấp danh mục. Trong thực tế, sự phân lớp có thể được sử dụng để khai thác dữ liệu. Các mối quan hệ giữa các phân mục trong bảng phân cấp danh mục có thể được sử dụng để kết nối dữ liệu. Vì vậy, dữ liệu ở giai đoạn này đủ thông minh để khai thác và kết nối với dữ liệu khác

Ontology và các luật. Ở giai đoạn này, các dữ liệu mới có thể được suy ra từ các dữ liệu đang tồn tại bằng cách sử dụng các luật logic. Điều cốt yếu ở đây là dữ liệu bây giờ đã đủ thông minh để được mô tả cùng với những mối quan hệ cụ thể, và bằng các hình thức tinh vi, phức tạp mà có thể áp dụng được các tính toán logic. Điều này cho phép tách dữ liệu thành các thành phần nhỏ hơn và có thể phân tích sâu hơn. Một ví dụ cho dữ liệu trong giai đoạn này là ta có thể tự động biến đổi một tài liệu trong một miền ứng dụng này thành một tài liệu tương đương trong một miền ứng dụng khác.

1.4.1. XML và Semantic Web

Cho dù HTML là rất phổ biến, nhưng nó hầu như chỉ được thiết kế cho sự biểu diễn đối với con người, và thật là khó để máy khai thác nội dung và

thực hiện xử lý tự động trên các tài liệu. Để giải quyết vấn đề này, World Wide Web Consortium (W3C) đã phát triển eXtensible Markup Language (XML) [17,18,29].

XML về cơ bản là một tập con của Standard Generalized Markup Language (SGML), là một chuẩn được sử dụng bởi cộng đồng xử lý text [18]. SGML là một meta-language, có nghĩa là nó có thể được sử dụng để định nghĩa các ngôn ngữ khác - các ứng dụng SGML. Ưu điểm của SGML là nó độc lập với môi trường, phân tách rõ ràng nội dung và định dạng, và có khả năng xác định liệu các tài liệu có tương thích với các qui tắc cấu trúc hay không. XML vẫn giữ nguyên những đặc tính này, nhưng bớt đi những thứ mà hiếm khi được sử dụng, dễ gây nhầm lẫn, hoặc khó cài đặt.

Công nghệ XML được xây dựng dựa trên các ký tự Unicode (Unicode character) và các URI (Uniform Resource Identfier). Các Unicode character cho phép XML được biên soạn dựa trên các ký tự chuẩn quốc tế. URI được sử dụng để xác định duy nhất các khái niệm (concept) của Sematic Web [24].

XML không phải là một ngôn ngữ, thực chất nó chỉ là một tập hợp các qui luật cú pháp để tạo ra ngôn ngữ đánh dấu mang tính chất ngữ nghĩa trong từng lĩnh vực cụ thể. Mặt khác có thể áp dụng XML để tạo ra một ngôn ngữ mới. Bất cứ một ngôn ngữ nào được tạo ra trên các luật XML (như MathXML) được gọi là một ứng dụng của XML [18].

XML là tầng cơ sở cú pháp của Semantic Web [18]. Tất cả các công nghệ khác mà mang đặc tính của Semantic Web đều được xây dựng dựa trên nền XML.

Cú pháp của XML khá giống với HTML. Điều này không có gì đáng ngạc nhiên vì HTML là một ứng dụng của SGML (ngôn ngữ cha của XML). Giống như HTML (và SGML), XML thêm các thẻ được bao bởi hai dấu ngoặc nhọn vào các dữ liệu văn bản, các thẻ này sẽ cung cấp các thông tin phụ thêm cho đoạn văn bản.

Ví dụ sau đây là một đoạn văn bản với các thẻ đánh dấu XML mô tả việc lưu trữ đĩa CD:

```
<?xml version=” 1.0 ” ?>
<catalog>
    <cd>
        <artist>Cracker</artist>
        <title>Kerosense Hat</title>
        <price currency=”USD”>15.99</price>
    </cd>
    <cd>
        <artist>Phair, Liz</artist>
        <title>Exile in Guyville</title>
        <price currency=”USD”>15.99</price>
    </cd>
    <cd>
        <artist>Soul Coughing</artist>
```

```
        <title>Irresistible Bliss</title>
        <price currency=”USD”>15.99</price>
    </cd>
</catalog>
```

Có ba loại thẻ trong XML: thẻ bắt đầu, thẻ kết thúc và thẻ thành phần. Thẻ bắt đầu đánh dấu bắt đầu mô tả một đối tượng, thẻ kết thúc đánh dấu sự kết thúc mô tả một đối tượng, mỗi thẻ thành phần mô tả một thuộc tính của đối tượng. Thẻ bắt đầu bao gồm một tên và một tập hợp các thuộc tính tuỳ chọn được bao bởi các dấu ngoặc nhọn. Mỗi thuộc tính là một cặp: tên/giá trị, được phân cách bởi dấu “=”. Trong ví dụ trên, thẻ *price* có thuộc tính là *currency*. Một thẻ kết thúc chứa tên giống như thẻ bắt đầu nhưng có dấu gạch chéo “/” đi trước và không có bất cứ một thuộc tính nào. Tất cả các thẻ bắt đầu phải kèm theo một thẻ kết thúc. Các thẻ thành phần giống như thẻ bắt đầu nhưng không có thẻ kết thúc. Thay vào đó, để kết thúc một thẻ thành phần thì dấu gạch chéo “/” được đặt ngay trước dấu đóng ngoặc “>”. Ví dụ, thẻ là một thẻ thành phần. Dữ liệu giữa một thẻ bắt đầu và một thẻ kết thúc được gọi là một *thành phần*. Một thành phần có thể là các thành phần khác, các đoạn văn bản, hoặc chính một đoạn thẻ bắt đầu và thẻ kết thúc khác.

Mặc dù tính mềm dẻo của XML làm cho nó có thể được soạn thảo với các nội dung tuỳ ý một cách nhanh chóng và dễ dàng, nhưng chính tính mềm dẻo này lại là sự khó khăn trong việc xử lý bằng máy tính. Không giống như HTML, XML không cung cấp ngữ nghĩa cho các thẻ, hầu hết các chương trình xử lý đều đòi hỏi tập các thẻ này đã được thống nhất ý nghĩa theo một vài qui

ước chuẩn. Để hỗ trợ việc xử lý bằng máy tính, XML cho phép định nghĩa ngữ pháp cho các thẻ. Những thông tin này chứa trong một file gọi là "document type definition" (DTD) [18,27]. DTD cung cấp cú pháp cho một tài liệu XML, nhưng nó không cung cấp ngữ nghĩa. Ý nghĩa của các thành phần trong DTD có thể được suy luận bởi con người dựa vào tên của nó. Nhưng các công cụ phần mềm thì không thể thu được ngữ nghĩa này một cách độc lập. Vì vậy việc trao đổi các tài liệu XML mà có hai DTD khác nhau trở thành một vấn đề khó khăn.

Một trong những vấn đề khó nhất là việc ánh xạ giữa các cách biểu diễn khác nhau của cùng một khái niệm, đấy chính là vấn đề thống nhất các DTD. Đầu tiên là việc xác định và ánh xạ sự khác nhau trong qui ước đặt tên. Cũng như ngôn ngữ tự nhiên, XML DTDs cũng có các tính chất đồng nghĩa và tính chất nhiều nghĩa của từ. Ví dụ <person> và <individual> có thể là cùng một khái niệm. Hay <spider> có thể chỉ khái niệm của một phần mềm máy tính hay là chỉ một loài động vật (con nhện). Một vấn đề thậm chí còn khó khăn hơn nữa là việc xác định và ánh xạ sự khác nhau về mặt cấu trúc. Chính vì tính mềm dẻo của XML đã làm cho việc thiết kế DTD có nhiều sự lựa chọn. Với cùng một khái niệm, các nhà thiết kế có thể mô tả bằng nhiều cách khác nhau. Ví dụ, ta có ba cách biểu diễn có thể cho tên của cùng một người:

<person>

<name>John Smith</name>

</person>

(Tên là một thành phần của người dưới dạng một chuỗi)

<person>

<name><fname>John</fname><lname> Smith</lname></name>

</person>

(Tên là một thành phần với nội dung là các thành phần)

<person name=”John Smith”>

(Tên là một thuộc tính)

Sự lựa chọn thứ nhất là tên đó là một chuỗi hay là một thành phần của chính cấu trúc đó. Sự lựa chọn thứ hai là liệu tên đó là một thuộc tính hay là một thành phần. Một trong những nguyên nhân dẫn đến vấn đề này là sự thiếu thông tin ngữ nghĩa trong XML. Không có một ý nghĩa cụ thể nào liên quan đến các thuộc tính hay nội dung của các thành phần. Chính sự thiếu thông tin ngữ nghĩa trong các XML DTD làm cho việc kết hợp các tài liệu XML trở nên khó khăn.

1.4.2. Ontology

XML mới chỉ cung cấp cơ sở về mặt cú pháp. Mặt khác, để chia sẻ các tài liệu XML mà đã có thêm nội dung ngữ nghĩa chỉ làm được khi cả hai bên đều hiểu ý nghĩa của các khái niệm ngữ nghĩa trong đó [24].

Ví dụ, nếu có một bên gán nhãn là <price> $1200 </price>, một bên gán nhãn là <cost> $1200 </cost>. Không có cách nào máy sẽ biết cả hai thứ kia là cùng một thứ trừ khi có thêm những công nghệ Sematic Web khác như Ontologies được thêm vào.

“Một ontology định nghĩa các từ vựng và các khái niệm được sử dụng để mô tả và biểu diễn trong một miền tri thức.”[20,24]

Một miền tri thức là các vấn đề xung quanh một chủ đề nào đó. Ví dụ: y học, quản lý buôn bán, sửa chữa ô tô, vật lý, tài chính, địa lý. Các sự mô tả trong một miền tri thức là sự thể hiện của các hoạt động. Ví dụ, mô tả trong lĩnh vực *sửa chữa ô tô*:

- Các thể loại xe (xe mui kín, xe thể thao, …)
- Các thể loại động cơ (gasoline, diesel, điện, động cơ lai).
- Hãng sản xuất (Ford, General Motor, Chevrolet, Nissan, Honda, Volvo, Volkswagen…)
- Những bộ phận tạo thành xe (động cơ, hệ thống phanh, hệ thống làm lạnh, hệ thống điện, thân xe…) và các tính chất của các bộ phận (một động cơ dung tích 4, 6, 8, 12 cylinder)

Điều quan trọng trong việc sửa chữa ô tô là làm thế nào để sửa các loại xe khác nhau, các bộ phận của mỗi loại xe, chẩn đoán và các dụng cụ để chẩn đoán và sửa chữa, ước tính giá thành của việc sửa chữa…Khi mô tả trong một miền tri thức, chúng ta mô tả các sự vật, hiện tượng, các thuộc tính của các sự vật-hiện tượng và mối quan hệ giữa chúng.

Một sự mô tả của một ontology bao gồm các thể loại khái niệm sau [5,28,20,22,24]:

- Các lớp (các sự vật nói chung) trong miền cần quan tâm.
- Các thể hiện (các sự vật cụ thể).
- Các mối quan hệ giữa các sự vật đó.
- Các thuộc tính (và các giá trị thuộc tính) của các sự vật.

- Các chức năng và các tiến trình liên quan đến sự vật.
- Các ràng buộc và các luật liên quan đến các sự vật

Cùng với việc mô tả trong một miền tri thức, chúng ta cũng cần biểu diễn các mô tả. Biểu diễn có nghĩa là ta mã hoá những mô tả này theo một phương pháp nào đó. Các mức độ biểu diễn cần thiết cho một mô hình biểu diễn bao gồm: cú pháp, ngữ nghĩa, và pragmatic [18,22].

Cú pháp: chỉ ra mối quan hệ giữa các ký hiệu (các từ vựng trong ngôn ngữ).

Ngữ nghĩa: chỉ ra mối quan hệ giữa các ký hiệu và các sự vật trong thế giới thực.

Pragmatic: dựa trên cú pháp và ngữ nghĩa để chỉ ra làm thế nào mà các ký hiệu có thể được sử dụng cho một mục đích cụ thể.

Ví dụ một ontology được biểu diễn bằng ngôn ngữ OIL [Horrocks et al, 2000]

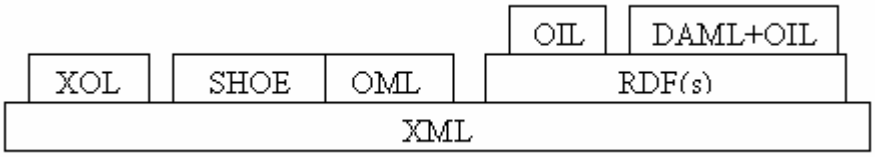
```
class-def animal                    % định nghĩa lớp động vật
class-def plant                     % định nghĩa lớp thực vật
        subclass-of NOT animal      % là một lớp không giao với lớp động vật
class-def tree
        subclass-of plant           % cây là một thể loại thực vật
class-def branch
        slot-constraint is-part-of  % cành cây là một bộ phận của cây
                has-value tree
class-def leaf
        slot-constraint is-part-of  % là là một bộ phận của cành cây
                has-value branch
class-def defined carnivore         % động vật ăn thịt là động vật
        subclass-of animal
        slot-constraint eats        % mà chỉ ăn các động vật khác
                value-type animal
class-def defined herbivore         % động vật ăn cỏ là động vật
        subclass-of animal
```

```
        slot-constraint eats          % mà chỉ ăn thực vật hoặc các bộ phận
                                      của thực vật
            value-type plant OR (slot-constraint is-part-of has-value
        plant)
class-def giraffe                     % hươu cao cổ là động vật
        subclass-of animal
        slot-constraint eats          % và chúng ăn lá
            value-type leaf
class-def lion
        subclass-of animal            % sư tử là động vật
        slot-constraint eats          % nhưng chúng ăn động vật ăn cỏ
            value-type herbivore
class-def tasty-plant                 % thực vật ngon là thực vật được ăn bởi
        subclass-of plant             % cả động vật ăn cỏ và động vật ăn thịt
        slot-constraint eaten-by
            has-value herbivore, carnivore
```

1.5. Các ngôn ngữ Ontology cho Semantic Web

1.5.1. Các ngôn ngữ

Cho tới nay, có nhiều ngôn ngữ Ontology cho Semantic Web đã được phát triển. Hầu hết các ngôn ngữ này dựa trên cú pháp XML, như XOL (Ontology Exchange Language), SHOE và OML (Ontology Markup Language), RDF (Resource Description Framework) và RDF Schema (các ngôn ngữ được đưa ra bởi W3C (World Wide Web Consortium)). Hai ngôn ngữ truyền thống được xây dựng dựa trên nền RDF và RDF Schema là OIL và DAML+OIL [5].



Hình 2: Một số ngôn ngữ ontology

Ontology Exchange Language (XOL) dựa trên XML. Cộng đồng thông tin sinh học ở Mỹ đã thiết kế XOL cho việc trao đổi các định nghĩa ontology giữa một tập hỗn tạp các hệ thống phần mềm trong lĩnh vực sinh học. Các nhà nghiên cứu đã tạo ra ngôn ngữ này sau khi thấy cần phải biểu diễn các thông tin sinh học chuyên môn của họ [5].

Simple HTML Ontology Extension (SHOE). Được phát triển bởi trường đại học Maryland. Nó được tạo ra như là sự mở rộng của HTML, kết hợp chặt chẽ các tri thức mang tính chất ngữ nghĩa trong các tài liệu HTML. Các tri thức được đánh dấu ngay trong các trang HTML. Với SHOE, các Agent có thể thu thập các thông tin giàu ý nghĩa về các trang Web và có thể cải thiện cơ chế tìm kiếm và thu thập tri thức. Tiến trình này bao gồm ba pha: định nghĩa một ontology, đánh dấu các trang HTML với các thông tin tương ứng trong ontology, và xây dựng một agent tự động tìm kiếm thông tin [5,20].

Ontology Markung Language (OML): được phát triển bởi trường đại học Washington, nó phần nào dựa trên SHOE. Vì vậy, OML và SHOE có rất nhiều đặc diểm chung [5].

Resourse Description Framework và RDF Schema: được phát triển bởi W3C để mô tả các tài nguyên Web, cho phép đặc tả ngữ nghĩa dữ liệu dựa trên XML đã được chuẩn hoá [29].

Ontology Interchange Language (OIL): được phát triển bởi dự án OntoKnowledge (www.ontoknowledge.org/OIL), cho phép việc trao đổi ngữ nghĩa giữa các kho dữ liệu Web. Cú pháp và ngữ nghĩa của nó là dựa trên OKBC, XOL và RDF) [12,30]**.**

DARPA Agent Markup Language + OIL (DAML+OIL): được phát triển bởi một tổ chức ở châu Âu (IST) theo dự án DARPA. DAML+OIL có cùng các đối tượng giống như OIL [15,30].

1.5.2. Đặc điểm chung của các ngôn ngữ

Mỗi ngôn ngữ ontology sẽ có một số đặc điểm riêng khác nhau, nhưng tri thức Ontology có thể được đặc tả bởi năm thành phần cơ bản sau: *concept* (thường được tổ chức phân cấp), *relation, function, axiom và instance [5,24]*.

a) Concept

Concept có thể là trừu tượng hoặc cụ thể, đơn hoặc phức, thực tế hoặc là tưởng tượng. Tóm lại, một concept có thể là bất cứ thứ gì mà được nói đến, vì vậy nó cũng có thể là sự mô tả của một công việc, một chức năng, một hành động…Concept còn được gọi là các lớp (class) như trong các ngôn ngữ XOL, RDF, OIL, DAML+OIL, các đối tượng (object) như trong OML, hoặc các phân mục (categories) như trong SHOE.

Concept bao gồm các thuộc tính (attribute). Thuộc tính còn được gọi là slot (như trong XOL), function (như trong OML), hay property (như trong RDF và DAML+OIL), binary relation và role (như trong SHOE và OIL). Các thuộc tính có các loại sau:

- *Instance attribute.* Các thuộc tính mà giá trị của nó có thể khác nhau đối với mỗi instance của một concept.
- *Class attribute.* Các thuộc tính mà giá trị của nó được kèm theo với mỗi concept. Có nghĩa là giá trị của nó sẽ là giống nhau cho tất cả các thể instance của một concept.

- *Local attribute.* là các thuộc tính có cùng tên được kèm theo cho concept khác nhau. Ví dụ: hai concept Bàn và Ghế có thể có cùng thuộc tính Màu sắc.
- *Global attribute.* là thuộc tính được áp dụng cho tất cả các concept của ontology đó.

Instance attribute và class attribute thường được sử dụng trong việc mô tả các concept. Sự cần thiết phải có các local attribute và global attribute hay không phụ thuộc vào nhu cầu biểu diễn tri thức trong từng ứng dụng.

Các class attribute (thuộc tính của lớp) có các thể loại sau:

- *Default slot value* (sử dụng để gán một giá trị cho một thuộc tính trong trường hợp không có một giá trị rõ ràng nào được định nghĩa cho thuộc tính đó).
- *Type* hay còn gọi là *range* (sử dụng để ràng buộc các thể loại của thuộc tính).
- *Cardinality constraints* (được sử dụng để ràng buộc số lượng lớn nhất và nhỏ nhất của các giá trị).

Các ràng buộc về type và cardinality của thuộc tính được sử dụng để qui định thể loại giá trị nào mà thuộc tính có thể có và có bao nhiêu giá trị mà thuộc tính đó có thể có. Ví dụ: một Sản phẩm thì chỉ có một Giá (thuộc tính này là một số nguyên) và có thể có từ 1 tới 5 Màu sắc (thuộc tính này có kiểu String). Giá trị default được sử dụng trong trường hợp chúng ta không có thông tin rõ ràng về giá trị của một thuộc tính. Ví dụ: ta có thể giả sử rằng giá

trị Khấu hao của một Sản phẩm là bằng 0 nếu nó không được gán một giá trị cụ thể nào.

Khái niệm **phân loại** được sử dụng để tổ chức tri thức ontology. Nó được sử dụng trong việc tổng quát hoá và cụ thể hoá các mối quan hệ thông qua việc áp dụng các đa thừa kế và đơn thừa kế. Ngôn ngữ có tồn tại *phân loại* thì phải có các định nghĩa sau:

- *Subclass of* (cũng còn được gọi là *subsumption relationship*) đặc tả những khái niệm tổng quát bằng những khái niệm cụ thể hơn.
- *Disjoint decomposition* (một sự phân chia mà tất cả các concept của nó thì là lớp con của một concept khác). Sự phân chia này không cần thiết phải là một sự phân chia đầy đủ. Điều này có nghĩa là có thể có một instance mà không phải là instance của một lớp con. Ví dụ: các concept Bàn và Ghế có thể là sự phân chia của concept Đồ gia dụng nhưng vẫn có những instance của Đồ gia dụng mà không thuộc về lớp Bàn hoặc Ghế (ví dụ như Tủ quần áo).
- *Exhaustive subclass decomposition.* là một sự phân chia đầy đủ, có nghĩa là bất kỳ một instance nào của concept cha cũng phải là một instance của một concept con nào đó. Ví dụ: Bộ nhớ máy tính bao gồm hai lớp con là Bộ nhớ trong và bộ nhớ ngoài.
- *Not subclass.* có thể được sử dụng để thể hiện rằng một concept thì không thể phân chia thành các concept nhỏ hơn nữa. Nó được sử dụng để biểu diễn cho các lớp con nguyên thuỷ.

b) Relation và function

Relation là một mối liên kết giữa các concept trong một lĩnh vực nào đó. Trong thực tế các relation có thể được định nghĩa bằng các thuộc tính (như trong XOL, RDF và DAML+OIL). Các relation còn được gọi là các role trong OIL.

Function là một loại đặc biệt của relation. Nó khác với relation ở chỗ giá trị của tham số cuối cùng trong số n tham số là duy nhất với mỗi tập n-1 tham số trước đó.

Ví dụ: ta có relation *Mua(Người mua, Sản phẩm, Số tiền)*. Và ta có hàm *Mua(Người mua, Sản phẩm, Số tiền, Đã trả hết tiền)*. Tham số cuối cùng là *Đã trả hết tiền* chỉ nhận hai giá trị là *True* hoặc *False*.

c) Axiom

Axiom là các câu luôn luôn đúng và có thể được sử dụng cho một vài mục đích như là ràng buộc thông tin, kiểm tra tính đúng đắn. Axiom còn được gọi là assertion (như trong OML). Axiom không được sử dụng rộng rãi trong khung cảnh các ứng dụng Semantic Web.

Chúng ta có thể hình dung Axiom như là các Axiom trong logic vị từ cấp 1. Ví dụ: $\forall p(p \Rightarrow p)$

d) Instance

Instance biểu diễn các thành phần trong một miền ứng dụng, đóng vai trò như là một sự cụ thể hoá của concept.

1.6. Kết luận chương 1

Sự phát triển của Internet dẫn đến nhu cầu cho sự ra đời của thế hệ tiếp sau của Web hiện tại: Semantic Web. Semantic Web ra đời gắn liền với công

nghệ XML và Ontology. XML là cơ sở cú pháp và Ontology là cơ sở ngữ nghĩa của Semantic Web. Thành phần cơ bản của Ontology là các lớp (class) hay còn gọi là các khái niệm (concept), các thuộc tính lớp và các mối quan hệ.

CHƯƠNG 2 - QUAN HỆ NGUYÊN NHÂN-KẾT QUẢ VÀ THUẬT TOÁN PHÁT HIỆN QUAN HỆ NGUYÊN NHÂN-KẾT QUẢ

2.1. Giới thiệu

Như đã biết, một trong những thành phần quan trọng nhất của ontology là các *concept* và các *relationship[5,6,18,24]*. Các concept là các khái niệm chỉ sự vật, hiện tượng,…và thường tương ứng với các danh từ [5,24]. Các relationship chỉ mối quan hệ giữa các concept. Các thành phần này được xây dựng càng chính xác và đầy đủ thì tri thức của Ontology càng được đánh giá tốt. Việc định nghĩa ra các concept và relationship có thể dựa trên các kinh nghiệm và sự tổng hợp tri thức của con người [20,24]. Tuy nhiên, sẽ là tốt hơn rất nhiều nếu như có một công cụ mà có khả năng hỗ trợ tự động tìm ra được các concept cũng như các mối quan hệ giữa các concept này nhằm hỗ trợ xây dựng ontology. Chương này sẽ trình bày một mô hình phân tích cấu trúc thể hiện của các quan hệ nguyên nhân-kết quả trong ngôn ngữ tự nhiên và một thuật toán đề xuất nhằm mục đích tìm ra được các mối quan hệ nguyên nhân-kết quả từ một tập dữ liệu văn bản. Thuật toán này có ý nghĩa hỗ trợ trong việc xây dựng tri thức của các Ontology.

2.2. Khái niệm về các mối quan hệ ngữ nghĩa trong ngôn ngữ tự nhiên

Trong lĩnh vực ngôn ngữ tự nhiên, các thể loại thông tin như từ vựng, cú pháp, ngữ nghĩa và tri thức đóng một vai trò quan trọng trong việc hình thành nên các câu [11]. Các nhà nghiên cứu đã chứng tỏ rằng tính mạch lạc của văn

bản có thể được giải thích bằng các quan hệ ngữ nghĩa. Ví dụ: mệnh đề phụ trong câu sau được liên kết bởi quan hệ nguyên nhân (hay còn gọi là quan hệ nguyên nhân-kết quả) chỉ ra bởi từ nối *"so":*

"It is raining heavily, so the lane is flooded."

(*"Trời mưa to nên đường bị ngập nước."*)

Phát hiện ra được các mối quan hệ trong văn bản là một điều hết sức quan trọng cho các mô hình mà muốn hiểu được ngôn ngữ của con người. Hơn thế nữa, các quan hệ về mặt ngữ nghĩa thể hiện các thành phần cốt lõi trong việc tổ chức của cơ sở tri thức ngữ nghĩa từ vựng.

Trong cơ sở tri thức ngữ nghĩa từ vựng, thông tin được biểu diễn dưới dạng các *khái niệm* được tổ chức trong một *cấu trúc phân cấp* và liên kết với nhau bởi các *mối quan hệ ngữ nghĩa [3,13]*. Các khái niệm có thể là một đơn vị text đơn giản như là các từ, tới một cấu trúc phức tạp hơn như là một mệnh đề danh từ phức tạp.

Một số quan hệ ngữ nghĩa quan trọng nhất trong ngôn ngữ tự nhiên là: quan hệ *tổng quát-cụ thể*, quan hệ *tổng thể-bộ phận*, quan hệ *nguyên nhân-kết quả*, quan hệ *đồng nghĩa*, quan hệ *trái nghĩa [11,13]*.

Quan hệ **tổng quát-cụ thể**: là một trong những quan hệ ngữ nghĩa cơ sở. Nó được sử dụng nhằm mục đích phân lớp các thực thể khác nhau để tạo ra một ontology có cấu trúc phân cấp. Một khái niệm được gọi là tổng quát của một khái niệm khác nếu nó tổng quát hơn khái niệm kia.

Ví dụ: Màu *"đỏ"* thì tổng quát hơn màu *"đỏ tươi"*.

Mặc dù bao gồm cả các danh từ và động từ, nhưng quan hệ tổng quát-cụ thể thường thích hợp cho các danh từ hơn.

Quan hệ **tổng thể-bộ phận**: là mối quan hệ về mặt ngữ nghĩa mà thể hiện liên kết tổng thể và bộ phận giữa hai khái niệm.

Ví dụ: *"tay"* là một bộ phận của *"cơ thể người"*.

Quan hệ **đồng nghĩa**: hai từ được coi là đồng nghĩa nếu chúng cùng ám chỉ cùng một khái niệm ngữ nghĩa. Tuy nhiên, một vài từ chỉ được coi là đồng nghĩa trong một khung cảnh cụ thể.

Quan hệ **trái nghĩa**: là quan hệ ngược lại với quan hệ đồng nghĩa. Và cũng như quan hệ đồng nghĩa. Cũng giống như quan hệ đồng nghĩa, một số từ chỉ được coi là trái nghĩa chỉ trong một vài khung cảnh cụ thể.

Quan hệ **nguyên nhân-kết quả**: là quan hệ bao gồm hai thành phần, một thành phần thể hiện nguyên nhân và một thành phần thể hiện kết quả.

Ví dụ:

"Lacking of calcium brings about rickets"

(*"Thiếu can xi dẫn đến bệnh còi xương"*).

2.3. Quan hệ nguyên nhân-kết quả

Quan hệ nguyên nhân-kết quả được xem như là một trong số những quan hệ ngữ nghĩa quan trọng nhất góp phần tạo nên tính mạch lạc của văn bản. Quan hệ nhân quả là một đặc điểm có mặt ở khắp các quá trình tự nhiên, và do vậy nó cũng được biểu diễn bằng ngôn ngữ của con người [16].

Nói theo nghĩa rộng, nguyên nhân ám chỉ cái cách để biết liệu một trạng thái của một sự việc có gây ra một trạng thái khác hay không. Mặc dù khái niệm nguyên nhân đã có từ rất cổ (từ thời Aristotle), nhưng trải qua thời gian, các nhà khoa học và các nhà triết học vẫn còn tranh luận với nhau về định nghĩa của nguyên nhân và khi nào thì hai trạng thái của một sự việc được gọi là có liên hệ nguyên nhân-kết quả với nhau.

Học thuyết về nguyên nhân rất rộng, và có lẽ đặc điểm thú vị nhất khi làm việc trên quan hệ nguyên nhân trong các thập kỷ qua là tính đa dạng của nó. Một vài học thuyết đã được phát triển và kết quả là rất nhiều công trình nghiên cứu được công bố. Sự bùng nổ của các hướng nghiên cứu này có thể giải thích phần nào là do sự đa dạng của các phối cảnh mà các nhà nghiên cứu đã sử dụng cũng như tính đa dạng của các miền nghiên cứu: *triết học, thống kê học, ngôn ngữ học, vật lý học, kinh tế học, sinh học, y học…*

Ví dụ, trong cuốn ”Knowledge Representation” của Sowa, trí tuệ nhân tạo (Artificial Intelligent) là một trong ba môn học kinh điển (trí tuệ nhân tạo, vật lý lý thuyết và triết học). Với môn học này, có rất nhiều câu hỏi thú vị về nguyên nhân đã được đặt ra để phát triển các học thuyết nhằm kích thích những hành vi trí tuệ tương tự với con người. Nhiều nghiên cứu về nguyên nhân trong trí tuệ nhân tạo đã được làm. Chẳng hạn như, *Planning* trong trí tuệ nhân tạo là vấn đề tìm kiếm một chuỗi các hoạt động nguyên thuỷ nhằm thu được một vài mục đích. Khả năng lý luận về mặt thời gian của các hành động là cơ sở cho bất kỳ một thực thể trí tuệ nào, thực thể mà cần thiết phải đưa ra một chuỗi các quyết định. Tuy nhiên, thật là khó để biểu diễn khái niệm một chuỗi các hành động đang diễn ra và khái niệm kết quả của chuỗi các hành động đó mà không sử dụng tới khái niệm nguyên nhân. Các hành động

planning cho các robot đòi hỏi việc lập luận về nguyên nhân theo thứ tự hành động và lượng thời gian tiêu tốn để thực hiện hành động đó. Xác định nguyên nhân của các trạng nào đó của các sự việc thì cũng ngụ ý rằng cần phải xem xét trạng thái trước nó về mặt thời gian.

2.4. Cấu trúc nguyên nhân-kết quả trong ngôn ngữ của con người

Cấu trúc nhân quả đóng một vai trò quan trọng trong lịch sử ngôn ngữ trong thời gian gần đây chủ yếu bởi vì các nghiên cứu của nó có liên quan đến việc tương tác giữa các thành phần đa dạng trong việc mô tả ngôn ngữ bao gồm: ngữ nghĩa, cú pháp và hình thái. Phần này tập trung vào các biểu thức ngôn ngữ đa dạng của nguyên nhân được sử dụng trong ngôn ngữ của con người.

Bất cứ một cấu trúc nguyên nhân-kết quả nào cũng đều bao gồm hai thành phần: *nguyên nhân* và *kết quả*.

Ví dụ:

"The bus fails to turn up. As the result, I'm late for a meeting"

(*"Vì xe buýt tới muộn nên tôi đi họp muộn"*)

Trong ví dụ trên, nguyên nhân được biểu diễn bởi hiện tượng xe buýt đến muộn, và kết quả là bị muộn buổi họp.

Có hai loại quan hệ nguyên nhân-kết quả: quan hệ nguyên nhân-kết quả *tường minh* và quan hệ nguyên nhân-kết quả không *tường minh*. Quan hệ nguyên nhân-kết quả tường minh thường có cấu trúc nguyên nhân rõ ràng: *vì-nên, do-nên*,…hoặc kèm theo các động từ gây nguyên nhân: *vì vậy, cho nên, gây ra*…Quan hệ nguyên nhân-kết quả không tường minh thì có cấu trúc phức

tạp hơn và khó nhận ra hơn. Để nhận biết được các quan hệ này, cần phải có thêm cả sự phân tích ngữ nghĩa và các tri thức cơ sở.

2.4.1. Cấu trúc nguyên nhân-kết quả tường minh

Các mẫu cú pháp-từ vựng của các quan hệ nguyên nhân-kết quả tường minh được chia thành các loại sau:

- Từ nối chỉ nguyên nhân.
- Động từ chỉ nguyên nhân.
- Câu phức với một cặp từ chỉ nguyên nhân.

2.4.1.1. Từ nối chỉ nguyên nhân

Từ nối chỉ nguyên nhân được chia thành các loại sau:

- Trạng từ chỉ nguyên nhân.
- Liên từ chỉ nguyên nhân

a) Trạng từ chỉ nguyên nhân

Là các cấu trúc liên kết hai câu đơn bằng một trạng từ nhằm mục đích tạo nên một mối quan hệ nguyên nhân.

Ví dụ:

"The teacher is so prissy. For this reason, Liên doesn't go to school"

(*"Cô giáo quá khó tính. Vì lí do này, Liên không đi học"*)

Một số trạng từ chỉ nguyên nhân thường gặp: "For this reason", "As a result", "The result that"… (*"vì lý do này", "kết quả là", "do vậy", "nhờ vậy"...)*

b) Liên từ chỉ nguyên nhân

Là cấu trúc liên kết giữa hai mệnh đề bằng một liên từ để tạo nên một quan hệ nguyên nhân-kết quả.

Ví dụ:

"It was cloudy, so the experiment was postponed"

(*"Trời nhiều mây nên cuộc thí nghiệm đã bị hoãn"*)

"The boy goes out because of the banking-dog"

(*"Cậu bé chạy ra ngoài sân vì thấy tiếng chó sủa"*)

Một số liên từ chỉ nguyên nhân thường gặp: "Because", "because of", "so", "so that", "for", "since", "as"… (*"vì", "do", "nhờ", "nhờ có", "cho nên"...)*

2.4.1.2. Động từ chỉ nguyên nhân

Nhiều nhà ngôn ngữ học quan tâm nhiều điến cấu trúc động từ chỉ nguyên nhân chủ yếu bởi vì những nghiên cứu này của họ có liên quan tới các cú pháp chuẩn và sự phân tích ngữ nghĩa của ngôn ngữ.

Theo Corina Roxana Girju [11], người đầu tiên đưa ra đề xuất phân lớp từ vựng cho các động từ nguyên nhân là nhà ngôn ngữ học người Nga V.P. Nedjalkov. Ở đây ông phân loại động từ nguyên nhân thành các dạng sau:

- Động từ nguyên nhân đơn giản.
- Động từ nguyên nhân bao hàm kết quả.
- Động từ nguyên nhân ám chỉ phương tiện (gây ra)

a) Động từ nguyên nhân đơn giản:

Là các động từ bao hàm ý nghĩa của quan hệ nguyên nhân-kết quả có dạng như *"cause" ,"lead to", "bring about", "generate", "make", "force", "allow"... ("gây ra", "dẫn đến", "sinh ra", "tạo ra", "làm cho"...)*

Ví dụ:

"Earthquakes generate tidal waves"

(*"Động đất gây ra sóng thần"*)

"Lacking of calcium might bring about rickets"

(*"Thiếu can xi có thể dẫn đến còi xương"*)

"Rain lead to flooded lanes"

(*"Trời mưa làm cho đường lội"*)

b) Động từ nguyên nhân bao hàm kết quả

Là những động từ thể hiện một hành động mà từ động từ đó chúng ta có thể biết được kết quả của hành động đó mà kết quả này không cần phải đề cập đến trong câu [11].

Ví dụ:

"The thieft killed the host"

(*"Tên trộm đã giết người chủ nhà"*)

(Với động từ *"giết"* chúng ta có thể biết là người chủ nhà đã chết)

"The artist burned his paintings which he drew yesterday"

(*"Người hoạ sỹ đã đốt những bức tranh mà anh ta đã vẽ ngày hôm qua."*)

(Với động từ *"đốt"* chúng ta biết được là những bức tranh mà người hoạ sỹ vẽ ngày hôm qua đã bị cháy hết).

Một số động từ nguyên nhân bao hàm kết quả: *"kill", "burn", "fire", "poison", "hit", "shoot"... ("giết", "đốt", "cháy", "đầu độc", "đánh", "bắn"...)*

c) Động từ nguyên nhân ám chỉ phương tiện (gây ra)

Là các động từ thể hiện một hành động mà từ động từ đó chúng ta có thể biết được phương tiện để gây ra hành động đó trong khi phương tiện này không cần phải được đề cập đến trong câu.

Ví dụ:

"Stepmother commonly poison her husband's stepchild"

(*"Gì ghẻ thường hay đầu độc những đứa con riêng của chồng"*)

(Với động từ *"đầu độc"* chúng ta có thể biết được các bà dì ghẻ đã dùng thuốc độc để đầu độc con chồng)

"He is swimming to the island"

(*"Anh ấy đang bơi ra ngoài đảo"*)

(Với động từ *bơi* chúng ta có thể biết được anh ý phải đang bơi trên một hồ *nước* trong khi trong câu không hề nhắc đến *nước*).

Một số động từ nguyên nhân ám chỉ phương tiện: *"poison", "swim", "shoot", "writte", "read"...("đầu độc", "bơi", "bắn", "viết", "đọc"...)*

2.4.1.3. Câu phức với một cặp từ chỉ nguyên nhân

Là cấu trúc câu ghép gồm hai mệnh đề được nối với nhau bằng một cặp từ nối để ám chỉ quan hệ nguyên nhân-kết quả giữa hai mệnh đề này.

Ví dụ:

"It is raining so heavily that the lane is flooded"

(*"Vì trời mưa to nên đường lội"*)

"If I have much money then I'll buy a beautiful house"

(*"Nếu tôi có nhiều tiền thì tôi sẽ mua một ngôi nhà thật đẹp"*)

Một số cặp từ nối chỉ nguyên nhân thường gặp [11]: *"If…then", "so… that"…("vì…nên…", "do…nên…", "nếu…thì…"…*)

2.4.2. Cấu trúc nguyên nhân không tường minh

Đây là thể loại khó nhất, nó đòi hỏi phải suy luận dựa trên các phân tích ngữ nghĩa và tri thức tổng thể.

Bao gồm các cấu trúc sau:

- Họ danh từ ghép

- Động từ ám chỉ nguyên nhân không tường minh.

a) Các họ danh từ ghép biểu diễn nguyên nhân

Các họ danh từ ghép là một trong những vấn đề khó nhất của việc xử lý ngôn ngữ tự nhiên, chủ yếu bởi vì chúng đòi hỏi việc phân tích ngữ nghĩa khá phức tạp. Các danh từ ghép là các mệnh đề danh từ được hình thành như là một sự mở rộng hay thừa kế của các danh từ gốc. Ví dụ: "giáo viên *tiếng Anh*", "tỉ lệ *gia tăng dân số*",… Sự nhập nhằng của các danh từ này đã làm cho việc

phân tích câu trở nên khó khăn hơn. Một từ vựng cơ sở có thể có nhiều hơn một nghĩa, vì vậy, một từ ghép thì lại càng có nhiều nghĩa hơn. Để có thể biên dịch chúng một cách đầy đủ, đòi hỏi phải có những tri thức ngôn ngữ mở rộng liên quan dến nội dung ngữ nghĩa của các thành phần trong câu và trong một ngữ cảnh nhất định.

Một trong số những quan hệ có thể liên kết hai danh từ trong một họ danh từ ghép là quan hệ nguyên nhân. Nó có dạng là một cụm danh từ được hình thành bởi hai cụm từ trong đó một cụm từ là nguyên nhân và một cụm từ là kết quả.

CT1 CT2 => CT1 là nguyên nhân của CT2 hoặc CT1 bị gây ra bởi CT2

Trong đó CT1 và CT2 là các cụm từ 1 và 2.

Ví dụ:

"Tetanus virus" ("Vi trùng *uốn ván")*

(Bệnh uốn ván bị gây ra bởi vi trùng)

b) Động từ chỉ nguyên nhân không tường minh

Đó là cấu trúc của một dãy các hành động thể hiện bằng các động từ mà hành động sau thì thường là kết quả của hành động trước. Trong cấu trúc này, chưa chắc đã xuất hiện các từ nối chỉ nguyên nhân.

Ví dụ:

"Feeling sorry for what he did, the burglar confessed to the policeman"

(*"Cảm thấy hối hận vì những gì mà mình đã làm, tên trộm đi đầu thú với cảnh sát"*).

(Hành động *đầu thú* là kết quả của hành động *hối hận*)

2.5. Thuật toán khai phá dữ liệu phát hiện quan hệ nguyên nhân-kết quả từ các văn bản

2.5.1. Giới thiệu

Vấn đề học ngôn ngữ tự nhiên là một chủ đề hay và đã được nghiên cứu từ nhiều năm nay. Nhóm nghiên cứu về học ngôn ngữ tự nhiên SIGNLL (Special Interest Group on Natural Language Learning) mỗi năm một lần tổ chức một hội thảo với các chủ đề xoay quanh vấn đề về học ngôn ngữ tự nhiên CoNLL (Conference of Natural Language Learning). Hội thảo lần thứ 8 tổ chức vào ngày 6-7 tháng 5 năm 2004 (CoNLL-2004) có chủ đề là *Sematic Role Labeling*.

Bài toán Sematic Role Labeling là bài toán yêu cầu gán nhãn ngữ nghĩa (sematic role) cho các thành phần cú pháp trong câu. Một *Semantic Role* là một mối quan hệ giữa các thành phần cú pháp trong câu và một thuộc tính ngữ nghĩa nào đó. Việc nhận ra và gán nhãn ngữ nghĩa cho các thành phần trong câu là một công việc quan trọng để trả lời cho các câu hỏi *"Ai", "Cái gì", "Khi nào", "Ở đâu", "Tại sao", ... ("Who", "What", "When", "Where", "Why", ...)*. Ví dụ, ta có câu sau đã được gán nhãn semantic roles:

[$_{A0}$ He] [$_{AM-MOD}$ would] [$_{AM-NEG}$ n't] [$_{V}$ **accept**] [$_{A1}$ anything of value] from [$_{A2}$ those he was writing about] .

Ở đây, các nhãn ngữ nghĩa đã được định nghĩa trong tập roleset tương ứng với các ký hiệu được định nghĩa trong PropBank Frames (qui định các ký hiệu cú pháp của ngân hàng dữ liệu PropBank) [19,20,21]:

V: động từ (verb)

A0: chủ ngữ điều khiển động từ *accept* **(**acceptor)

A1: vị ngữ bị điều khiển bởi động từ **(**thing accepted)

A2: vị ngữ phụ sau giới từ (accepted-from)

AM-MOD: động từ tình thái (modal)

AM-NEG: phủ định (negative)

Đây là một bài toán lớn và đã có nhiều công trình được trình bày tại hội thảo nhằm đưa ra các giải pháp cho vấn đề này như các bài báo: *Hierarchical Recognition of Propositional Arguments with Perceptrons* của các tác giả Xavier Carreras and Llu´ıs M`arquez (TALP Research Centre,Technical University of Catalonia) và Grzegorz Chrupała (GRIAL Research Group, University of Barcelona); *Semantic Role Labeling by Tagging Syntactic Chunks* của các tác giả Kadri Hacioglu[1], Sameer Pradhan[1], WayneWard[1], James H. Martin[1], Daniel Jurafsky[2] ([1]University of Colorado at Boulder, [2]Stanford University); *Semantic Role Labeling using Maximum Entropy Model* của các tác giả Joon-Ho Lim, Young-Sook Hwang, So-Young Park, Hae-Chang Rim (Department of Computer Science & Engineering Korea University); *Semantic Role Labeling Via Generalized Inference Over Classifiers* của tác giả Vasin Punyakanok, Dan Roth, Wen-tau Yih, Dav Zimak Yuancheng Tu (Department of Computer Science Department of Linguistics, University of Illinois at Urbana-Champaign). Tuy nhiên, tất cả các thuật toán được đề xuất này có độ chính xác vẫn chưa cao (precision <75% và recall <70%).

Mặt khác, Corina Roxana Girju [11] đưa ra một thuật toán tìm ra các động từ thể hiện quan hệ nguyên nhân và các động từ thể hiện quan hệ tổng thể-bộ phận. Trong công trình của mình, Corina Roxana Girju đã đi sâu nghiên cứu về cấu trúc ngôn ngữ tự nhiên, thuật toán của tác giả nhằm mục đích tìm kiếm câu có cấu trúc nguyên nhân-kết quả và tổng thể-bộ phận, sau đó đánh giá mức độ quan trọng của các động từ chính trong câu bằng cách thống kê tần suất xuất hiện của chúng trong một số lượng lớn các văn bản.

Thuật toán được chúng tôi đưa ra là một cải tiến của thuật toán của Corina Roxana Girju [11]. Chúng tôi cũng tìm kiếm các câu có cấu trúc nguyên nhân-kết quả như cách mà Roxana Girju đã làm, nhưng sau đó không xác định tần suất xuất hiện của động từ mà thống kê tần suất xuất hiện của chính các cặp danh từ chỉ nguyên nhân-kết quả trong câu (còn tác giả Corina Roxana Girju thì lại lấy ra động từ để thống kê tần suất xuất hiện của động từ). Cặp danh từ nào có tần suất xuất hiện càng nhiều thì xác suất mang quan hệ ngữ nghĩa nguyên nhân-kết quả của chúng càng cao. Bài toán này là một phần nhỏ của bài toán Semantic Role. Cụ thể là chúng tôi chỉ tập trung giải quyết việc gán nhãn những động từ chỉ nguyên nhân đơn giản (động từ chỉ nguyên nhân tường minh).

2.5.2. Thuật toán phát hiện quan hệ nguyên nhân-kết quả

Như chúng tôi đã giới thiệu và phân tích ở trên, quan hệ nguyên nhân-kết quả thể hiện trong ngôn ngữ tự nhiên vô cùng phong phú, đa dạng và phức tạp. Chỉ riêng việc phân tích câu để xác định ngữ nghĩa của câu thuộc cấu trúc nhân quả nào cũng đã là một trong những dạng bài toán khó nhất của xử lý ngôn ngữ tự nhiên. Vì vậy, trong thuật toán này, không bao trùm toàn bộ mọi

cấu trúc phức tạp của quan hệ nguyên nhân mà chỉ quan tâm đến cấu trúc nguyên nhân tường minh thể hiện ở động từ chỉ nguyên nhân. Các trường hợp khác của quan hệ nguyên nhân thì không được xét đến ở đây.

Quan hệ nguyên nhân tường minh với một động từ chỉ nguyên nhân có thể biểu diễn dưới dạng:

<DT1 - *động từ chỉ nguyên nhân* - DT2>

Trong đó:

DT1 và DT2 là các danh từ (hoặc ngữ danh từ). Chúng có thể tương ứng với các concept của ontology.

Ngữ danh từ là một nhóm các từ mà kết thúc bằng một danh từ. Nó có thể chứa quán từ (the, a, this, …) ở đầu, chứa các tính từ, trạng từ, và danh từ. Ngữ danh từ không được bắt đầu bằng một giới từ.

Thủ tục phát hiện quan hệ nhân quả.

Khái quát thuật toán:

Đầu vào: danh sách các động từ chỉ nguyên nhân.

Đầu ra: danh sách các cặp quan hệ nguyên nhân- kết quả có dạng (DT1, DT2)

Bước 1: Với mỗi văn bản trong tập dữ liệu. Chọn ra các câu có cấu trúc <DT1-động từ-DT2> từ các văn bản.

Trong đó, DT1 và DT2 là các danh từ (hoặc ngữ danh từ).

Bước 2: So sánh động từ trong câu đã chọn với các động từ chỉ nguyên nhân trong bảng động từ chỉ nguyên nhân. Nếu động từ này trùng với một trong các động từ chỉ nguyên nhân trong bảng thì xét cặp (DT1, DT2):

- Nếu cặp danh từ này đã có trong cơ sở dữ liệu thì tăng giá trị tần suất xuất hiện của chúng lên 1.

- Nếu cặp danh từ này chưa tồn tại trong cơ sở dữ liệu thì thêm mới nó vào cơ sở dữ liệu.

Bước 3: lặp lại bước hai với tất cả các câu có dạng <DT1- động từ-DT2> trong văn bản đó.

Bước 4 : Quay trở lại thực hiện bước 1 với mỗi văn bản trong tập dữ liệu.

Bước 5: Sắp xếp các cặp (DT1, DT2) thu được theo thứ tự giảm dần của tần xuất xuất hiện.

Bước 6: Chọn ra m cặp đầu tiên trong cơ sở dữ liệu. Đó là những cặp quan hệ nhân quả cần tìm.

Chi tiết thuật toán:

In put: V là tập chứa các động từ chỉ nguyên nhân.

Out put: O là một tập gồm các cặp có dạng (DT1, DT2) là các cặp thể hiện quan hệ nguyên nhân-kết quả.

1. C := Φ là tập hợp sẽ chứa các cặp (DT1, DT2, i) với DT1, DT2 là các danh từ chỉ nguyên nhân và kết quả và i là tần xuất xuất hiện của cặp danh từ đó.
2. *For* mỗi văn bản Di trong CSDL

    2.1 *For* mỗi câu Sj trong văn bản Di

        2.1.1 Nếu Sj là câu có dạng <danh từ 1- động từ - danh từ 2 >

2.1.1.1 Tách ra cặp (DT1, DT2) với DT1= danh từ 1 và DT2= danh từ 2.

2.1.1.2 Gán v:= động từ.

2.1.1.3 Nếu v đã có trong trong V

2.1.1.3.1 Nếu (DT1, DT2) đã có trong C thì tăng tần suất xuất hiện của nó lên 1.

2.1.1.3.1 Nếu (DT1, DT2) chưa có trong C thì gán C:= C U (DT1, DT2, 1).

3. Sắp xếp tập C theo thứ tự giảm dần của tần suất xuất hiện.

4. Chọn ra *m* cặp quan hệ đầu tiên trong C làm kết quả trả về trong tập O.

Chú ý: Một điều quan trọng cần chú ý với thủ tục trên là với câu có dạng <DT1-động từ chỉ nguyên nhân- DT2> thì DT1 có thể là nguyên nhân của DT2 hoặc DT2 là nguyên nhân của DT1. Nhưng cặp quan hệ nguyên nhân-kết quả thu được (DT1, DT2) thì phải có một dạng thống nhất là DT1 là nguyên nhân và DT2 là kết quả. Vì vậy chúng ta cần xác định rõ loại động từ gây nguyên nhân là loại động từ nào: <Nguyên nhân- động từ- kết quả> hay <kết quả - động từ- nguyên nhân>, để từ đó gán cặp (DT1, DT2) cho thích hợp. Để giải quyết vấn đề này có thể thêm cho mỗi

động từ nguyên nhân một thuộc tính thể hiện tính chất trên.

2.6. Kết luận chương 2

Chương này trình bày khái niệm, ý nghĩa và phân tích chi tiết cấu trúc của quan hệ nguyên nhân-kết quả được thể hiện trong ngôn ngữ của con người. Từ đó đưa ra một thuật toán nhằm phát hiện ra các cặp nguyên nhân-kết quả từ một tập hợp các văn bản text. Chương trình cài đặt thử nghiệm cho thuật toán và việc đánh giá kết quả thuật toán sẽ được trình bày ở chương tiếp theo.

CHƯƠNG 3 - KẾT QUẢ THỬ NGHIỆM THUẬT TOÁN

3.1. Giới thiệu

Chương trình thử nghiệm cài đặt cho thuật toán khai phá dữ liệu phát hiện quan hệ nhân quả trong văn bản được viết bằng ngôn ngữ **Java** và kết nối với cơ sở dữ liệu **Oracle**.

Chương trình bao gồm 1100 dòng lệnh trong năm file:

- File chương trình chính: Phối hợp các lớp và chạy chương trình.
- Lớp *ConnectDBClass*: chứa các thủ tục tiện ích để kết nối vào CSDL.
- Lớp *ConvertFileClass*: chứa các thủ tục để chuyển từ định dạng dữ liệu gốc của Pern Tree Bank [7,8] thành định dạng có thể xử lý được.
- Lớp *ReadFileClass*: chứa các thủ tục đọc file phân tích câu tách động từ, danh từ để cho vào CSDL.

Chương trình viết theo mục đích riêng và phải phân tích file theo định dạng dữ liệu của Pern Tree Bank nên không sử dụng mã nguồn có sẵn.

Dữ liệu sử dụng để thử nghiệm cho thuật toán là một **corpus** được trích ra từ ngân hàng dữ liệu Penn TreeBank II (http://www.cis.upenn.edu/~treebank). Ngân hàng dữ liệu này bao gồm khoảng 1 triệu câu, được lấy từ tạp chí Wall Street Journal xuất bản năm 1989.

3.2. Định dạng file dữ liệu

Dữ liệu Penn Tree Bank nằm trong 2300 file. Mỗi file chứa một tập hợp các câu đã được đánh dấu cú pháp sẵn theo định dạng của Penn TreeBank [7,8].

Ví dụ, câu sau đã được đánh dấu cú pháp đầy đủ:

```
The          DT    B-NP  (S*    O
$            $     I-NP    *    O
1.4          CD    I-NP    *    O
billion      CD    I-NP    *    O
robot        NN    I-NP    *    O
spacecraft   NN    I-NP    *    O
faces        VBZ   B-VP    *    O
a            DT    B-NP    *    O
six-year     JJ    I-NP    *    O
journey      NN    I-NP    *    O
to           TO    B-VP  (S*    O
explore      VB    I-VP    *    O
Jupiter      NNP   B-NP    *    B-LOC
and          CC    O       *    O
its          PRP$  B-NP    *    O
16           CD    I-NP    *    O
known        JJ    I-NP    *    O
moons        NNS   I-NP    *S)  O
.            .     O       *S)  O
```

Các ký hiệu của một câu được đưa ra bằng cách sử dụng phương pháp biểu diễn theo cột phân cách nhau bằng các dấu cách. Mỗi cột mã hoá một ký hiệu bằng các thẻ đánh dấu tương ứng với ký hiệu đó.

Với mỗi câu, bao gồm những cột sau:

1. *Words*.

2. *Part of speech tags*.

3. *Chunks in IOB2 format.*

4. *Clauses in Start-End format*.

5. *Named Entities in IOB2 format*.

Words chứa danh sách các từ đơn của câu.

Part of speech tags biểu diễn từ loại của từng từ đơn tương ứng trong cột Word. Một số định dạng từ loại:

JJ: tính từ.

JJR: tính từ so sánh hơn.

JJS: tính từ so sánh bậc nhất.

RB: trạng từ.

RBR: trạng từ so sánh hơn.

RBS: trạng từ so sánh bậc nhất.

CC: từ nối.

CD: từ chỉ số lượng.

DT: quán từ.

NN: danh từ đơn.

NNS: danh từ số nhiều.

NNP: danh từ riêng số ít.

NNPS: danh từ riêng số nhiều.

VB: động từ, dạng nguyên thể.

VBD: động từ, dạng quá khứ.

VBG: động từ, dạng tiếp diễn hoặc danh động từ.

Định dạng IOB2 biểu diễn các đoạn nối tiếp nhau. Các từ mà không thuộc đoạn nào thì nhận giá trị thẻ O. Các từ bên trong một đoạn loại $k, thì từ đầu tiên ứng với thẻ có dạng là "B-$k" (Begin), và các từ tiếp ứng với thẻ có dạng là "I-$k" (Inside).

Một số ký hiệu hay sử dụng của định dạng IOB2:

ADJ tính từ (adjective).

ADJP ngữ giới từ (adjective phrase)

ADV trạng từ (adverb)

ART quán từ (article)

N danh từ (noun)

NP ngữ danh từ (noun phrase)

S câu (sentence)

V động từ (verb)

VP ngữ động từ (verb phrase)

Định dạng *Start-End* biểu diễn các cụm từ (phrases) lồng vào nhau. Mỗi thẻ biểu diễn mở đầu và kết thúc của một cụm từ, nó có dạng STARTS*ENDS. Thẻ START có dạng "($k", nó biểu diễn vị trí bắt đầu của một cụm từ của thể loại $k. Thẻ END có dạng "$k)", biểu diễn vị trí kết thúc của cụm từ thể loại $k. Sự kết nối của các cấu trúc thẻ thì tạo nên một cấu trúc ngoặc. Ví dụ, thẻ

"*" biểu diễn một từ mà không phải là từ bắt đầu hay kết thúc của một cụm từ; thẻ "(A0*A0)" biểu diễn một từ mà tạo thành đối số A0; thẻ "(S (S*S)" biểu diễn một từ mà cấu thành một mệnh đề cơ sở (nhãn S) và bắt đầu một mệnh đề mức cao hơn.

3.3. Chương trình thử nghiệm

Chương trình thử nghiệm cài đặt thử nghiệm cho thuật toán phát hiện quan hệ nguyên nhân-kết quả chạy trên tập dữ liệu đã được phân tích cú pháp sẵn của Penn TreeBank như đã mô tả ở trên.

Chương trình chạy trên máy tính IBM Pentium 4, CPU 2.4 GHz, 500 Mb RAM. Tổng số thời gian mỗi lần chạy chương trình với tập dữ liệu được mô tả ở trên là 8h24'.

Các động từ chỉ nguyên nhân sử dụng cho chương trình là các động từ chỉ nguyên nhân được lấy ra từ WordNet 2.1 (http://wordnet.princeton.edu/).

STT	Động từ
1	Induce
2	Cause
3	Make
4	Result (in/from)
5	Lead (to)
6	Produce
7	Generate
8	Create
9	Bring (about)

Bảng 1: Các động từ nguyên nhân lấy ra từ WordNet

WordNet là một hệ thống tham khảo từ vựng trực tuyến được thiết kế bởi một nhóm nghiên cứu trường đại học **Princeton University**

(http://www.princeton.edu/main/). Hệ thống này đã và đang được sử dụng bởi nhiều nhóm nghiên cứu có liên quan.

3.4. Kết quả thực nghiệm

Kết quả tìm được tổng cộng 34 033 cặp danh từ (hay ngữ danh từ).

Trong đó,

+ Có 2 cặp danh từ (hay ngữ danh từ) có tần suất xuất hiện nhiều nhất là 9 lần. Đó là các cặp: company-sale (*công ty kinh doanh- việc buôn bán*), smoking-lung cancer (*hút thuốc- bệnh ung thư phổi*).

+ Có 4 cặp có tần suất xuất hiện 8 lần. Đó là các cặp: smoking-pulmonary problem (*hút thuốc- các bệnh về phổi*), traffic-noise (*giao thông-tiếng ồn*), Standard & Poor-underwriter (cặp này không có nghĩa), environmental change-erosion (*thay đổi của môi trường- sự xói mòn*).

Ta có bảng kết quả như sau:

Tần suất xuất hiện	**Số cặp danh từ/ngữ danh từ**	**Tỉ lệ % trên tổng số các cặp tìm thấy**
9	2	0.005 %
8	4	0.012 %
7	8	0.024 %
6	23	0.068 %
5	30	0.081%
4	99	0.29 %
3	263	0.77 %

Tần suất xuất hiện	Số cặp danh từ/ngữ danh từ	Tỉ lệ % trên tổng số các cặp tìm thấy
2	502	1.48 %
1	33077	97.2 %

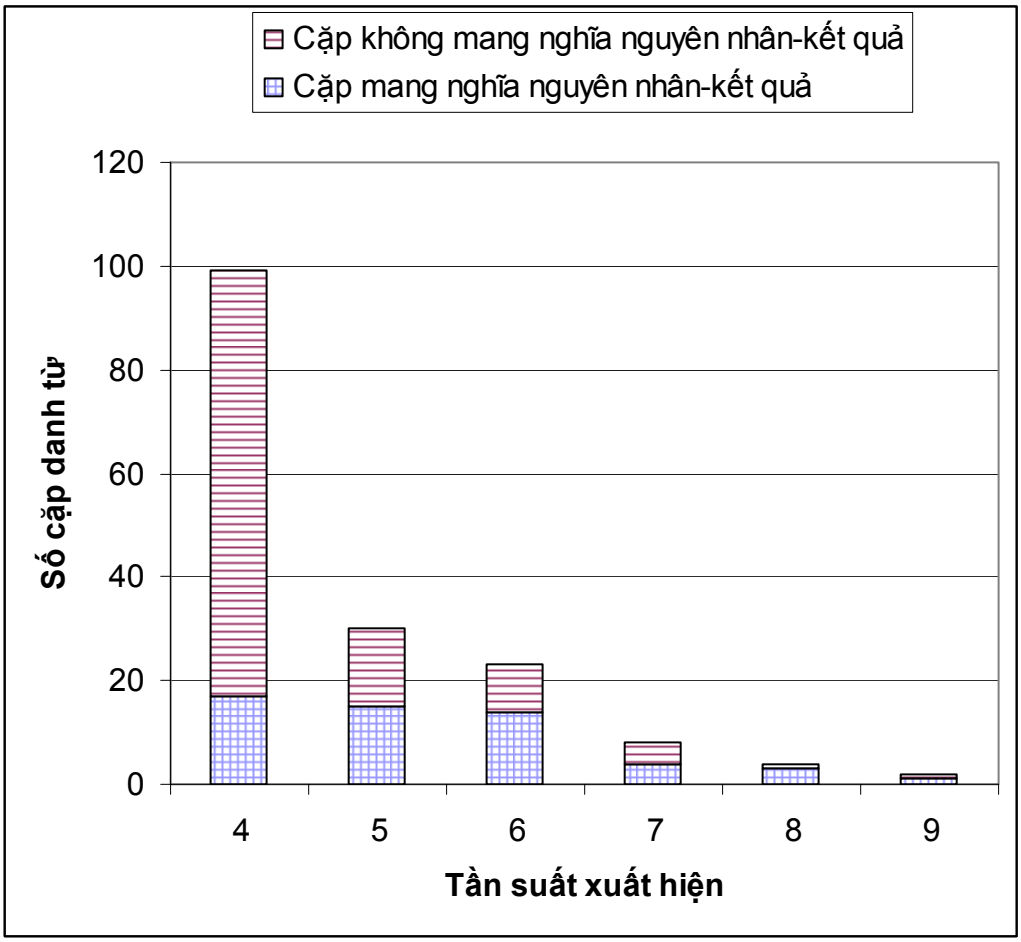
Bảng 2: Tỉ lệ phần trăm của các cặp danh từ tìm thấy theo tần suất xuất hiện.

Tính tỉ lệ phần trăm của số cặp danh từ (hay ngữ danh từ) có ý nghĩa nguyên nhân-kết quả theo từng tần suất xuất hiện ta có bảng sau:

Tần suất xuất hiện	Số cặp danh từ/ngữ danh từ	Số cặp danh từ/ngữ danh từ mang ý nghĩa nguyên nhân-kết quả	Tỉ lệ % số cặp mang ý nghĩa nguyên nhân-kết quả
9	2	1	50 %
8	4	3	75 %
7	8	4	50 %
6	23	14	61 %
5	30	15	50 %
4	99	17	17.2 %

Bảng 3: tỉ lệ phần trăm các cặp mang nghĩa nguyên nhân-kết quả theo tần suất xuất hiện.

Bảng trên được biểu diễn dưới dạng đồ thị như sau:



Hình 3: đồ thị tỉ lệ các cặp danh từ mang nghĩa nguyên nhân-kết quả theo tần suất xuất hiện.

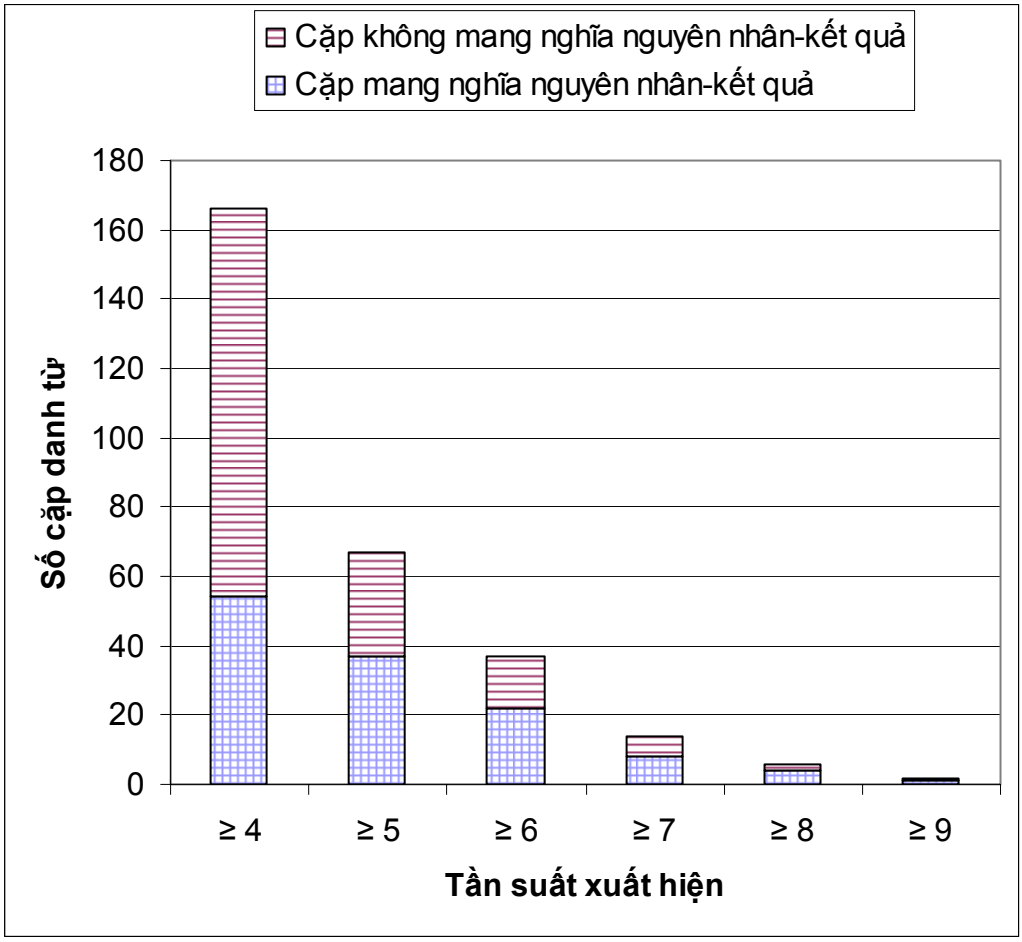
Tính tỉ lệ phần trăm số cặp danh từ (hay ngữ danh từ) mang ý nghĩa nguyên nhân-kết quả theo tần suất xuất hiện lớn hơn một ngưỡng nào đó ta có bảng kết quả sau:

Tần suất xuất hiện	Số cặp danh từ/ngữ danh từ	Số cặp danh từ/ngữ danh từ mang ý nghĩa nguyên nhân-kết quả	Tỉ lệ % số cặp mang ý nghĩa nguyên nhân-kết quả
≥ 9	2	1	50 %

Tần suất xuất hiện	Số cặp danh từ/ngữ danh từ	Số cặp danh từ/ngữ danh từ mang ý nghĩa nguyên nhân-kết quả	Tỉ lệ % số cặp mang ý nghĩa nguyên nhân-kết quả
≥ 8	6	4	66.7 %
≥ 7	14	8	57.1 %
≥ 6	37	22	59.4 %
≥ 5	67	37	55.2 %
≥ 4	166	54	32.5 %

Bảng 4: tỉ lệ các cặp danh từ mang nghĩa nguyên nhân-kết quả có tần suất lớn hơn một giá trị ngưỡng.

Bảng trên được biểu diễn dưới dạng đồ thị:



Hình 4: đồ thị thể hiện tỉ lệ các cặp danh từ có nghĩa nguyên nhân-kết quả có tần xuất lớn hơn một giá trị ngưỡng.

3.5. Nhận xét

Bảng kết quả cho thấy với những cặp có tần suất xuất hiện lớn thì tỉ lệ phần trăm các cặp mang ý nghĩa nguyên nhân-kết quả càng cao.

Với những cặp có tần suất xuất hiện lớn hơn 5 lần thì tỉ lệ này đều > 50 %.

Tỉ lệ chính xác vẫn chưa cao (< 70 %) nhưng kết quả đạt được đã cho thấy có thể dựa vào thuật toán đề xuất để tìm ra những cặp danh từ (hoặc ngữ

danh từ) có quan hệ ngữ nghĩa nguyên nhân-kết quả. Đây chính là mục đích của luận văn này.

3.6. Kết luận chương 3

Chương này là kết quả cài đặt thử nghiệm của thuật toán được trình bày ở chương 2. Chương trình cài đặt viết bằng ngôn ngữ Java, chạy trên ngân hàng dữ liệu đã được phân tích cú pháp sẵn Penn Tree Bank. Sử dụng các động từ chỉ nguyên nhân được lấy ra từ WordNet 2.1, chương trình đã tìm thấy 34 033 cặp danh từ (hay ngữ danh từ). Trong số các cặp có tần suất xuất hiện >= 4 có 32.5 % là các cặp mang ý nghĩa nguyên nhân-kết quả.

KẾT LUẬN

Như vậy, kết quả thực nghiệm của thuật toán đã tìm được 54 cặp danh từ (hay ngữ danh từ) mang ý nghĩa nguyên nhân-kết quả trong số 166 cặp kết quả tìm thấy mà có tần suất xuất hiện ≥ 4. Những thông tin tìm được của thuật toán sẽ là các thông tin rất hữu ích trong việc xây dựng ontology hay việc xây dựng các ứng dụng khác của Semantic Web.

Luận văn mới chỉ giới hạn việc tìm quan hệ ngữ nghĩa ở cấu trúc quan hệ nguyên nhân-kết quả. Để phát triển, có thể áp dụng tương tự thuật toán vào các loại quan hệ ngữ nghĩa khác như tổng thể-bộ phận, khái quát-cụ thể bằng cách phân tích cấu trúc của các quan hệ này trong câu.

Ngoài việc ứng dụng kết quả của thuật toán tìm quan hệ ngữ nghĩa vào việc xây dựng Ontology cho Semantic Web. Kết quả của thuật toán còn có thể được ứng dụng trong các lĩnh vực khác. Ví dụ như trong việc xây dựng máy tìm kiếm để thực hiện trả lời câu hỏi *Who, What, When, Where*...

Việc đánh giá mức độ thể hiện ý nghĩa nguyên nhân, kết quả của cặp danh từ (hay ngữ danh từ) của thuật toán mới chỉ dựa vào tần suất xuất hiện trong các văn bản. Việc đánh giá này có thể mở rộng lên bằng cách gán cho mỗi cặp một trọng số. Trọng số này sẽ được tính thông qua các thông số như: tần suất xuất hiện, mức độ quan trọng của động từ chỉ nguyên nhân mà nó liên kết…

Kết quả thực nghiệm của thuật toán chưa cho độ chính xác cao (< 70 %), do chạy trên một tập dữ liệu chưa lớn lắm, nhưng đã cho thấy kết quả của thuật toán có thể được sử dụng để tham khảo và xây dựng các mối quan hệ và tìm ra các concept trong quá trình xây dựng Ontology.

TÀI LIỆU THAM KHÁO

Tiếng Việt

[1]. Đặng Tiểu Hùng (2004), *Phương pháp biểu diễn ngữ nghĩa lân cận siêu liên kết cho máy tìm kiếm VietSeek,* Luận văn thạc sỹ, Khoa Công Nghệ-Đại học Quốc gia Hà nội, tr 6-42.

[2]. Đoàn Sơn (2001), *Các phương pháp biểu diễn và ứng dụng trong khai phá dữ liệu văn bản,* Luận văn thạc sỹ, Khoa Công Nghệ-Đại học Quốc gia Hà nội, tr 16-32.

[3]. Phạm Thanh Nam, Bùi Quang Minh, Hà Quang Thụy (2004). *Giải pháp tìm kiếm trang Web tương tự trong máy tìm kiếm VietSeek*. Tạp chí Tin học và Điều khiển học (nhận đăng 1-2004)

[4]. Phan Xuân Hiếu (2003), *Khai phá song song luật kết hợp mờ,* Luận văn thạc sỹ, Khoa Công Nghệ- Đại học Quốc gia Hà nội, tr 9-16, tr 42-58.

Tiếng Anh

[5]. Asuncion Gomez-Perez and Oscar Corcho (January / February 2002), *Ontology Languages for the Semantic Web,* IEEE intelligent systems, http://computer.org/intelligent.

[6]. Aubrey E.Hill (1998), *Automated knowledge acquisition of case-based semantic networks for interative enhancement of the dataming proccess,* Doctor of Philosophy, University of Alabama at Birmingham, pp 14-32.

[7]. Beatrice Santorini (1990), *Part-of-Speech Tagging Guidelines for the Penn TreeBank Project,* Penn Treebank II Project, http://www.cis.upenn.edu/~treebank.

[8]. Beatrice Santorini (1991), *Bracking Guidelines for Penn TreeBank Project,* Penn Treebank II Project, http://www.cis.upenn.edu/~treebank.

[9]. Chiristopher D. Manning, Hinrich Schuze (1999), *Foundations of Statistical Natural Language Processing,* The MIT Press, Cambridge, Massachusets London, England.

[10]. Choochart Haruechaiyasak (2003), *A dataming and Semantic Web frameworks for building a web based recomender system,* Doctor of Philosophy, the University of Miami, pp 31-44, pp 50-59.

[11]. Corina Roxana Girju (2002), *Text mining for semantic relations,* Doctor of Philosophi in computer science, University of texas at Dallas, pp 25-63, pp 86-106.

[12]. Dieter Fensel and Frank van Harmelen (March/April 2001), *OIL: an ontology infrastructure for the Semantic Web,* IEEE intelligent systems, http://computer.org/intelligent.

[13]. Đoàn Thiện Thuật (2001), *A concise Vietnamese grammar for non-native speakers.* Nhà xuất bản thế giới 2001, pp 6-15, pp 20-29.

[14]. Ha Quang Thuy, Nguyen Tri Thanh (2003). *A web site representation method using concept vectors and web site classifications*. Gửi đăng Tạp chí Tin học và Điều khiển học tháng 10-2003.

[15]. I.Horrocks and F.van Harmelen (draft report, 2001), *Reference Description of the DAML+OIL Ontology Markup Language,* www.daml.org/2000/12/reference.html

[16]. J. Han and M. Kamber (2000), *Data Mining: Concepts and Techniques,* Morgan Kaufmann, ch 1, pp 3-31.

[17]. Jeff Heflin, James Hender (2000), *Semantic Interoperablity on the Web,* University of Mary Land, http://www.cs.umd.edu/~heflin.

[18]. Jeffrey Douglas Heflin (2001), *Toward the Semantic Web: a knowledge representation in a dynamic, distributated environment,* Doctor of Philosophy, University of Maryland, pp 40-83.

[19]. Jingkun Hu (2004), *Visual Modeling of XML constraints based on a new extensible constraint Markup Language,* Doctor of Philosophy, Pace University, pp 9-44 .

[20]. Jonh Davies, Dieter Fensel, Frank van Harmelen (2003), *Towards the Semantic Web Ontology-driven Knoledge Management,* John Wiley & Sons Ltd, pp 1-9, pp 16,17,18

[21]. Lan Eric Gibson (2001), *Data mining Analysis of digital library database usage partern as a tool facilitating efficient user navigation,* Doctor of Philosophy, the University of Alabama, pp 23-42.

[22]. Maedche, Alexander D (2002), *Ontology learning for the Semantic Web,* Kluwer Academic Publisher, pp 10-34.

[23]. Marie Meteer, et al (1995), *Dysfluency Annotation Stylebook for the Switchboard Corpus,* Penn Treebank II Project, http://www.cis.upenn.edu/~treebank.

[24]. Michael C.Dacota, Leo J. Obrst, Kevin T. Smith (2003), *The Semantic Web,* Wiley Publisher, ch 1,2, 7.

[25]. Paul Kingsbury, Martha Palmer, and Mitch Marcus (2002), *Adding Sematic Annotation to Penn TreeBank*, In Proceedings of the Human Language Technology Conference, San Diego, California.

[26]. Scott Owen Farrar (2003), *An ontology for linguistics on the Semantic Web,* Doctor of Philosophy, Arizona State University, pp 12-14.

[27]. Sean Luke, Lee Spector, David Rager , *Ontology-Based Knowled Discovery on the World Wide Web,* http://www.cs.umd.edu/~seanl.

[28]. Sean Luke, Lee Spector, David Rager, James Hendler, *Ontology-based Web Agents*, ARPA/ Rome Laboratory Planning Initiative.

[29]. Stefan Decker[1], Frank van Harmelen[3,4], Jeen Broekstra[4,], Michael Erdmann[5], Dieter Fensel[3], Ian Horrocks [2], Michel Klein[3], Sergey Melnik[1] (2003), *The Semantic Web - on the respective Roles of XML and RDF*, IEEE intelligent systems, http://computer.org/intelligent.

[30]. Syed Ahmed (2003), *Ontologies of electronic devicesn in DAML+OIL for automated product design services in the Semantic Web,* Master of engineering in Telecommunication Technology Management, Caleton University, Ottawa Canada, pp 4-89.

[31]. Youngchoon Park (2002), *A frame work for discription, sharing and retrievel of semantic visual information,* Doctor of Philosophy, Arizona State University, pp 1-94.

[32]. CoNLL Share Task: http://www.lsi.upc.edu/~srlconll/

PHỤ LỤC: Kết quả thực nghiệm với các cặp danh từ có tần suất xuất hiện lớn hơn 4 lần.

Chương trình chạy trên tập dữ liệu Penn Tree Bank tìm ra các cặp danh từ có tần suất xuất hiện ≥ 4 sau:

STT	Danh từ	Danh từ		Tần suất xuất hiện
1	Company	Sale		9
2	Smoking	lung cancer	Y	9
3	Smoking	pulmonary problem	Y	8
4	Traffic	Noise	Y	8
5	Standard & Poor	underwriter		8
6	environmental change	erosion	Y	8
7	daylight-saving time	Extra hour	Y	7
8	over age	retirement	Y	7
9	Jewel	robbery	Y	7
10	net income	Share		7
11	Group	Share		7
12	Investors Service Inc.	underwriter		7
13	Bank	provision	Y	7
14	Investor	Stock		7
15	Bad road	traffic jam	Y	6
16	War	Death	Y	6
17	Poverty	malaria	Y	6
18	open-market	investment	Y	6
19	poor rain	slower agriculture	Y	6
20	each index	100		6
21	Chicago Board	Trade		6
22	program trading	market		6
23	Trader	market		6
24	HIV positive	sickness	Y	6
25	good command	victory	Y	6
26	dramatic environmental change	warmer climate	Y	6
27	environmental change	ecosystem change	Y	6

STT	Danh từ	Danh từ		Tần suất xuất hiện
28	Soil	good crop	Y	6
29	Fight	wounded people	Y	6
30	Recklessness	Failure	Y	6
31	Company	Stock		6
32	Billion	Dollar		6
33	bank	paid-up capital	Y	6
34	underwriter	Merrill Lynch Capital Markets		6
35	investor	recession		6
36	Congress	hard decision	Y	6
37	Remic issuance	program		6
38	market	Price		5
39	arms race	poverty	Y	5
40	environmental stress	Breast cancer	Y	5
41	high blood pressure	heart disease	Y	5
42	each index	the close		5
43	problem	problem	Y	5
44	company	Cent		5
45	Cow	Caft	Y	5
46	Merc	Trade		5
47	company	Debt		5
48	president	chief executive officer	Y	5
49	virus	infection	Y	5
50	Fog	delayed flight	Y	5
51	damage	Bay Area		5
52	temperature increase	ice-melting	Y	5
53	loan	Bank	Y	5
54	index	equaling		5
55	major technological breakthrough	annual cost concession	Y	5
56	volcanic effect	warming	Y	5
57	undersea earthquake	tsunamis	Y	5
58	president	company		5
59	Warner	producer		5
60	IBM	equipment	Y	5
61	charge	Share		5

STT	Danh từ	Danh từ		Tần suất xuất hiện
62	charge	Cent		5
63	spokesman	company		5
64	Fannie Mae	program		5
65	money	bank		5
66	sale	company	Y	5
67	issue	Merrill Lynch Capital Markets		5
68	the head coach	a national championship		4
69	chip	image		4
70	provision	bank	Y	4
71	bank	bank		4
72	company	cost		4
73	report	smoking		4
74	Buy-out	buy-out		4
75	great disservice	scotch and water		4
76	public	scotch and water		4
77	dollar	U.S.		4
78	group	investor		4
79	company	ton		4
80	sale	share		4
81	Clean Water Act	scotch and water		4
82	president	Congress		4
83	Congress	president		4
84	scotch and water	hairyknuckled knock		4
85	scotch and water	Sierra Club	Y	4
86	scotch and water	door		4
87	Trader	money	Y	4
88	president	power	Y	4
89	future	investor		4
90	announcement	market		4
91	time	time		4
92	carelessful driver	accident	Y	4
93	Fed	interest rate		4
94	sleeping pill	sleep	Y	4
95	individual stock	average		4

STT	Danh từ	Danh từ		Tần suất xuất hiện
96	magnitude	hazard		4
97	K mart	number one job		4
98	poverty	sickness	٢	4
99	company	market		4
100	K mart	market-share loss		4
101	K mart	discount store		4
102	motor vehicle accident	spinal cord injury	٢	4
103	chief executive officer	company		4
104	price	average		4
105	Buy-out group	bid		4
106	company	plant		4
107	close	trading		4
108	sale	asset		4
109	planner	business		4
110	Early intervention	problem		4
111	money	retirement		4
112	money	first home		4
113	retirement	purchase		4
114	money	purchase	٢	4
115	Way	computer		4
116	earthquake	market		4
117	market	volatility		4
118	Different tactic	money	٢	4
119	California	state official		4
120	computer	phone line		4
121	Way	quake		4
122	Californians	computer		4
123	nation	troubled thrift		4
124	Earthquake	Damage	٢	4
125	quake	computer		4
126	announcement	close		4
127	portfolio	investor		4
128	Two-third	investor		4
129	company	announcement		4

STT	Danh từ	Danh từ		Tần suất xuất hiện
130	shock wave	market		4
131	market	investor	√	4
132	department	bill		4
133	course	firm		4
134	market	firm	√	4
135	Firm	profit	√	4
136	hard decision	right		4
137	percentage basis	share		4
138	Fear	market		4
139	loss	third quarter		4
140	inflation	recession	√	4
141	right	appropriate material and advice		4
142	right	decision		4
143	Germany Fund Inc.	share		4
144	Plan	company		4
145	gainer	share		4
146	right	life		4
147	right	way		4
148	right	rest		4
149	Congress	right	√	4
150	offering	program		4
151	responsibilitie	guardian		4
152	hard decision	complaint	√	4
153	hard decision	fact		4
154	group	alleged earlier violation		4
155	total volume	program		4
156	group	so-called prior-notice requirement		4
157	guardian	stability		4
158	guardian	price level		4
159	guardian	measure		4
160	provision	paid-up capital		4

Ghi chú: những cặp được đánh dấu "v" là những cặp mang ý nghĩa quan hệ nguyên nhân-kết quả.