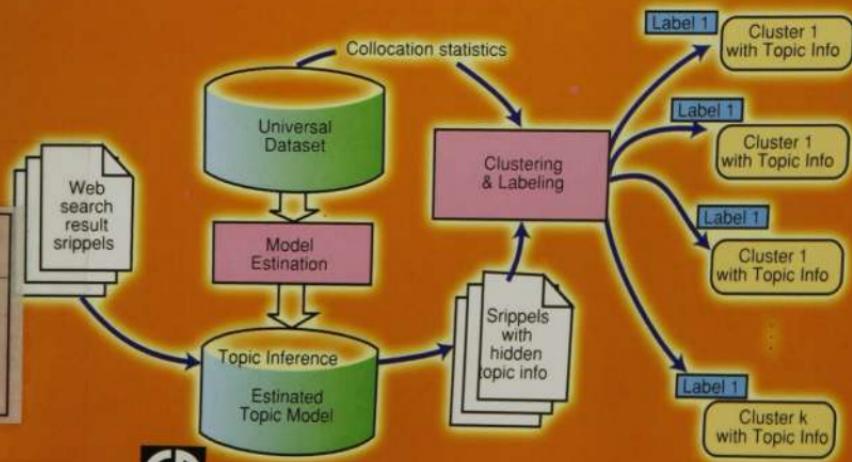


CK.0000046762

HÀ QUANG THỦY (Chủ biên)
N HIẾU - ĐOÀN SƠN - NGUYỄN TRÍ THÀNH
NGUYỄN THU TRANG - NGUYỄN CẨM TÙ

Giáo trình KHAI PHÁ DỮ LIỆU WEB



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

HÀ QUANG THỦY (Chủ biên)
PHAN XUÂN HIẾU – ĐOÀN SƠN – NGUYỄN TRỊ THÀNH
NGUYỄN THU TRANG – NGUYỄN CẨM TÚ

Giáo trình KHAI PHÁ DỮ LIỆU WEB



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

Công ty Cổ phần sách Đại học - Dạy nghề – Nhà xuất bản Giáo dục Việt Nam
giữ quyền công bố tác phẩm.

MỤC LỤC

| | <i>Trang</i> |
|--|--------------|
| LỜI GIỚI THIỆU | 3 |
| Chương 1 MỘT SỐ NỘI DUNG CƠ BẢN VỀ KHAI PHÁ DỮ LIỆU | 9 |
| 1.1. Khai phá dữ liệu và phát hiện tri thức trong cơ sở dữ liệu | 9 |
| 1.2. Khai phá dữ liệu và xử lý cơ sở dữ liệu truyền thống | 20 |
| 1.3. Một số lĩnh vực ứng dụng khai phá dữ liệu điển hình | 22 |
| 1.4. Kiểu dữ liệu trong khai phá dữ liệu | 24 |
| 1.5. Các bài toán khai phá dữ liệu điển hình | 26 |
| 1.6. Tình liên ngành của khai phá dữ liệu | 30 |
| 1.7. Khuynh hướng phát triển của khai phá dữ liệu | 33 |
| Câu hỏi và bài tập | 38 |
| Chương 2 TỔNG QUAN VỀ KHAI PHÁ WEB | 39 |
| 2.1. Giới thiệu về khai phá Text | 39 |
| 2.2. Giới thiệu về khai phá Web | 48 |
| 2.3. Khai phá sử dụng Web | 56 |
| 2.4. Khai phá cấu trúc Web | 66 |
| Câu hỏi và bài tập | 68 |
| Chương 3 MỘT SỐ KIẾN THỨC TOÁN HỌC CHO KHAI PHÁ DỮ LIỆU WEB | 69 |
| 3.1. Mô hình đồ thị | 70 |
| 3.2. Học máy xác suất Bayes | 79 |
| 3.3. Thuật toán Viterbi | 88 |
| Câu hỏi và bài tập | 93 |
| Chương 4 MỘT SỐ VẤN ĐỀ VỀ XỬ LÝ NGÔN NGỮ TIẾNG VIỆT CHO KHAI PHÁ VĂN BẢN | 94 |
| 4.1. Giới thiệu | 94 |
| 4.2. Kho dữ liệu | 96 |
| 4.3. Quan hệ ngữ nghĩa trong văn bản | 96 |
| 4.4. Xử lý ngôn ngữ tiếng Việt | 104 |
| 4.5. Giới thiệu một số nghiên cứu xử lý tiếng Việt | 119 |
| Câu hỏi và bài tập | 120 |
| Chương 5 CÁC PHƯƠNG PHÁP BIỂU DIỄN VĂN BẢN | 121 |
| 5.1. Phân tích văn bản | 121 |
| 5.2. Các mô hình biểu diễn văn bản | 125 |
| 5.3. Các phương pháp lựa chọn các từ trong biểu diễn văn bản | 129 |
| 5.4. Thu gọn đặc trưng biểu diễn | 132 |
| 5.5. Phương pháp biểu diễn trang Web | 139 |
| Câu hỏi và bài tập | 142 |
| Chương 6 HỆ THỐNG TÌM KIẾM | 143 |
| 6.1. Tìm kiếm trên Web | 143 |
| 6.2. Máy tìm kiếm | 146 |
| 6.3. Cấu trúc và hoạt động của một máy tìm kiếm | 151 |
| 6.4. Crawling trang Web | 153 |
| 6.5. Phân tích và đánh giá số | 167 |

| | | |
|-------------------|---|------------|
| 6.6 | Tính hạng trang Web | 173 |
| 6.7. | Máy tìm kiếm thực thể | 183 |
| | Câu hỏi và bài tập | 185 |
| Chương 7. | PHÂN CỤM VĂN BẢN | 186 |
| 7.1. | Giới thiệu | 186 |
| 7.2. | Thuật toán phân cụm k-means | 191 |
| 7.3. | Thuật toán phân cụm phân cấp từ dưới lên | 197 |
| 7.4. | Thuật toán phân hoạch từ trên xuống | 201 |
| 7.5. | Gán nhãn cho các cụm | 202 |
| 7.6 | Đánh giá thuật toán phân cụm | 204 |
| 7.7 | Mô hình phân cụm kết quả tìm kiếm và gán nhãn cụm tiếng Việt | 211 |
| | Câu hỏi và bài tập | 219 |
| Chương 8. | PHÂN LỚP VĂN BẢN | 220 |
| 8.1. | Giới thiệu | 220 |
| 8.2. | Một số thuật toán phân lớp có giám sát | 223 |
| 8.3. | Học bản giám sát và một số thuật toán phân lớp bản giám sát | 232 |
| | Câu hỏi và bài tập | 241 |
| Chương 9. | TRÍCH CHỌN THÔNG TIN TRÊN WEB | 242 |
| 9.1. | Giới thiệu | 242 |
| 9.2. | Các phương pháp trích chọn thông tin từ văn bản Web phi cấu trúc | 251 |
| 9.3. | Các phương pháp trích chọn thông tin chủ đề trên Web | 267 |
| | Câu hỏi và bài tập | 274 |
| Chương 10. | WEB NGỮ NGHĨA | 275 |
| 10.1. | Giới thiệu Web ngữ nghĩa | 275 |
| 10.2. | Kiến trúc của Web ngữ nghĩa | 277 |
| 10.3. | Các ngôn ngữ nền tảng cho Web ngữ nghĩa | 280 |
| 10.4. | Tiệm cận tới Web ngữ nghĩa | 292 |
| | Câu hỏi và bài tập | 299 |
| | TÀI LIỆU THAM KHẢO | 300 |

LỜI GIỚI THIỆU

Trong cuốn sách nổi tiếng "*Data Mining – Concepts and Techniques*" hai tác giả Jiawei Han và Micheline Kamber nhận định rằng, tình trạng "giàu về dữ liệu mà nghèo về thông tin" là một động lực phát triển lĩnh vực khai phá dữ liệu và phát hiện tri thức trong cơ sở dữ liệu (CSDL). Hoạt động nghiên cứu và triển khai xây dựng các hệ thống tự động nhận ra các mẫu có giá trị, mới, hữu ích tiềm năng và hiệu được trong khối dữ liệu đồ sộ, nhằm bổ sung tài nguyên tri thức cho con người là hết sức cần thiết và có ý nghĩa trong quá trình hình thành và phát triển kinh tế tri thức.

Ngày nay, World Wide Web đã trở thành một kho tài nguyên dữ liệu không lồ về mọi lĩnh vực; kho tài nguyên dữ liệu này đang không ngừng tăng trưởng với tốc độ cao. Kho tài nguyên dữ liệu Web tiềm ẩn nhiều mảnh thông tin quý giá đối với hoạt động của cộng đồng nói chung và từng cá thể nói riêng. Các hệ thống khai phá dữ liệu Web đã trở thành các công cụ lá chắn cho tài nguyên Web "kho trối chung vô tận của riêng mình" (Cao Bá Quát) thực sự phát huy hiệu quả tới cộng đồng và tới mỗi cá thể trong cộng đồng. Phù hợp với sự phát triển của Web, hoạt động nghiên cứu và triển khai và khai phá dữ liệu Web không ngừng được tăng trưởng. Hiệp hội các nhà khoa học về Phát hiện tri thức và Khai phá dữ liệu (The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining, viết tắt là SIGKDD) đã tập hợp được nhiều nhà khoa học trong đó có nhiều nhà khoa học máy tính nổi tiếng thế giới. Từ năm 1995 tới nay, hoạt động diễn hình nhất của SIGKDD là tổ chức Hội nghị Khoa học quốc tế thường niên *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Khai phá dữ liệu Web đã trở thành một trong những nội dung nhận được nhiều quan tâm nhất tại *ACM SIGKDD Conference on Knowledge Discovery and Data Mining* và các hội nghị khoa học quốc tế lớn khác.

Từ năm 2006, "*Khai phá dữ liệu Web*" đã là một môn học trong Chương trình đào tạo ngành Công nghệ thông tin (CNTT) và ngành Hệ thống thông tin (HTTT) tại Khoa Công nghệ Thông tin, Trường Đại học Công nghệ (ĐHCHN), Đại học Quốc gia Hà Nội (ĐHQGHN). *Giáo trình Khai phá dữ liệu Web* này được tập hợp và hoàn thiện từ nội dung các bài giảng trong thời gian vừa qua, nhằm cung cấp một tài liệu hoàn chỉnh phục vụ hoạt động giảng dạy và học tập môn học này tại Khoa CNTT. Trong

DHCN cả ở bậc đại học và sau đại học. Các nội dung trong giáo trình không chỉ đáp ứng yêu cầu đào tạo về lĩnh vực khoa học và công nghệ liên quan, mà còn cung cấp một số kiến thức và kỹ năng mở rộng và chuyên sâu phục vụ nhu cầu nghiên cứu và phát triển lĩnh vực khai phá dữ liệu Web không chỉ tại Trường DHCN mà còn ở các cơ sở đào tạo và nghiên cứu khác trong nước.

Giáo trình gồm 10 chương, nội dung sơ bộ như sau:

Chương 1 – Một số nội dung cơ bản về khai phá dữ liệu cung cấp các kiến thức cơ bản nhất về lĩnh vực khai phá dữ liệu và phát hiện tri thức trong các CSDL, nhằm giúp độc giả nắm bắt được bản chất của các khái niệm cơ bản trong khai phá dữ liệu, phân biệt các khái niệm này với một số khái niệm liên quan và một số bài toán cơ bản nhất và xu hướng phát triển của khai phá dữ liệu, phát hiện tri thức trong các CSDL.

Chương 2 – Tổng quan về khai phá Web cung cấp các kiến thức cơ bản nhất về khai phá Text và khai phá Web, nhằm giúp độc giả nắm bắt được các nội dung cơ bản của khai phá Text và khai phá Web. Chương này cũng trình bày cơ bản về khai phá cấu trúc Web và khai phá sử dụng Web.

Chương 3 – Một số kiến thức toán học cho khai phá dữ liệu Web nhằm mục tiêu cung cấp một số kiến thức nền tảng về toán học cho khai phá dữ liệu Web. Lý thuyết đồ thị và lý thuyết xác suất thâm nhập sâu rộng vào khai phá dữ liệu Web theo các góc độ mô hình, giải pháp và kỹ thuật có nguồn gốc từ bản chất tự nhiên và xã hội của Web.

Chương 4. Một số vấn đề về xử lý ngôn ngữ tiếng Việt cho khai phá văn bản cung cấp một số kiến thức nền tảng về xử lý ngôn ngữ tự nhiên nói chung và xử lý tiếng Việt nói riêng, cho phép nâng cao hiệu quả của các giải pháp khai phá Web tiếng Việt.

Chương 5 – Các phương pháp biểu diễn văn bản trình bày bài toán các khuôn dạng biểu diễn dữ liệu cho các thuật toán khai phá dữ liệu.

Chương 6 – Hệ thống tìm kiếm, Chương 7 – Phân cụm văn bản, Chương 8 – Phân lớp Web, Chương 9 – Trích chọn thông tin trên Web trình bày về bốn bài toán chủ yếu của khai phá dữ liệu Web. Các khái niệm liên quan, các mô hình biểu diễn, các thuật toán, các kỹ thuật và các phương pháp đánh giá hiệu quả được giới thiệu và phân tích.

Chương 10 – Web ngữ nghĩa trình bày về Web ngữ nghĩa, thể hệ mới của Web gồm khái niệm, kiến trúc, các ngôn ngữ và quá trình tiệm cận tới Web ngữ nghĩa.

Trong quá trình biên soạn giáo trình này, chúng tôi được khai thác nguồn tài nguyên phong phú, bao gồm nhiều bài báo khoa học, các tiện ích và sản phẩm phần mềm thuộc lĩnh vực khai phá Web. Đây là một thuận lợi

lớn về nguồn chất liệu biên soạn giáo trình. Nhóm tác giả xin bày tỏ lời cảm ơn chân thành tới TS. Nguyễn Lê Minh, Nghiên cứu sinh Nguyễn Viết Cường hiện đang công tác tại Viện Khoa học và Công nghệ tiên tiến Nhật Bản và Nghiên cứu sinh Đặng Thanh Hải hiện đang công tác tại Đại học Antwerp – Bỉ về việc cộng tác triển khai các hoạt động nghiên cứu liên quan. Nhóm tác giả đánh giá cao và chân thành cảm ơn tập thể cán bộ, sinh viên thuộc Phòng thí nghiệm Công nghệ tri thức, Trường ĐHCN đã cộng tác nghiên cứu, triển khai các đề tài KC.01.02/06-10, NCCB 203904 QC.07.13, QC.07.06. Giáo trình này là một sản phẩm của Phòng thí nghiệm Công nghệ tri thức, Bộ môn HTTT được hoàn thành nhân dịp 10 năm truyền thống của Trường ĐHCN (tháng 10/2009). Trong môi trường của một trường đại học định hướng nghiên cứu, các tác giả đã và đang nhận được sự tham gia đóng góp tích cực từ đội ngũ người học trong việc đàm bảo tính cập nhật về nội dung và tính hiệu quả về cấu trúc của giáo trình. Một số nghiên cứu của nhóm tác giả được trình bày trong giáo trình này là kết quả cộng tác nghiên cứu của chúng tôi với Cố Giáo sư Susumu Horiguchi tại Viện Khoa học & Công nghệ tiên tiến Nhật Bản và Đại học Tohoku.

Nhóm tác giả cũng gặp một số khó khăn khi biên soạn giáo trình. Khó khăn thứ nhất là vấn đề lựa chọn thuật ngữ tiếng Việt. Đối với lĩnh vực kinh tế Web, việc lựa chọn thuật ngữ tiếng Việt là rất khó khăn, vì đây là lĩnh vực nghiên cứu còn rất mới không chỉ ở Việt Nam mà còn trên thế giới. Vì vậy, ngay một số thuật ngữ tiếng Anh cũng có một vài phương án trình bày và nhiều nghĩa. Khó khăn thứ hai là về tính hoàn thiện nội dung trong giáo trình đối với một lĩnh vực nghiên cứu mới với nội dung rất phong phú. Dù nhóm tác giả đã cố gắng thu thập, nghiên cứu và tổng hợp, song giáo trình khó tránh khỏi khiếm khuyết. Chúng tôi rất mong nhận được các ý kiến đóng góp từ các nhà khoa học, các giảng viên và người học để giáo trình ngày càng thêm hoàn thiện.

Mọi ý kiến đóng góp xin gửi về: Công ty CP Sách Đại học – Dạy nghề NXB Giáo dục Việt Nam, 25 Hàn Thuyên – Hà Nội.

Hà Nội, tháng 9 năm 2009
CÁC TÁC GIÀ

Chương 1

MỘT SỐ NỘI DUNG CƠ BẢN VỀ KHAI PHÁ DỮ LIỆU

1.1. Khai phá dữ liệu và phát hiện tri thức trong cơ sở dữ liệu

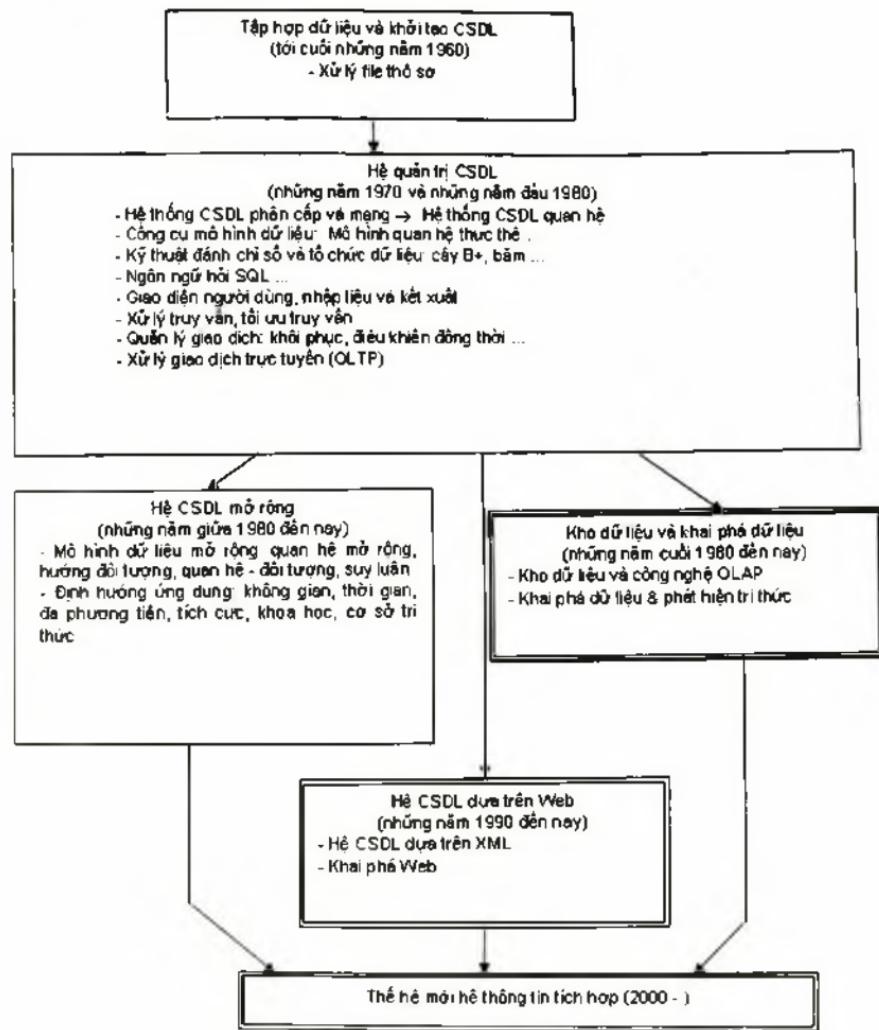
Theo J. Han và M. Kamber [HK0106], quá trình tiến hoá của lĩnh vực công nghệ cơ sở dữ liệu (CSDL) được mô tả như Hình 1.1, trong đó công nghệ khai phá dữ liệu (Data Mining) được coi là giai đoạn tiến hoá mới của công nghệ CSDL. Quá trình tiến hoá này được bắt đầu từ cuối những năm 1980 và không ngừng được phát triển về bề rộng và chiều sâu.

Trước tiên, xét sơ bộ về mục đích nghiên cứu của lĩnh vực khai phá dữ liệu. Theo Fayyad và cộng sự [FPS96], việc nghiên cứu, phát triển lĩnh vực khai phá dữ liệu và phát hiện tri thức trong CSDL (Knowledge Discovery in Databases: KDD) nhằm giải quyết tình trạng "*ngập tràn thông tin mà thiếu thông tri thức*". Số lượng thông kê dưới đây được đưa ra vào năm 2006 [Pia06] đã minh chứng cho tình trạng "*ngập tràn thông tin*" là hiện nay tồn tại nhiều kho chứa dữ liệu đã trở nên khổng lồ mà hàng ngày dung lượng của chúng còn được tăng trưởng với tốc độ cao. Về dữ liệu Web, diễn hình là *Alexa*, sau 7 năm đã có 500TB (terabyte), *Google* đã lưu trữ hơn 4 tỷ trang Web với dung lượng nhiều trăm terabytes, *IBM WebFountain* với hơn 160TB, *Internet Archive*⁽¹⁾ xấp xỉ 300TB,... Về CSDL, diễn hình là *Max Planck Institute for Meteorology* có tới hơn 220TB. *Yahoo!* có hơn 100TB còn *AT&T* có gần 100TB⁽²⁾. Theo ước lượng của UC Berkeley 2003 thì có tới 5 exabytes (5 triệu terabytes) dữ liệu mới được khởi tạo trong năm 2002. Mục đích của việc thu thập và lưu trữ các kho chứa dữ liệu khổng lồ được liệt kê ở trên không ngoài mục đích khai phá dữ liệu, nhằm phát hiện các tri thức mới giúp ích cho hoạt động của con người trong tập hợp dữ liệu. Chẳng hạn, từ một giải pháp phân lớp trong khai phá dữ liệu Web (Web Mining), có thể phát triển thành một thành phần của máy tìm kiếm (Search Engine).

⁽¹⁾ <http://www.archive.org>.

⁽²⁾ http://www.wintercorp.com/Vldb/2005_TopTen_Survey/TopTenWinners_2005.asp.

để khi một trang Web mới được tải về, máy tìm kiếm sẽ tự động phân nó vào một lớp trang Web đã được xác định; việc phân lớp đó sẽ tạo ra thuận lợi cho việc tìm kiếm về sau của người dùng. Trong tình trạng kích thước Web đã và đang có độ tăng trưởng cao, việc phân lớp tự động như vậy thực sự rất có ý nghĩa.



Hình 1.1. Tiến hóa công nghệ CSDL

Lĩnh vực khai phá dữ liệu và phát hiện tri thức trong CSDL đã tập hợp các phương pháp, thuật toán và kỹ thuật từ nhiều chuyên ngành nghiên cứu

khác nhau như thu nhận mẫu, CSDL, thống kê, trí tuệ nhân tạo, thu nhận tri thức trong hệ chuyên gia,... cùng hướng tới một mục tiêu thống nhất, trích lọc được các "tri thức" từ dữ liệu trong các CSDL không lô. Tinh phong phú và đa dạng của lĩnh vực khai phá dữ liệu dẫn đến một thực trạng là, tồn tại các quan niệm khác nhau về chuyên ngành khoa học – công nghệ gần gũi nhất với lĩnh vực đó. Giáo trình này tán thành quan niệm của J. Han và M. Kamber, coi lĩnh vực khai phá dữ liệu là giai đoạn phát triển mới của công nghệ CSDL và có liên quan mật thiết với nhiều liên ngành. Như vậy, có thể gắn lĩnh vực này với chuyên ngành hệ thống thông tin.

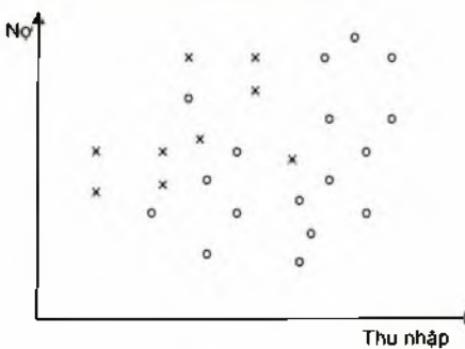
Ví dụ 1.1. (Frawley, Piatetski-Shapiro và Matheus [FPS96])

Hình 1.2 trình bày một tập dữ liệu giả định về vay nợ ngân hàng, gồm 23 trường hợp được biểu diễn trong không gian hai chiều. Mỗi điểm trên đồ thị biểu diễn một trường hợp vay nợ ở ngân hàng trong quá khứ. Trục hoành biểu diễn thu nhập, trục tung biểu diễn tổng nợ cá nhân của người đi vay (tiền thế chấp, tiền chi trả ô tô,...). Dữ liệu được phân thành hai lớp: lớp \times gồm những người thiếu khả năng trả nợ ngân hàng, lớp \circ gồm những người có tình trạng tốt.

Khái niệm 1.1 [FPS96]

Phát hiện tri thức trong cơ sở dữ liệu (đôi khi còn được gọi là khai phá dữ liệu) là một quá trình không thường nhân ra những mẫu có giá trị mới, hữu ích tiềm năng và hiểu được trong dữ liệu.

Là lĩnh vực nghiên cứu và triển khai được phát triển rất nhanh chóng, có phạm vi rất rộng lớn, lại được rất nhiều nhóm nghiên cứu tại nhiều trường đại học, viện nghiên cứu, công ty ở nhiều quốc gia trên thế giới quan tâm, cho nên tồn tại rất nhiều cách tiếp cận khác nhau đối với lĩnh vực phát hiện tri thức trong CSDL. Chính vì lý do đó, trong nhiều tài liệu, như đã nói ở trên, các nhà khoa học đã dùng nhiều thuật ngữ khác nhau, mà các thuật ngữ này được coi là mang cùng nghĩa với KDD như chiết lọc tri thức (knowledge extraction), phát hiện thông tin (information discovery), thu hoạch thông tin (information harvesting), khai thác dữ liệu (data archaeology)



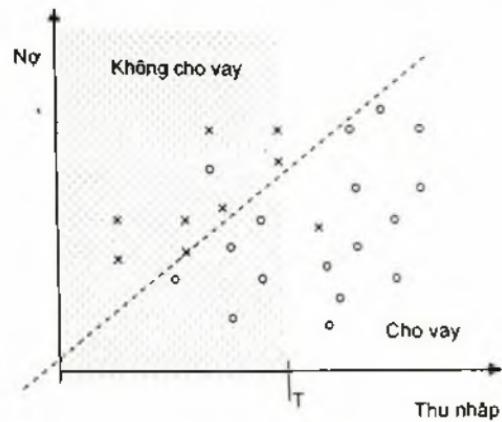
Hình 1.2. Tập dữ liệu có hai lớp \times và \circ

xử lý mẫu dữ liệu (data pattern processing),... Hơn nữa, trong nhiều trường hợp, hai khái niệm "Phát hiện tri thức trong cơ sở dữ liệu" và "khai phá dữ liệu" còn được dùng thay thế nhau [FPS96]. Hai khái niệm khai phá dữ liệu và phát hiện tri thức trong các CSDL thường cặp đôi với nhau.

J. Han và M. Kamber quan niệm rằng, cụm từ tiếng Anh "Data Mining" chưa diễn tả đầy đủ và toàn diện ý nghĩa của lĩnh vực nghiên cứu – triển khai mà nó mang tên. Một cách tương ứng trong tiếng Việt, cụm từ "khai phá dữ liệu" cũng được nhiều nhà khoa học Việt Nam bắn khoan vì cho rằng, cụm từ này chưa bao hàm được hết nội dung ngữ nghĩa cần diễn tả. Tuy nhiên, tương ứng với cụm từ tiếng Anh "Data Mining" (mang nội dung được J. Han và M. Kamber xác định), trong giáo trình này chúng tôi chọn thuật ngữ tiếng Việt là "khai phá dữ liệu" vì thuật ngữ tiếng Việt đã trở thành phổ biến trong các tài liệu tiếng Việt liên quan hiện nay.

Một số thuật ngữ có trong khái niệm 1.1 ở trên cần được giải thích là "dữ liệu", "mẫu", "có giá trị", "mới", "hữu ích", "hiệu được".... Dưới đây trình bày một số giải thích sơ bộ về các khái niệm, nhằm làm tinh minh thêm ngữ nghĩa của khái niệm KDD trong khái niệm 1.1.

- *Dữ liệu* (chính xác hơn là *tập dữ liệu*) được hiểu như là một tập F gồm hữu hạn các *trường hợp* (*sự kiện*). Theo nội dung của phát hiện tri thức trong các CSDL, dữ liệu phải bao gồm nhiều trường hợp. Trong ví dụ 1.1, F là tập hợp gồm 23 trường hợp (bán ghi) với 3 trường thông tin (thuộc tính) tương ứng chứa các giá trị về *số nợ*, *thu nhập* và *tình trạng vay nợ*. Trong bài toán khai phá văn bản, tập dữ liệu F chính là tập hợp các văn bản có thể có trong miền ứng dụng. Trong bài toán khai phá luật kết hợp giao dịch, tập F bao gồm tất cả các giao dịch có thể có được trong miền áp dụng của bài toán.



Hình 1.3. Nguadro đơn T theo thu nhập để phân lớp cho vay (Lưu ý: đường nghiêng rời nét cho quyết định tốt hơn)

• *Mẫu*. Trong quá trình KDD, người ta sử dụng một ngôn ngữ L để biểu diễn các tập con các sự kiện (dữ liệu) thuộc vào tập sự kiện F, theo đó mỗi biểu thức E trong ngôn ngữ L sẽ biểu diễn một tập con F_E tương ứng các sự kiện trong F. E được gọi là *mẫu* nếu nó đơn giản hơn (theo một ngữ cảnh nào đó) so với việc liệt kê các sự kiện thuộc F_E . Chẳng hạn, biểu thức " $\text{THUNHẬP} < \$t$ " (mô hình chứa một biến THUNHẬP) trong mệnh đề "*Nếu* $\text{THUNHẬP} < \$t$ *thì* người vay nợ roi vào tình trạng không thể chi trả" sẽ là một mẫu khi cho biến t nhận một giá trị thích hợp. Như trình bày bằng đồ thị tại Hình 1.3, khi biến t nhận một giá trị cụ thể T, mẫu này (biểu diễn mọi trường hợp có $\text{THUNHẬP} < T$) hiển nhiên là gọn hơn so với việc liệt kê 14 trường hợp cụ thể. Tương tự, nếu F là tập các trang Web trong kho lưu trữ của một máy tìm kiếm (chẳng hạn Google), thì mẫu "tài liệu có chứa từ cụm từ "Search Engine"" sẽ biểu diễn một tập bao gồm một số lượng rất lớn các tài liệu Web có chứa cụm từ "Search Engine" đó.

• Quá trình KDD thường bao gồm *nhiều bước* như *chuẩn bị dữ liệu*, *tìm kiếm mẫu*, *ước lượng trí thức*, *tinh chế sự tương tác nội* tại sau khi chuyển dạng dữ liệu. Quá trình được thừa nhận là *không tầm thường* theo nghĩa là quá trình đó không chi nhiều bước, mà còn được thực hiện lặp, quan trọng hơn là quá trình đó bao hàm một mức độ tìm kiếm tự động. Chẳng hạn, trong *Ví dụ 1.1*, khi tính toán ý nghĩa về thu nhập của một người, nếu chỉ thông qua các tác động đơn giản mà chúng ta thu nhận được một kết luận nào đó có thể là hữu ích thì đúng với cho rằng, đó đã là một khám phá (hoặc đúng cho rằng một tri thức đã được phát hiện).

• *Có giá trị*: Mẫu được phát hiện cần phải có *giá trị* đối với các dữ liệu mới (xuất hiện trong tương lai) theo một mức độ chân thực nào đấy. Tính chất "có giá trị" được hiểu theo nghĩa liên quan tới một *độ đo tính có giá trị* (*chân thực*) là một hàm C ánh xạ một biểu thức thuộc ngôn ngữ biểu diễn mẫu L tới một không gian đo được (bộ phận hoặc toàn bộ) M_C . Một biểu thức E trong L biểu diễn một tập con $F_E \subset F$ có thể được gán một độ đo chân thực c = C(E, F).

Chẳng hạn, nếu đường biên xác định mẫu " $\text{THUNHẬP} < \$t$ " như chỉ dẫn trong Hình 1.3 được dịch sang phái (biến THUNHẬP nhận giá trị lớn hơn) thì độ chân thực của mẫu mới sẽ bị giảm xuống, bởi vì nó đã bao gói thêm các tình huống vay tối lại bị đưa vào vùng không cho vay nợ.

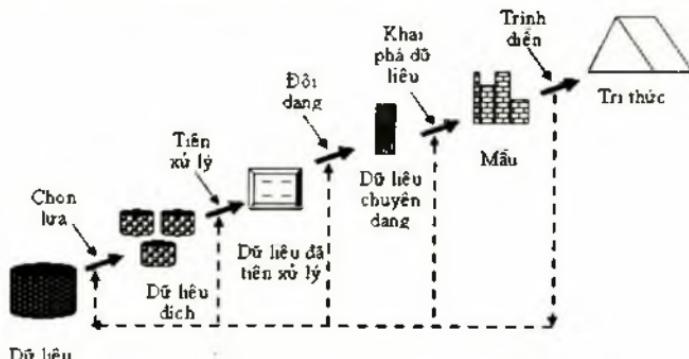
Tương tự, mẫu "*Nếu* $a * \text{THUNHẬP} + b * NQ < 0$ (*thuộc* mô hình tuyến tính hai biến THUNHẬP và NQ trong $\alpha * \text{THUNHẬP} + \beta * NQ$) *thì* người vay nợ roi vào tình trạng không thể chi trả" biểu diễn một nửa mặt phẳng phía trên của đường rời nét trong Hình 1.3 sẽ cho độ chân thực cao hơn (hay được coi là "có giá trị hơn") so với mọi mẫu thuộc mô hình một biến " $\text{THUNHẬP} < \$t$ ".

• **Tính mới:** Mẫu phải là mới trong một miền xem xét nào đó, ít nhất là hệ thống đang được xem xét. **Tính mới** có thể đo được khi quan tâm tới sự thay đổi trong dữ liệu (bằng việc so sánh giá trị hiện tại với giá trị quá khứ hoặc giá trị kỳ vọng) hoặc tri thức (tri thức mới quan hệ như thế nào với các tri thức đã có). Tổng quát, điều này có thể được đo bằng một hàm $N(E, F)$, hoặc là độ đo về tính mới, hoặc là độ đo kỳ vọng.

• **Hữu ích tiềm năng:** Mẫu cần có khả năng chỉ dẫn tới các tác động hữu dụng và được đo bởi một hàm tiện ích. Chẳng hạn, hàm U ánh xạ các biểu thức trong L tới một không gian đo có thứ tự (bộ phận hoặc toàn bộ) M_U , theo đó $u = U(E, F)$. Ví dụ, trong tập dữ liệu vay nợ, hàm này có thể là *sự tăng hy vọng theo sự tăng lãi của nhà băng* (tính theo đơn vị tiền tệ) kết hợp với quy tắc quyết định được trình bày trong Hình 1.3.

• **Có thể hiểu được:** Một mục tiêu của KDD là tạo ra *các mẫu mà con người hiểu chúng dễ dàng hơn* các dữ liệu nền (dữ liệu sẵn có trong hệ thống). **Có thể hiểu được** là tiêu chí khó để được một cách chính xác, cho nên thường tính chất "có thể hiểu được" được thay bằng một độ đo về sự dễ hiểu. Tồn tại một số độ đo về sự dễ hiểu, các độ đo như vậy được sắp xếp từ cú pháp (tức là cỡ của mẫu theo bit) tới ngữ nghĩa (tức là dễ dàng để con người nhận thức được theo một tác động nào đó). Bởi lý do đó, giả định rằng tính hiểu được là *đo được* bằng một hàm S ánh xạ biểu thức E trong L tới một không gian đo được có thứ tự (bộ phận hoặc toàn bộ) M_S ; theo đó, $s = S(E, F)$.

• **Độ hấp dẫn:** Một tiêu chí quan trọng được gọi là *độ hấp dẫn*, thường được coi như một *độ đo tổng thể về mẫu* là sự kết hợp của các tiêu chí *giá trị, mới, hữu ích và dễ hiểu*. Một số hệ thống KDD thường sử dụng một hàm hấp dẫn dưới dạng hiển là $i = I(E, F, C, N, U, S)$ thực hiện ánh xạ một biểu thức trong L vào một không gian đo được M_i . Một số hệ thống KDD khác lại có thể xác định giá trị hấp dẫn của mẫu một cách trực tiếp thông qua thứ tự của các mẫu được phát hiện.



Hình 1.4. Quá trình phát hiện tri thức trong CSDL

Trong thực tiễn giải quyết các bài toán khai phá dữ liệu, người ta thường chỉ quan tâm đến độ hấp dẫn, còn các độ đo khác được mặc định coi là thành phần của độ hấp dẫn. Cụ thể là, khi thi hành một loại bài toán phát hiện tri thức cụ thể, một số độ đo tương ứng được tính toán nhằm xác định độ hấp dẫn của tri thức ("mẫu", "luật") đang được xem xét. Chẳng hạn, trong bài toán khai phá luật kết hợp, hai độ đo được xem xét đó là *độ hỗ trợ* (xác định phạm vi ảnh hưởng của luật) và *độ tin cậy* (xác định tính tin cậy của luật) hợp thành độ hấp dẫn của luật kết hợp đã được khai phá. Tương tự, trong bài toán phân lớp, người ta sử dụng hai độ đo cơ bản là *độ hồi phục* (khả năng bao gói ví dụ đúng) và *độ chính xác* (khả năng chính xác khi xác định ví dụ đúng); đồng thời, một số độ đo mang ý nghĩa kết hợp từ hai độ đo này cũng được sử dụng.

- *Tri thức*: Một mẫu $E \in L$ được gọi là *tri thức* nếu như đối với một lớp người sử dụng nào đó, chỉ ra được một ngưỡng $i \in M_i$ mà $\text{độ hấp dẫn } I(E, F, C, N, U, S) > i$.

Chú ý rằng, khái niệm "tri thức" trên không mang một nghĩa tuyệt đối, mà phụ thuộc vào quan điểm của người sử dụng hệ thống KDD ("một lớp người sử dụng nào đó"). Như một nội dung của sự kiện, nó chỉ là một định hướng cho người sử dụng và được xác định bằng bất kỳ hàm và ngưỡng nào được người sử dụng chọn. Chẳng hạn, trong bài toán khai phá luật kết hợp, chúng ta chỉ quan tâm tới các "*tập phô biến*" là những tập có độ hỗ trợ vượt qua một ngưỡng *minsup* nào đó. Hơn nữa, chỉ các luật kết hợp có độ tin cậy vượt quá ngưỡng *minconf* mới được khai phá để cung cấp tri thức tới người sử dụng. Các ngưỡng *minsup* và *minconf* có thể được thay đổi theo lựa chọn của người sử dụng.

Một cách hình thức, thuyết minh cụ thể của định nghĩa trên về "tri thức" là chọn ngưỡng nào đó $c \in M_C$ (về tính "*có giá trị*"), $s \in M_S$ (về tính "*có thể hiệu được*") và $u \in M_U$ (về tính "*hữu ích*") và khi đó gọi mẫu E là *tri thức* nếu và chỉ nếu:

$$C(E, F) > c \text{ và } S(E, F) > s \text{ và } U(E, F) > u$$

Thông qua việc đặt các ngưỡng thích hợp với mục đích phát hiện tri thức, người sử dụng có thể nhấn mạnh một dự báo chính xác hoặc các mẫu hữu ích (vượt qua một độ đo đánh giá nào đó) qua những độ đo liên quan. Rõ ràng là, tồn tại một không gian vô hạn cho phép ánh xạ I xác định "tri thức cần phát hiện". Quyết định như vậy là tự do đối với người sử dụng và được đặc trưng đối với từng miền ứng dụng.

Ken McGarry [Gar05] trình bày một nghiên cứu tổng quan về việc sử dụng các độ đo hấp dẫn được dùng phổ biến trong phát hiện tri thức trong CSDL. Có thể phân chung theo lớp độ đo hướng mục tiêu, lớp độ đo hướng

chủ đề và lớp độ đo cho luật kết hợp. Tác giả nhận xét rằng, tồn tại rất nhiều các độ đo hướng chủ đề để đáp ứng miền rộng lớn các ứng dụng, và vì vậy rất thuận tiện để chọn ra một độ đo phù hợp đối với một miền ứng dụng đã cho.

Những điều trình bày trên cho thấy vai trò của hệ thống KDD cũng như vai trò của người sử dụng trong một phiên làm việc của mình, tạo nên sự cộng tác giữa người sử dụng và hệ thống KDD. Trong sự cộng tác đó, hệ thống KDD tạo thuận tiện cho người sử dụng có cách thức linh hoạt dùng các nguồn để được cung cấp "tri thức" từ hệ thống phù hợp với những dự đoán chủ quan của mình. Như vậy, có thể thấy rằng, cùng dùng một phần mềm KDD, song mỗi người sử dụng lại có thể khai thác nó theo cách thức riêng của mình.

Theo B.Kovalerchuk và E.Vityaev [KV01], Friedman đã tổng hợp một số quan niệm sau đây liên quan về khái niệm "khai phá dữ liệu":

- Quá trình không tầm thường để nhận biết từ dữ liệu ra các mẫu có giá trị, mới, hữu dụng và hiểu được (Fayyad);
- Quá trình trích lọc các thông tin chưa biết trước, có thể nhận thức được, có thể tác động được từ CSDL lớn và sử dụng chúng để tạo ra quyết định công tác (Zekulin);
- Tập các phương pháp được dùng trong quá trình phát hiện tri thức nhằm tường minh các quan hệ và các mẫu chưa biết trước chưa trong dữ liệu (Ferruzza);
- Quá trình hỗ trợ quyết định khi tìm kiếm những mẫu thông tin chưa biết và hữu ích từ CSDL lớn (Parsaye).

Giáo trình này tiếp nhận quan điểm của Fayyad, Piatetsky-Shapiro, Smyth, như đã trình bày trong *Khái niệm 1.1*, chúng ta coi KDD là một quá trình bao gồm nhiều bước thực hiện, trong đó, khai phá dữ liệu là một bước thực hiện chính yếu. Cách hiểu như vậy đã quy định có sự phân biệt giữa hai khái niệm khai phá dữ liệu và KDD.

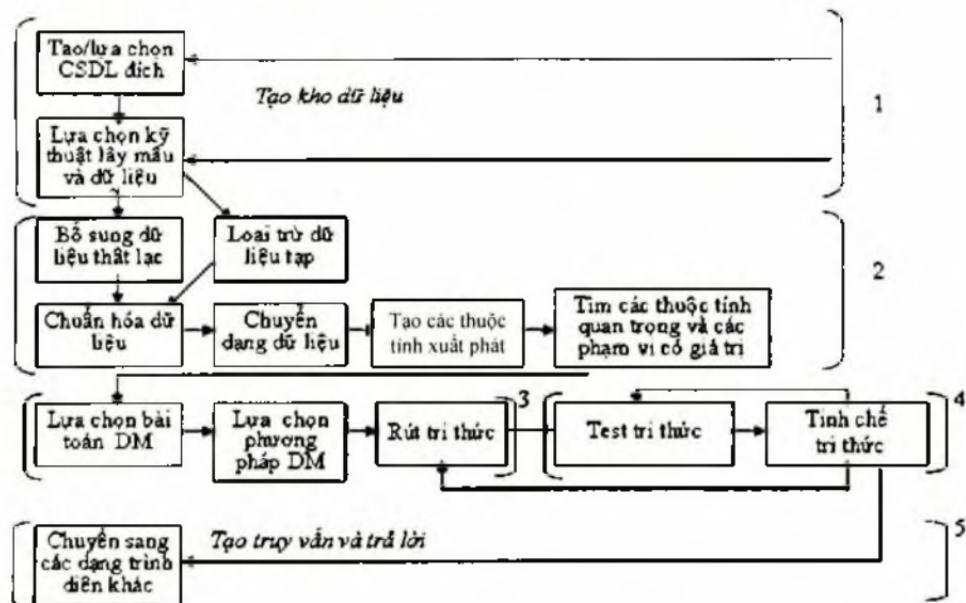
Khái niệm 1.2 (Frawley, Piatetski-Shapiro và Matheus [FPS96])

Khai phá dữ liệu là một bước trong quá trình Phát hiện tri thức trong cơ sở dữ liệu, thi hành một thuật toán khai phá dữ liệu để tìm ra các mẫu từ dữ liệu theo khuôn dạng thích hợp.

Tương ứng với sơ đồ mô tả chi tiết quá trình KDD (Hình 1.5), các nhóm bước thực hiện sau đây được tiến hành trong quá trình phát hiện tri thức trong CSDL:

- (1) *Mở rộng hiểu biết* về miền ứng dụng, về các tri thức với độ ưu tiên thích hợp và về mục đích của người dùng cuối. Có thể coi nội dung công việc này tương ứng với nội dung khảo sát bài toán trong quá trình xây dựng một hệ thống thông tin nói chung.

Khởi tạo tập dữ liệu đích, tạo kho dữ liệu: chọn tập dữ liệu "và/hoặc" hướng trọng tâm tới tập con các biến hoặc mẫu dữ liệu mà trên đó công việc phát hiện tri thức được tiến hành. Tri thức miền ứng dụng có được thông qua việc mở rộng hiểu biết về miền ứng dụng nói trên đóng vai trò là nền tảng tri thức để khởi tạo tập dữ liệu đích, kho dữ liệu.



Hình 1.5. Mô tả chi tiết các bước trong quá trình KDD

(2) *Làm sạch và tiền xử lý dữ liệu:* thực hiện các thao tác cơ sở như giải quyết thiếu vắng giá trị, loại bỏ nhiễu hoặc yếu tố ngoại lai, kết nối các thông tin cần thiết tới mô hình hoặc loại bỏ nhiễu, quyết định chiến lược nhằm nắm bắt các trường dữ liệu (các thuộc tính), tính toán dây thông tin thời gian và sự biến đổi được định trước.

Chất lượng của hệ thống khai phá dữ liệu phụ thuộc vào chất lượng của dữ liệu đầu vào. Mục tiêu của *làm sạch dữ liệu* nhằm đảm bảo dữ liệu đầu vào có chất lượng tốt.

Thu gọn và trình diễn dữ liệu có mục tiêu tìm được các đặc trưng hữu ích nhằm trình bày mỗi phụ thuộc dữ liệu theo mục đích của bài toán. Thu gọn dữ liệu được thi hành về chiều ngang (giảm số lượng đối tượng), chiều dọc (giảm số lượng trường dữ liệu) hoặc cả hai nhằm làm cho kích thước dữ liệu được xử lý, tăng tốc độ hoạt động của hệ thống. Sử dụng các phương pháp thu gọn hoặc biến đổi chiều nhằm rút gọn số lượng các biến cần quan tâm hoặc để tìm ra các mô

tà bất biến đối với dữ liệu nhằm trình diễn dữ liệu phù hợp nhất. Do khối lượng dữ liệu trong bài toán KDD là rất lớn, nên việc thi hành bước này là rất cần thiết. Khi thu gọn theo chiều ngang cần lưu ý là tập dữ liệu được chọn lựa sau khi thu gọn phải có *tính đại diện cho tập toàn bộ dữ liệu của miền ứng dụng*. Việc chọn lựa dữ liệu vào xây dựng mô hình khai phá dữ liệu (xây dựng nhà kho dữ liệu) thông thường cần được tiến hành theo một phương pháp đảm bảo tính "ngẫu nhiên" khi chọn lựa dữ liệu trong miền ứng dụng. Tương tự, khi thu gọn theo chiều dọc cần lưu ý các thuộc tính còn lại phải đảm bảo tính đại diện cho đối tượng trong bài toán khai phá dữ liệu đang xem xét. Trong không ít bài toán khai phá dữ liệu, khi thu gọn theo chiều dọc lại nhận được kết quả tốt hơn không chỉ về thời gian và không gian, mà còn cả về chất lượng của bài toán khai phá dữ liệu khi đạt được độ chính xác cao hơn vì đã loại bỏ được một số thuộc tính gây nhiễu. Phương pháp *phân từ chính* (PCA) thường được sử dụng trong bài toán thu gọn theo chiều dọc.

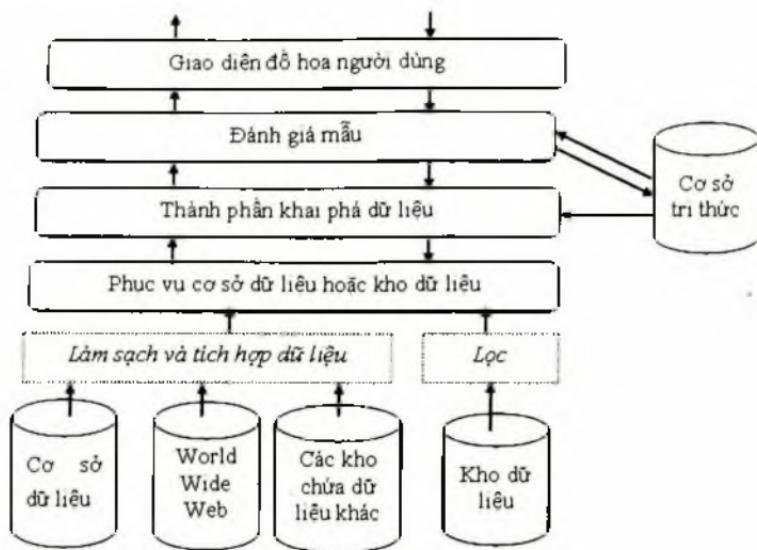
- (3) *Chọn bài toán khai phá dữ liệu*: quyết định mục tiêu của quá trình KDD là loại bài toán cụ thể nào, chẳng hạn như phân lớp, hồi quy, phân đoạn,...

Chọn lựa các phương pháp khai phá dữ liệu: lựa chọn phương pháp dùng để tìm mẫu trong dữ liệu. Nội dung này bao gồm cả việc quyết định các mô hình và tham số có thể được chấp nhận và phương pháp khai phá dữ liệu phù hợp với tiêu chuẩn tổng thể của quá trình KDD.

Thi hành thuật toán khai phá dữ liệu: tiến hành việc dò tìm các mẫu cần quan tâm dưới dạng trình bày riêng biệt, hoặc một tập các trình bày như quy tắc phân lớp, cây, hồi quy, phân đoạn,... Trong bước này, sự hỗ trợ của người dùng vẫn đóng một vai trò quan trọng.

- (4) *Giai thích mẫu* đối với các mẫu được khám phá, có thể quay về một cách hợp lý tới bất kỳ bước nào từ bước đầu tiên tới bước thi hành thuật toán khai phá dữ liệu để thực hiện lặp.
- (5) *Hợp nhất các tri thức* đã được khám phá, kết hợp các tri thức này thành một hệ thống trình diễn hoặc được biên soạn dễ dàng và kết xuất thành những thành phần hấp dẫn. Kiểm tra và giải quyết xung đột đối với tri thức được trích chọn.

Trong quá trình phát hiện tri thức trong các CSDL như mô tả ở trên có sự tham gia của các kho dữ liệu (Data Warehouse), nội dung về kho dữ liệu sẽ được giới thiệu ở phần sau.



Hình 1.6. Kiến trúc điển hình của hệ thống khai phá dữ liệu

Kiến trúc một hệ thống khai phá dữ liệu: Kiến trúc điển hình của một hệ thống khai phá dữ liệu được trình bày trong hình 1.6 [HK0106]. Trong kiến trúc hệ thống này, các nguồn dữ liệu cho các hệ thống khai phá dữ liệu bao gồm hoặc CSDL, hoặc Kho dữ liệu, hoặc World Wide Web, hoặc kho chứa dữ liệu kiểu bất kỳ khác, hoặc tổ hợp các kiểu đã liệt kê nói trên. Cơ sở tri thức chứa các tri thức miền ứng dụng hiện có, được sử dụng trong thành phần hệ thống khai phá dữ liệu để làm tăng tính hiệu quả của thành phần này. Một số tham số của thuật toán khai phá dữ liệu tương ứng sẽ được tinh chỉnh theo tri thức miền sẵn có từ cơ sở tri thức trong hệ thống. Cơ sở tri thức còn được sử dụng trong việc đánh giá các mẫu đã khai phá được, xem chúng có thực sự hấp dẫn hay không, trong đó có việc đổi chứng mẫu mới với các tri thức đã có trong cơ sở tri thức. Nếu mẫu khai phá được là thực sự hấp dẫn thì chúng được bổ sung vào cơ sở tri thức để phục vụ cho hoạt động tiếp theo của hệ thống. Như vậy, nguồn tri thức bổ sung vào cơ sở tri thức ở đây không chỉ từ lập luận logic theo các hệ toán logic để có tri thức mới, không chỉ do con người hiểu biết về thế giới khách quan để bổ sung vào, mà còn là tri thức được phát hiện một cách tự động từ nguồn dữ liệu.

1.2. Khai phá dữ liệu và xử lý cơ sở dữ liệu truyền thống

Như đã giới thiệu, khai phá dữ liệu là một thế hệ phát triển mới trong thời gian gần đây của công nghệ CSDL. Điều đó có nghĩa là, có mối quan hệ gần gũi giữa bài toán khai phá dữ liệu và bài toán xử lý (tác nghiệp) CSDL truyền thống trong mối liên quan tới một đối tượng chung là CSDL. Tuy nhiên, hai bài toán này cũng có sự phân biệt. Điểm khác biệt đầu tiên giữa khai phá dữ liệu và xử lý CSDL truyền thống là đối tượng tác động của bài toán khai phá dữ liệu phải là các CSDL, các kho dữ liệu có dung lượng rất lớn; trong khi đó bài toán tác nghiệp CSDL truyền thống liên quan tới các CSDL với mọi kích thước. Thêm nữa, những nội dung dưới đây cung cấp thêm các thông tin bổ sung về bài toán khai phá dữ liệu [KV01].

Hệ quản trị CSDL truyền thống được định hướng việc tìm kiếm tới:

– *Ghi nhận riêng lẻ*, chẳng hạn như cần tìm kiếm câu trả lời cho truy vấn "Hãy hiển thị số tiền của Ông Nguyễn Văn A có trong ngày 5 tháng Giêng năm nay". Việc tìm kiếm các ghi nhận riêng lẻ thường được chỉ dẫn là xử lý giao dịch trực tuyến (on-line transaction processing - OLTP).

– *Ghi nhận thống kê*, chẳng hạn như để trả lời câu hỏi "Có bao nhiêu nhà đầu tư nước ngoài mua cổ phiếu X trong tháng trước?". Việc tìm kiếm ghi nhận thống kê thường được chỉ dẫn là hệ thống hỗ trợ quyết định thống kê (stastical decision support system - DSS).

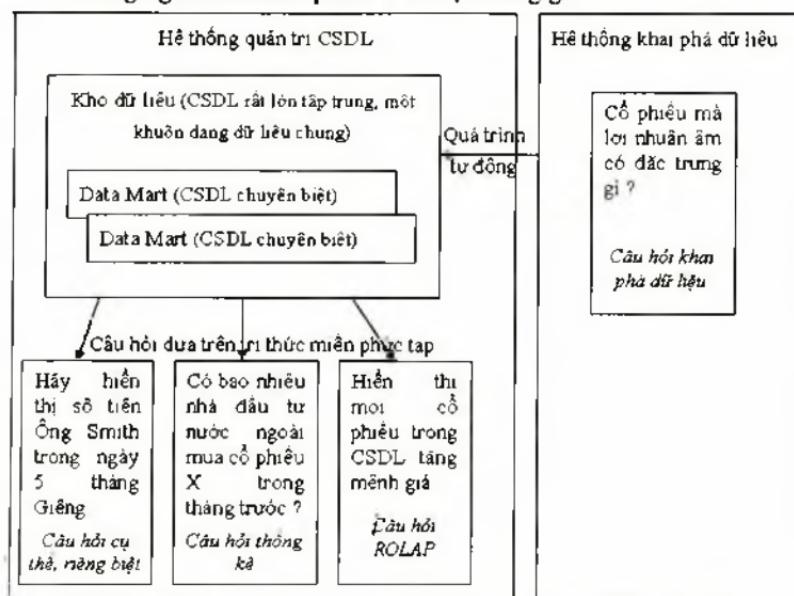
– *Ghi nhận về dữ liệu đa chiều*, chẳng hạn như để đáp ứng yêu cầu "Hiển thị mọi cổ phiếu trong CSDL với mệnh giá tăng". Việc tìm kiếm các ghi nhận dữ liệu đa chiều thường được hiểu là cung cấp, xử lý, phân tích trực tuyến (on-line analytic processing - OLAP) và xử lý phân tích trực tuyến quan hệ (relational OLAP - ROLAP).

Để các loại truy vấn (như những truy vấn nói trên) đặt ra được vẫn để cần giải quyết một cách đúng đắn, và qua đó tạo ra được các quyết định hữu ích thì cần phải công nhận đã tồn tại *một giàn thiết về tri thức miền phức hợp "đầy đủ"* (sophisticated domain knowledge) mà các loại truy vấn nói trên được đưa ra dựa trên cơ sở tri thức miền đó. Trong CSDL quan hệ, tập các phụ thuộc hàm, các luật suy diễn Armstrong là một bộ phận của tri thức miền ứng dụng nói trên. Tuy nhiên, với các CSDL lớn có dung lượng tới hàng trăm Gigabytes (GB) thì rất khó khăn để công nhận một tri thức miền phức hợp đầy đủ.

Phương pháp khai phá dữ liệu hỗ trợ việc mở rộng mục tiêu của CSDL truyền thống bằng cách cho phép tìm kiếm các câu trả lời cho các truy vấn

tuy tho sơ, song lại quan trọng, có tác dụng cải tiến miền tri thức (trong trường hợp này tri thức miền phức hợp được coi là chưa đầy đủ) như:

- Các cổ phiếu tăng giá có đặc trưng gì?
- Tỷ giá US\$ – DMark có đặc trưng gì?
- Hy vọng gì về cổ phiếu X trong tuần tiếp theo?
- Trong tháng tiếp theo, sẽ có bao nhiêu đoàn viên công đoàn không trả được nợ của họ?
- Những người mua sản phẩm Y có đặc trưng gì?



Hình 1.7. Mối quan hệ giữa hệ thống CSDL và hệ thống khai phá dữ liệu

Trả lời các truy vấn trên đường như là đã khám phá ra được các quy tắc (luật) tiềm ẩn trong dữ liệu và trên cơ sở các quy tắc đó mà đưa ra được các dự báo. Những quy tắc được khám phá là *không tuyệt đối, không mang tính "bất di bất dịch"* mà có tính chất "*đa số trường hợp là đúng*" và có thể thay đổi từ thời điểm này đến thời điểm khác. Chẳng hạn như, luật kết hợp "có đến 80% người nếu đã mua bia thì cũng mua thêm mực hoặc lạc rang" được phát hiện cho thấy tại thời điểm đang xem xét phần đông người mua bia thì cũng mua thêm mực hoặc lạc rang. Có thể đến thời điểm nào đó khác trong tương lai, khi mà thị hiếu của người uống bia có sự thay đổi, theo đó họ sẽ không mua mực hoặc lạc rang nữa thì trong CSDL giao dịch sẽ không tiềm ẩn "luật" nói trên nữa.

Mỗi quan hệ giữa hệ thống quản trị CSDL với hệ thống khai phá dữ liệu được mô tả trong Hình 1.7 [KV01]. Như vậy, trong khai phá dữ liệu thì *giai thiết đã biết về một tri thức miền phức tạp "đầy đủ" không còn là yếu tố cốt lõi*, và quá trình phát hiện tri thức có tác dụng bổ sung thêm các tri thức "mới" vào miền tri thức đó.

1.3. Một số lĩnh vực ứng dụng khai phá dữ liệu điển hình

Theo J. Han và M. Kamber [HK0106], ứng dụng của KDD được chia thành hai lớp chính, bao gồm lớp các ứng dụng phân tích dữ liệu – hỗ trợ quyết định và lớp các lĩnh vực ứng dụng khác.

Lớp các ứng dụng trong phân tích dữ liệu – hỗ trợ quyết định bao gồm các ứng dụng trong phân tích và quản lý thị trường, phân tích và quản lý rủi ro, khám phá ngoại lai và các mảng không hữu ích. Dữ liệu trong các ứng dụng này là khá phong phú, có được từ các giao dịch thẻ tín dụng, nghiên cứu đời sống cộng đồng....

Bảng 1.1. Xu thế phát triển của các lĩnh vực khai phá dữ liệu điển hình [Pia06]

| Field | Y03 | Y04 | Y06 | Growth | Trend |
|---|------|------|------|--------|-------|
| e-commerce & Web mining | 11% | 14% | 30% | 137% | up |
| Security / Anti-terrorism & Government / Military | 4.6% | 9.3% | 11% | 58% | up |
| Travel/Hospitality | 3.8% | 2.3% | 4.5% | 47% | up |
| Investment / Stocks | 6.1% | 10% | 10% | 21% | ~ |
| Fraud Detection | 18% | 22% | 22% | 8% | ~ |
| Direct Marketing/ Fundraising | 22% | 22% | 20% | -9% | ~ |
| Manufacturing | 3.8% | 10% | 6.4% | -10% | ~ |
| Other | 9.2% | 22% | 14% | -13% | ~ |
| Retail | 13% | 10% | 10% | -15% | ~ |
| Telecom | 16% | 14% | 13% | -15% | ~ |
| Biotech/Genomics | 21% | 21% | 16% | -25% | down |
| Banking / Credit Scoring | 28% | 34% | 20% | -35% | down |
| Insurance | 13% | 17% | 11% | -38% | down |
| Science | 18% | 23% | 11% | -47% | down |
| Medical/ Pharma | 12% | 17% | 7.3% | -51% | down |

Một số mục tiêu khai phá dữ liệu như tìm ra các nhóm khách hàng định hướng tiếp thị dựa trên các đặc trưng về niềm hứng thú, mức thu nhập,... cũng như phân tích thị trường chéo như tìm ra các mối liên kết, đồng quan hệ trong việc bán hàng để dự báo theo các kết hợp đó.

Một số ứng dụng điển hình nhất là phân tích hướng khách hàng theo từng loại sản phẩm để định hướng tiếp thị phù hợp, phân tích nhu cầu khách hàng, định danh loại sản phẩm thích hợp cho từng lớp khách hàng để đưa ra chiến lược kinh doanh đối với nhóm khách hàng mới, đưa ra các báo cáo tóm tắt đa chiều cũng như những thông tin tóm tắt về mặt thống kê,...

Ngoài ra, ứng dụng trong lập kế hoạch tài chính và đánh giá lưu lượng tiền tệ,... trong tài chính – ngân hàng cũng được phát triển. Trong công tác lập kế hoạch tài nguyên cũng đã xuất hiện nhiều ứng dụng của KDD. Hơn nữa, đã có nhiều cách tiếp cận khác nhau nhằm phát hiện tri thức đã được sử dụng trong các ứng dụng như vậy.

Trong nhóm phân tích dữ liệu và hỗ trợ quyết định, KDD còn được ứng dụng khá rộng rãi trong lĩnh vực bảo hiểm y tế, phục vụ thẻ tín dụng, viễn thông, thể thao, chỉnh phục vũ trụ,...

Lớp các lĩnh vực *ứng dụng điển hình khác* bao gồm khai phá Text, khai phá Web, khai phá dữ liệu dòng, khai phá dữ liệu sinh học,... Một số sản phẩm điển hình về khai phá Text và khai phá Web đã khẳng định được tính hiệu quả, chẳng hạn các sản phẩm TextAnalyst*, TextracterTM, WebAnalyst và PolyAnalyst,... của công ty Megaputer⁽¹⁾, hoặc WebFountain của IBM,...

Sự phát triển nhanh chóng của khai phá dữ liệu làm cho miền ứng dụng của lĩnh vực này ngày càng thêm phong phú và đa dạng, chẳng hạn, theo quan niệm của J. Han và M. Kamber về các khu vực ứng dụng khai phá dữ liệu đã có sự thay đổi từ phiên bản 2001 tới phiên bản 2006 [HK0106]. Trong phiên bản 2006, J. Han và M. Kamber coi rằng, các lĩnh vực điển hình của khai phá dữ liệu là phân tích dữ liệu tài chính, công nghiệp bán lẻ, công nghiệp truyền thông, phân tích dữ liệu sinh học, ứng dụng các ngành khoa học khác, sự xâm nhập sai trái,...

Còn theo Gregory Piatetsky – Shapiro [Pia06], các miền ứng dụng điển hình của khai phá dữ liệu là:

– *Ứng dụng trong khoa học* như thiên văn học, tin sinh học, y học (sáng chế các dược phẩm),...

– *Ứng dụng trong thương mại* như quản lý quan hệ khách hàng (Customer Relationship Management: CRM), phát hiện gian lận, thương

⁽¹⁾ <http://www.megaputer.com/>

mại điện tử, sản xuất, thể thao – giải trí, dịch vụ viễn thông, tiếp thị định hướng, bảo hiểm y tế,...

– Ứng dụng trong World Wide Web như máy tìm kiếm, quảng cáo trực tuyến, khai phá Web và khai phá text...

– Ứng dụng trong hoạt động chính quyền như phát hiện tội phạm, phát hiện lừa đảo thuê thu nhập cá nhân,...

Bảng 1.1 mô tả một số lĩnh vực ứng dụng khai phá dữ liệu điện hình và xu thế tăng trưởng các ứng dụng đối với từng miền trong số đó.

1.4. Kiểu dữ liệu trong khai phá dữ liệu

Về nguyên lý chung, nguồn dữ liệu được sử dụng để tiến hành khai phá dữ liệu nhằm phát hiện trí thức là rất phong phú và đa dạng, trong đó điện hình nhất là CSDL quan hệ, kho dữ liệu, CSDL giao dịch, các hệ thống dữ liệu và thông tin mở rộng khác.

• Cơ sở dữ liệu quan hệ

Thứ nhất, tính phổ biến của hệ thống CSDL quan hệ hiện nay tạo ra một hệ quả tự nhiên quy định CSDL quan hệ là một nguồn đầu vào điện hình nhất, được quan tâm trước hết của khai phá dữ liệu. *Thứ hai*, một trong những mẫu được quan tâm là mẫu về các loại "quan hệ" mà với bản chất của mình, hệ thống CSDL quan hệ tiềm ẩn các mẫu dạng như thế. Như đã biết trong lý thuyết CSDL, hệ thống CSDL quan hệ thường bao gồm một tập các bảng (hai chiều dọc và ngang). Theo chiều dọc, bảng gồm một số cột (còn được gọi là *thuộc tính*, *trường* hay *đặc trưng*) và theo chiều ngang bảng chứa một tập rất lớn các dòng (còn được gọi *ban ghi* hay *bộ*). Số lượng cột của bảng còn được gọi là *số chiều*. Hệ thống CSDL quan hệ còn bao gồm một mô hình ngữ nghĩa mà thông thường là mô hình thực thể – quan hệ.

• Kho dữ liệu

Theo J. Han và M. Kamber, tồn tại nhiều cách hiểu về kho dữ liệu, nhưng cách hiểu phổ dụng nhất là theo định nghĩa của W.H. Inmon (một chuyên gia hàng đầu về kho dữ liệu). Theo W.H. Inmon [Inm02], "kho dữ liệu là tập hợp các dữ liệu định hướng theo chủ đề, được tích hợp lại, có tính phiên bản theo thời gian và kiên định được dùng để hỗ trợ việc tạo quyết định quản lý". Tên gọi của bốn thuộc tính "định hướng theo chủ đề", "được tích hợp lại", "có tính phiên bản theo thời gian" và "kiên định" ở trên của kho dữ liệu mới chỉ cung cấp một số nét cơ bản nhất về các đặc trưng của kho dữ liệu. W.H. Inmon (cũng như J. Han và M. Kamber) đã giải thích nội dung chi tiết về bốn thuộc tính này.

Kho dữ liệu là một kết quả xuất hiện trong quá trình tiền hoá các hệ hỗ trợ quyết định. Thuật ngữ "tạo kho dữ liệu" (Data warehousing) được dùng để chỉ quá trình xây dựng và sử dụng kho dữ liệu. Như vậy, quá trình phát hiện tri thức trong CSDL tiếp nhận đầu vào là các hệ thống CSDL, các nhà kho tổ chức dữ liệu từ các nguồn và các dữ liệu mô tả. Cần chú ý rằng, để đáp ứng bốn thuộc tính trên, kho dữ liệu được coi chỉ bao gồm các dữ liệu được coi là "có chất lượng" thông qua các khâu chọn lựa, tiền xử lý và có thể bao gồm cả khâu chuyên dạng trong quá trình phát hiện tri thức trong CSDL (Hình 1.4).

Các nghiên cứu và triển khai liên quan tới kho dữ liệu chỉ dẫn khuynh hướng hiện tại của các hệ thống thông tin quản lý (MIS: Management Information Systems) phổ biến là nhằm vào việc thu thập, làm sạch dữ liệu giao dịch và tạo cho chúng độ linh hoạt khi tìm kiếm trực tuyến. Một tiệm cận phổ biến đối với phân tích kho dữ liệu gọi là OLAP (On-Line Analytical Processing), thông qua một tập các nguyên lý được Codd đề xuất vào năm 1993. Các bộ công cụ OLAP chủ trọng tới việc cung cấp tới SQL các tiện ích phân tích dữ liệu đa chiều chất lượng cao bằng cách tính toán gian lược và phân tách nhiều chiều. Cả phát hiện tri thức lẫn OLAP được coi là hai khía cạnh quan hệ mật thiết nhau, được tích hợp trong một thế hệ mới các bộ công cụ trích lọc và quản lý thông tin.

Đồng thời với sự phát triển của công nghệ kho dữ liệu, các hệ thống tích hợp các nguồn dữ liệu cá dữ liệu trong quá khứ lẫn dữ liệu tác nghiệp đã được xây dựng. Nhiều hệ thống khai phá dữ liệu có đầu vào từ siêu dữ liệu (metadata) cùng các dữ liệu nguồn trong các kho dữ liệu.

• Cơ sở dữ liệu giao dịch

Một lớp bài toán khai phá dữ liệu phổ biến là khai phá quan hệ kết hợp, trong đó điển hình là bài toán khai phá luật kết hợp, được xuất phát từ việc xem xét các CSDL giao dịch (bán hàng). Dữ liệu giao dịch chính là dữ liệu nguyên thủy xuất hiện trong định nghĩa về luật kết hợp cùng với các độ đo của luật như độ hỗ trợ và độ tin cậy. Khi mở rộng dữ liệu từ dữ liệu giao dịch sang dữ liệu vô hướng, hoặc dữ liệu phức tạp hơn có trong các CSDL quan hệ, các giải pháp khai phá luật kết hợp được cải tiến để thích ứng với sự biến đổi này (bao gồm bước chuyên dạng dữ liệu trong quá trình phát hiện tri thức từ các CSDL). Các giải pháp ứng dụng lý thuyết tập mờ (chẳng hạn, [JEM03, IIP03, STH06]) và lý thuyết tập thô (chẳng hạn, [Zia94, SH98, Ale99, SZ00, Li07]) tương ứng với việc mở rộng miền dữ liệu cần khai phá đã được tiến hành trong nhiều công trình nghiên cứu.

• Các hệ thống dữ liệu mở rộng

Trong quá trình phát triển, các phương pháp và thuật toán khai phá dữ liệu thích hợp đối với các CSDL mở rộng và các kiểu kho chứa dữ liệu được

dễ xuất. Các phương pháp và thuật toán này phù hợp với dữ liệu trong CSDL hướng đối tượng, CSDL không gian – thời gian, CSDL tạm thời, dữ liệu chuỗi thời gian (bao gồm dữ liệu tài chính), dữ liệu dòng, CSDL Text và CSDL đa phương tiện, CSDL hỗn tạp và CSDL thừa kế, và World Wide Web.

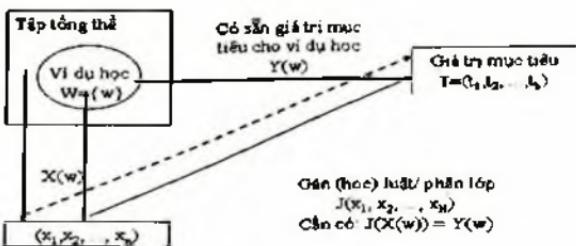
Hệ thống CSDL quan hệ – đối tượng có thể được coi là sự bổ sung theo tiếp cận hướng đối tượng tới các hệ thống CSDL quan hệ. Mô hình dữ liệu quan hệ – đối tượng mô tả ngữ nghĩa của hệ thống CSDL quan hệ – đối tượng, được phát triển từ mô hình quan hệ với việc bổ sung các kiểu dữ liệu giàu ngữ nghĩa. Thực thể từ mô hình quan hệ thực thể được phát triển thành đối tượng trong mô hình quan hệ đối tượng.

1.5. Các bài toán khai phá dữ liệu diễn hình

Khai phá dữ liệu là lĩnh vực nghiên cứu mang tính thực tiễn cao, đồng thời lại đòi hỏi một nền tảng toán học mạnh trong việc xây dựng các mô hình toán học phù hợp nhất cho miền dữ liệu của bài toán đang được quan tâm. Bước khai phá dữ liệu trong quá trình KDD thường áp dụng một phương pháp khai phá dữ liệu cụ thể, liên quan đến các khái niệm mẫu và mô hình. Như đã được giới thiệu trong mục 1.1, mẫu là một biểu thức trong một ngôn ngữ mô tả L nào đó được chọn. Mô hình được coi là một biểu thức tổng quát trong ngôn ngữ mô tả L nói trên; tính tổng quát của mô hình được thể hiện thông qua các tham số mô hình, trong trường hợp đó, một mẫu là một thể hiện của mô hình. Chẳng hạn, biểu thức $\alpha x^2 + \beta x$ (với hai tham số α và β) là mô hình, còn $3x^2 + x$ là một mẫu trong mô hình đó (đối với mẫu này thì các tham số mô hình α và β đã được cho giá trị cụ thể là $\alpha = 3$ và $\beta = 1$).

Nhiệm vụ của bài toán khai phá dữ liệu từ một tập dữ liệu quan sát (tập các sự kiện) đã có, thì hoặc cần phải xác định mô hình phù hợp với tập dữ liệu quan sát đó, hoặc cần tìm ra các mẫu từ tập dữ liệu đó.

Bài toán khai phá dữ liệu thường hướng tới một trong hai loại mô hình đó là *mô hình theo tiếp cận thống kê* (mô hình thống kê) hoặc *mô hình lôgic*. Mô hình thống kê được định hướng tới loại mô hình bao hàm các yếu tố chưa xác định, chẳng hạn như mô hình $ax + e$, trong mô hình này thì x là biến trong ngôn ngữ mô tả L , còn e có thể là biến ngẫu nhiên Gauss (thể hiện tính chưa xác định của mô hình). Ngược lại, mô hình lôgic định hướng tới loại mô hình xác định hoàn toàn, chẳng hạn ax , trong đó không thừa nhận yếu tố không rõ ràng khi mô hình hoá. Mô hình thống kê được dùng hầu khắp đối với các ứng dụng khai phá dữ liệu thực tế.



Hình 1.8. Sơ đồ biểu diễn mô hình học máy: cần học là đường nét rờ
 (Lưu ý, học máy không giám sát (phân cụm) không có giá trị mục tiêu cho ví dụ học
 (không có hai đường连线 hướng tới giá trị mục tiêu))

Hầu hết các phương pháp khai phá dữ liệu đã được xây dựng có nội dung từ các phương pháp học máy, thiết kế mẫu và thống kê (phân lớp, phân đoạn, mô hình đồ thị...). Thuật toán giải quyết mỗi bài toán nói trên cuốn hút một phạm vi người quan tâm đa dạng, bao gồm cả các chuyên gia phân tích dữ liệu lẫn những người chưa hề có kinh nghiệm.

Ở mức cao – tổng quát, hai mục tiêu chủ yếu của khai phá dữ liệu là dự báo và mô tả, mà chúng ta coi hai mục tiêu này tương ứng với hai bài toán tổng quát của khai phá dữ liệu. Bài toán dự báo sử dụng một số biến (hoặc trường) trong CSDL để dự đoán về hoặc giá trị chưa biết (đã có), hoặc giá trị sẽ có trong tương lai của các biến. Bài toán mô tả hướng tới việc tìm ra các mẫu mô tả dữ liệu. Dự đoán và mô tả có tầm quan trọng khác nhau đối với các thuật toán khai phá dữ liệu riêng. Trong ngữ cảnh KDD, vẫn đề mô tả có khuynh hướng quan trọng hơn vẫn đề dự báo, điều này là trái ngược với nội dung chủ yếu của các ứng dụng nhận dạng mẫu và học máy, thì vẫn đề dự báo là quan trọng hơn. Điều có vẻ trái ngược đó có thể được giải thích khi xem xét, phân tích nội dung của chính khái niệm "phát hiện tri thức trong CSDL"; khái niệm này đã bao hàm tính huống sẵn có dữ liệu để phát hiện các mẫu tiềm ẩn trong dữ liệu đó, các mẫu tiềm ẩn đó liên quan tới bài toán mô tả dữ liệu. Mặt khác, mô tả được mô hình dữ liệu thì cũng rất thuận tiện cho dự báo.

Ở mức chi tiết – cụ thể, dự báo và mô tả được thể hiện thông qua các bài toán cụ thể như mô tả khái niệm, quan hệ kết hợp, phân cụm, phân lớp, hồi quy, mô hình phụ thuộc, phát hiện biến đổi và độ lệch, và một số bài toán cụ thể khác như trình bày dưới đây.

• Mô tả khái niệm

Nội dung của bài toán mô tả khái niệm là tìm ra các đặc trưng và tính chất của khái niệm (dùng để "mô tả" khái niệm đó). Diễn hình nhất trong

lớp bài toán này là các bài toán như tóm quát hoá, tóm tắt, phát hiện các đặc trưng dữ liệu ràng buộc,... Bài toán tóm tắt là một bài toán mô tả diễn hình, áp dụng các phương pháp để tìm ra một mô tả có dạng đối với một tập con dữ liệu. Một ví dụ diễn hình về bài toán tóm tắt là bài toán *tính kí vọng và độ lệch chuẩn* của một tập dữ liệu trong thống kê xác suất; hai giá trị này chính là hai đặc trưng diễn hình nhất về một hiện tượng có dãy giá trị thể hiện mà chúng ta đã quan sát được.

Nhiều phương pháp đã được biện luận đối với việc thu nhận được các quy tắc tóm tắt, kỹ thuật hiện thị đa biến, phát hiện quan hệ hàm giữa các biến. Kỹ thuật tóm tắt thường được áp dụng trong phân tích dữ liệu thăm dò có tương quan và tự động hoá sinh ra các thông báo.

Trong khai phá Text và khai phá Web, tóm tắt văn bản là một biểu hiện cụ thể của tóm tắt, theo đó từ một văn bản đã có, cần tìm ra văn bản ngắn gọn (với độ dài 100 từ, 200 từ hoặc 500 từ) mà vẫn giữ được ngữ nghĩa cơ bản của văn bản gốc.

• *Quan hệ kết hợp*

Phát hiện mối quan hệ kết hợp trong tập dữ liệu là một bài toán quan trọng trong khai phá dữ liệu. Một trong những mối quan hệ kết hợp diễn hình là quan hệ kết hợp giữa các biến dữ liệu, trong đó bài toán khai phá luật kết hợp là một bài toán diễn hình. Bài toán khai phá luật kết hợp (thuộc lớp phát hiện quan hệ kết hợp) thực hiện việc phát hiện ra mối quan hệ giữa các tập thuộc tính (các tập biến) có dạng $X \rightarrow Y$, trong đó X, Y là hai tập thuộc tính. Về hình thức, luật kết hợp có dạng giống như phụ thuộc hàm trong CSDL quan hệ, tuy nhiên, nó không được định sẵn từ tri thức miền.

Trong khai phá text và khai phá Web tồn tại nhiều bài toán phát hiện quan hệ kết hợp, diễn hình như bài toán phát hiện quan hệ ngữ nghĩa (chẳng hạn như quan hệ nhân – quả, quan hệ toàn bộ – bộ phận, quan hệ chung – riêng,...) trong văn bản (hoặc trong tập văn bản), bài toán phát hiện mối quan hệ giữa nội dung trang Web người sử dụng đang quan tâm tới các trang Web mà họ có thể sẽ hướng tới,...

• *Phân lớp*

Phân lớp (Classification/Categorization) thực hiện việc xây dựng (mô tả) các mô hình (hàm) dự báo, nhằm mô tả hoặc phát hiện các lớp hoặc khái niệm cho các dự báo tiếp theo. Một số phương pháp diễn hình là cây quyết định, luật phân lớp, mạng neuron. Nội dung của phân lớp chính là một hàm ảnh xạ các dữ liệu vào một trong một số lớp đã biết. Ví dụ, phân lớp một văn bản (bao gồm cả trang Web) vào một trong một số lớp văn bản (trang Web) đã biết, phân lớp khuynh hướng trong thị trường tài chính, phát hiện tự động các đối tượng đáng quan tâm trong CSDL ảnh lớn,... Hình 1.8 mô

tả sơ bộ về bài toán phân lớp (thường được tương ứng với học có giám sát), theo đó đường ngang liền nét cho biết đã biết thuộc tính lớp đối với một tập hợp dữ liệu nào đó (tập dữ liệu học). Nội dung chi tiết hơn về bài toán phân lớp sẽ được trình bày chi tiết hơn trong Chương 7 – *Phân lớp văn bản*.

• *Phân cụm*

Phân cụm (Clustering) thực hiện việc nhóm dữ liệu thành các “cụm” (có thể coi là các lớp mới) để có thể phát hiện được các mẫu phân bố dữ liệu trong miền ứng dụng. Phân cụm là một bài toán mô tả hướng tới việc nhận biết một tập hữu hạn các cụm hoặc các lớp để mô tả dữ liệu. Các cụm (lớp) có thể tách rời nhau và toàn phần (tạo nên một phân hoạch cho tập dữ liệu), hoặc được trình bày đẹp hơn như phân lớp có thứ bậc hoặc có thể chồng lên nhau (giao nhau). Ví dụ, bài toán phát hiện các nhóm người tiêu dùng trong CSDL tiếp thị, hoặc nhận biết các loại quang phổ trong tập phép đo không gian hồng ngoại,... Thông thường, mục tiêu định hướng của bài toán phân cụm là cực đại tính tương đồng giữa các phần tử trong mỗi cụm và cực tiểu tính tương đồng giữa các phần tử thuộc các cụm khác nhau.

Trong nhiều trường hợp, phân cụm còn được gọi là *học máy không giám sát* (unsupervised learning) và phân lớp còn được gọi là *học máy có giám sát* (supervised learning). Mô hình học máy (có giám sát và không giám sát) được trình bày trong Hình 1.8 [KV01].

Trong một số ứng dụng, bài toán phân đoạn (segmentation) cần được giải quyết. Về bản chất, phân đoạn là tổ hợp của phân cụm và phân lớp, trong đó phân cụm được tiến hành trước và sau đó là phân lớp. Chương 6 – *Phân cụm văn bản* sẽ mô tả chi tiết hơn về bài toán phân cụm.

• *Hồi quy*

Hồi quy là một bài toán điển hình trong phân tích thống kê và dự báo, trong đó tiến hành việc dự đoán các giá trị của một hoặc một số biến phụ thuộc vào giá trị của một tập hợp các biến độc lập [HD03]. Mô hình hồi quy là khá thông dụng trong dự báo dài hạn. Trong khai phá dữ liệu, bài toán hồi quy được quy về việc học một hàm ánh xạ dữ liệu nhằm xác định giá trị thực của một biến theo một số biến khác. Tình huống ứng dụng hồi quy rất đa dạng, chẳng hạn như dự đoán số lượng sinh vật phát quang trong khu rừng nhờ đo vi sóng các sensor từ xa, hoặc ước lượng xác suất người bệnh có thể chết theo kết quả test triệu chứng, hoặc dự báo nhu cầu người tiêu dùng đối với một sản phẩm mới được coi như một hàm của quảng cáo tiêu dùng, hoặc dự báo chuỗi thời gian mà các biến đầu vào được coi như bản trễ thời gian của biến dự báo,...

• Mô hình phụ thuộc

Bài toán xây dựng mô hình phụ thuộc hướng tới việc tìm ra một mô hình mô tả sự phụ thuộc có ý nghĩa giữa các biến. Mô hình phụ thuộc gồm hai mức: mức cấu trúc của mô hình mô tả (thường dưới dạng đồ thị), trong đó các biến là phụ thuộc bộ phận vào các biến khác; trong khi mức định lượng của mô hình mô tả sức mạnh của tính phụ thuộc khi sử dụng việc do tính theo giá trị số. Ví dụ, lưới phụ thuộc xác suất cần đảm bảo tính độc lập điều kiện nhằm định rõ diện mạo cấu trúc của mô hình và xác suất, hoặc tương quan để mô tả sức mạnh của tính phụ thuộc. Phân tích khuynh hướng và tiên hoà cũng được coi thuộc vào loại khai phá mô hình phụ thuộc. Trong phân tích khuynh hướng và tiên hoà, các phương pháp phân tích xu thế, khai phá mẫu kế tiếp, phân tích dựa trên tính tương tự,... thường được áp dụng.

• Phát hiện biến đổi và độ lệch

Tập trung vào việc phát hiện hầu hết sự thay đổi có ý nghĩa dưới dạng độ do đã biết trước hoặc giá trị chuẩn, cung cấp những tri thức về sự biến đổi và độ lệch cho người dùng. Bài toán phát hiện biến đổi và độ lệch còn được ứng dụng trong bước tiền xử lý trong quá trình phát hiện tri thức trong CSDL. Chính vì lý do đó, cần tránh suy nghĩ cho rằng, sự biến đổi và độ lệch mang ý nghĩa "không chính quy" mà phải quan niệm sự biến đổi và độ lệch đó (có thể là bất thường) là một nội dung băn chất của dữ liệu.

Ngoài ra có thể kể tới bài toán phân tích định hướng mẫu và một số bài toán khai phá dữ liệu kiêu thống kê khác.

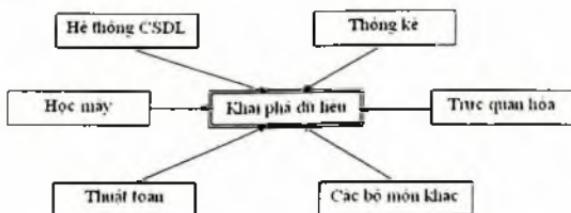
1.6. Tinh liên ngành của khai phá dữ liệu

KDD nhận được sự quan tâm đặc biệt của các nhà nghiên cứu trong các lĩnh vực học máy, thu nhận mẫu, CSDL, thống kê, trí tuệ nhân tạo, thu nhận tri thức đối với hệ chuyên gia được trình bày trong Hình 1.9 [HK0106]. Hệ thống KDD lôi cuốn các phương pháp, thuật toán và kỹ thuật từ các lĩnh vực rời rạc nhau này. Mục tiêu thống nhất là trích lọc tri thức từ dữ liệu trong ngữ cảnh các CSDL lớn.

Một số lập luận trong phần trước [HK0106] đã chỉ dẫn rằng, khai phá dữ liệu là bước phát triển mới của công nghệ CSDL, vì vậy nhiều nội dung trong khai phá dữ liệu là gần gũi với CSDL. Đồng thời, một số đặc điểm phân biệt giữa hệ thống CSDL truyền thống với hệ thống khai phá dữ liệu cũng đã được thảo luận, trong đó dấu hiệu phân biệt điển hình nhất giữa nội dung nghiên cứu là quan niệm về một già thiết sẵn có một tri thức miễn ứng dụng đầy đủ.

Tài nguyên dữ liệu đầu vào cho các hệ thống khai phá dữ liệu gồm có các CSDL, các kho dữ liệu và các loại nguồn chứa dữ liệu khác. Chính vì lý

do đó, trong không ít trường hợp, lĩnh vực *kho dữ liệu* (data warehouse) được coi là một bộ phận của lĩnh vực khai phá dữ liệu và phát hiện tri thức trong CSDL.



Hình 1.9. Tính đa/liên ngành của khai phá dữ liệu

Như đã được trình bày, quá trình phát hiện tri thức làm việc với tập hợp dữ liệu lớn mà trong nhiều trường hợp tập dữ liệu trở nên khổng lồ. Phạm vi tác động to lớn và đa dạng đòi hỏi các thuật toán khai phá dữ liệu phải dùng dồn và hiệu quả; chính vì điều đó cho nên, rất nhiều thuật toán khai phá dữ liệu đã được đề xuất. Xindong Wu và cộng sự [WQK08] cung cấp một danh sách gồm mươi thuật toán khai phá dữ liệu nổi tiếng nhất, đó là các thuật toán C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naive Bayes và CART. Một số nội dung cơ bản nhất về các thuật toán này được giới thiệu trong các phần nội dung liên quan trong tài liệu này.

Đối với các lĩnh vực *học máy* và *thu nhận mẫu*, sự đan xen với khai phá dữ liệu (và KDD) trải theo các nghiên cứu về lý thuyết và thuật toán đối với các hệ thống trích lọc mẫu và mô hình dữ liệu (chú yếu đối với các phương pháp khai phá dữ liệu). Các phương pháp học máy giám sát (phân lớp), không giám sát (phân cụm), bán giám sát (phân lớp và phân cụm) đã rất phổ biến trong khai phá dữ liệu, nhằm lựa chọn mô hình và xác định tham số mô hình trong các hệ thống KDD. Trọng tâm của KDD đối với việc mở rộng các lý thuyết và thuật toán học máy hướng tới bài toán tìm ra các mẫu đặc biệt (những mẫu mà trong một số ngữ cảnh còn được gọi là *tri thức hữu dụng* hoặc *hấp dẫn*) trong các tập hợp dữ liệu có dung lượng lớn của thế giới thực. Như vậy, khai phá dữ liệu mở rộng nội dung học máy thông qua các công việc lựa chọn dữ liệu đầu vào, trình diễn mẫu, đánh giá mẫu đầu ra,... trong ngữ cảnh miền dữ liệu cần xử lý có dung lượng rất lớn.

KDD cũng có rất nhiều điểm chung với chuyên ngành *thống kê*, đặc biệt là phân tích dữ liệu thăm dò (EDA: Exploratory Data Analysis) cũng như dự báo [HD03]. Hệ thống KDD thường gắn kết với các thủ tục thống kê đặc biệt đối với mô hình dữ liệu và nằm bắt nhiều trong một khung cảnh phát hiện tri thức tổng thể. Các phương pháp khai phá dữ liệu dựa theo

thống kê nhận được sự quan tâm đặc biệt. Tuy nhiên, cần phân biệt giữa bài toán thống kê và bài toán khai phá dữ liệu. Chẳng hạn, trong bài toán kiểm định giả thiết thống kê [HHN04], cho trước một giả thiết thống kê và công việc cần tiến hành là kiểm tra xem tập hợp toàn bộ các dữ liệu quan sát được có phù hợp với giả thiết thống kê nói trên hay không, hay cũng vậy, giả thiết thống kê có đúng trên toàn bộ dữ liệu quan sát được hay không. Nếu kiểm định cho kết quả không phù hợp, có nghĩa là giả thiết thống kê là không đúng trên tập dữ liệu quan sát. Như vậy, tính đúng đắn của giả thiết thống kê được xem xét trên tập toàn bộ các dữ liệu quan sát đã có. Trong trường hợp bài toán học khai phá dữ liệu, mô hình kết quả khai phá dữ liệu là không được xác định trước. Mô hình kết quả cần phải phù hợp với tập toàn bộ dữ liệu của miền ứng dụng mà không phải chỉ với tập dữ liệu quan sát được (tập dữ liệu quan sát được chỉ là một bộ phận mà thường là rất nhỏ so với miền dữ liệu của thế giới thực, xem Hình 1.8), do đó cần đảm bảo các tham số mô hình không phụ thuộc vào cách chọn tập dữ liệu học. Chính vì lý do cốt lõi này mà bài toán học khai phá dữ liệu đòi hỏi đáp ứng yêu cầu là tập dữ liệu cần có tính "đại diện" cho toàn bộ dữ liệu trong miền ứng dụng và tập dữ liệu kiểm tra cần phải được chọn một cách độc lập với tập dữ liệu học. Một số dấu hiệu phân biệt khác về mặt thuật ngữ cũng được lưu ý, chẳng hạn khai phá dữ liệu dùng các thuật ngữ *biến ra/biến mục tiêu, thuật toán khai phá dữ liệu, thuộc tính/đặc trưng, bản ghi*,... trong khi đó xử lý thống kê dùng các thuật ngữ tương ứng là *biến phụ thuộc, thủ tục thống kê, biến giải thích, quan sát*,...

Như đã được khẳng định tại các phần trên là, không phải tất cả các mẫu đều hữu dụng và hệ thống cần đưa ra các tiêu chí để lọc các mẫu được coi là hấp dẫn nhất. Thông thường các hệ thống sử dụng một ngưỡng hấp dẫn cực tiêu cho các mẫu được coi là tri thức. Chẳng hạn, trong bài toán phát hiện luật kết hợp, người ta chỉ giữ lại các luật vượt qua ngưỡng độ hỗ trợ tối thiểu và độ tin cậy tối thiểu. Ngay cả trong trường hợp đó, không phải mọi "tri thức" được hệ thống coi là "hữu dụng" đều hoàn toàn phù hợp với người sử dụng. Bước *trực quan hóa* trong quá trình KDD hiển thị các tri thức được hệ thống phát hiện một cách trực quan nhất để tạo thuận lợi cho người sử dụng (through qua tri thức và kinh nghiệm) lựa chọn ra các tri thức thực sự hữu dụng cho mục đích ứng dụng của người sử dụng.

Phát hiện máy với mục tiêu là phát hiện các luật kinh nghiệm từ quan sát và thử nghiệm: *mô hình nhận qua* phát hiện các kết luận của mô hình nhận qua từ dữ liệu là những lĩnh vực nghiên cứu có mối liên hệ với nhau.

1.7. Khuynh hướng phát triển của khai phá dữ liệu

Như đã được giới thiệu, không những trở thành một lĩnh vực khoa học – công nghệ thời sự, mà khai phá dữ liệu vẫn đang được phát triển rất mạnh mẽ. Thuật ngữ *khai phá dữ liệu* cũng như lĩnh vực khai phá dữ liệu đã trở nên nổi bật, và vì vậy, thuật ngữ *data mining* và thuật ngữ *machine learning* (một thuật ngữ có quan hệ mật thiết với khai phá dữ liệu) đã được ghi nhận vào danh sách top 20 thuật ngữ khoa học hàng đầu do trang Web ResearcherID⁽¹⁾ liệt kê. Hiệp hội các nhà khoa học về Phát hiện tri thức và Khai phá dữ liệu (The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining, viết tắt là SIGKDD) được thành lập và hoạt động. Ban điều hành của SIGKDD gồm một số nhà khoa học hàng đầu thế giới về lĩnh vực này, do Piatetsky – Shapiro⁽²⁾ chủ trì. Từ năm 1995, hoạt động diễn hình nhất của SIGKDD là tổ chức Hội nghị khoa học quốc tế thường niên ACM SIGKDD Conference on Knowledge Discovery and Data Mining.

Khuynh hướng phát triển của khai phá dữ liệu còn quan hệ mật thiết với khuynh hướng phát triển của khoa học máy tính.

• Khuynh hướng phát triển của khoa học máy tính

Trong [Hop07], John E. Hopcroft trình bày về khuynh hướng phát triển của khoa học máy tính. Ông đề cập tới một số yếu tố nổi bật sau đây của xã hội điện tử (e-society) trong tương lai tác động tới sự chuyển biến của khoa học máy tính:

- Tính sẵn sàng máy tính cả theo không gian và cả theo thời gian;
- Tính đáp ứng tốc độ xử lý đối với mọi nghiệp vụ văn phòng (soạn thảo văn bản, email, chat, bảng tính);
- Tính tích hợp máy tính và truyền thông;
- Tính sẵn sàng dữ liệu dạng số hoá;
- Tính kết nối mạng của mọi thiết bị.

Trong nghiên cứu của mình, J. E. Hopcroft sử dụng kết quả nghiên cứu về khai phá dữ liệu văn bản của Rich Caruana cùng cộng sự [CJG06]. Bài toán mà Rich Caruana và cộng sự giải quyết (được giới thiệu chi tiết hơn tại Chương 2) được mô tả sơ bộ như sau: Cho trước một tập hợp (khoảng 300000) tài liệu khoa học (công trình nghiên cứu) cần phát hiện ra các chủ đề khoa học chủ chốt và qua đó dự báo được xu hướng nghiên cứu, phát triển các chủ đề khoa học mới thuộc lĩnh vực khoa học máy tính. Giải pháp

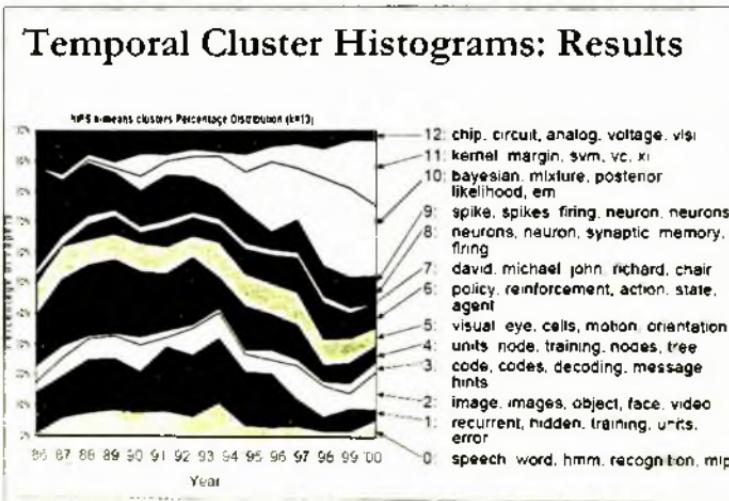
⁽¹⁾ www.researcherid.com

⁽²⁾ <http://www.kdnuggets.com/gps.html>

tiến hành không cần khai thác các chi dẫn của các công trình, nghĩa là chỉ sử dụng nội dung các công trình. Hình 1.10 mô tả một kết quả nghiên cứu của Rich Caruana và cộng sự, theo đó phát hiện ra 13 cụm chủ đề và cung cấp ý tưởng về xu hướng phát triển của 13 cụm chủ đề nói trên. Từ kết quả nghiên cứu nói trên của Rich Caruana cùng cộng sự và một số công trình liên quan khác, J. E. Hopcroft giới thiệu một số nội dung lý thuyết cần được quan tâm để làm nền tảng khoa học giải quyết các bài toán thi hành xã hội điện tử như sau:

– Lý thuyết, mô hình và giải pháp tìm kiếm. *Thứ nhất*, câu hỏi tìm kiếm đã có sự thay đổi về chất từ câu hỏi mang tính cụ thể, thông kê sang câu hỏi mang tính tự vấn và đòi hỏi sự phân tích phức hợp như: "Với tôi, mua ô tô loại nào là thích hợp?", "Hãy xây dựng một lịch sử có chủ giải về lý thuyết đô thị", "Tôi nên vào trường đại học nào?", "Các lĩnh vực của khoa học máy tính đã phát triển như thế nào?..." *Thứ hai*, không gian tìm kiếm là rộng lớn và câu hỏi được đặt ra mọi lúc, mọi nơi.

– Mạng và cảm biến. Trong một môi trường có tính sẵn sàng theo không gian và thời gian, hoạt động có tính ngẫu nhiên, giao tiếp với môi trường thông qua các cảm biến và kết nối mạng các mức thành phần (mức cảm biến, mức mạng các mạng con, mức các thành phần lớn và cực lớn...) cần được mô hình hóa với các giải pháp tích hợp hiệu quả.



Hình 1.10. Tình hình phát triển một số nhóm chủ đề trong khoa học máy tính qua phân cụm tài liệu khoa học [CJG06]

– Xử lý dữ liệu nhiều chiều đồ sộ và chứa nhiều nhiễu. Tính đồ sộ của dữ liệu nằm trong xu thế bùng nổ thông tin như đã biết. Dữ liệu cần có nhiều chiều để biểu diễn sát thực hơn về thực tại. Tính ngẫu nhiên cùng với tính phức tạp của hệ thống dẫn đến việc dữ liệu có thể có chứa nhiều nhiễu.

– Mô hình và giải pháp tích hợp hệ thống và tài nguyên dữ liệu. Dù sử dụng phương pháp xây dựng hệ thống nào (chức năng, đối tượng, kết hợp...) thì cách tiếp cận dựa trên thành phần đã trở thành cách tiếp cận chung, rất hữu hiệu, đặc biệt là đối với các hệ thống lớn.

Một trong những mô hình toán học điển hình nhất liên quan tới các nội dung lý thuyết nêu trên là đồ thị lớn. Một ví dụ đơn giản là đồ thị Web được đề cập trong các máy tìm kiếm hiện nay đã có số đỉnh lên tới hàng tỷ nút. Tính sẵn sàng, mọi lúc, mọi nơi đòi hỏi mô hình hệ thống được thiết lập dưới dạng đồ thị sẽ có số nút rất lớn. Hơn nữa, các đồ thị lớn này cần là các đồ thị ngẫu nhiên. Lời giải cho các đồ thị lớn hiện nay được sự quan tâm đặc biệt.

• *Khuynh hướng phát triển của khai phá dữ liệu*

Trang Web <http://www.kdnuggets.com/> do Piatetsky – Shapiro chủ trì là một trong những trang Web điển hình về lĩnh vực khai phá dữ liệu và phát hiện tri thức trong CSDL. Nhiều thông tin cập nhật nhất về lĩnh vực được thông báo tại trang Web này, đặc biệt là các kết quả thăm dò, cung cấp một số thông tin hữu ích liên quan tới khuynh hướng phát triển của lĩnh vực khai phá dữ liệu. Một số nội dung cụ thể về khuynh hướng nghiên cứu của khai phá dữ liệu được đề cập dưới dạng bài toán thách thức trong các hội nghị khoa học về khai phá dữ liệu, chẳng hạn như [ASG06, Son07].

Theo J. Han và M. Kamber [HK0106], xu hướng phát triển khai phá dữ liệu đã và đang là các nội dung nghiên cứu có tính thời sự, rất đa dạng và phong phú. Việc phát triển các phương pháp và hệ thống khai phá dữ liệu dù sức mạnh và hiệu quả, xây dựng các môi trường khai phá dữ liệu tương tác và tích hợp, thiết kế các ngôn ngữ khai phá dữ liệu, áp dụng các kỹ thuật khai phá dữ liệu để giải quyết các bài toán ứng dụng lớn là những bài toán quan trọng trong nghiên cứu và triển khai về khai phá dữ liệu. Trong [YW06], Qiang Yang và Xindong Wu giới thiệu về 10 bài toán thách thức trong lĩnh vực khai phá dữ liệu, đã và đang cuốn hút các xu hướng nghiên cứu và triển khai đối với lĩnh vực này.

Từ các nghiên cứu tổng hợp của J. Han, M. Kamber và Qiang Yang – Xindong Wu, chúng ta có thể thấy một số xu hướng phát triển nghiên cứu và triển khai điển hình nhất về khai phá dữ liệu như sau:

– Phát triển một lý thuyết thống nhất về khai phá dữ liệu. Như đã được trình bày, lĩnh vực khai phá dữ liệu được ứng dụng rộng rãi, nhận được sự

quan tâm của đông đảo các nhà khoa học thuộc các lĩnh vực nghiên cứu rất đa dạng, vì vậy trình độ phát triển hiện thời của mỗi một nghiên cứu về khai phá dữ liệu lại mang tính quá đặc thù. Rất nhiều kỹ thuật được thiết kế cho các bài toán riêng lẻ, chẳng hạn như phân lớp hoặc phân cụm, mà không có một cơ sở lý thuyết thống nhất.

Dù chưa hình thành được một khung lý thuyết thống nhất, song nhiều nghiên cứu về khai phá dữ liệu dựa trên lý thuyết xác suất, lý thuyết đồ thị, lý thuyết tối ưu... đã được tiến hành và thông qua đó, một số thành phần về khung lý thuyết chung đã dần được hình thành. Theo cách tiếp cận mô hình Internet và Web dựa trên lý thuyết xác suất, Pierre Baldi và cộng sự đã hệ thống hoá các phương pháp và thuật toán khai phá dữ liệu Web [BFS03]. Dragomir R. Radev quan tâm tới tình hình nghiên cứu về mô hình đồ thị Web thông qua việc ghi nhận các công trình nghiên cứu liên quan [Rad09]. Trong [Zhu08], Xiaojin Zhu cung cấp một cái nhìn khái quát về tình hình nghiên cứu học máy bán giám sát, mà trong đó có một khối lượng đáng kể các mô hình dựa trên lý thuyết đồ thị và lý thuyết xác suất.

Một khung lý thuyết thống nhất được hình thành làm nền tảng lý thuyết cho các bài toán khai phá dữ liệu đa dạng như phân cụm, phân lớp, luật kết hợp... cũng như cho các cách tiếp cận khác nhau của khai phá dữ liệu sẽ tạo tiền đề thúc đẩy các nghiên cứu thuộc lĩnh vực này. Đồng thời, khung lý thuyết thống nhất được xây dựng cũng hỗ trợ hoạt động phát triển dài hạn của khai phá dữ liệu.

– Mở rộng miền ứng dụng khai phá dữ liệu cả về bề rộng và chiều sâu. Phát triển các ứng dụng khai phá dữ liệu được mở rộng tới thương mại điện tử, tiếp thị điện tử và trở thành trào lưu trong dịch vụ bán lẻ; đồng thời, được tăng cường sử dụng trong nhiều lĩnh vực khác như phân tích tài chính, viễn thông, sinh dược phẩm và các ngành khoa học khác. Xu thế trình độ kinh tế tri thức của xã hội ngày càng được tăng cường là tiền đề cho việc mở rộng miền ứng dụng của khai phá dữ liệu.

Theo hướng kết hợp với các lĩnh vực khoa học khác, có thể thấy xu thế đang phát triển mạnh mẽ các ứng dụng khai phá dữ liệu vào bài toán Y – Sinh học, Môi trường, An toàn, Toàn vẹn dữ liệu và nhiều bài toán khác. Chẳng hạn, hiện nay lĩnh vực khai phá dữ liệu Web đã liên quan mật thiết với thông tin học thông qua độ đo *Webometrics* (một dạng của *Bibliometrics* và *Informetrics* trong thông tin học) [Payn08]. Đồng thời, xu hướng tích hợp khai phá dữ liệu từ các miền ứng dụng khác nhau ngày càng trở nên rõ nét. Theo xu hướng đó, xuất hiện các bài toán khai phá tri thức phức hợp từ các dữ liệu phức tạp mà các dữ liệu này được biểu diễn dưới dạng đồ thị hoặc dưới dạng các quan hệ phức hợp.

– Phát triển các phương pháp khai phá dữ liệu có tính khá cỡ và tương tác. Sự tăng trưởng khối lượng các dữ liệu có rất nhiều chiều và dòng dữ liệu tốc độ cao. Phù hợp với sự bùng nổ thông tin và nhu cầu phát triển ứng dụng khai phá dữ liệu, việc đề xuất các thuật toán khai phá dữ liệu có chức năng tự tương tác và tương tác lẫn nhau đã có tính ban chất. Trong một số ứng dụng, chẳng hạn, trong khai phá text hoặc phân tích an toàn hệ thần kinh, số chiều của dữ liệu lên tới từ hàng trăm triệu tới hàng tỷ đặc trưng. Trong một số ứng dụng khác, chẳng hạn, trong các bài toán nghiên cứu về thiên văn hoặc về mạng máy tính, dòng dữ liệu là rất lớn (có thể lên tới hàng trăm TB tại thời điểm hiện nay). Công nghệ khai phá dữ liệu hiện tại vẫn quá chậm để chủ động được đối với các dữ liệu lớn như vậy. Mặt khác, khai phá dữ liệu dựa trên ràng buộc là một định hướng quan trọng nâng cao năng lực tổng thể của quá trình khai phá dữ liệu có sự tăng cường tương tác với người sử dụng.

– Phát triển các mô hình và phương pháp tích hợp khai phá dữ liệu vào các hệ thống CSDL, hệ thống kho dữ liệu và hệ thống CSDL Web. Các hệ thống này đã trở thành trào lưu của các hệ thống xử lý thông tin. Chẳng hạn, bài toán tích hợp Web với kho dữ liệu bao gồm nhiều nội dung của khai phá nội dung Web để xây dựng được kho dữ liệu với nguồn dữ liệu giàu có của Web. Vẫn đề quan trọng khi tích hợp khai phá dữ liệu ở đây là phải đảm bảo rằng, các phục vụ khai phá dữ liệu được coi là các thành phần phân tích dữ liệu ban chất của hệ thống cần phải được tích hợp một cách trọn vẹn với môi trường xử lý thông tin.

– Chuẩn hóa các ngôn ngữ khai phá dữ liệu cùng với các phương tiện chuẩn hóa khác làm thuận tiện hơn việc phát triển có tính hệ thống các giải pháp khai phá dữ liệu, tính liên thao tác của các hệ thống và chức năng khai phá dữ liệu phức hợp. Một số kết quả ở mức sản phẩm công nghệ điện hình theo hướng này có OLE DB (*Object Linking and Embedding, Database*) dùng cho khai phá dữ liệu của Microsoft, PMML (*Predictive Model Markup Language*) của Data Mining Group (DMG) và CRISP-DM (*Cross Industry Standard Process for Data Mining*) của nhóm phát triển CRISP-DM (<http://www.crisp-dm.org/>).

– Khai phá dữ liệu động, không cân bằng và nhạy cảm về chi phí. Mô hình khai phá dữ liệu cần gắn kết với thời gian, vì dữ liệu là không tĩnh và thay đổi theo thời gian. Theo cách thông thường, mô hình được học cần phù hợp theo thời gian, khi có dữ liệu hiện thời cần học tiếp mô hình cho các khai phá tiếp theo, có nghĩa là mô hình cũng có tính xu hướng. Một khuynh hướng của khai phá dữ liệu là mô hình được xây dựng bao hàm được tính xu hướng càng nhiều càng tốt. Tương tự về khai phá dữ liệu đối với dữ liệu không cân bằng, nhạy cảm về chi phí.

– Khai phá dữ liệu trong một khung cảnh mạng, trong đó có các mạng xã hội trên Internet hoặc các mạng máy tính (khai phá dữ liệu tốc độ cao đối với dòng dữ liệu tốc độ cao). Liên quan mật thiết tới khai phá dữ liệu trong khung cảnh mạng là các bài toán khai phá dữ liệu phân tán và khai phá dữ liệu đa tác từ, cũng như khai phá dữ liệu liên quan tới các quá trình.

– Tăng cường tính trực quan hóa trong khai phá dữ liệu là giải pháp hiệu quả, nhằm làm cho quá trình phát hiện tri thức từ tập dữ liệu được thi hành bằng các bộ công cụ trực quan hóa và dễ dàng tích hợp được với các thành phần khai phá dữ liệu.

Những nội dung được trình bày trên đây về khuynh hướng phát triển của khai phá dữ liệu minh chứng thêm cho khẳng định rằng, lĩnh vực này đang phát triển rất mạnh mẽ.

Câu hỏi và bài tập

1. Phân biệt bài toán quản trị CSDL tác nghiệp với bài toán khai phá dữ liệu.
2. Phân tích vai trò của cơ sở tri thức trong một hệ thống khai phá dữ liệu (Hình 1.6).
3. Phân biệt bài toán khai phá dữ liệu với bài toán kiểm nghiệm giả thiết thống kê.
4. Han và Kamber [HK0106] quan niệm khai phá dữ liệu và phát hiện tri thức trong CSDL là bước phát triển mới của công nghệ CSDL. Hãy lập luận làm sáng tỏ quan niệm trên
5. Trình bày một số mẫu truy vấn trong hệ thống quản trị CSDL và hệ thống khai phá dữ liệu. Phân tích làm sáng tỏ các mẫu truy vấn trong hệ thống khai phá dữ liệu là phức tạp hơn mẫu truy vấn trong hệ thống quản trị CSDL.
6. Hệ thống khai phá dữ liệu có nhất thiết có nguồn đầu vào là kho dữ liệu hay không? Phân tích một số lợi điểm khi hệ thống khai phá dữ liệu có nguồn dữ liệu đầu vào chỉ là các kho dữ liệu.
7. Phân tích về tính "không tầm thường" của quá trình phát hiện tri thức trong CSDL.
8. Phân biệt bài toán khai phá dữ liệu mô tả với bài toán khai phá dữ liệu dự báo.
9. Phân tích tầm quan trọng của khâu làm sạch dữ liệu và tiền xử lý dữ liệu trong quá trình khai phá dữ liệu và trình bày sơ bộ về nội dung của khâu này.
10. Phân tích về sự cần thiết phải tiến hành tính toán giá trị một số đồ đo nào đó trong các bài toán khai phá dữ liệu.

Chương 2

TỔNG QUAN VỀ KHAI PHÁ WEB

2.1. Giới thiệu về khai phá Text

Khai phá Text được bắt nguồn từ các hoạt động xử lý văn bản mà con người đã tiến hành thường xuyên và không ngừng trong suốt lịch sử phát triển của loài người kể từ khi xuất hiện chữ viết; đây là các hoạt động như tóm tắt văn bản, phát hiện nội dung nổi bật, khai thác nội dung của một tác phẩm văn học,... Có thể lấy ví dụ như *Truyện Kiều* của Nguyễn Du, dù đã có rất nhiều công trình nghiên cứu về *Truyện Kiều* đã được công bố; tuy nhiên, các nghiên cứu về *Truyện Kiều* vẫn tiếp tục được tiến hành và có thêm nhiều phát hiện mới lý thú. Từ bản chất da nghĩa, phi ngữ cảnh của ngôn ngữ tự nhiên, có thể nói rằng, các bài toán xử lý văn bản là vô cùng, vô tận, đặc biệt trong hoàn cảnh bùng nổ văn bản điện tử hiện nay. Đặc điểm này vừa là một thuận lợi lớn, song cũng vừa là một khó khăn không nhỏ đối với các nghiên cứu về khai phá Text. Hiện nay, 90% lượng thông tin hiện có là thông tin không cấu trúc; cả theo tỷ lệ phân trăm và theo số lượng tuyệt đối thì lượng thông tin không cấu trúc đang tăng trưởng hàng ngày [Sch09]. Điều đó chứng tỏ rằng, nguồn dữ liệu dâu vào của khai phá Text ngày càng được tăng lên với tốc độ tăng trưởng cao.

Trước hết chúng ta tìm hiểu về khái niệm khai phá Text. Khác với tính phong phú về cách diễn đạt khái niệm khai phá dữ liệu, khái niệm khai phá Text lại nhận được sự thống nhất cao, chẳng hạn như [Ana01, Wit06, Sch09]; điều đó cũng hé lộ sức tự nhiên khi đã thừa nhận nội dung và ý nghĩa của khái niệm khai phá dữ liệu. Thông nhất như vậy được lý giải bằng việc cộng đồng nghiên cứu đã thống nhất về cách hiểu rằng, khai phá Text là dạng khai phá dữ liệu trên dữ liệu Text. Trong giáo trình này, cụm từ "khai phá dữ liệu văn bản" hay "khai phá văn bản" được hiểu là mang nội dung của khái niệm "khai phá Text".

2.1.1. Khái niệm

Khai phá Text là quá trình trích chọn ra các *tri thức mới, có giá trị và tác động được* đang tiềm ẩn trong các văn bản để sử dụng các tri thức này vào việc tổ chức thông tin tốt hơn nhằm hỗ trợ con người.

Về bản chất, khai phá Text là sự kết hợp giữa khai phá dữ liệu và xử lý ngôn ngữ tự nhiên (NLP: Natural Language Processing). Chính vì lẽ đó, nhiều nghiên cứu thuộc lĩnh vực NLP được coi là thành phần của khai phá Text và ngược lại, nhiều kết quả nghiên cứu về khai phá Text lại được coi là các sản phẩm phát triển từ NLP⁽¹⁾. Một số nội dung cơ bản nhất về NLP và xử lý tiếng Việt được trình bày trong Chương 4.

2.1.2. Quá trình khai phá Text

Quá trình khai phá Text là cụ thể hoá quá trình khai phá dữ liệu nói chung đối với dữ liệu Text. Với giả thiết đã xác định được: (1) bài toán khai phá Text và (2) miền dữ liệu Text thuộc miền ứng dụng, quá trình khai phá Text trải qua các bước như sau:

– Thu thập dữ liệu Text thuộc miền ứng dụng. Ở bước này, có hai điều cần được lưu ý. *Thứ nhất*, chỉ cần thu thập dữ liệu Text thuộc miền ứng dụng mà không phải là tập tất cả các văn bản có thể có của thế giới thực. Chẳng hạn, trong bài toán khai phá Text của Rich Caruana cùng cộng sự [CJG06], miền ứng dụng quy định rằng, tập dữ liệu chỉ là tập tất cả các công trình khoa học; còn trong bài toán khai phá dữ liệu Text thuộc lĩnh vực y tế và chăm sóc sức khỏe thì chỉ cần quan tâm thu thập các văn bản về y tế và chăm sóc sức khỏe. *Thứ hai*, yêu cầu cốt lõi của bước thu thập dữ liệu là tập dữ liệu Text thu thập được phải đại diện được cho toàn bộ dữ liệu Text thuộc miền ứng dụng. Chẳng hạn, tập dữ liệu trang Web mà máy tìm kiếm Google thu thập được cho là đại diện cho toàn bộ tập mọi trang Web trên Internet; tính đại diện đó được đảm bảo bằng thuật toán thu thập trang Web (thuật toán Crawling) của Google phù hợp với mô hình sinh các trang Web trên Internet. Mô hình sinh trang Web, tính ngẫu nhiên của việc thu thập dữ liệu là các yếu tố cần được quan tâm trong thuật toán thu thập trang Web. Nên nhớ rằng, tập trang Web mà Google thu thập được (còn được gọi là được Google đánh chỉ số) dù rất đồ sộ, song không phải là toàn bộ mọi trang Web có thể có.

– Biểu diễn dữ liệu Text thu thập được sang khuôn dạng phù hợp với bài toán khai phá Text. Chương 5 sẽ trình bày một số phương pháp biểu diễn Text, trong đó một số công cụ và tài nguyên thiết yếu của Xử lý ngôn ngữ tự nhiên sẽ được sử dụng. Biểu diễn dữ liệu Text càng phù hợp với bài toán khai phá Text, thì chất lượng của kết quả khai phá Text càng được nâng cao.

– Lựa chọn tập dữ liệu đầu vào cho thuật toán khai phá dữ liệu. Trong hầu hết trường hợp, tập dữ liệu thuộc miền ứng dụng đã thu thập được là rất

⁽¹⁾ <http://www.ics.mq.edu.au/~swan summarization projects full.htm>

lớn, và vì vậy, trong nhiều trường hợp là vượt quá khả năng xử lý (về không gian, về thời gian) đối với các thuật toán khai phá dữ liệu. Chính vì lý do đó, cần chọn ra từ tập dữ liệu thu thập được một tập con để thực hiện bài toán khai phá dữ liệu. Các yếu tố đảm bảo tính đại diện của tập dữ liệu thu thập được cũng được áp dụng trong các giải pháp lựa chọn tập dữ liệu đầu vào cho thuật toán khai phá dữ liệu.

- Thực hiện thuật toán khai phá dữ liệu đối với tập dữ liệu đã được lựa chọn để tìm ra các mẫu, các tri thức. Chẳng hạn, đối với bài toán phân lớp văn bản, mẫu (tri thức) được tích hợp thành bộ phân lớp kết quả và bộ phân lớp này sẽ được sử dụng vào việc phân lớp đối với các văn bản mới.

- Thực hiện việc khai thác sử dụng các mẫu, các tri thức nhận được từ quá trình khai phá Text vào thực tiễn hoạt động.

2.1.3. Đặc trưng của khai phá Text

Đặc thù về đối tượng dữ liệu văn bản tạo ra một số đặc điểm của khai phá Text là khác biệt so với khai phá dữ liệu thông thường. Bang 2.1 trình bày một số đối sánh về một số đặc trưng giữa khai phá Text với khai phá dữ liệu nói chung [Ana01]. Qua nội dung trình bày trong Bang 2.1, có thể thấy khai phá Text là lĩnh vực khai phá dữ liệu được phát triển gần đây, có thị trường rộng lớn, đối tượng là các văn bản không có cấu trúc, với một số mục tiêu và phương pháp đặc thù

Bảng 2.1. So sánh đặc điểm khai phá dữ liệu với khai phá Text [Ana01]

| Dấu hiệu phân biệt | Khai phá dữ liệu | Khai phá Text |
|--------------------|---|--|
| Đối tượng dữ liệu | Dữ liệu số/phân loại | Văn bản |
| Cấu trúc đối tượng | CSDL quan hệ | Text dạng tự do |
| Mục tiêu | Đưa báo, đoán nhận | Tìm kiếm thông tin liên quan, hiểu ngữ nghĩa, phân lớp/phân bố |
| Phương pháp | Học máy DT, NN, GA, MBR | Chi số, xử lý mạng nơron, ngôn ngữ, kiến trúc |
| Kích cỡ thị trường | Trăm nghìn phân tích viên từ công ty lớn và vừa | Hàng triệu người dùng từ hàng và cá nhân |
| Tình trạng | Quảng bá từ năm 1994 | Mới quảng bá từ năm 2000 |

2.1.4. Một số bài toán điển hình

Một số bài toán điển hình nhất trong khai phá Text là [Lew91, Ana01]:

– **Tìm kiếm (Search and retrieval):** Quá trình tìm kiếm văn bản theo yêu cầu của người dùng. Nội dung của bài toán này là chọn ra một tập con các văn bản trong kho chứa các văn bản để hiển thị cho người dùng toàn bộ hoặc bộ phận các văn bản đáp ứng yêu cầu của người dùng. Trong trường hợp toàn bộ các văn bản đã được tập hợp vào kho chứa (CSDL văn bản), thì bài toán chỉ bao gồm công việc lấy ra (Retrieval) từ CSDL các văn bản đáp ứng truy vấn; ngược lại cần phải tìm kiếm (Search) văn bản từ mọi nguồn và lấy ra các văn bản đáp ứng. Yêu cầu của người sử dụng được thể hiện dưới dạng truy vấn, trong đó dạng đơn giản nhất là từ khoá (keyword hoặc term), hiểu theo nghĩa "hãy tìm ra các văn bản có chứa từ khoá nói trên". Truy vấn có thể được trình bày dưới dạng phức tạp là "biểu thức" tổ hợp các đặc trưng trình bày văn bản trong ứng dụng cụ thể.

Các phương pháp tìm kiếm văn bản điện hình là dựa theo chỉ số (như Excite, Altavista...), dựa trên kiến trúc (như Yahoo, Lycos, Megaputer theo nền kiến trúc), dựa theo phân tích nhị phân và gốc từ (như HotBot, dt-Search), dựa trên ngữ nghĩa và ngôn ngữ (như các sản phẩm của Megaputer theo nền ngữ nghĩa và ngôn ngữ) [FF02, Ana01, Sla02, YN99].

Máy tìm kiếm (search engine) là một dạng điện hình về hệ thống tìm kiếm trang Web, mà theo mỗi truy vấn của người sử dụng thì máy tìm kiếm sẽ chỉ ra một danh sách các văn bản (xếp giảm dần theo độ quan trọng) đáp ứng truy vấn của người sử dụng. Trường hợp đặc biệt khi mà truy vấn chỉ là từ khoá (hoặc một tập từ khoá), thì văn bản kết quả tìm ra chỉ cần chứa từ khoá (hay tập từ khoá) xuất hiện trong truy vấn. *Chương 6 – Hệ thống tìm kiếm* sẽ trình bày chi tiết về máy tìm kiếm.

– **Phân tích ngữ nghĩa (Semantic analysis):** Quá trình đưa ra cách "hiểu" về văn bản thông qua mối liên quan ngữ nghĩa của văn bản với tập khái niệm cho trước [BFS03, Ana01, YN99]. Một bài toán nền tảng của phân tích ngữ nghĩa là phát hiện quan hệ ngữ nghĩa trong văn bản [Gir08].

– **Phân cụm:** Quá trình nhóm các văn bản trong một tập văn bản thành các "cụm", trong đó nội dung các văn bản trong cùng một cụm là "gắn gần" nhau theo một độ đo nào đó [BBM02, Ana01]. *Chương 7 – Phân cụm văn bản* trình bày chi tiết về bài toán phân cụm.

– **Phân lớp:** Quá trình tự động xếp văn bản vào một trong một số lớp văn bản đã xác định từ trước. Các giải pháp phân lớp tự động văn bản thường được thiết kế dựa trên các phương pháp học máy (Cây quyết định, Bayes, k-nghề láng giềng gần nhất). Như đã được giới thiệu, phân đoạn là quá trình kết hợp của hai quá trình phân cụm và phân lớp; trước tiên tiến hành phân cụm, sau đó tiến hành phân lớp [Ana01, Slat02]. Các nội dung cơ bản về phân lớp văn bản được trình bày trong *Chương 8 – Phân lớp văn bản*.

- Trích chọn đặc trưng (Feature extraction): Quá trình phát hiện và lưu trữ lại những thành phần ngôn ngữ cần quan tâm được xuất hiện trong văn bản. Phát hiện từ mang nghĩa (term), cụm từ mang nghĩa (feature) xuất hiện trong văn bản, biểu diễn văn bản theo các thành phần mang nghĩa đó hoặc sử dụng chúng trong các CSDL. Web được coi là thuộc vào công việc trích chọn đặc trưng. Trong một số trường hợp, các đặc trưng chưa xác định trước, việc xác định chúng đồng thời với việc phân tích nội dung văn bản: tuy nhiên, trong một số trường hợp khác, các đặc trưng (như tên người, công việc,...) có thể được xác định trước, việc phân tích văn bản cho phép phát hiện sự xuất hiện (có tần số) các đặc trưng đó trong văn bản [Slat02, Cha03].

Cần phân biệt khái niệm trích chọn đặc trưng ở đây với khái niệm lựa chọn (rút gọn) đặc trưng trong biểu diễn trang Web, trong đó việc lựa chọn đặc trưng được định hướng vào việc rút gọn tập các đặc trưng đã cho nhằm giảm bớt độ phức tạp xử lý trong khai phá Web.

- Tóm tắt văn bản (Document Abstract/Summarization): Từ một văn bản nguồn, cần xây dựng một văn bản dịch "ngắn hơn" mà vẫn giữ nguyên (về cơ bản) được ngữ nghĩa của văn bản nguồn. Tóm tắt văn bản còn được quan niệm như "hiểu văn bản" (Document Understanding).

Tồn tại hai dạng điển hình trong việc xây dựng văn bản tóm tắt. Dạng đầu tiên (Abstract) là đơn giản hơn, trong đó văn bản tóm tắt gồm một số câu sẵn có của văn bản nguồn, hay văn bản tóm tắt là văn bản nguồn loại bỏ đi một số (có thể nhiều/rất nhiều) câu. Dạng thứ hai (Summarization) phức tạp hơn, mỗi câu trong văn bản tóm tắt được xây dựng từ việc kết hợp các cụm từ xuất hiện ở một số câu trong văn bản nguồn. Thông thường các hệ thống sử dụng giải pháp kết hợp hai dạng này.

Tóm tắt đa văn bản (multi – document summarization) giải quyết bài toán vấn đề xây dựng một văn bản tóm tắt cho một tập hợp các văn bản là một bài toán thường gặp trong khai phá Web, chẳng hạn như, sau khi phân cụm trang Web thì một yêu cầu đặt ra là mô tả tóm tắt nội dung các văn bản trong cụm này.

Viện Quốc gia về Chuẩn hoá và Công nghệ Mỹ (The National Institute of Standards and Technology) rất quan tâm tới các hoạt động nghiên cứu và triển khai về hiểu văn bản. Hoạt động điển hình của cơ quan này là tổ chức hội nghị khoa học quốc tế thường niên về hiểu văn bản với tên gọi là Document Understanding Conference (DUC) trong giai đoạn 2001 – 2007 và do tổ chức và Text Analysis Conference (TAC) từ năm 2008 tới nay. Gần đây nhất, TAC 2009 Workshop¹¹ được tổ chức trong các ngày 16 – 17/11/2009 tại Gaithersburg, Maryland USA. Tại các hội nghị DUC và

¹¹ <http://www.nist.gov/tac/2009/workshop/index.html>

TAC, nhiều công trình nghiên cứu về tóm tắt văn bản (bao gồm cả tóm tắt da văn bản) có giá trị được công bố.

– Ngoài ra, các bài toán Xây dựng ontology (Ontology building). Dẫn đường văn bản cũng nhận được nhiều sự quan tâm [BFS03, Ana01]

Trong thực tế, tồn tại rất nhiều bài toán liên quan tới khai phá Text. Dưới đây là một bài toán ví dụ.

Ví dụ 2.1. Trong một seminar khoa học tại Việt Nam¹, GS. John Edward Hopcroft⁽²⁾ (một chuyên gia hàng đầu của Mỹ về lĩnh vực CNTT) đã trình bày báo cáo "Future Directions in Computer Science" [Hop07] và ông đã giới thiệu công trình nghiên cứu của Rich Caruana cùng cộng sự [CJG06]. Bài toán trong công trình nghiên cứu của Rich Caruana cùng cộng sự được mô tả như sau: Cho trước một tập (khoảng 300000) công trình nghiên cứu khoa học (bài đăng tạp chí, báo cáo hội nghị, luận án tiến sĩ) đã được công bố. Từ nội dung văn bản của mỗi công trình nghiên cứu, chúng ta nhận được tên tác giả (các tác giả), các tài liệu tham khảo, nơi công bố (tên tạp chí, hội nghị, hội thảo....).

Sử dụng nội dung các tài liệu và chỉ một loại siêu dữ liệu là tên các tác giả của tài liệu, Rich Caruana cùng cộng sự đặt ra các mục tiêu cơ bản cần hướng tới là:

- + Tìm ra diễn biến quá trình phát triển theo thời gian của các chủ đề khoa học theo một số tiêu chí như ty lệ các tài liệu theo các chủ đề, các chủ đề nổi bật mới, thời điểm một chủ đề cụ thể đạt đỉnh cao nhất, chủ đề nào đang tàn lụi..., và theo đó, tìm ra được các chủ đề có vai trò chủ chốt trong tập hợp các chủ đề.

- + Nhận biết được các tài liệu có uy thế là tài liệu giới thiệu các ý tưởng mới và có chỉ số ảnh hưởng lớn.

- + Nhận biết được tác giả có uy thế là tác giả có ảnh hưởng lớn đối với sự phát triển của các chủ đề.

Có thể thấy rằng, tồn tại không ít bài toán khai phá Text được phát sinh trong công trình của Rich Caruana cùng cộng sự. Hình 1.10 đã trình bày một trong những kết quả diễn hình của Rich Caruana cùng cộng sự cho thấy xu hướng phát triển của 13 nhóm chủ đề nghiên cứu khoa học – công nghệ chính yếu. Biểu đồ trên Hình 1.10 cho thấy một số nhóm chủ đề nghiên cứu hiện đang trong giai đoạn phát triển tốt (*nhóm 10*: bayesian, mixture, posterior, likelihood, em; *nhóm 9*: spike, spikes, firing, neuron, neurons;

¹ Seminar được tổ chức vào ngày 19-8-2008 tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

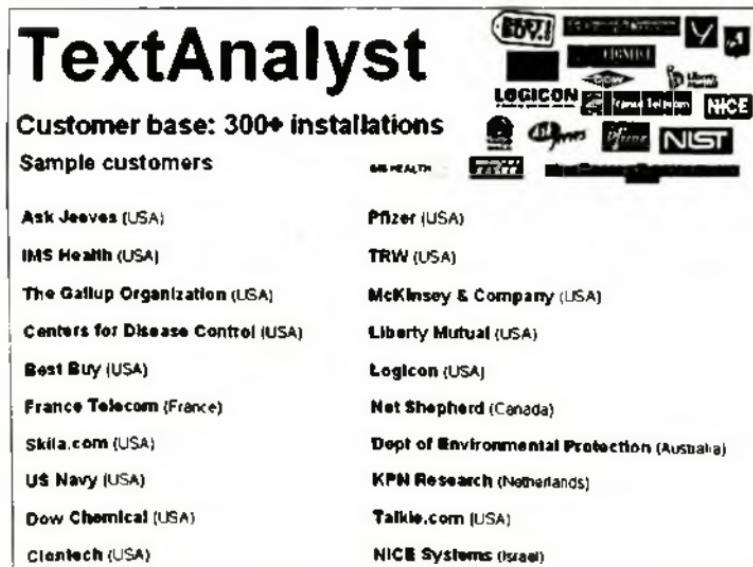
² <http://www.cs.cornell.edu/jeh>

³ <http://www.cs.cornell.edu/Chinab2007.ppt>

*nhóm 2: image, images, object, face, video), một số nhóm chủ đề hiện đang phát triển song song có xu hướng chung lại (*nhóm 12: chip, circuit, analog, voltage, vlsi; nhóm 4: units, node, training, nodes, tree*) và các nhóm chủ đề còn lại đang phát triển bình thường.* Lưu ý là, một số nhóm chủ đề có xu hướng chung lại song vẫn có số lượng lớn công trình nghiên cứu được công bố, điển hình là nhóm chủ đề thứ 12: chip, circuit, analog, voltage, vlsi.

Tồn tại nhiều cơ sở nghiên cứu, nhóm nghiên cứu về khai phá Text trên thế giới và một trong số đó là Trung tâm Khai phá Text quốc gia Anh (U.K. National Text Mining Centre¹¹). Trung tâm này đã công bố một số công trình nghiên cứu và sản phẩm về khai phá Text [AN06], và khẳng định là một trong những đơn vị đầu tiên nghiên cứu về khai phá Text.

Kết quả nghiên cứu về khai phá text còn được thi hành bằng các sản phẩm phần mềm hoàn chỉnh, hoặc các tiện ích thành phần trong các sản phẩm hoàn chỉnh; chẳng hạn, các sản phẩm của Công ty Megaputer. Hình 2.1 trình bày một số khách hàng đối với sản phẩm TextAnalyst của Công ty Megaputer. Danh sách các khách hàng của Megaputer cho thấy phần mềm khai phá Text có thể được tích hợp thành các thành phần trong nhiều ứng dụng đa dạng.



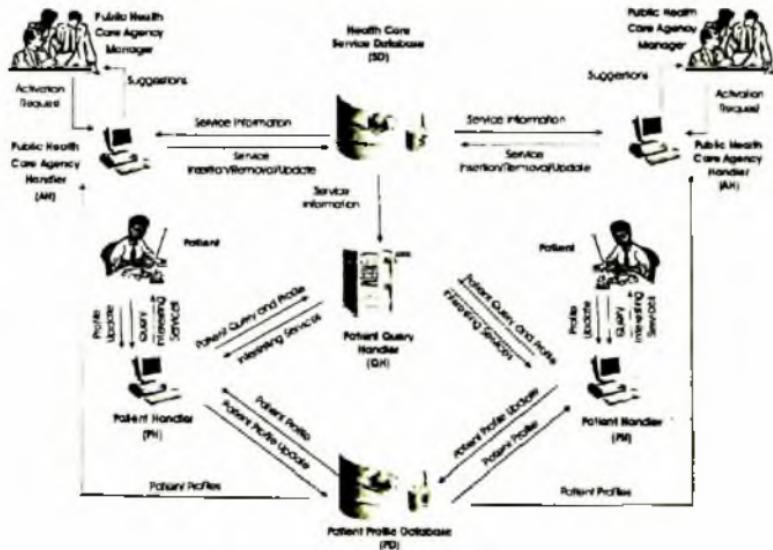
Hình 2.1. Các khách hàng sử dụng sản phẩm TextAnalyst của Công ty Megaputer

¹¹ <http://www.nactem.ac.uk/>

2.1.5. Một số lĩnh vực ứng dụng khai phá Text điện hình

Trong [MY09], một số lĩnh vực ứng dụng điện hình của khai phá Text đã được trình bày bao gồm các lĩnh vực Y tế và Chăm sóc sức khỏe, Sinh học, An toàn thông tin, Phát triển phần mềm, Khoa học và một số ứng dụng trong khuôn khổ của khai phá Web.

– Y tế và Chăm sóc sức khỏe là một lĩnh vực ứng dụng điện hình nhất của khai phá Text hiện nay chính bởi tính quan trọng của lĩnh vực này trong đời sống loài người. Tập hợp các văn bản sinh ra trong hoạt động Y tế và Chăm sóc sức khỏe hàm chứa nhiều thông tin tiềm ẩn, hữu ích phục vụ con người. Chẳng hạn, một bài toán điện hình trong lĩnh vực này là xem xét tập các văn bản lưu trữ hoạt động khám và điều trị bệnh nhân của các bác sĩ, tập văn bản này có số lượng rất lớn. Một mặt, tập văn bản này không chỉ cung cấp các kinh nghiệm chuyên môn của các bác sĩ trong quá trình khám và điều trị, mà còn cung cấp thông tin về quá trình phát triển chuyên môn của từng bác sĩ. Mặt khác, trong tập các văn bản như thế cũng tiềm ẩn các thông tin liên quan về mối quan hệ bệnh – thuốc điều trị – bệnh nhân.

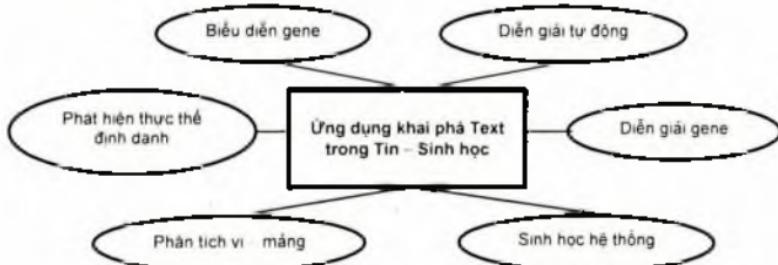


Hình 2.2. Hệ thống hỗ trợ quyết định trong chăm sóc sức khỏe [MY09]

Sophia Ananiadou [Ana08] cung cấp một danh mục các công bố khoa học về khai thác về khai phá Text trong Y – Sinh học, một số hội thảo điện

hình, nhận dạng thực thể tên và quản lý thuật ngữ, khai phá quan hệ – trích chọn sự kiện, chủ giải và kho dữ liệu Y – Sinh học, phân cụm từ và phân tích cú pháp. Danh mục này cho thấy vẫn đề khai phá Text trong Y – Sinh học không chỉ hấp dẫn mà còn có xu thế ngày càng phát triển. Nhiều công trình nghiên cứu đã giới thiệu các hệ thống ứng dụng trong thực tiễn. Chẳng hạn trong [MY09], P.D. Meo và cộng sự giới thiệu một hệ thống khai phá Text để hỗ trợ quyết định (trong hoạt động chăm sóc sức khỏe) (Hình 2.2).

– Gần gũi với ứng dụng trong lĩnh vực Y tế và Chăm sóc sức khỏe, ứng dụng khai phá Text trong Tin – Sinh học cũng đang phát triển. Các thuật toán khai phá dữ liệu cho xâu ký tự cũng được cài tiến vào xâu các gen sinh học và thu được các kết quả thú vị. Yanliang Qi [MY09] liệt kê các ứng dụng điển hình của khai phá Text trong Tin – Sinh học (Hình 2.3). Theo Sophia Ananiadou và John McNaught [AN06], việc tổ chức, tìm kiếm, khám phá hoặc truyền dẫn tri thức sinh học nhận được sự quan tâm đặc biệt của nhiều nhóm nghiên cứu trên thế giới. Kho chứa văn bản về sinh học PubMed (NIH-NLM's Medline) chứa tới 18 triệu văn bản tóm tắt, bao gồm nhiều tư liệu nguồn quý giá về tương tác protein.



Hình 2.3. Các ứng dụng điển hình của khai phá Text trong Tin – Sinh học [MY09]

– Công trình của Rich Caruana cùng cộng sự [CJG06] mô tả một ví dụ ứng dụng của khai phá Text trong khoa học.

– Shuting Xu và Xin Luo [MY09] trình bày kết quả nghiên cứu về các vấn đề hiện tại và phân tích tương lai của khai phá Text đối với các ứng dụng an toàn thông tin. Các dạng dữ liệu điển hình để khai phá dữ liệu là e-mail, trang Web, tin tức, thông điệp diễn đàn Web/thông điệp khẩn cấp. Các kiểu tri thức cần được phát hiện về kết nối giữa các con người/các nhóm/các đối tượng, các ứng xử/thị hiếu lạm thường của con người, chủ đề/dề tài của văn bản, tác giả của văn bản,... Mạng xã hội là nền tảng tốt đồi với các giải pháp ứng dụng khai phá Text trong an toàn thông tin.

- Ứng dụng của khai phá Text trong lĩnh vực phát triển phần mềm liên quan tới phát hiện sự "nhái" phần mềm trong công nghệ phần mềm và hệ thống nhúng. Theo A. Dreweke và cộng sự [MY09], có tới 30% mã của các hệ thống phần mềm lớn là "nhái". khai phá Text theo dấu chỉ định và dựa trên đồ thị tỏ ra hữu hiệu để nhận diện các thành phần nhái trong hệ thống phần mềm. Đồng thời, khai phá Text cũng được ứng dụng để phát hiện các đoạn mã lặp, giúp mã chương trình được thu gọn hơn.

- Các ứng dụng của khai phá Text được tích hợp trong khai phá Web sẽ được giới thiệu ở mục tiếp theo.

2.2. Giới thiệu về khai phá Web

Khoảng gần chục năm trở lại đây, bắt đầu từ khoảng năm 2000, khai phá Text và khai phá Web có tốc độ phát triển vượt bậc. Những nội dung nghiên cứu về lĩnh vực khai phá Text và khai phá Web đã và đang nhận được sự quan tâm đặc biệt của nhiều nhà khoa học, nhiều nhóm nghiên cứu trên toàn thế giới. Hàng năm, nhiều hội thảo khoa học quốc tế được tổ chức, nhiều công trình khoa học được công bố trên các tạp chí khoa học, các trang thông tin điện tử [Knd].

Hơn mươi lăm năm vừa qua, sự phát triển nhanh chóng của mạng Internet và Intranet đã sinh ra một khối lượng không lồ các dữ liệu dạng siêu văn bản (dữ liệu Web). Internet đã trở thành một kênh thông tin rất quan trọng về khoa học, kinh tế, thương mại; đồng thời, nó cũng đã là một kênh quang cáo thông dụng và hiệu quả. Một trong những lý do cho sự phát triển như vậy là chi phí thấp để duy trì một trang Web trên Internet. So sánh với những dịch vụ khác như đăng tin hay quảng cáo trên một tờ báo hay tạp chí truyền thông, thì một trang Web là rẻ hơn rất nhiều và phân phối nhanh chóng hơn tới hàng triệu người dùng trên khắp mọi nơi trên thế giới. Internet như một xã hội ảo, bao gồm các thông tin về mọi mặt của đời sống kinh tế, xã hội trên toàn thế giới được trình bày dưới dạng văn bản, hình ảnh, âm thanh.... Chính vì lý do đó, Web đã trở thành một hệ thống rất lớn, được phân bổ rộng khắp, cung cấp thông tin trên mọi lĩnh vực khoa học, xã hội, thương mại, văn hoá... là một nguồn tài nguyên giàu có cho khai phá dữ liệu.

Phần dữ liệu quan trọng trên Web là văn bản, và vì vậy, khai phá Text là một nội dung đương nhiên trong khai phá Web. Tuy nhiên, khai phá Web còn bao hàm nhiều nội dung khác. Khai phá Web ngoài việc cần khai phá dữ liệu từ nội dung các trang văn bản, còn phải khai thác được các nguồn tài nguyên đa dạng khác trên Web cũng như khai thác được mối quan hệ giữa chúng. Có thể coi khai phá Web là sự hội tụ giữa khai phá dữ liệu, xử lý

ngôn ngữ tự nhiên và Word-Wide-Web, bao gồm rất nhiều lĩnh vực nghiên cứu như CSDL, trí tuệ nhân tạo, truy xuất thông tin và nhiều lĩnh vực khác. Các công nghệ Agent-base, truy xuất thông tin dựa trên khái niệm, truy xuất thông tin sử dụng lập luận dựa trên trường hợp riêng (case-based reasoning) và tính hạng văn bản dựa trên các đặc trưng, các siêu liên kết,... thường được xem là các thành phần trong khai phá Web. Các chủ đề trong khai phá Web vẫn tiếp tục được mở rộng, và chính vì vậy, định nghĩa khai phá Web vẫn còn tiếp tục được xem xét bổ sung.

2.2.1. Khái niệm

Như vậy chúng ta có thể hiểu rằng, khai phá Web như là việc *trích chọn ra các thành phần được quan tâm hay được đánh giá là có ích cùng các thông tin tiềm năng từ các tài nguyên hoặc các hoạt động liên quan tới World Wide Web* [BFS03, Sla02].

Một cách trực quan có thể quan niệm khai phá Web là sự kết hợp giữa khai phá Text với Công nghệ Web, hay cụ thể hơn là:

$$\text{Khai phá Web} = \text{Khai phá dữ liệu} + \text{Xử lý ngôn ngữ tự nhiên} + \text{World Wide Web}$$

Hiện tại, phần nội dung diễn hình nhất trong trang Web là văn bản, vì vậy, khai phá văn bản Web là một thành phần cơ bản của khai phá Web. Tuy nhiên, với sự tiến bộ không ngừng của công nghệ Internet, nhu cầu về khai phá dữ liệu đối với các dữ liệu đa phương tiện khác như hình vẽ, tiếng nói, ca nhạc, phim,... đã không ngừng được phát triển cả về chiều rộng lẫn chiều sâu. Có nhiều công trình nghiên cứu về lĩnh vực mới mẻ này, chẳng hạn như [Tu04].

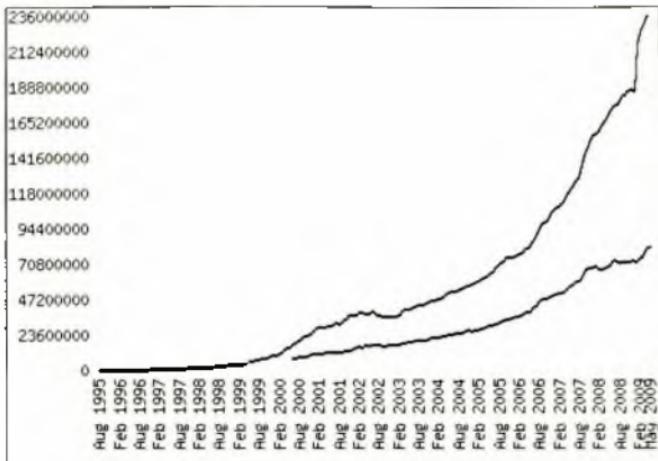
2.2.2. Đặc trưng của khai phá Web

Một số đặc điểm dưới đây cho thấy một số thách thức cũng như thuận lợi đối với khai phá Web [Sla02].

- **Web quá lớn để tổ chức thành kho dữ liệu**

Các CSDL truyền thống có kích thước không lớn lắm, thường được lưu trữ tập trung; trong khi đó, kích thước Web rất lớn, tới hàng terabytes và thay đổi liên tục, không những thế còn phân tán trên rất nhiều máy tính khắp trên thế giới. Hình 2.4 cho số liệu thống kê tại thời điểm tháng 5/2009, cho thấy có gần 236 triệu Website trên Internet. Trong năm tháng đầu năm 2009, trung bình mỗi tháng có tới 2.8 triệu Website mới xuất hiện. Theo kết quả thống kê vào tháng 01/2005, có hơn 11 tỷ trang Web được đánh chỉ số [Knd513]. Google đã lưu trữ hơn 4 tỷ trang Web với dung lượng hàng trăm terabytes [GPS06]. Kích thước trung bình của mỗi trang là 5 – 10KB thì

tổng kích thước của các trang Web được đánh chỉ số đã lên tới ít nhất là 55TB. Tỷ lệ tăng của các trang Web rất nhanh, hai năm gần đây số các trang Web tăng gấp đôi và còn tiếp tục tăng trong hai năm tới. Nhiều tổ chức và xã hội đặt hầu hết những thông tin công cộng của họ lên Web. Như vậy, việc xây dựng một kho dữ liệu để lưu trữ, sao chép hay tích hợp các dữ liệu trên Web là gan như không thể.



Hình 2.4. Tổng số website là 235.890.526 sites (tháng 5/2009)

• Độ phức tạp của trang Web là rất lớn

Dữ liệu trong các CSDL truyền thống thường là loại dữ liệu đồng nhất (về ngôn ngữ, định dạng,...), còn dữ liệu Web thì hoàn toàn không đồng nhất. Dữ liệu Web bao gồm rất nhiều loại ngôn ngữ khác nhau (cả ngôn ngữ diễn tả nội dung lẫn ngôn ngữ lập trình), nhiều loại định dạng khác nhau (text, HTML, PDF, hình ảnh, âm thanh,...), nhiều loại từ vựng khác nhau (địa chỉ email, các liên kết, các mã nén – zipcode, số điện thoại,...). Nói cách khác, các trang Web thiếu một cấu trúc thông nhất. Chúng được coi như “một thư viện kỹ thuật số rất rộng lớn”; tuy nhiên, lượng không lồ các tài liệu trong thư viện lại không được sắp xếp theo một tiêu chuẩn chuyên biệt nào, không theo phạm trù nào.... Điều này là một thử thách rất lớn cho việc tìm kiếm thông tin cần thiết trong một thư viện như thế.

• Web là một nguồn tài nguyên thông tin có độ thay đổi cao

Web không chỉ có thay đổi về độ lớn, mà thông tin trong chính các trang Web cũng được cập nhật liên tục. Theo kết quả nghiên cứu [Sla02]

hơn 500.000 trang Web trong hơn bốn tháng thì có tới 23% các trang thay đổi hàng ngày, và khoảng hơn 10 ngày thì 50% các trang trong tên miền đó biến mất, nghĩa là địa chỉ URL của nó không còn tồn tại nữa. Tin tức, thị trường chứng khoán, các công ty quảng cáo và trung tâm phục vụ Web thường xuyên cập nhật trang Web của họ. Thêm vào đó sự kết nối thông tin và sự truy cập bản ghi cũng được cập nhật.

• *Web phục vụ một cộng đồng người dùng rộng lớn và đa dạng*

Internet hiện nay nối với khoảng 50 triệu trạm làm việc [Sla02], và cộng đồng người dùng vẫn đang nhanh chóng lan rộng. Mỗi người dùng có một kiến thức, mỗi quan tâm, sở thích khác nhau. Nhưng hầu hết người dùng không có kiến thức tốt về cấu trúc mạng thông tin, hoặc không có ý thức cho những tìm kiếm, rất dễ bị "lạc" trong khối dữ liệu khổng lồ của mạng, hoặc nhầm chán khi tìm kiếm mà chỉ nhận được những mảng thông tin không mấy hữu ích.

• *Chỉ một phần rất nhỏ của thông tin trên Web là thực sự hữu ích*

Theo thống kê [Sla02], 99% của thông tin Web là vô ích với 99% người dùng Web. Vì vậy, nhiều phần Web không đáng quan tâm lại có trong kết quả nhận được khi tìm kiếm. Cần tìm giải pháp khai phá Web để nhận được trang Web chất lượng cao nhất theo tiêu chuẩn quan tâm của người dùng.

Những đặc điểm trên đây cho thấy sự khác biệt đáng kể giữa việc tìm kiếm trong một CSDL truyền thống với việc tìm kiếm trên Internet. Tuy nhiên, các thách thức trên cũng thúc đẩy hoạt động nghiên cứu khai phá dữ liệu các tài nguyên trên Internet.

Đồng thời, khai phá Web cũng nhận được một số thuận lợi do sự giàu có thông tin trên Web:

– Trang Web được cấu trúc theo quy định của ngôn ngữ định dạng, theo đó, ngoài phần nội dung (văn bản, hình ảnh và dữ liệu đa phương tiện khác), trang Web còn chứa các thẻ thi hành cấu trúc các nội dung của trang Web đó. Chính vì lý do đó, trang Web được coi như một loại *dữ liệu bán cấu trúc*. Dữ liệu bán cấu trúc từ trang Web (tính bán cấu trúc của trang Web) tạo cho khai phá Web có được một số thuận lợi nhất định khi so sánh với khai phá Text.

– Web bao gồm không chỉ có các trang, mà còn có cả các liên kết trỏ từ trang này tới trang khác. Khi một tác giả tạo một liên kết từ trang của ông ta tới một trang A, có nghĩa là A là trang có hữu ích với vấn đề đang bàn luận. Nếu một trang càng nhiều liên kết từ trang khác trỏ đến, chứng tỏ trang đó quan trọng. Ví vậy, các thông tin liên kết trang cung cấp một lượng thông tin giàu có về mối liên quan, chất lượng và cấu trúc của nội dung trang Web, và vì thế là một nguồn tài nguyên lớn cho khai phá Web.

- Một máy chủ Web thường đăng ký một bản ghi đầu vào (Weblog entry) cho mọi lần truy cập trang Web. Nó bao gồm địa chỉ URL, địa chỉ IP, timestamp. Dữ liệu Weblog cung cấp lượng thông tin giàu có về những trang Web động. Thực hiện phân tích các hồ sơ truy cập này có thể rút ra những thông kê về xu hướng truy cập Web, cấu trúc Web và nhiều thông tin hữu ích khác.

2.2.3. Phân loại khai phá Web

Hình 2.5 trình bày sơ đồ phân loại các lĩnh vực nghiên cứu điện hình trong khai phá Web. Người ta thường phân khai phá Web thành ba lĩnh vực chính: khai phá nội dung Web (web content mining), khai phá cấu trúc Web (web structure mining) và khai phá sử dụng Web (web usage mining).



Hình 2.5. Phân loại khai phá Web

• Khai phá nội dung Web

Phần lớn các tri thức của World Wide Web được chứa trong nội dung văn bản. Tuy nhiên, văn bản không phải là toàn bộ nội dung Web, vì rằng, trên Web còn có các dữ liệu đa phương tiện khác như hình ảnh, âm thanh, video... Khai phá dữ liệu hình ảnh, âm thanh, video,... hiện đang là hướng nghiên cứu thời sự. Tuy nhiên, giải trình này tập trung định hướng vào khai phá dữ liệu văn bản, cho nên, những nội dung liên quan đến khai phá dữ liệu đa phương tiện như hình ảnh, âm thanh, video,... sẽ chỉ được đề cập một cách sơ bộ. Khai phá nội dung Web là các quá trình xử lý để lấy ra các tri thức từ nội dung các trang văn bản hoặc mô tả của chúng.

Có hai chiến lược khai phá nội dung Web, đó là: (1) khai phá trực tiếp nội dung của trang Web; (2) tối ưu kết quả trả về của các công cụ tìm kiếm, chẳng hạn như máy tìm kiếm.

- *Khai phá nội dung trang Web:* Liên quan tới việc truy xuất các thông tin từ các văn bản có cấu trúc, văn bản siêu liên kết, hay các văn bản bán cấu

trúc. Vấn đề này liên quan chủ yếu tới việc khai phá bùn thâm nội dung các văn bản, và đây là nội dung cơ bản nhất cần quan tâm trong khai phá Web. Các chương tiếp theo sẽ trình bày nội dung; các phương pháp giải quyết các bài toán phân cụm, phân lớp, trích chọn thông tin và các bài toán liên quan trong khai phá nội dung trang Web.

- *Tối ưu hóa kết quả tra từ các công cụ tìm kiếm:* Trong các công cụ tìm kiếm Web, sau khi đã tìm ra được những trang Web thỏa mãn yêu cầu người dùng, còn một công việc không kém phần quan trọng, đó là phải sắp xếp, chọn lọc kết quả theo mức độ phù hợp với yêu cầu người dùng. Một mặt, việc sắp xếp các trang Web kết quả được thi hành theo hướng tính độ quan trọng của các trang Web đó và trình diễn chúng theo thứ tự giảm dần về độ quan trọng cao (Chương 6). Mặt khác, cần tiến hành phân cụm tập các trang Web trả về và tạo nhãn cụm cung cấp tới người dùng. Quá trình này thường sử dụng các thông tin như tiêu đề trang, URL, content-type, các liên kết trong trang Web,... để tiến hành phân cụm và đưa ra tập con các kết quả tốt nhất cho người dùng. Một ví dụ về bài toán phân cụm Web các kết quả trả về của máy tìm kiếm tiếng Việt được giới thiệu ở Chương 7.

• Khai phá cấu trúc Web

Nhờ vào các kết nối giữa các văn bản siêu liên kết, World Wide Web chứa đựng nhiều thông tin hơn so với tập các văn bản nội dung trang Web. Chẳng hạn, số lượng liên kết trả lời một trang Web được coi như một chỉ số về mức độ quan trọng của trang Web đó; đồng thời, các liên kết đi từ một trang Web chỉ dẫn rắng, các trang dịch có nội dung liên quan tới các chủ đề đang được đề cập trong trang hiện tại. Khai phá cấu trúc Web là các quá trình xử lý, nhằm rút ra các tri thức từ cách tổ chức và liên kết giữa các tham chiếu của các trang Web.

Một nội dung nghiên cứu quan trọng cần được quan tâm trong khai phá cấu trúc Web là mô hình sinh đồ thị Web. Pierre Baldi và cộng sự [BFS03] đã trình bày và phân tích một số mô hình sinh đồ thị Web và các mạng liên quan. Trong các mô hình này, một thuộc tính căn bản là luật số lớn.

• Khai phá sử dụng Web

Khai phá sử dụng Web hay khai phá hồ sơ Web (web log mining) là việc xử lý để lấy ra các thông tin hữu ích trong các hồ sơ truy cập Web. Thông thường, các Web server thường ghi lại và tích lũy các dữ liệu về các tương tác của người dùng mỗi khi nó nhận được một yêu cầu truy cập. Việc phân tích các hồ sơ truy cập Web của các Web site khác nhau sẽ dự đoán các tương tác của người dùng khi họ tương tác với Web cũng như tìm hiểu cấu trúc của Web, từ đó cải thiện các thiết kế của các hệ thống liên quan. Tương tự như khai phá cấu trúc Web, mô hình ứng xử của người dùng trên Internet và mô hình sử dụng Web cần được quan tâm đề cập [BFS03].

Có hai xu hướng chính trong khai phá sử dụng Web là *phân tích các mẫu truy cập* (General Access Pattern Tracking) và *phân tích các xu hướng cá nhân* (Customized Usage tracking).

– *Phân tích các mẫu truy cập*: Phân tích các hồ sơ Web để biết được các mẫu và các xu hướng truy cập. Các phân tích này có thể giúp cấu trúc lại các site trong các phân nhóm hiệu quả hơn, hay xác định các vị trí quảng cáo hiệu quả nhất, cũng như gắn các quảng cáo sản phẩm nhất định cho những người dùng nhất định để đạt được hiệu quả cao nhất....

– *Phân tích các xu hướng cá nhân*: Mục đích là để chuyên biệt hóa các Web site cho các lớp đối tượng người dùng. Các thông tin được hiển thị, độ sâu của cấu trúc site và định dạng của các tài nguyên, tất cả đều có thể chuyên biệt hóa một cách tự động cho mỗi người dùng theo thời gian dựa trên các mẫu truy cập của họ.

Việc phân loại khai phá Web (nội dung, liên kết, sử dụng) như mô tả tại Hình 2.5 mang tính tương đối. Trong thực tiễn, ứng dụng khai phá Web được tích hợp từ một hoặc một vài loại khai phá Web nói trên.

2.2.4. Số bộ về các bài toán khai phá Web

Mỗi bài toán của khai phá Web đều cần bao hàm tính đặc thù của Web. Có thể phân chia các bài toán khai phá Web thành hai loại là:

- *Các bài toán chung của khai phá dữ liệu Text với việc bổ sung các yếu tố của miền ứng dụng dữ liệu Web*

Các bài toán Phân lép, Phân cụm và Phân đoạn trong khai phá Web tương tự như các bài toán tương ứng trong khai phá Text, song có bổ sung đặc thù của Web là nội dung trang Web lại có siêu liên kết hướng tới các trang Web khác. Trong nhiều trường hợp, các bài toán này còn được làm phù hợp với môi trường online của Internet: chẳng hạn như, bài toán phân cụm, phân lớp đối với một tập các trang Web là kết quả trả về từ máy tìm kiếm.

Các bài toán Phái hiện ràng buộc (Associating) và Phái hiện luật kết hợp (Association Rule) liên quan tới không chỉ các yếu tố trong nội dung văn bản mà còn liên quan tới các yếu tố đặc thù của Web: chẳng hạn, sự ràng buộc của các trang Web, sự ràng buộc giữa người sử dụng với các trang Web mà họ thường quan tâm trong phiên làm việc, hoặc sự ràng buộc giữa nhóm người sử dụng với tập các trang Web mà mọi thành viên trong nhóm cùng quan tâm.

- *Các bài toán khai phá dữ liệu mang tính đặc thù của Web*

Bài toán Dự báo (Predicting) khai thác yếu tố thời gian liên quan tới thời điểm xuất hiện trang Web để có thể dự báo xu thế về các đặc trưng như nội dung, về cấu trúc và hình thức trình bày của các trang Web xuất hiện

trong thời gian tới. Khai phá xâu sử dụng Web trong một phiên làm việc cũng là một bài toán nhận được sự quan tâm của nhiều nhà khoa học.

Các bài toán Dự báo nhu cầu (Response prediction) và Đánh giá khách hàng khai thác Web (Customer valuation) liên quan đến đối tượng sử dụng CSDL trang Web.

Một số bài toán điển hình nhất của khai phá dữ liệu Web (tìm kiếm, phân cụm, phân lớp, trích chọn thông tin) sẽ được giới thiệu trong các chương tiếp theo.

2.2.5. Khuynh hướng khai phá Web

Khuynh hướng khai phá Web có xuất phát điểm từ quá trình phát triển của lĩnh vực này, trong đó có nguyên nhân từ việc ứng dụng rộng rãi và hiệu quả của lĩnh vực khai phá Web trong thực tiễn kinh tế – xã hội. Quan hệ hữu cơ mật thiết của nghiên cứu khoa học và triển khai ứng dụng, việc chuyên hoá nhanh chóng và hiệu quả từ các kết quả nghiên cứu lý thuyết với nền tảng toán học tối thành việc ra đời và phát triển thành công nhiều ứng dụng của khai phá dữ liệu Web là xu thế phát triển của khai phá Web hiện nay.

Theo Sergey Ananyan [Ana01], kích cỡ thị trường của khai phá dữ liệu văn bản tăng trưởng theo đơn vị triệu người tiến hành hoạt động đó. Đồng thời, hàng tỷ người dùng Internet được thu hút bởi các tiện ích khai phá dữ liệu văn bản khi làm việc với các hệ thống tìm kiếm và các ứng dụng phong phú khác.

Nhiều sản phẩm điển hình về khai phá dữ liệu văn bản, chẳng hạn như các sản phẩm TextAnalyst*, Textracter™, WebAnalyst và PolyAnalyst,... của Công ty Megaputer Intelligence, Inc⁽¹⁾ [Kis95, Ana01], WebFountain của IBM,... đã chiếm được thị phần không nhỏ khi được tích hợp vào nhiều ứng dụng của các doanh nghiệp phát triển phần mềm.

Thành công kỳ diệu của các hệ thống *máy tìm kiếm* (search engine) (điển hình như máy tìm kiếm Google) đã khẳng định tính cần thiết của các nghiên cứu về khai phá dữ liệu văn bản. Hệ thống tìm kiếm Google do hai nghiên cứu sinh tại Stanford University là Sergey Brin và Lawrence Page sáng lập vào năm 1998 [BP98]. Đến năm 2006, hệ thống này đã được đánh giá có giá trị lên tới hàng chục tỷ đô la Mỹ. Hiện nay, thị trường cung cấp dịch vụ thông qua các máy tìm kiếm đã trở thành một thị trường rất lớn, mà theo dự báo sẽ đạt tới hàng trăm tỷ đô la Mỹ vào năm 2010⁽²⁾⁽³⁾, và vì vậy, nó đã thu hút sự quan tâm của nhiều công ty CNTT hàng đầu trên thế giới.

⁽¹⁾ Megaputer Intelligence, Inc. (<http://www.megaputer.com>)

⁽²⁾ http://www.keynote.com/news_events/releases_2006/06jan18.html: Google Poses Strong challenge to Leader Baidu in China, Reports Keynote

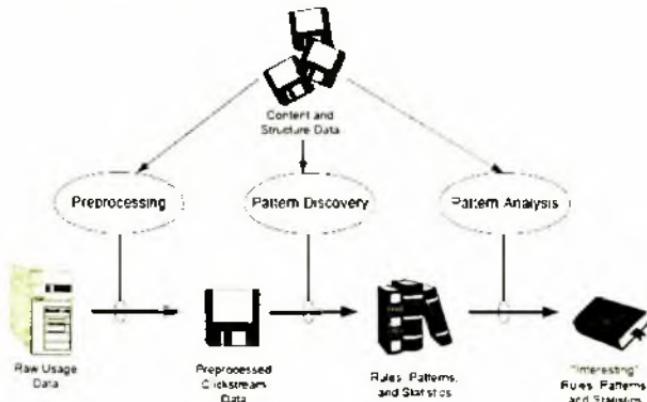
⁽³⁾ <http://searchenginewatch.com/searchday/article.php/3575926>: The State of Search Engine Marketing

Vấn đề bandedia hóa trên Internet luôn song hành với khai phá dữ liệu văn bản Web. Bandedia hóa trên Internet liên quan tới nhiều lĩnh vực nghiên cứu, trong đó liên quan mật thiết nhất tới các lĩnh vực *xử lý ngôn ngữ tự nhiên, an toàn và mã hoá thông tin* (Information security and cryptography)⁽¹⁾.

Trong [Cha03], Soumen Chakrabarti giới thiệu xu hướng phát triển của khai phá Web trong tương lai liên quan tới trích chọn thông tin: các vấn đề của xử lý ngôn ngữ tự nhiên (Mạng từ vựng và Ontology; Từ và phân đoạn; Phân tích cú pháp và trình diễn tri thức); hệ thống hỏi – đáp và các vấn đề về mô tả, về cá thể hóa và kết hợp nhóm.

2.3. Khai phá sử dụng Web

Khai phá sử dụng Web là một trong ba loại hình chính của khai phá Web. Quá trình khai phá sử dụng Web được mô tả trong Hình 2.6 [Coo00]. Các vấn đề cơ bản cần quan tâm trong khai phá sử dụng Web là nguồn tài nguyên dữ liệu và mô hình dữ liệu, tiền xử lý dữ liệu, phát hiện mẫu và phân tích mẫu [Pia06, Coo00].



Hình 2.6. Quá trình khai phá sử dụng Web [Coo00]

Robert Walker Cooley [Coo00] giới thiệu các nội dung cụ thể thuộc về các vấn đề cần quan tâm ở trên là:

- *Nguồn dữ liệu:* có ở các logfile (tại máy phục vụ, máy khách, máy trung gian) hoặc có trong các CSDL khách hàng.

⁽¹⁾ The OpenNet Initiative (<http://www.opennetinitiative.net>)

– *Mô hình (thực thể) dữ liệu*: người sử dụng, khung nhìn trang Web, file trang Web, trình duyệt, phục vụ Web, phục vụ nội dung, phiên người sử dụng, phiên phục vụ....

– *Tiền xử lý dữ liệu*: dữ liệu cấu trúc, dữ liệu nội dung, xử lý văn bản, rút gọn đặc trưng và tiền xử lý dữ liệu đối với mô hình dữ liệu.

– *Phát hiện mẫu*: bài toán phát hiện mẫu ở đây bao gồm hầu hết các bài toán khai phá dữ liệu điển hình là phân tích thống kê, phát hiện luật kết hợp, phân cụm, phân lớp, mẫu tuân tự và mô hình phụ thuộc.

Vẫn đề đầu tiên mang đặc thù của khai phá sử dụng Web, các vấn đề còn lại đều mang nội dung chung của quá trình phát hiện trí thức. Nguồn tài nguyên dữ liệu bao gồm các file biên bản sử dụng Web tại máy chủ Web, tại máy khách và các vị trí trung gian (khai phá mẫu truy cập) và các CSDL người dùng. Như đã được giới thiệu, khai phá sử dụng Web được phân thành khai phá mẫu truy cập và khai phá xu hướng cá nhân (hoặc cá nhân hoá việc sử dụng Web). Nhiều ứng dụng khai phá Web đã tích hợp hai loại khai phá sử dụng Web này.

2.3.1. Phân tích mẫu truy nhập Web

Khác với bài toán phân tích xu hướng cá nhân trong khai phá sử dụng Web sẽ được trình bày ở mục tiếp theo quan tâm tới cá nhân người dùng hoặc một nhóm người dùng; bài toán phân tích mẫu truy cập Web quan tâm đến khai phá những mẫu có tính phổ dụng của tập người dùng khi truy nhập Web, có thể coi tập người dùng là đối tượng phục vụ của bài toán phân tích mẫu truy nhập Web.

Thông tin truy nhập của người dùng được Web server ghi nhận lại trong Web server log theo mẫu log chung (Common Log Format: CLF), hoặc mẫu log chung mở rộng (Extended CLF: ECLF). Thông tin được lưu giữ liên quan đến phiên truy nhập của người dùng, thường bao gồm các thông tin là địa chỉ IP của máy người dùng (cho biết tên máy chủ người dùng), thời điểm bắt đầu truy nhập, nhu cầu người dùng (phương thức, địa chỉ Web, giao thức), mã trạng thái đáp ứng yêu cầu (bình thường, truy nhập không hoàn chỉnh, không tìm thấy...), kích thước dữ liệu truy nhập, chỉ dẫn về địa chỉ URI, đi tới yêu cầu này, công cụ truy nhập Web của người dùng. Hình 2.7 trình bày một số bản ghi ECLF và một số trường dữ liệu log. Theo hướng tiếp cận này, hệ thống khai phá sử dụng Web không đòi hỏi thông tin về người dùng. Như đã trình bày ở trên, đơn vị dữ liệu xác định một lần sử dụng Web chính là một phiên làm việc người dùng, dữ liệu về phiên làm việc đã được ghi trong logfile của hệ thống.

| Client Address | Userid | Time | Method/URL/Protocol | Status/Size | Referrer | Useragent |
|----------------|--------|------------------------------|---------------------------|-------------|------------|---|
| 123.456.78.9 | -- | [28/Apr/1998 03:04:41 -0500] | GET /index.html HTTP/1.0 | 200 3200 | - | Microsoft Internet Explorer/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:05:34 -0500] | GET /index.html HTTP/1.0 | 200 2050 | index.html | Microsoft Internet Explorer/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:06:02 -0500] | POST /getbinary1 HTTP/1.0 | 200 5688 | index.html | Microsoft Internet Explorer/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:06:58 -0500] | GET /index.html HTTP/1.0 | 200 3200 | - | Microsoft Internet Explorer/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:07:42 -0500] | GET /index.html HTTP/1.0 | 200 2050 | index.html | Mozilla/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:09:50 -0500] | GET /index.html HTTP/1.0 | 200 1820 | index.html | Mozilla/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:10:02 -0500] | GET /index.html HTTP/1.0 | 200 2270 | index.html | Mozilla/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:10:45 -0500] | GET /index.html HTTP/1.0 | 200 9430 | index.html | Mozilla/4.0 (Win95_1) |
| 123.456.78.9 | -- | [28/Apr/1998 03:12:23 -0500] | GET /index.html HTTP/1.0 | 200 7220 | index.html | Mozilla/4.0 (Win95_1) |
| 206.49.78.2 | -- | [28/Apr/1998 05:05:22 -0500] | GET /index.html HTTP/1.0 | 200 3200 | - | Mozilla/3.0 (Win95_1) |
| 206.49.78.3 | -- | [28/Apr/1998 05:06:03 -0500] | GET /index.html HTTP/1.0 | 200 1680 | index.html | Mozilla/3.0 (Win95_1) |

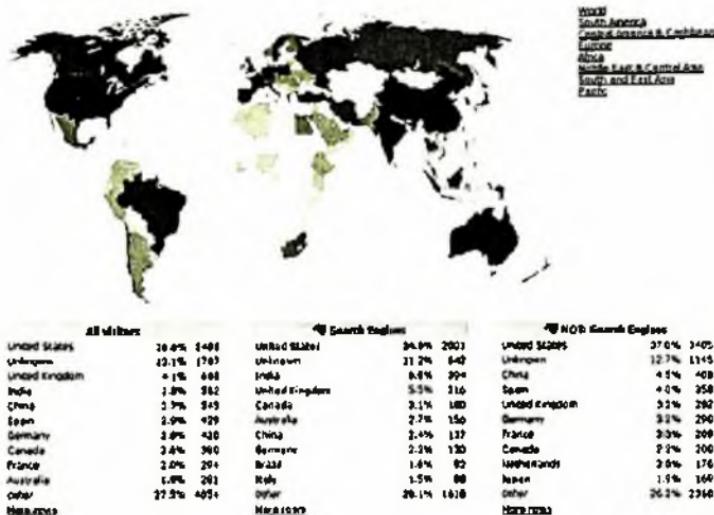
Hình 2.7. Một ví dụ về mẫu log chung [Coo00]

Từ dữ liệu được lưu giữ tại Web server log (và các log khác), các nhà khai phá dữ liệu có thể khai tạo các CSDL, các kho dữ liệu và tiến hành nhiều bài toán khai phá dữ liệu như tìm mối quan hệ nội dung giữa các trang Web (thông qua mối liên hệ giữa địa chỉ URL tới và trang Web), thói quen truy nhập các vùng người dùng, xu hướng truy nhập người dùng,... Như được trình bày trong [Pia06], ngoài phương thức sử dụng các tiện ích của hệ điều hành để phân tích Weblog, khai phá sử dụng Web còn sử dụng một số công cụ hỗ trợ khác, chẳng hạn như gawk, phiên bản phần mềm tự do (GNU)⁽¹⁾ của công cụ awk (mang tên những người thiết kế nó là Alfred V. Aho, Peter J. Weinberger và Brian W. Kernighan). Hình 2.8 trình bày một kết quả phân tích Weblog [Pia06]. Mẫu được khai phá từ logfile được sử dụng theo nhiều phương án và một phương án sử dụng diễn hình là dẫn dắt online người sử dụng. Trên cơ sở thành viên truy nhập hiện tại của người dùng, căn cứ vào các luật truy cập (tri thức) đã được phát hiện, hệ thống dẫn dắt người dùng phù hợp với luật truy cập (cũng được coi là phù hợp với tư duy hiện thời của người dùng).

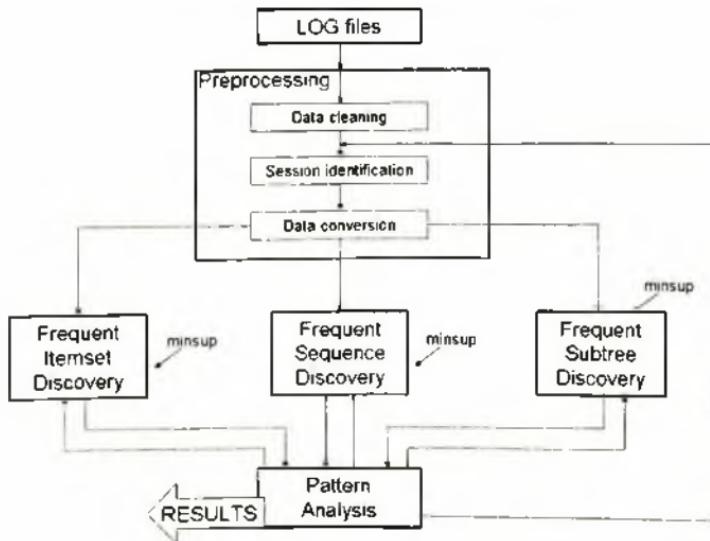
Renáta Ivánesy và István Vajk [IV06] nghiên cứu về khai phá mẫu tuân tự các logfile đối với tập trang Web, dãy trang Web và đô thị Web. Hình 2.9 trình bày sơ đồ khai phá sử dụng Web phát hiện ba loại mẫu nói trên. Các tác giả sử dụng thuật toán Itemset Code để khai phá tập trang Web, thuật toán SM-Tree để khai phá dãy trang Web và thuật toán PD-Tree để khai phá đô thị Web.

(1) <http://www.gnu.org/software/gawk/>

Connections with Most Visitors



Hình 2.8. Kết quả phân tích nguồn truy nhập theo các quốc gia từ trang Web KDnuggets, tuần 21-27/5/2006 [Pla06]



Hình 2.9. Sơ đồ khai phá mẫu tuần tự từ logfile [IV06]

Tồn tại hướng nghiên cứu khai phá sử dụng Web không chỉ quan tâm tới việc phát hiện các tri thức (diễn hình là luật kết hợp) tiềm ẩn trong nội dung các trang Web được lưu giữ tạm thời tại Web server, mà còn cho phép nhận diện được hành vi sử dụng Web của người dùng.

Thuật toán khai phá sử dụng Web còn cần phải quan tâm tới một số vấn đề liên quan khác. Chẳng hạn, cần thi hành các giải pháp lưu trữ các trang Web "một cách có lợi nhất" trong hoàn cảnh hạn chế về dung lượng bộ nhớ được quy định cho vùng Web-cache. Một số giải pháp điều phối cache như GDSIZE, GDSF, LFUDA, LRU,... được thi hành nhằm loại bỏ các trang Web "không hữu ích" ở vùng Web-cache; các chiến lược này hoạt động tương tự như các chiến lược loại bỏ trang bộ nhớ trong hệ điều hành. Nghiên cứu của Qiang Yang và Haining Henry Zhang [YZ03] khẳng định, chiến lược caching GDSF (Greedy-Dual-Size-Frequency) có nhiều lợi thế, vì vậy là một phương án lựa chọn tốt.

Một số kết quả khai phá sử dụng Web được giới thiệu chi tiết trong [Ber00, MS99, Pia06, Coo00, BFS03]. Trong [BFS03], Pierre Baldi và cộng sự đã trình bày các mô hình và nền tảng ứng xử của người dùng trên Web. Loại mô hình diễn hình nhất là mô hình Markov với giả thiết Markov bậc 1. Theo các tác giả, dữ liệu tại máy khách hàng, chẳng hạn, hoạt động truy nhập Web của người dùng được các phần mềm máy khách ghi nhận cũng là nguồn tài nguyên rất có ích cho khai phá sử dụng Web.

Khai phá mẫu truy cập theo luật kết hợp:

Loại mẫu diễn hình nhất trong phân tích mẫu truy nhập Web là luật kết hợp. Phản này giới thiệu sơ bộ về luật kết hợp và thuật toán Apriori khai phá luật kết hợp. Thuật toán Apriori và các biến thể của nó được sử dụng rộng rãi để giải quyết bài toán phát hiện luật kết hợp từ logfile, chẳng hạn như [IV06, FL03, MD01].

• Luật kết hợp

Cho một tập mục $I = \{i_1, i_2, \dots, i_n\}$, mỗi phần tử thuộc I được gọi là *một mục*. Mục còn được gọi là *thuộc tính* và I cũng được gọi là *tập các thuộc tính*. Mỗi tập con trong I được gọi là *một tập mục*, số lượng các phần tử trong tập mục được gọi là *độ dài của tập mục*. Cho một CSDL giao dịch $D = \{t_1, t_2, \dots, t_m\}$, trong đó mỗi t_j là một giao dịch và là một tập con thuộc I . Về mặt thực tiễn, mỗi giao dịch t_j là một danh sách các mục (tên mặt hàng) trong giao dịch (phiếu giao dịch hàng). Ở đây số lượng giao dịch (lực lượng) của D ký hiệu là $|D|$ hoặc $\text{card}(D)$ là rất lớn.

Cho X, Y là hai tập mục (hai tập con của I). Luật kết hợp được ký hiệu là $X \rightarrow Y$, trong đó $X \cap Y = \emptyset$, thể hiện mối ràng buộc của tập mục Y theo tập mục X theo nghĩa " X kéo theo Y " ra sao về sự xuất hiện trong các giao

dịch. Tập mục X được gọi là xuất hiện trong giao dịch t nếu như $X \subseteq t$, và có thể được diễn giải là "mọi tên mặt hàng trong X đều xuất hiện trong phiếu giao dịch t".

Giá trị của luật kết hợp $X \rightarrow Y$ được thể hiện thông qua hai độ đo là độ hỗ trợ $\text{supp}(X \rightarrow Y)$ và độ tin cậy $\text{conf}(X \rightarrow Y)$:

- Độ hỗ trợ của một tập mục X (ký hiệu $\text{supp}(X)$) được định nghĩa là:

$$\text{supp}(X) = |\{t \in D : X \subseteq t\}| / |D|$$

- $\text{supp}(X \rightarrow Y) = \text{supp}(XY)$ là tỷ lệ giao dịch có chứa $(X \cup Y)$ trong tập D.

- $\text{conf}(X \rightarrow Y) = \text{supp}(X \rightarrow Y) / \text{supp}(X)$ là tỷ lệ tập giao dịch có chứa $(X \cup Y)$ so với tập giao dịch có chứa X.

Từ định nghĩa ta có: $0 \leq \text{supp}(X \rightarrow Y) \leq 1$ và $0 \leq \text{conf}(X \rightarrow Y) \leq 1$. Theo quan niệm xác suất, độ hỗ trợ là xác suất xuất hiện tập mục $X \cup Y$, còn độ tin cậy là xác suất có điều kiện xuất hiện Y khi đã xuất hiện X.

Luật kết hợp $X \rightarrow Y$ được coi là một "tri thức" ("mẫu có giá trị") nếu xảy ra đồng thời

$$\text{supp}(X \rightarrow Y) \geq \text{minsup} \text{ và } \text{conf}(X \rightarrow Y) \geq \text{minconf}$$

với minsup và minconf là hai ngưỡng cho trước. Tập mục X có độ hỗ trợ qua ngưỡng minsup ($\text{supp}(X) \geq \text{minsup}$) được gọi là *tập phô biến*.

Mục tiêu của khai phá luật kết hợp là tìm ra tất cả các luật kết hợp có giá trị. Để giải quyết bài toán trên, trước hết cần tìm ra mọi tập phô biến, mỗi tập phô biến đóng vai trò của tập XY trong luật kết hợp $X \rightarrow Y$.

• Thuật toán Apriori

Thuật toán Apriori là một thuật toán điển hình tìm luật kết hợp [WKQ08]. Thuật toán dựa theo tính chất Apriori phát biểu rằng: "tập con bất kỳ của một tập phô biến cũng là một tập phô biến", tính chất này hiển nhiên đúng. Nội dung quan trọng nhất của thuật toán Apriori là tìm ra được tất cả các tập phô biến có thể có trong D. Thuật toán hoạt động theo quy tắc quy hoạch động, nghĩa là từ các tập $F_i = \{c_i \mid c_i \text{ tập phô biến}, |c_i| = i\}$ gồm mọi tập mục phô biến có độ dài i với $1 \leq i \leq k$, đi tìm tập F_{k+1} gồm mọi tập mục phô biến có độ dài $k + 1$. Trong thuật toán, các tên mục i_1, i_2, \dots, i_n ($n = |D|$) được sắp xếp theo một thứ tự cố định (thường được đánh chỉ số 1, 2, ..., n). Mô tả thuật toán Apriori như sau:

Thuật toán Apriori [WKQ08]:

Input: – Cơ sở dữ liệu giao dịch D = {t | t giao dịch}

– Độ hỗ trợ tối thiểu $\text{minsup} > 0$

Output: – Tập hợp tất cả các tập phô biến

```

0 mincount = minsup * |D|;
1 F1 = {các tập phô biến có độ dài 1}
2 for (k=1; Fk ≠ ∅; k++) do begin
3 Ck+1 = apriori-gen (Fk); // sinh mới ứng viên độ dài k+1
4 for t ∈ D do begin
5 Ct = {c ∈ Ck+1 | c ⊆ t}; //mọi ứng viên chứa trong t
6 for c ∈ Ct do
7 c.count++;
8 end
9 Fk+1 = {c ∈ Ck+1 | c.count ≥ mincount};
10 end
11 Answer  $\cup_k F_k$ ;

```

Thủ tục con Apriori-gen có nhiệm vụ sinh ra các tập mục ứng viên có độ dài k + 1 từ F_k (các tập phô biến có độ dài k) được thi hành qua hai bước chính như sau:

– *Bước nối*: Sinh các tập mục R_{k+1} là ứng viên tập phô biến có độ dài k + 1 bằng cách kết hợp hai tập phô biến P_k và Q_k có độ dài k và trùng nhau ở k - 1 mục đầu tiên:

$$R_{k+1} = P_k \cup Q_k = \{i_1, i_2, \dots, i_{k-1}, i_k, i_k\} \text{ với} \\ P_k = \{i_1, i_2, \dots, i_{k-1}, i_k\} \text{ và } Q_k = \{i_1, i_2, \dots, i_{k-1}, i_k\}$$

trong đó $i_1 \leq i_2 \leq \dots \leq i_{k-1} \leq i_k \leq i_k$.

– *Bước tia*: Giữ lại tất cả các R_{k+1} thỏa mãn tính chất Apriori ($\forall X \subseteq R_{k+1} \text{ và } |X| = k \Rightarrow X \in F_k$), nghĩa là đã loại (tia) bỏ đi mọi ứng viên R_{k+1} không đáp ứng tính chất này.

Trong mỗi bước k, thuật toán Apriori đều phải duyệt CSDL D. Khi động, duyệt D để có được F₁. Các bước k sau đó, duyệt D để tính số lượng giao dịch t thỏa mãn từng ứng viên c của C_{k+1} (mỗi giao dịch t chỉ xem xét một lần cho mọi ứng viên c).

Kết quả của thuật toán Apriori là tập F = F₁ ∪ F₂ ∪ ... ∪ F_k, trong đó k là số được xác định qua vòng lặp từ 2 đến 10 của thuật toán.

Sau đó, $\forall c \in F$ cho c đóng vai trò như X ∪ Y của luật kết hợp ($X \rightarrow Y$) thực hiện việc tách c thành hai tập mục con rời nhau X và Y (c = X + Y) và tính độ tin cậy $conf(X \rightarrow Y) = supp(c)/supp(X) = c.count/X.count$.

Ví dụ: Trong [IV06], Renáta Ivánčsy và István Vajk sử dụng các thuật toán biến thể từ Apriori để phát hiện luật kết hợp và luật tuần tự từ dữ liệu tại logfile (Hình 2.10).

| | | | | |
|---|-------------|--------|----------------------------------|-------|
| opinion & misc & travel | → on-air | 90.26% | misc → local | 2.07% |
| news & misc & business & bbs | → frontpage | 90.24% | frontpage → frontpage → sports | 2.02% |
| news & business & sports & bbs | → frontpage | 90.00% | local → frontpage | 1.63% |
| news & misc & business & sports | → frontpage | 89.68% | on-air → misc → on-air | 1.72% |
| news & tech & living & business & sports | → frontpage | 89.00% | on-air → frontpage | 1.69% |
| frontpage & tech & living & business & sports | → news | 87.87% | on-air → news | 1.51% |
| frontpage & opinion & living & sports | → news | 87.81% | news → frontpage → news | 1.49% |
| frontpage & tech & opinion & living | → news | 87.80% | local → news | 1.46% |
| frontpage & tech & on-air & business & sports | → news | 87.58% | frontpage → frontpage → business | 1.35% |
| news & misc & sports & bbs | → frontpage | 87.56% | news → sports | 1.33% |
| news & tech & on-air & business & sports | → frontpage | 87.43% | health → local | 1.16% |
| news & living & business & sports | → frontpage | 87.18% | misc → frontpage → frontpage | 1.16% |
| news & business & sports & bbs | → frontpage | 86.70% | on-air → local | 1.15% |
| misc & living & travel | → on-air | 86.55% | misc → on-air → misc | 1.15% |
| tech & living & sports & bbs | → frontpage | 86.52% | frontpage → frontpage → living | 1.14% |
| tech & business & sports & bbs | → frontpage | 86.40% | local → frontpage → frontpage | 1.13% |
| news & misc & living & business | → frontpage | 86.22% | health → misc | 1.12% |
| on-air & business & sports & bbs | → frontpage | 86.22% | misc → on-air → on-air | 1.10% |
| news & tech & misc & bbs | → frontpage | 86.18% | local → misc → local | 1.09% |
| on-air & misc & business & sports | → frontpage | 86.15% | misc → news | 1.06% |
| tech & misc & travel | → on-air | 86.08% | news → living | 1.06% |
| tech & living & business & sports | → frontpage | 86.08% | on-air → misc → on-air → misc | 1.00% |
| news & living & sports & bbs | → frontpage | 86.06% | | |
| misc & business & sports | → frontpage | 86.06% | | |
| frontpage & tech & opinion & sports | → news | 86.06% | | |
| news & opinion & living & sports | → frontpage | 86.06% | | |
| misc & business & travel | → on-air | 86.06% | | |
| news & tech & misc & business | → frontpage | 86.03% | | |
| misc & business & bbs | → frontpage | 86.03% | | |
| tech & living & sports & bbs | → frontpage | 86.03% | | |
| local & misc & business & sports | → news | 86.03% | | |
| news & opinion & business & bbs | → frontpage | 86.03% | | |
| news & misc & living & sports | → frontpage | 86.03% | | |
| news & on-air & business & sports | → frontpage | 86.01% | | |

a)

b)

Hình 2.10. Kết quả phát hiện luật kết hợp và luật tuần tự từ logfile [IV06]

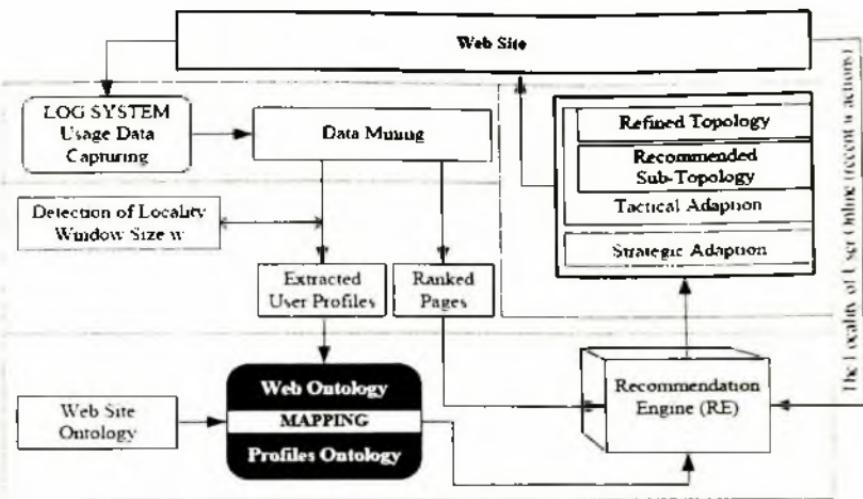
2.3.2. Phân tích xu hướng cá nhân

Như đã giới thiệu, phân tích xu hướng cá nhân nhằm tối tính cá nhân hoá, vì vậy dữ liệu cũng cần có tính cá nhân hoá, hoặc ở logfile ở máy khách, hoặc ở CSDL khách hàng, hoặc dữ liệu thu nhận online với khách hàng. Phần này giới thiệu một số nội dung phân tích xu hướng cá nhân không có CSDL khách hàng và các hệ thống tư vấn khách hàng.

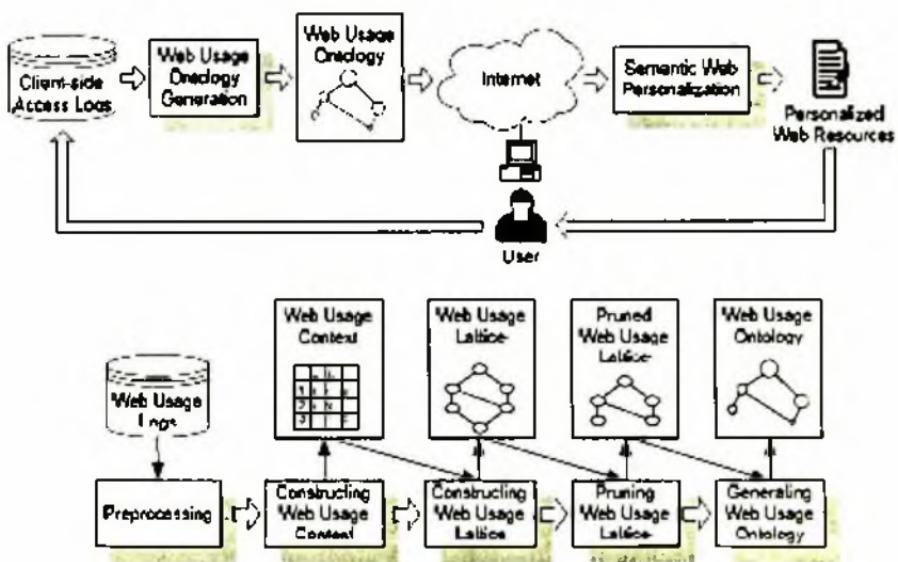
- *Phân tích xu hướng cá nhân từ máy khách*

Hình 2.11 trình bày hệ thống khai phá sử dụng Web có sử dụng dữ liệu người dùng ở máy khách của Tarmo Robal và Ahto Kalja [RK07]. Thông tin người dùng được các phần mềm hệ thống tại máy khách được trích chọn và dữ liệu sử dụng hệ thống tư vấn cho từng người dùng cụ thể.

Trong [ZHF05], Baoyao Zhou và cộng sự đề xuất hệ thống dựa theo logfile xây dựng các ontology sử dụng Web để tư vấn người sử dụng hệ thống (Hình 2.12).



Hình 2.11. Sinh tư vấn dựa trên trích chọn tiêu sử dụng [RK07]



Hình 2.12. Hệ thống khai phá sử dụng Web tư vấn hướng cá nhân: Kiến trúc hệ thống (trên) và sinh ontology sử dụng Web (dưới) [ZHF05]

Một số thông tin hành vi người dùng cũng được một số hệ thống khai thác nhằm khai phá chuỗi hành vi của người dùng và từ đó có dự báo hành vi tiếp theo của người dùng để chuẩn bị sẵn các tài nguyên phù hợp với thao tác tiếp theo của người dùng.

Hệ thống tư vấn khách hàng là một ứng dụng điển hình của khai phá Web trong hoạt động tư vấn khách hàng. Trong hệ thống này, CSDL khách hàng lưu trữ về thông tin khách hàng đăng ký. Thông tin có được từ CSDL khách hàng cho phép:

- Kết nối được các phiên làm việc của cùng một khách hàng và vì vậy, tạo thuận tiện trong việc khảo sát mỗi quan hệ khách hàng – mặt hàng.
- Kết nối được nhóm khách hàng có cùng một (hoặc một nhóm) thuộc tính như giới, độ tuổi, nghề nghiệp, thu nhập.... Trong một số hệ thống, một số thuộc tính mô tả thị hiếu của khách hàng cũng được đưa vào CSDL.

Trong [BFS03], Pierre Baldi và cộng sự dành một chương trình bày về các mô hình và ứng dụng thương mại trên Web. Dữ liệu về khách hàng có tại máy phục vụ, tại máy khách hoặc tại các vị trí trung gian (chẳng hạn, tại vị trí cung cấp dịch vụ Internet như tại máy phục vụ proxy). Ứng dụng điển hình là các hệ thống tư vấn khách hàng tự động. Lọc cộng tác là cách tiếp cận chính yếu nhất trong hệ thống, theo đó, hệ thống sử dụng các chọn lựa của các cá nhân trong quá khứ để dự báo sự chọn lựa mới và đưa ra một tư vấn mới. Hai mô hình lọc điển hình theo cách tiếp cận này được khảo sát, đó là mô hình lọc cộng tác người láng giềng gần nhất và mô hình lọc cộng tác dựa trên mô hình.

Tư tưởng của mô hình lọc cộng tác người láng giềng gần nhất cũng rất đơn giản. Đối với một người dùng a , trước hết tìm ra tập các người dùng "tương tự với a " trong dữ liệu lọc, sau đó sử dụng chọn lựa đối với một thuộc tính của các người dùng tương đồng với a để dự báo chọn lựa của người dùng a đối với thuộc tính đó. Trong mô hình lọc cộng tác người láng giềng gần nhất, cần giải quyết các bài toán xác định trọng số trong phương trình dự báo, thu gọn số chiều bài toán, tính toán và phân cụm.

Trong mô hình lọc cộng tác dựa trên mô hình, trên cơ sở thừa kế các mẫu lựa chọn của người dùng trong quá khứ, cần xây dựng không trực tuyến một mô hình kỳ vọng về mối quan hệ giữa các mục hàng. Sau đó, mô hình này được sử dụng trực tuyến để dự báo sự chọn lựa của người dùng mới. Mô hình hướng tới yêu cầu tính toán thời gian thực, trong đó, với một mô hình đã được xây dựng, thời gian dự báo không phụ thuộc vào số lượng khách hàng có trong CSDL. Mô hình lọc cộng tác dựa trên mô hình được phân loại thành mô hình mật độ kết nối và mô hình phân bố có điều kiện. Tồn tại một số mô hình trộn giữa lọc nội dung và lọc cộng tác.

2.4. Khai phá cấu trúc Web

Theo Pierre Baldi và cộng sự [BFS03], Internet được nhìn nhận dưới dạng đồ thị theo nhiều khung nhìn khác nhau. Theo khung nhìn vật lý, đồ thị có các đỉnh là các đối tượng vật lý thực sự và các cung là các đường vật lý liên kết các đỉnh. Theo khung nhìn trừu tượng hơn, đồ thị có các đỉnh là các trang Internet và các cung là các liên kết giữa các đỉnh này. Đây cũng là khung nhìn của Khai phá cấu trúc Web (và khai phá Web). Trên Internet còn có nhiều hệ thống được nhìn nhận dưới dạng đồ thị như mạng địa chỉ e-mail, mạng blog, mạng người dùng trong một diễn đàn. Khi quan niệm Internet là một xã hội ảo, thì mọi mạng tồn tại trên Internet đều được coi là mạng xã hội.

Khai phá cấu trúc Web sử dụng các kiến trúc liên kết Web để phát hiện được mô hình về cấu trúc liên kết của Web, dựa trên kiến trúc topo của các liên kết với mô tả hoặc không [Lee05, BFS03]. Khai phá cấu trúc Web gồm hai loại cơ bản, đó là khai phá đồ thị Web và khai phá cấu trúc trang Web.

2.4.1. Khai phá đồ thị Web

Khai phá đồ thị Web là bài toán cơ bản nhất và cũng điển hình nhất trong khai phá cấu trúc Web. Đồ thị Web được coi như một ví dụ về mạng xã hội, một đối tượng nghiên cứu hiện đang rất được quan tâm. Nhắc lại, trong đồ thị Web thì trang Web là đỉnh và một trang Web có cung tới một trang Web khác khi mà trong nội dung của nó có liên kết từ sang trang Web kia. Đồ thị Web được xem xét dưới dạng có hướng hoặc vô hướng tùy thuộc vào bài toán được đặt ra.

Một bài toán kinh điển trong đồ thị Web là bài toán Tính hạng (độ quan trọng) trang Web. Hàng trang Web được sử dụng trong nhiều tình huống. Chẳng hạn, hạng trang Web được dùng để dẫn dắt đường đi trên Web, những trang có hạng cao được dẫn dắt đi thăm trước. Trong máy tìm kiếm, hạng trang Web dẫn dắt thứ tự hiển thị kết quả tìm kiếm, theo đó, trang Web có hạng cao hơn được hiển thị trước trang Web có hạng thấp hơn. Tính hạng trang Web có liên quan tới mô hình sinh trang Web trong hệ thống Web [Hav02, Hop08, BFS03].

Tính hạng còn được ứng dụng trong bài toán phát hiện địa chỉ mail spam trong mạng e-mail, theo đó các địa chỉ e-mail có hạng thấp có khả năng cao là một địa chỉ spam.

Pierre Baldi và cộng sự [BFS03] cung cấp một cái nhìn tổng quan về đồ thị Web cùng một số vấn đề cơ bản nhất trong khai phá đồ thị Web. Trong giáo trình này, một số kiến thức cơ sở toán học về đồ thị Web được trình bày ở Chương 3 và bài toán tính hạng trang Web được trình bày ở Chương 6.

2.4.2. Khai phá cấu trúc trang Web

Trang Web là một đối tượng dữ liệu bán giám sát, cấu trúc của trang Web tuân theo quy định của ngôn ngữ định dạng trang Web (chẳng hạn HTML). Khai phá cấu trúc trang Web thực hiện việc phát hiện các mẫu từ tập các kiến trúc trang Web. Đối tượng dữ liệu trong trường hợp này là khung trang Web gồm các đối tượng "thẻ" và cấu trúc giữa các thẻ này. Kết quả của khai phá cấu trúc trang Web được sử dụng để hỗ trợ các bài toán khai phá dữ liệu Web khác. Trong nhiều ứng dụng, khai phá cấu trúc trang Web được kết hợp cùng các loại khai phá Web khác, đặc biệt là khai phá nội dung trang Web. Đã có công trình nghiên cứu có giá trị về khai phá cấu trúc trang Web, chẳng hạn [ACM03, AM03, HA09, LGZ03, RGS04, WHW09].

Davi de Castro Reis và cộng sự [RGS04] nghiên cứu về bài toán trích chọn tự động tin tức trên Web theo cách tiếp cận khoảng cách cây. Các tác giả chọn cấu trúc dạng cây để biểu diễn trang Web và sử dụng độ *do chi phí chuyển đổi cây* (Tree Edit Distance) để đánh giá độ tương tự về cấu trúc của các trang Web. Với nền tảng là thuật toán RTDM (Restricted Top – Down Mapping), Davi de Castro Reis và cộng sự đề xuất mô hình trích chọn tin tức từ các công thông tin (portal) báo điện tử gồm các bước:

(1) Dùng kỹ thuật phân cụm cấu trúc để phân cụm các trang báo điện tử (độ *do khoang cách* giữa hai trang Web là chi phí chuyển đổi cây cấu trúc trang Web);

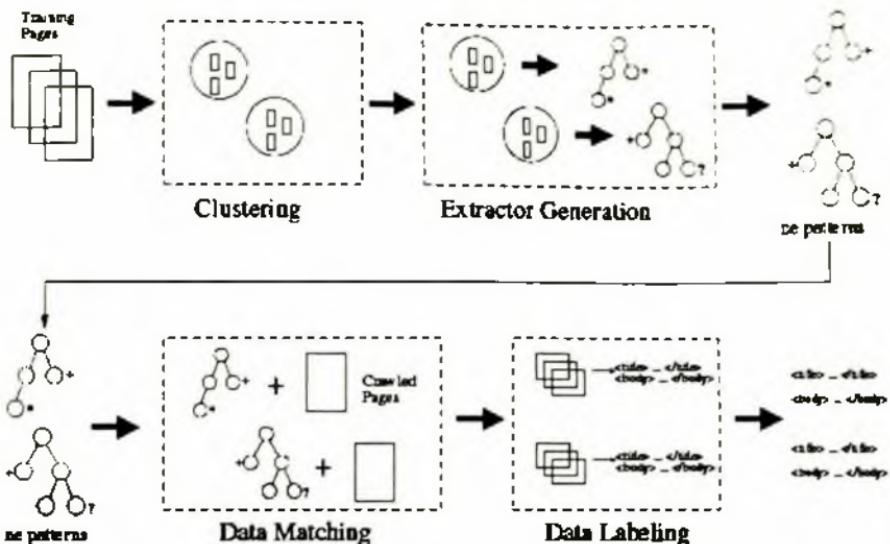
(2) Sinh các mẫu trích chọn dưới dạng cây;

(3) Dổi sánh dữ liệu (kiểm tra đánh giá mẫu trích chọn) cũng dựa theo thuật toán RTMD, trong đó định giá cho các thao tác thay thế (Vertex Replacement), chèn đỉnh (Vertex Insertion) và loại bỏ đỉnh (Vertex Removal);

(4) Áp dụng mẫu để trích chọn tin tức.

Hình 2.13 trình bày sơ đồ quá trình trích chọn tự động tin qua 4 bước nói trên. Mô hình trên đã được thi hành thành công trong sản phẩm Viennews "Các kênh báo điện tử trên thiết bị điện thoại di động thông minh" đạt giải ba cuộc thi Trí tuệ Việt Nam năm 2006.

Trong [AM03], Arvind Arasu và Hector Garcia-Molina cũng khai thác các khía cạnh tạo trang Web để tìm ra các mẫu của các trang sách được trưng bày tại các công ty bán sách trực tuyến. L. Arllota và cộng sự [ACM03], Bing Liu và cộng sự [LGZ03], P. S. Hiremath, Siddhu P. Algur [IAA09], Junfeng Wang và cộng sự [WHW09] đề xuất các mô hình khai phá cấu trúc trang Web có kết hợp với nội dung trang Web.



Hình 2.13. Các bước chính của mô hình trích chọn tự động tin trên Web [RGS04]

Câu hỏi và bài tập

1. Phân biệt sự khác nhau giữa bài toán khai phá dữ liệu thông thường, bài toán khai phá dữ liệu Text và khai phá dữ liệu Web.
2. Nêu các phương pháp khai phá dữ liệu Text và Web cũng như những ứng dụng của chúng trong thực tế.
3. Phân tích và cho ví dụ về ứng dụng thực tế của các bài toán khai phá Web.

Chương 3

MỘT SỐ KIẾN THỨC TOÁN HỌC CHO KHAI PHÁ DỮ LIỆU WEB

Trong Chương 1 chúng ta đã nghiên cứu và khẳng định rằng, lĩnh vực khai phá dữ liệu vừa có tính ứng dụng thực tiễn cao, vừa đòi hỏi nền tảng toán học mạnh.

Để cập về nền tảng lý thuyết cần cho sự phát triển ngành khoa học máy tính trong giai đoạn bùng nổ Internet, John E. Hopcroft nhấn mạnh các nội dung về đồ thị ngẫu nhiên (Random Graph), chuyên dịch pha (Phase Transitions), các thành phần không lồ (Giant Components), phân tích phổ (Spectral Analysis), hiện tượng thế giới nhỏ (Small-World phenomena), đồ thị tăng trưởng (grown graphs) [Hop07]. Có thể thấy lý thuyết đồ thị và lý thuyết xác suất được John E. Hopcroft coi như nền tảng kiến thức vững chắc cho việc nghiên cứu và phát triển lĩnh vực khoa học máy tính. Trong khai phá dữ liệu Web, lý thuyết đồ thị và lý thuyết xác suất còn mang một ý nghĩa lớn hơn, đặc biệt là đối với việc phát triển các mô hình và giải pháp khai phá dữ liệu Web. Hầu hết các mô hình và giải pháp khai phá Web là trực tiếp, hoặc gián tiếp có nền tảng dựa trên lý thuyết đồ thị và lý thuyết xác suất [BFS03, Zhu08, Rad09].

Mô hình dựa trên đồ thị xuất hiện trong nhiều lĩnh vực thuộc lĩnh vực khai phá dữ liệu Web. Đơn giản và dễ nhận biết nhất chính là *đồ thị Web* với các đỉnh là các trang Web và các cung là các siêu liên kết giữa các trang Web với nhau. Cùng với sự phát triển không ngừng của Internet và Web, các mô hình mạng phức hợp liên quan (diễn hình như mạng xã hội, mạng e-mail, mạng block...) xuất hiện ngày càng nhiều. Trong nhiều năm qua, Dragomir R. Radev ghi nhận danh mục các công trình khoa học được công bố liên quan tới mô hình đồ thị trong khai phá Web trong các phiên bản "Bibliography Webgraph Papers". Kết quả của tác giả cho thấy, số lượng các công trình nghiên cứu về khai phá Web được ghi nhận lần lượt là 496 (tháng 5/2005), 1212 (tháng 5/2007), 1361 (tháng 5/2008), 1457 (tháng 1/2009) và 1471 (tháng 8/2009). Nội dung cập nhật trong các phiên bản thống kê gần đây chủ yếu là bổ sung các công trình khoa học mới. Chúng tôi cho rằng, tập hợp nội dung các văn bản thuộc các danh mục nói trên cũng chứa đựng các thông tin tiềm ẩn, có giá trị về lĩnh vực nghiên cứu.

Tương tự, trong "Semi-Supervised Learning Literature Survey", Xiaojin Zhu [Zhu08] khảo sát và tập hợp danh mục các công trình nghiên cứu về học bán giám sát. Các mô hình khai phá Web đa dạng và phong phú được xây dựng dựa trên lý thuyết xác suất (diễn hình là các mô hình dựa trên xác suất Bayes như mô hình Markov ẩn, mô hình Entropy cực đại, mô hình trường ngẫu nhiên có điều kiện,...) cũng là một nội dung cơ bản trong các danh mục. Khác với công việc mà Dragomir R. Radev tiến hành trong [Rad09] là chỉ lên danh mục các công trình liên quan, Xiaojin Zhu còn giới thiệu một số nội dung cơ bản và đặc trưng về tiếp cận học bán giám sát. Nội dung các phiên bản danh mục của Xiaojin Zhu cũng tiềm ẩn tri thức liên quan về học giám sát có thể khai phá được.

Cuốn sách "*Modeling the Internet and the Web: Probabilistic Methods and Algorithms*" của Pierre Baldi và cộng sự [BFS03] tổng hợp nội dung các công trình nghiên cứu về khai phá Web với cách tiếp cận các mô hình và thuật toán dựa trên nền tảng của lý thuyết xác suất. Các nội dung nền tảng toán học của khai phá Web, mà diễn hình là lý thuyết đồ thị và lý thuyết xác suất đóng vai trò chủ chốt của cuốn sách.

Các khảo sát trên đây cho thấy cả ba lĩnh vực nghiên cứu hợp thành Khai phá Web là Khai phá dữ liệu, Xử lý ngôn ngữ tự nhiên và World Wide Web đều được mô hình hóa và sử dụng các giải pháp từ lý thuyết đồ thị và lý thuyết xác suất. Đây là hai nền tảng toán học chủ yếu để phát triển các mô hình và là công cụ hiệu quả trong khai phá Web. Nguyên nhân làm cho hai nền tảng lý thuyết này chiếm vai trò quan trọng như vậy trong khai phá dữ liệu Web xuất phát từ bản chất tự nhiên và xã hội của hệ thống Web.

Mục này giới thiệu một số kiến thức cơ bản nhất của hai lý thuyết nói trên liên quan tới khai phá dữ liệu Web.

3.1. Mô hình đồ thị

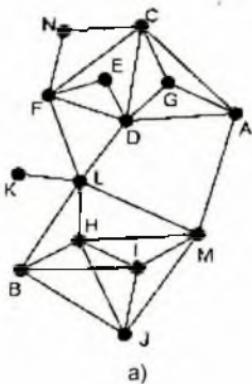
3.1.1. Một số kiến thức cơ bản

Lý thuyết đồ thị được ứng dụng rộng rãi trong khai phá Web, từ việc áp dụng trong mô hình hóa các đối tượng trong miền ứng dụng tới việc cai tiền các giải pháp khai phá Web. Phần này cung cấp một số nội dung cơ bản về lý thuyết đồ thị.

Định nghĩa 3.1: Đồ thị toán học là một cặp $G = \langle V, E \rangle$, trong đó V là tập các *đỉnh* (còn gọi là *mùt*), còn E là tập các *cung* (còn gọi là *cạnh*). Một cách hình thức, trong đồ thị $G = \langle V, E \rangle$:

- Tập $V = \{v\}$ được gọi là *tập đỉnh*;
- Tập $E \subseteq V \times V = \{e = (u, v) : u, v \in V\}$ được gọi là *tập cung*. Đối với cung $e = (u, v)$ thì đỉnh u và đỉnh v được gọi là *kề* với cung e ; hay cũng vậy, cung e là *kề* với đỉnh u và đỉnh v .

Nếu tập đỉnh V là hữu hạn thì đồ thị được gọi là *đồ thị hữu hạn*; nếu mọi cặp (u, v) không có thứ tự thì gọi G là *đồ thị vô hướng*; trong trường hợp ngược lại, gọi G là *đồ thị có hướng*. Cung (u, v) được gọi là đi ra từ u và đi vào tới v . Trong khai phá dữ liệu Web, chỉ có các đồ thị hữu hạn được quan tâm và chúng bao gồm cả hai loại có hướng và vô hướng.



a)

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| B | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| C | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| D | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| E | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| G | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| H | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| I | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| J | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| L | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| M | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| N | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

b)

Hình 3.1. Biểu diễn hình học và ma trận kề của đồ thị

Có một số phương pháp biểu diễn đồ thị, trong đó phương pháp hình học là một phương pháp biểu diễn điện hình nhất. Trong biểu diễn hình học, mỗi đỉnh của đồ thị được biểu diễn bằng một điểm hình học và mỗi cung (u, v) được biểu diễn bằng đoạn nối từ đỉnh u tới đỉnh v . Nếu đồ thị có hướng thì đoạn nối từ u tới v có chiều di vào v ; ngược lại, khi đồ thị không có hướng, nghĩa là cặp (u, v) cũng là cặp (v, u) , thì đoạn nối (u, v) không có chiều mũi tên. Hình 3.1 cho một ví dụ một đồ thị vô hướng có 14 đỉnh $\{A, B, C, D, E, F, G, H, I, J, K, L, M, N\}$ với tập hợp các cung được biểu diễn hình học như Hình 3.1a.

Biểu diễn hình học cho một cách nhìn trực quan về đồ thị, song không thể xử lý tính toán được. Biểu diễn ma trận kề là một biểu diễn điện hình của đồ thị dễ có thể xử lý được trong máy tính. Ma trận kề của đồ thị G có n đỉnh là một ma trận vuông có n chiều $A_G(n \times n)$, trong đó các đỉnh được tương ứng từ trái qua phải (theo chiều ngang) và từ trên xuống dưới (theo chiều dọc) với cùng một thứ tự. Nếu có cung di từ u tới v thì phần tử $A_G(u, v) = 1$; trong trường hợp còn lại, $A_G(u, v) = 0$. Biểu diễn ma trận kề

của đồ thị ví dụ G đã cho được trình bày trong Hình 3.1b. Chú ý rằng, ma trận kè của đồ thị vô hướng luôn là một ma trận đối xứng.

Độ của một đỉnh là số cung kè với nó. Trong đồ thị có hướng, với mỗi đỉnh thì số lượng cung đi tới đỉnh được gọi là *độ vào* của đỉnh, còn số lượng cung đi ra từ đỉnh được gọi là *độ ra* của đỉnh.

Đường đi là dãy các đỉnh, trong đó với mọi đỉnh (trừ đỉnh cuối cùng) đều tồn tại cung đi ra từ nó và đi tới đỉnh kế tiếp trong dãy. Trong đồ thị ở Hình 3.1, dãy AGDCGD là đường đi với các cung lần lượt là (A, G), (G, D), (D, C), (C, G), (G, D), trong đó A là đỉnh đầu, còn D là đỉnh cuối. Nếu mọi đỉnh (ngoại trừ đỉnh đầu) của một đường đi mà không được gặp quá một lần trên đường đi thì gọi đường đi đó là *đường đi đơn*. Trong đồ thị ở Hình 3.1, dãy AGDC là đường đi đơn với các cung lần lượt là (A, G), (G, D) và (D, C).

Một chu trình đơn là đường đi đơn với đỉnh đầu trùng với đỉnh cuối.

Độ dài của một đường đi là số lượng cung xuất hiện trên đường đi đó. Giữa hai đỉnh u và v, đường đi có độ dài ngắn nhất có đỉnh đầu là u và đỉnh cuối là v thì gọi là *đường đi ngắn nhất* từ u tới v. Khoảng cách lý thuyết đồ thị giữa hai đỉnh là độ dài đường đi ngắn nhất giữa chúng. Đường đi ngắn nhất từ đỉnh u tới đỉnh v không là duy nhất, nhưng khoảng cách lý thuyết đồ thị từ u tới v là duy nhất.

Một đồ thị được gọi là *liên thông* nếu với hai đỉnh u và v bất kỳ, luôn tồn tại đường đi đơn từ u tới v. Độ dài lớn nhất của tập các đường đi đơn trong đồ thị là $n - 1$, trong đó n là số lượng đỉnh của đồ thị.

Một đỉnh v được gọi là *đạt được từ một đỉnh u* nếu tồn tại đường đi đơn đi từ u tới v. Đồ thị con của đồ thị $G = (V, E)$ là đồ thị $G' = (V', E')$, trong đó tập V' là tập con các đỉnh trong V , còn E' là tập tất cả các cung thuộc E nối các đỉnh thuộc V' . Gọi đồ thị con G' là *thành phần liên thông* của G nếu G' là đồ thị con liên thông cực đại theo nghĩa không tồn tại một đồ thị con liên thông của G mà chứa thực sự G' .

Điểm cát của một đồ thị là đỉnh $v \in V$ mà nếu bỏ đỉnh v đó đi cùng với mọi cung kè đỉnh v thì số lượng thành phần liên thông của đồ thị được tăng lên thực sự.

Cầu của đồ thị là một cung trong đồ thị mà bỏ cung đó đi thì số lượng các thành phần liên thông của đồ thị được tăng thực sự.

Trong thực tế, nhiều bài toán quan trọng trong khai phá Web liên quan tới việc bài toán của đồ thị. Chẳng hạn, trong các hệ thống tìm kiếm trang Web (Chương 6), bài toán tính hạng hay "độ quan trọng" của một trang Web được xem xét trong đồ thị Web có hướng. Trong đồ thị Web có hướng, mỗi đỉnh là một trang Web, nếu như tồn tại liên kết ngoài từ trang Web u tới trang Web v thì có một cung đi từ đỉnh u tới đỉnh v. Thuật toán tính hạng

trang nguyên thuỷ [PBM98] đưa về bài toán tìm một vector riêng của ma trận biểu diễn đồ thị Web. Trong trường hợp cần xem xét đồ thị Web có trọng số, thì các giá trị độ ra và độ vào của đỉnh được sử dụng.

3.1.2. Đồ thị ngẫu nhiên

Trong không ít nghiên cứu, đồ thị Web được xem xét như một đồ thị ngẫu nhiên. Tính ngẫu nhiên trong đồ thị ngẫu nhiên cho phép mô hình hoá được biến động ngẫu nhiên của cả hệ thống Web. Đồ thị ngẫu nhiên được Paul Erdos và Alfréd Rényi đề xuất vào năm 1959 [ER61]. Sự tiến hoá nấm pha của đồ thị ngẫu nhiên theo tỷ lệ giữa số lượng cung trên số lượng đỉnh có thể được áp dụng vào việc mô hình hoá sự biến động của đồ thị Web. Phần này giới thiệu sơ bộ về đồ thị ngẫu nhiên như đã được đề cập trong các nghiên cứu về mô hình Web. Một số nội dung ở đây có liên quan tới lý thuyết xác suất sẽ được giải thích trong mục 3.2.

Cho trước hai số nguyên dương n và N , trong đó n là số lượng đỉnh và N là số lượng cung của các đồ thị ngẫu nhiên sẽ được xem xét.

Đặt $\Xi_{n, N} = \{G_{n, N} = \langle V, E_{n, N} \rangle; V = \{v_1, v_2, \dots, v_n\}; E_{n, N} \subseteq V \times V \text{ mà } |E_{n, N}| = N\}$, có nghĩa là, $\Xi_{n, N}$ là tập các đồ thị $G_{n, N}$ có cùng tập đỉnh V với n đỉnh và các tập cung khác nhau $E_{n, N}$ với N cung. Các đồ thị $G_{n, N}$ phần tử của $\Xi_{n, N}$ là vô hướng, không có cung bội (Định nghĩa 3.1 đã ngầm định đồ thị không có cung bội), không bị giằng. Với các ràng buộc nói trên, tổng số cung nối có thể có giữa các đỉnh trong V là $\binom{n}{2}$ và có $\binom{\binom{n}{2}}{N}$ cách chọn các bộ N cung khác nhau trong tập $\binom{\binom{n}{2}}{N}$ cung có thể. Như vậy, $|\Xi_{n, N}| = \binom{\binom{n}{2}}{N}$

và nói chung thì đây là một số rất lớn.

Định nghĩa 3.2a (Đồ thị ngẫu nhiên [ER61]): Đồ thị ngẫu nhiên $G_{n, N}$ được xác định như một phần tử của $\Xi_{n, N}$ được chọn một cách ngẫu nhiên, trong đó các phần tử trong $\Xi_{n, N}$ là đồng khả năng được chọn với xác suất

$$\frac{1}{\binom{\binom{n}{2}}{N}}$$

Định nghĩa 3.2b (Đồ thị ngẫu nhiên theo quá trình ngẫu nhiên [ER61]): Đồ thị ngẫu nhiên là một quá trình thống kê ngẫu nhiên với:

– Khi $t = 1$: chọn cung e_1 từ $\binom{n}{2}$ cung đồng khả năng có thể có kết nối n định của V;

– Khi $t = 2$: chọn cung e_2 từ $\binom{n}{2} - 1$ cung đồng khả năng còn lại (khác với e_1);

– ...

– Khi $t = k + 1$: chọn cung e_{k+1} từ $\binom{n}{2} - k$ cung đồng khả năng còn lại (khác với e_1, e_2, \dots, e_k);

– ...

– Ký hiệu $G_{n,N}$ là đồ thị chứa các đỉnh thuộc V và các cung e_1, e_2, \dots, e_N .

Paul Erdos và Alfréd Rényi chỉ ra rằng, hai định nghĩa nói trên là tương đương. Định nghĩa thứ hai có ý nghĩa sử dụng cao hơn khi mà số hiệu của các cung trong đồ thị ngẫu nhiên được tương ứng với thời điểm và điều đó tạo thuận lợi cho việc nghiên cứu tỷ mỉ sự tiến hóa của đồ thị ngẫu nhiên, tức là làm sáng tỏ bước tiếp bước cấu trúc của $\Xi_{n,N}$ khi N tăng lên. Điều đó giải thích được quá trình tiến hóa của đồ thị ngẫu nhiên được sử dụng như mô hình về sự tiến hóa của rất nhiều hệ thống trong thế giới thực, thậm chí bao hàm cả sự phát triển các quan hệ xã hội. Ngày nay, quá trình tiến hóa của đồ thị ngẫu nhiên được sử dụng như một mô hình biểu diễn tốt các đồ thị, các mạng có trên hệ thống Internet.

Nghiên cứu của Paul Erdos và Alfréd Rényi [ER61] chứng tỏ rằng, khi số đỉnh n là một số nguyên dương khá lớn và cho số cung N tăng dần từ 1 tới $\binom{n}{2}$, sự tiến hóa của $G_{n,N}$ trải qua 5 pha phân biệt rõ ràng, mỗi pha có một số tính chất đặc trưng riêng. Đó là các pha:

(1) $N(n) = O(n)$:

(2) $N(n) \approx cn$, với $0 < c < 1/2$;

(3) $N(n) \approx cn$, với $c \geq 1/2$;

(4) $N(n) \approx cn \cdot \log n$, với $c \leq 1/2$;

(5) $N(n) \approx (n \cdot \log n)w(n)$, với $w(n) \rightarrow \infty$ khi $n \rightarrow \infty$.

Tương ứng với mỗi một trong năm pha tiến hóa trên, đồ thị ngẫu nhiên tuân theo một số phân bố xác suất như phân bố chuẩn, phân bố Poisson hoặc phân bố hàm mũ.

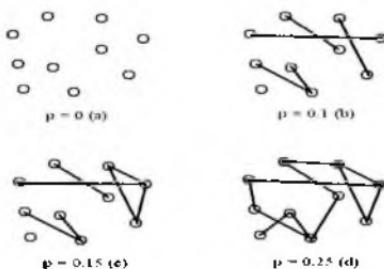
Dưới đây là một ví dụ minh họa sự biến động của đồ thị ngẫu nhiên theo sự biến động của xác suất p lựa chọn cung.

Ví dụ 3.1. Minh họa hình ảnh của đồ thị ngẫu nhiên.

Giả thiết rằng, có một số N rất lớn ($N \gg 1$) các nút đặt rải rác trên sân nhà. Giả sử sử dụng một sợi dây buộc hai nút bất kỳ với xác suất p thành một cặp nút. Khi đó, tổng số cung là $pN(N - 1)/2$ (Hình 3.2). Mục tiêu chính của lý thuyết đồ thị ngẫu nhiên là xác định tại liên kết nào xác suất p của một thuộc tính cụ thể của đồ thị sẽ xuất hiện gần như là nhiều nhất.

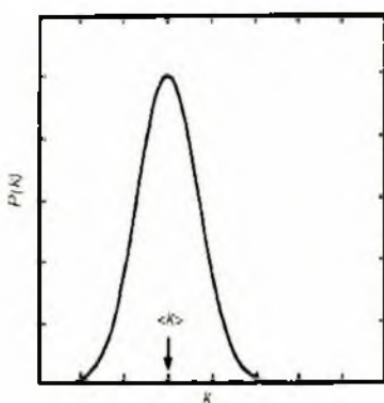
Một điều khá đặc biệt đó là, các tính chất chính và quan trọng của các đồ thị ngẫu nhiên có thể xuất hiện khá dột ngột (tương tự như sự biến động khi chuyển pha). Chẳng hạn, nếu nâng một nút lên thì liệu sẽ có bao nhiêu nút bị nâng theo? Paul Erdos và Alfréd Rényi chỉ ra rằng, nếu xác suất p lớn hơn một ngưỡng p_c ($p_c \sim (\ln N)/N$) thì hầu hết mọi nút trong đồ thị ngẫu nhiên là được kết nối, điều này có nghĩa là nhật được tất cả các nút bằng cách nâng một nút lên.

Bậc trung bình của một đồ thị ngẫu nhiên là $\langle k \rangle = p(N - 1) \approx pN$. Gọi L_{rand} là độ dài đường dẫn trung bình của mạng ngẫu nhiên. Bằng quan sát, có thể thấy sẽ có $\langle k \rangle^{L_{rand}}$ các đỉnh trong đồ thị ngẫu nhiên có khoảng cách L_{rand} hoặc rất gần với đại lượng này. Do vậy, $N \sim \langle k \rangle^{L_{rand}}$, điều này có nghĩa là $L_{rand} \sim \ln N / \langle k \rangle$. Sự gia tăng của hàm loga trong độ dài đường dẫn trung bình với độ lớn của đồ thị là một ánh hưởng phổ biến của small-world (một thuộc tính trong mạng xã hội). Bởi vì $\ln N$ tăng chậm hơn so với N , nó cho phép chiều dài trung bình phải khá nhỏ, thậm chí ngay cả trong một mạng khá lớn. Mặt khác, trong mạng ngẫu nhiên hoàn toàn (ví dụ mạng của những người bạn), xác suất mà hai người bất kỳ là bạn của nhau không lớn hơn xác suất hai người được chọn ngẫu nhiên trong mạng là bạn của nhau. Vì thế, độ phân cụm của mô hình Paul Erdos và Alfréd Rényi là

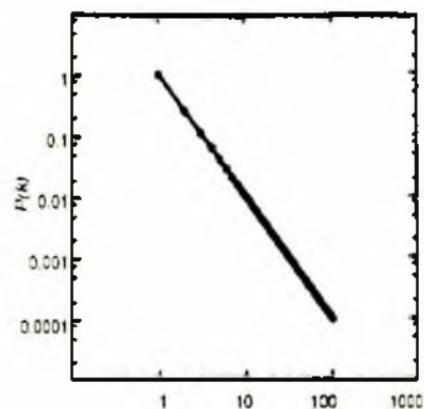


Hình 3.2. Sự phát triển của một đồ thị ngẫu nhiên:
(a) khởi tạo 10 nút; (b) nối cặp nút với xác suất $p = 0,1$;
(c) nối các cặp nút với xác suất $p = 0,15$ và (d) nối các cặp nút với xác suất $p = 0,25$

$C = p = \langle k \rangle / N \ll 1$. Điều này có nghĩa là, mạng ngẫu nhiên trên diện rộng nói chung là không bị phân cụm. Trong thực tế, với N lớn, thuật toán của Paul Erdos và Alfréd Rényi sinh ra một mạng đồng nhất có các liên kết tuân theo phân phối Poisson (Hình 3.3). Phân bố hàm mũ (Hình 3.4) cũng là một khả năng khi số lượng cung đạt một giá trị rất lớn.



Hình 3.3. Phân bố Poisson



Hình 3.4. Phân bố hàm mũ

Thông qua việc tổng hợp kết quả nghiên cứu của nhiều tác giả, Pierre Baldi và cộng sự đã làm sáng tỏ về mối liên hệ giữa khái niệm đồ thị ngẫu nhiên với đồ thị Web, liên kết theo luật số lớn, mạng thế giới nho, các mô hình sinh đồ thị Web và các mạng liên quan trên Internet [BFS03].

3.1.3. Mạng xã hội

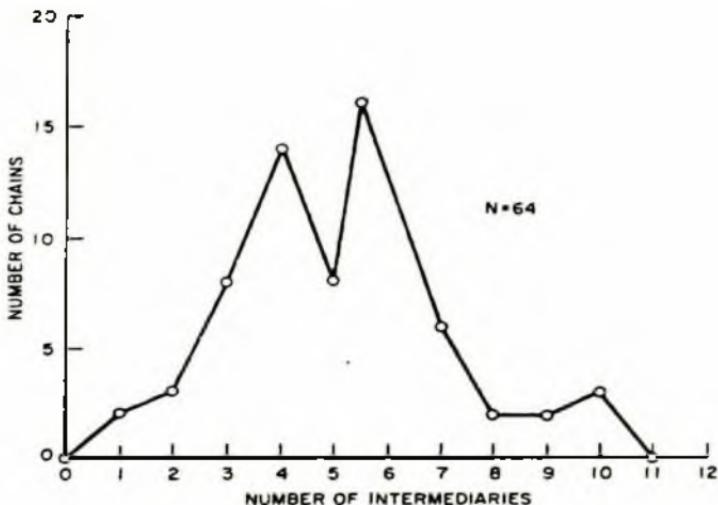
Mạng xã hội là mạng của một nhóm người hoạt động và các mối quan hệ gắn kết họ với nhau. Những người hoạt động trong mạng có thể là những cá nhân hoặc tập thể (các đơn vị như các phòng ban, các tổ chức, các gia đình,...). Những người này trao đổi tài nguyên với nhau và chính điều đó đã gắn kết họ lại với nhau trong một mạng xã hội. Tài nguyên ở đây bao gồm dữ liệu, thông tin, sản phẩm và các dịch vụ, hỗ trợ xã hội hoặc hỗ trợ tài chính. Mỗi loại tài nguyên trao đổi được xem như một mối liên kết của mạng xã hội và những cá nhân duy trì mối liên hệ này được tương ứng với việc duy trì một cung. Sức bền của cung này được sắp xếp từ yếu đến mạnh phụ thuộc vào số lượng và các kiểu của nguồn tài nguyên mà các thành viên trao đổi, mức độ thường xuyên trao đổi và sự thân mật trong quá trình trao đổi giữa họ.



Hình 3.5. Một mạng xã hội kiểu e-mail

Nguồn: <http://www.uvm.edu/~pdodds/teaching/courses/2008-01UVM-295/docs/2008-01UVM-295smallworldnetworks-slides-handout.pdf>

Các mối quan hệ trao đổi thường được tiến hành trong một số lượng dân số lựa chọn nhất định. Những nhà phân tích trong lĩnh vực mạng dựa vào các quan hệ giữa các thành viên của một cộng đồng, các hàng xóm, một nhóm hoặc một lớp để hiểu cách thức các mạng xác định dân số hay các nhóm nhỏ bên trong một mạng lớn. Cách thức mà một người kết nối với một người khác thể hiện cấu trúc nền tảng của mạng, bao gồm những người thuộc và không thuộc vào một mạng và các kiểu trao đổi nào để xác định một mạng. Mạng này được duy trì bởi sự trao đổi của các tài nguyên đơn lẻ hay rất nhiều tài nguyên lớn tương ứng với các nút mạnh hay yếu. Ví dụ, các nhà phân tích có thể dò tìm sự trao đổi thông tin về công việc của những người quen biết nhưng không mấy thân thiện, mối quan hệ trong dòng tộc hoặc mối quan hệ giữa những người công nhân. Các mạng xã hội được lần đầu bởi những sự chuyên đổi này chỉ ra cách các nguồn tài nguyên di chuyển trong một mạng, cách mà các tác nhân xác định vị trí để tác động tới nguồn tài nguyên trao đổi và các kiểu của tài nguyên trao đổi rất quan trọng trong những môi trường khác nhau.



Hình 3.6. Tần suất độ dài của chuỗi đầy đủ [TM69]

Vấn đề nghiên cứu cấu trúc của mạng xã hội đã gây ra sự chú ý và quan tâm sâu sắc của các nhà nghiên cứu trong nhiều năm qua. Đầu tiên là thí nghiệm của Stanley Milgram [TM69] được cuốn hút vào việc khám phá ra độ dài đường dẫn giữa mọi người trong một mạng xã hội trên diện rộng. Xuất phát điểm từ câu hỏi "*Xác suất biết lẫn nhau của hai người được chọn từ một quần thể lớn, chẳng hạn như nước Mỹ, là bao nhiêu?*". Kết quả thí nghiệm của Stanley Milgram về phân bố tần suất độ dài chuỗi đầy đủ (completed chain) được trình bày trong Hình 3.6, trong đó trực hoành chỉ độ dài của chuỗi đầy đủ (là số lượng các truy vấn trực tiếp từ người khởi đầu tới người đích). Độ dài trung bình của các chuỗi đầy đủ là 5.2 và điều đó có thể được giải thích là "cần dùng khoảng 5.2 giao tiếp là một người bất kỳ có thể giao tiếp với một người bất kỳ khác". Độ dài 5.2 là nhỏ, vì vậy từ kết quả thí nghiệm dẫn đến một giả thuyết về "*thế giới nhỏ*", tương tự như cách nói quen thuộc của người Việt Nam là "*qua đất tròn*". Mặc dù thí nghiệm của Stanley Milgram với số lượng 279 người khởi đầu (nhỏ) cùng với các giả thiết đi kèm trong thí nghiệm, song giả thuyết về đường kính nhỏ của các mạng xã hội vẫn còn có giá trị. Trên thực tế, người ta đã tìm hiểu được nhiều mạng xã hội theo mãn giả thuyết đường kính nhỏ của S. Milgram, bao gồm các mạng cộng tác khoa học và đồ thị các cuộc gọi điện thoại....

Sự quan tâm nghiên cứu về mạng xã hội của các nhà khoa học đã thu nhận được nhiều phát minh khoa học mới về mạng xã hội trong nhiều thập kỷ qua, được mô hình và phân tích bằng các công cụ của lý thuyết đồ thị. Qua những nghiên cứu này, người ta đã chứng minh được mạng xã hội thực

tiễn có xu hướng có cấu trúc của mạng bát ngẫu nhiên; ngoài ra, nó còn mang hai tính chất nổi bật và quan trọng nhất của mạng phức hợp đó là thuộc tính small-world và thuộc tính độ phân phối theo hàm mũ của mạng scale-free.

3.2. Học máy xác suất Bayes

3.2.1. Một số kiến thức cơ sở

Phần này trình bày một số kiến thức cơ bản trong lý thuyết xác suất theo định hướng ứng dụng từ cách tiếp cận của khai phá Web.

Định nghĩa 3.3: Không gian do được (Ω, \mathcal{B}) là một cặp của không gian các đối tượng trong miền ứng dụng (không gian mẫu) Ω với một σ -trường \mathcal{B} các tập con của Ω , trong đó mỗi tập con trong Ω được gọi là *một sự kiện*; σ -trường (còn gọi là σ -đại số) \mathcal{B} là một tập hợp các tập con của Ω thỏa mãn các tính chất sau:

$$-\quad \Omega \in \mathcal{B} \quad (3.1)$$

$$-\quad \text{Nếu } F \in \mathcal{B} \text{ thì } (\Omega \setminus F) \in \mathcal{B} \quad (3.2)$$

$$-\quad \text{Nếu } F_i \in \mathcal{B} (i = 1, 2, \dots) \text{ thì } \bigcup_i F_i \in \mathcal{B} \quad (3.3)$$

Như vậy có nghĩa là, trong σ -trường \mathcal{B} thì không gian mẫu Ω là một phần tử của nó (3.1), tập bù của một tập thuộc \mathcal{B} cũng thuộc \mathcal{B} (3.2) và hợp của các tập thuộc \mathcal{B} cũng thuộc \mathcal{B} (3.3). Theo định luật De-Morgan của lý thuyết tập hợp, ta nhận được:

$$\text{Nếu } F_i \in \mathcal{B} (i = 1, 2, \dots) \text{ thì } \bigcap_{i=1}^{\infty} F_i = \left(\bigcup_{i=1}^{\infty} F_i^c \right)^c \in \mathcal{B} \quad (3.4)$$

trong đó F^c là tập bù của tập F theo Ω .

Với một tập mẫu Ω , tồn tại một lớp rộng lớn các σ -trường trên nó, trong đó có hai σ -trường đặc biệt đó là (Ω, \emptyset) và $(\Omega, 2^\Omega)$, ở đây (Ω, \emptyset) là "nhỏ nhất", còn $(\Omega, 2^\Omega)$ là "lớn nhất" (2^Ω ký hiệu tập hợp mọi tập con có thể có của Ω bao gồm cả chính nó và tập rỗng; về lực lượng thì $|2^\Omega| = 2^{|\Omega|}$).

Nếu trong công thức (3.3), giới hạn là hợp hữu hạn các tập con, thì khi đó \mathcal{B} được gọi là *một trường*. Về mặt nghiên cứu, trường cho phép các tính toán xác định, còn σ -trường cho phép xem xét đáng diệu của các dãy vô hạn các sự kiện thuộc không gian mẫu. Tồn tại hai trường hợp riêng về dãy đếm được vô hạn đơn diệu là:

$$G_1 \supset G_2 \supset \dots \supset G_k \supset G_{k+1} \supset \dots \supset (\cap G_i) = G$$

$$\text{và } F_1 \subset F_2 \subset \dots \subset F_k \subset F_{k+1} \subset \dots \subset (\cup F_i) = F$$

Định nghĩa 3.4: Không gian xác suất là một bộ ba (Ω, \mathcal{B}, P) , trong đó (Ω, \mathcal{B}) là một không gian đo được, còn P là một độ đo trên σ -trường \mathcal{B} ($P: \mathcal{B} \rightarrow \mathbb{R}$ các số thực) thỏa mãn các điều kiện:

- Không âm: $P(F) \geq 0, \forall F \in \mathcal{B}$ (3.5)

- Chuẩn hoá: $P(\Omega) = 1$ (3.6)

- Cộng tính đếm được:

$$\text{Nếu } F_i \in \mathcal{B}, F_i \cap F_j = \emptyset, \forall i, j \text{ thì } P\left(\bigcup_i F_i\right) = \sum_i P(F_i) \quad (3.7)$$

Lưu ý rằng, nếu bỏ qua điều kiện chuẩn hóa xác suất (3.6) thì bộ ba (Ω, \mathcal{B}, P) được gọi là *không gian đo được* với độ đo P . Giá trị $P(F)$ được gọi là xác suất của sự kiện F . Nếu không nhấn mạnh tới ngữ nghĩa sự kiện là một tập con trong không gian mẫu, ký hiệu e thường được dùng để chỉ các sự kiện. Xem xét các dãy vô hạn đơn điệu các sự kiện, ta có được các trường hợp xác suất giảm tới 0 (nếu giao là \emptyset) hoặc xác suất tăng tới 1 (nếu có hợp là Ω).

Định nghĩa 3.5: Một hệ thống động là một bộ bốn $(\Omega, \mathcal{B}, P, T)$, trong đó:

- (Ω, \mathcal{B}, P) là một không gian xác suất;
- T là một ánh xạ trong Ω , $T: \Omega \rightarrow \Omega$, thỏa mãn điều kiện $\forall F \in \mathcal{B}$ thì $T^{-1}F = \{\omega: T(\omega) \in F\} \in \mathcal{B}$ và được gọi là phép biến đổi đo được.
Trong một số trường hợp đơn giản gọi T là phép biến đổi.

Tính "động" trong hệ thống động cho phép mô tả được tính động của các mô hình ứng dụng thông qua các phép biến đổi T tương ứng.

Định nghĩa 3.6: Cho không gian xác suất (Ω, \mathcal{B}, P) và một không gian đo được khác (A, \mathcal{B}_A) . Biến ngẫu nhiên (hoặc phép đo) được xác định trên (Ω, \mathcal{B}, P) và nhận giá trị trên (A, \mathcal{B}_A) là một ánh xạ (hoặc một hàm) $f: \Omega \rightarrow A$ bảo đảm tính chất:

$$\forall F \in \mathcal{B}_A: f^{-1}(F) = \{\omega: f(\omega) \in F\} \in \mathcal{B} \quad (3.8)$$

Tính chất (3.8) có thể phát biểu là "đối ảnh của một sự kiện trong (A, \mathcal{B}_A) cũng là một sự kiện trong (Ω, \mathcal{B}, P) ". Từ tính chất này có thể di dời ý tưởng chuyển việc khảo sát trên \mathcal{B}_A (cụ thể là khảo sát phân hoạch trong A) về việc khảo sát trên \mathcal{B} . Với một biến ngẫu nhiên f , độ đo xác suất P trong không gian xác suất (Ω, \mathcal{B}, P) cảm sinh ra một độ đo xác suất P' trong không gian xác suất (A, \mathcal{B}_A) , trong đó $\forall F \in \mathcal{B}_A: P'(F) = P(f^{-1}(F)) = P(\{\omega: f(\omega) \in F\})$. Ta nhận được (A, \mathcal{B}_A, P') cũng là một không gian xác suất. Trong trường hợp không gây nhầm lẫn, dùng chính ký hiệu P thay cho P' . Với phần tử $a \in A$ (hoặc tập con $C \subseteq A$), gọi *dai lượng* $P(a)$ (hoặc $P(C)$) là xác suất để biến ngẫu nhiên f nhận giá trị a (hoặc thuộc vào tập C).

Trường hợp đặc biệt (song diễn hình do tính thông dụng) của biến ngẫu nhiên là tập $A = \mathbb{R}$ (tập các số thực) và \mathcal{B}_A là trường Borel (σ -trường nhỏ nhất chứa các khoảng số thực). Theo cách gọi truyền thống, thì biến ngẫu nhiên được gọi khác nhau tùy thuộc vào không gian giá trị A , cụ thể được gọi là "*biến ngẫu nhiên*" khi A là trường số thực, là "*vector ngẫu nhiên*" khi A là không gian $\mathbb{O}colit$ hoặc là "*quá trình ngẫu nhiên*" khi A là không gian dãy (hoặc không gian sóng).

Một biến ngẫu nhiên giá trị thực f ($f: \Omega \rightarrow \mathbb{R}$, với \mathbb{R} là tập số thực) còn được gọi là *một phép đo*.

Định nghĩa 3.7: Quá trình ngẫu nhiên thời gian rời rạc (thông dụng được gọi là quá trình ngẫu nhiên) là một dãy các biến ngẫu nhiên $\{X_n | n \in J\}$, với J là tập chỉ số đếm được} cùng xác định trên một không gian xác suất.

Dãy các số nguyên $Z = \{\dots, -2, -1, 0, 1, 2, \dots\}$ và dãy các số tự nhiên $Z^+ = \{0, 1, 2, \dots\}$ là hai tập chỉ số diễn hình; trong trường hợp đó, quá trình ngẫu nhiên với tập chỉ số Z được gọi là *quá trình ngẫu nhiên hai phía*; còn với tập chỉ số Z^+ được gọi là *quá trình ngẫu nhiên một phía*.

Xem xét quá trình ngẫu nhiên một phía được hình thành theo cách thức sau đây. Phép biến đổi T được thực hiện lặp và theo chỉ số i nhận được dãy phép biến đổi T^i như sau:

- $T^2: \Omega \rightarrow \Omega$, trong đó $T^2(\omega) = T(T(\omega))$;
- $T^n: \Omega \rightarrow \Omega$, trong đó $T^n(\omega) = T(T^{n-1}(\omega))$.

Nếu f là một biến ngẫu nhiên A -giá trị (miền giá trị là tập A) được xác định trên (Ω, \mathcal{B}) thì hàm $fT^n: \Omega \rightarrow A$ được xác định bằng $fT^n(\omega) = f(T^n(\omega))$ cũng là một biến ngẫu nhiên với mọi số tự nhiên n . Như vậy, một hệ thống động cùng với một biến ngẫu nhiên (phép đo) f sẽ xác định được một quá trình ngẫu nhiên một phía $\{X_n\}_{n \in \mathbb{Z}^+}$ với $X_n(\omega) = f(T^n(\omega))$.

• Ký vọng

Cho (Ω, \mathcal{B}, m) là một không gian xác suất. Như đã giới thiệu, một biến ngẫu nhiên có giá trị thực f được gọi là *một phép đo*. Trong các nội dung tiếp theo, ta hạn chế chỉ xem xét phép đo f với miền giá trị (còn gọi bằng ký hiệu *chuẩn*) gồm hữu hạn các giá trị không âm. Trong trường hợp này, phép đo f còn được gọi là *biến ngẫu nhiên rời rạc*, hoặc *phép đo rời rạc*, hoặc *phép đo số*, hoặc *hàm đơn giản* (như cách gọi thông dụng của toán học).

Giả sử miền giá trị của f là $f(\Omega) = \{b_i | i = 1, 2, \dots, N\}$, trong đó b_i là khác biệt nhau từng đôi một. Ký hiệu các tập $F_i = f^{-1}(b_i) = \{x: f(x) = b_i\}$, $i = 1, 2, N$. Do f là phép đo, nên mọi $F_i \in \mathcal{B}$ và các F_i là rời nhau từng đôi một và hợp của toàn bộ F_i cho không gian Ω . Điều đó có nghĩa là $\{F_i | i = 1, 2, \dots, N\}$ tạo nên một phân hoạch của Ω . Như vậy, phép đo f có thể được biểu diễn thành:

$$f(x) = \sum_{i=1}^n b_i I_{F_i}(x) \quad (3.9)$$

trong đó $b_i \in R$, $F_i \in \mathcal{B}$, còn I_{F_i} là hàm đặc trưng trên F_i .

Định nghĩa 3.8: Kỷ vọng (trung bình mẫu, trung bình xác suất, số trung bình) của một phép đo rời rạc $f: \Omega \rightarrow R$ được xác định bởi công thức (3.9) đối với độ đo xác suất m (ký hiệu là $E_m f$) được định nghĩa là:

$$E_m f = \sum_{i=1}^n b_i m(F_i) \quad (3.10)$$

Như một hệ quả trực tiếp, ta có $E_m I_F = m(F)$.

• Entropy

Cho $(\Omega, \mathcal{B}, P, T)$ là một hệ thống động, f là phép đo xác định trên Ω và f xác định một quá trình ngẫu nhiên một phía $f_n = fT^n$; $n = 0, 1, 2, \dots$ trong đó f_0 dùng để chỉ một phép biến đổi mà chưa áp dụng f lần nào ($\forall \omega \in \Omega: f_0(\omega) = \omega$). Phép đo f có bảng ký hiệu mẫu là $A = \{a_1, a_2, \dots, a_A\}$. Gọi đại lượng $P(\{\omega \in \Omega: f(\omega) = a\}) = P(f = a)$ là xác suất biến ngẫu nhiên f nhận giá trị a .

Định nghĩa 3.9: Đại lượng entropy $H_P(f)$ của một biến ngẫu nhiên trên không gian xác suất (Ω, \mathcal{B}, P) với bảng ký hiệu mẫu A được xác định:

$$H_P(f) = - \sum_{a \in A} P(f = a) \ln P(f = a) \quad (3.11)$$

trong đó coi rằng $0 \ln(0) = 0$.

Trong khoa học máy tính, logarit cơ số 2 thường được dùng thay cho logarit tự nhiên. Đơn vị theo logarit tự nhiên được gọi là "nat" (thường được dùng trong toán học), còn đơn vị do theo logarit cơ số 2 được gọi là "bit" (thường được dùng trong ngữ cảnh trực quan).

Trong các ngữ cảnh ngầm định, công thức (3.11) thường được viết là:

$$H(f) = - \sum_{a \in A} p_f(a) \ln p_f(a) \quad (3.12)$$

Giá trị $H(f)$ là một độ đo tương ứng với biến ngẫu nhiên f . Trong lý thuyết thông tin, entropy được gắn với ý nghĩa độ đo bất định của biến ngẫu nhiên f trong một không gian xác suất.

Tương tự, entropy của một sự kiện e có xác suất $P(e)$ được tính theo công thức:

$$H(e) = -P(e) \ln P(e) - (1 - P(e)) \ln (1 - P(e)) \quad (3.13)$$

Entropy của các sự kiện có giá trị không âm, entropy của biến có là tổng các biến có độc lập bằng tổng entropy của các biến có thành phần.

Định nghĩa 3.10: Entropy có điều kiện của biến cỗ X khi biến cỗ Y đã xuất hiện được định nghĩa như sau:

$$\begin{aligned} H(X|Y) &= H(X, Y) - H(Y) \\ \text{hay } H(X|Y) &= - \sum_{x,y} P(x = a, y = b) \ln P(x = a, y = b) \\ &= - \sum_{x,y} p_{x,y}(x, y) \ln p_{x|y}(x, y) \end{aligned} \quad (3.14)$$

trong đó $p_{x,y}(x, y)$ là hàm mật độ xác suất đối với (X, Y) và $p_{x|y}(x|y)$ là hàm mật độ xác suất có điều kiện. Ta có $0 \leq H(X|Y) \leq H(X)$.

Đại lượng thông tin quan hệ trung bình giữa hai biến cỗ X và Y được xác định như sau:

$$\begin{aligned} I(X, Y) &= H(X) + H(Y) - H(X, Y) \\ &= H(X) - H(X|Y) = H(Y) - H(Y|X). \end{aligned}$$

Theo ngữ nghĩa của hàm mật độ xác suất, thì đại lượng trên được tính như sau:

$$\begin{aligned} I(X, Y) &= \sum_{x,y} P(X = x, Y = y) \ln \frac{P(X = x, Y = y)}{P(X = x)P(Y = y)} \\ &= \sum_{x,y} p_{x,y}(x, y) \ln \frac{p_{x,y}(x, y)}{p_x(x)p_y(y)} = \sum_{x,y} p_{x,y}(x, y) \ln \frac{p_{x|y}(x|y)}{p_x(x)p_y(y)} \\ &= \sum_{x,y} p_{x,y}(x, y) \ln \frac{p_{y|x}(y|x)}{p_y(y)} \end{aligned}$$

Giá trị đại lượng thông tin quan hệ giữa hai biến cỗ được ứng dụng trong một số lớp bài toán khai phá dữ liệu, chẳng hạn, trong thuật toán cây quyết định, thuật toán cực đại entropy.

Robert M. Gray [Gra08] cung cấp một cách hệ thống các nội dung về entropy và thông tin.

3.2.2. Học máy xác suất Bayes

Như đã biết, nhiều nội dung trong khai phá Web liên quan tới mô hình xác suất do mô hình xác suất thường được sử dụng để đặc trưng cho mô hình Web [BFS03] về các phương diện lý thuyết, thực nghiệm và triển khai. Lý thuyết và mô hình xác suất cho phép làm phù hợp một cách chặt chẽ mô hình miền ứng dụng với các sự kiện và tạo nên chỉ dẫn tới các sự kiện như vậy từ các dữ liệu nhận được từ quan sát.

Cụ thể hơn, giả sử quan tâm tới một sự kiện e là một định đê hay một khẳng định từ thế giới thực. Ví dụ, e là khẳng định "*tới tháng 5 năm 2009, có hơn 235 triệu Website trên Internet*". Khẳng định e được gọi là "xác định tốt" là đúng hay sai căn cứ vào việc xác định có tính lý luận về việc bằng cách nào chúng ta liệt kê ra được số lượng Website trên Internet đến thời điểm tháng 5 năm 2009 là lớn hơn 235 triệu.

Cách thông dụng nhất thường được sử dụng để đưa ra khẳng định trên là trích dẫn gián tiếp tới kết quả nghiên cứu do các tổ chức, cá nhân có uy tín công bố. Chẳng hạn, ở đây ta hướng tới Netcraft survey⁽¹⁾, với việc công nhận cách thức Netcraft survey tiến hành là đáng tin cậy. Tuy nhiên, cũng có thể tiến hành theo một cách thức công phu hơn; chẳng hạn như, thông qua số liệu thống kê từ một số máy tìm kiếm điển hình. Xác suất $P(e)$ có thể được xem như một giá trị số phản ánh mức độ đúng/sai của sự kiện e trong thế giới thực, được tính theo các thông tin hiện có. Giá trị này được coi như "độ tin tưởng", thường được gọi là *mô tả Bayes* (the Bayesian interpretation) của xác suất $P(e)$ về sự kiện e (định đê hay khẳng định được đưa ra). Để chính xác hơn, sử dụng xác suất có điều kiện đối với sự kiện e , được ký hiệu là $P(e|\theta)$, mô tả tổng quát về độ tin tưởng, trong đó θ là tri thức nền mà dựa trên đó cho phép ta tin cậy về sự kiện e . Trong nhiều trường hợp, bỏ qua biểu diễn về điều kiện nền θ (chi viết $P(e)$) với một quan niệm ngầm định rằng điều kiện nền θ luôn luôn được coi là đã có.

Mô hình tần số là một mô hình cổ điển liên quan tới xác định xác suất $P(e)$ xuất hiện sự kiện e , trong đó xác suất xuất hiện biến cố e là tần số thử nghiệm thành công quan sát được trên một số hữu hạn các phép thử nghiệm được lặp đi lặp lại. Theo mô hình này, để xác định độ tin cậy của định đê "*tới tháng 5 năm 2009, có tới hơn 235 triệu Website trên Internet*", cần tiến hành lặp đi lặp lại các thử nghiệm đếm toàn bộ số Website trong tháng 5 năm 2009. Kết quả là xác suất $P(e)$ được tính bằng tỷ số giữa số lượt thử nghiệm có giá trị đếm vượt quá 235 triệu trên tổng số lượt đếm thử nghiệm.

Mô tả Bayes theo *mô hình xác suất $P(e|\theta)$* là tổng quát hoá của mô hình tần số nói trên. Mô tả Bayes rất hữu dụng do nó cho phép đưa ra các khẳng định định đê "về số lượng Website đang tồn tại" mà không cần thi hành lặp đi lặp lại các thử nghiệm đếm toàn bộ số Website như đã làm trong mô hình tần số.

Lý luận trên đây còn chứng tỏ rằng, độ tin tưởng có thể được trình diễn bằng các số thực và được thay đổi theo tỷ lệ đê chuẩn hoá vào đoạn $[0, 1]$. Đồng thời, độ tin tưởng bắt buộc phải tuân theo các quy tắc xác suất, mà nói riêng là phải tuân theo định lý Bayes. Các quy tắc biến đổi cơ bản xác suất

⁽¹⁾ (http://news.netcraft.com/archives/web_server_survey.html)

như xác suất có điều kiện hay luật của xác suất tổng vẫn còn nguyên giá trị ngữ nghĩa như trong lý thuyết xác suất.

Tiếp cận Bayes cho phép coi xác suất trên đây như là một thực thể động theo nghĩa nó được cập nhật/biến đổi khi có thêm dữ liệu mới xuất hiện. Điều đó cho phép thay đổi độ tin tưởng của con người khi có thêm dữ liệu mới quan sát được. Cơ sở của lý luận như vậy xuất phát từ nhận định (trong một số trường hợp được coi như tiên đề) là, khi có thêm dữ liệu từ miền ứng dụng thì tri thức về miền ứng dụng cũng được cải tiến. Người ta thường thay chỉ dẫn $P(e|\theta)$ bằng chỉ dẫn $P(e|D)$, trong đó D là một tập dữ liệu đã quan sát được (là tài nguyên để có tri thức nền). Theo định lý Bayes:

$$P(e|D) = \frac{P(D|e)P(e)}{P(D)} \quad (3.15)$$

Để nắm bắt được nội dung cơ bản của công thức này trong ứng dụng, cần thiết phải giải thích về ý nghĩa của mỗi toán hạng trong đẳng thức. Trong (3.15), $P(e)$ là độ tin cậy về sự kiện e trước khi ta nhận được dữ liệu, vì vậy gọi nó là xác suất tiên nghiệm (*prior probability*) của e hoặc độ tin cậy tiên nghiệm vào e . Chẳng hạn như, lấy lại ví dụ e là khẳng định "*tới tháng 5 năm 2009, có tới hơn 235 triệu Website tên Internet*" thì $P(e)$ phản ánh độ tin tưởng của chúng ta là khẳng định này là đúng. Khi ta nhận được số liệu thống kê vào thời điểm bắt đầu tháng 6/2009 từ các máy tìm kiếm thông dụng, ta coi là có điều kiện (dữ liệu) D . Có thể xem xét số liệu về số lượng Website nhận được qua kết quả thống kê từ các máy tìm kiếm như các cột dưới của số lượng thực sự các Website trên Internet. Đến thời điểm đó, xác suất $P(e|D)$ phản ánh độ tin tưởng hậu nghiệm vào e (xác suất hậu nghiệm) được cập nhật trong điều kiện dữ liệu D có được tính toán theo công thức (3.15) thông qua định lý Bayes.

Về phái của công thức (3.15) chứa xác suất tiên nghiệm, thoả mãn đương nhiên xác suất hậu nghiệm tương ứng với xác suất tiên nghiệm. Về này còn chứa xác suất $P(D|e)$, được gọi là khả năng (*likelihood*) của dữ liệu, tức là xác suất dữ liệu xảy ra dưới giả thiết là sự kiện e đã đúng (biên cỗ e đã xuất hiện). Để tính toán được đại lượng likelihood, bắt buộc phải thừa nhận một mô hình xác suất gắn kết với mệnh đề e đang được quan tâm với dữ liệu D đã cho. Đây là bản chất của học máy xác suất.

Trong ví dụ về việc đưa ra số lượng Website, cần chỉ ra một mô hình cho phép cung cấp một phân bố xác suất về số lượng Website mà mỗi máy tìm kiếm trong điều kiện biên cỗ có điều kiện là đúng, theo nghĩa thực tế có hơn 235 triệu Website tồn tại tới trước ngày 1/6/2009. Mô hình này có thể là *phức tạp* khi mô tả cách thức các máy tìm kiếm tìm ra các Website hoặc có thể là *đơn giản* khi chỉ là một hàm xấp xỉ số lượng tổng cộng các Website trên Internet theo số lượng Website mà mỗi máy tìm kiếm phát hiện được.

Trong công thức (3.15), đại lượng likelihood $P(D|e)$ phản ánh độ tương tự ra sao của dữ liệu nhận được, khi đã cho e và một mô hình gắn kết e với dữ liệu. Nếu $P(D|e)$ là rất nhỏ, điều đó được giải thích rằng, mô hình được gắn một xác suất nhỏ tới dữ liệu nhận được. Chẳng hạn, nếu như số lượng các Website được các máy tìm kiếm thu thập theo giá thiết chí có giá trị là vài triệu so với số lượng hàng trăm triệu thì khả năng e có xác suất nhỏ.

Có thể sử dụng giả thiết thay thế e bằng biến cỗ ngược e^* , trong đó phải tin cậy là cả hai điều kiện

$$P(e) + P(e^*) = 1 \text{ và } P(e|D) + P(e^*|D) = 1$$

được đảm bảo như các tiên đề cơ sở của xác suất. Hàng số "chuẩn hoá" Bayes của công thức (3.15) có thể tính được với lưu ý rằng

$$P(D) = P(D|e)P(e) + P(D|e^*)P(e^*)$$

Rõ ràng là $P(e|D)$ phụ thuộc vào cả xác suất tiên nghiệm và likelihood theo nghĩa "cạnh tranh" với giả thiết ngược liên quan tới xác suất tiên nghiệm của e^* và likelihood đối với e^* , thì độ tin tưởng hậu nghiệm sẽ được tăng lên.

Do các xác suất cỗ giá trị rất nhỏ và phép cộng để thực hiện hơn phép nhân, người ta thường sử dụng dạng logarithm hoá. Logarithm hai vê của (3.15), ta có:

$$\log(P(e|D)) = \log(P(D|e)) + \log(P(e)) - \log(P(D)) \quad (3.16)$$

Để áp dụng được công thức (3.15) hoặc (3.16) tới bất kỳ lớp nào của mô hình, chỉ cần đặc tả được xác suất tiên nghiệm $P(e)$ và giá trị likelihood $P(D|e)$ của dữ liệu.

Sau khi cập nhật độ tin tưởng vào biến cỗ e , từ $P(e)$ tới $P(e|D)$, có thể tiếp tục quá trình này với việc hợp nhất với dữ liệu mới khi chúng xuất hiện. Ví dụ, sau này nhận thêm dữ liệu về số lượng Website trên Internet từ các khao sát khác, và gọi tập dữ liệu mới này là D_2 . Sử dụng luật Bayes, nhận được công thức:

$$P(e|D, D_2) = \frac{P(D_2|e, D)P(e|D)}{P(D_2|D)} \quad (3.17)$$

Qua so sánh hai công thức (3.17) và (3.16), nhận thấy rằng, xác suất hậu nghiệm cũ $P(e|D)$ lại đóng vai trò xác suất tiên nghiệm mới khi tập dữ liệu mới D_2 xuất hiện.

Sử dụng xác suất tiên nghiệm là điểm mạnh của tiếp cận Bayes, do nó cho phép sử dụng tri thức tiên nghiệm với việc tinh chỉnh vào quá trình mô hình hoá. Nói chung, hiệu lực của xác suất tiên nghiệm giảm khi số lượng của các điểm dữ liệu tăng. Về mặt hình thức, điều đó được giải thích là già

trị log-likelihood $\log P(D|e)$ tăng một cách tuyến tính theo số lượng các điểm dữ liệu trong D , trong khi đó giá trị tiên nghiệm $\log P(e)$ vẫn là hằng số. Điểm chú ý cuối cùng song lại rất quan trọng là, năng lực của các tiên nghiệm khác nhau, giống như mô hình khác nhau và các lớp mô hình, có thể được ẩn định nội tại trong khung cảnh Bayes bằng cách so sánh các xác suất tương ứng.

Việc tính toán giá trị likelihood (hoặc log-likelihood) thường là một nội dung chính khi giới thiệu về một mô hình cụ thể. Một số nội dung chi tiết về tính toán giá trị này được trình bày tại Chương 9.

3.2.3. Ước lượng giá trị tham số

Trong toán học ứng dụng, việc giải quyết một bài toán có thể được phân thành hai bài toán con chính như sau:

– Lựa chọn loại mô hình hay dạng hàm. Loại mô hình thường được biểu diễn qua một bộ tham số. Chẳng hạn, nếu bài toán phù hợp với phân bố chuẩn thì bộ tham số là kỳ vọng và độ lệch bình phương trung bình. Trong trường hợp bài toán phù hợp với loại mô hình máy vector hỗ trợ (SVM), thì bộ tham số chính là các hệ số của hàm phân biệt [Vap98]. Việc lựa chọn kiểu mô hình thường dựa trên tri thức về miền ứng dụng mà chúng ta có được, trong đó phải kể đến tập dữ liệu đã quan sát được. Các quy luật tự nhiên, các loại mô hình tương ứng với bài toán tương tự được xem xét, các giả thiết thống kê được kiểm nghiệm, trong đó kiểm nghiệm giả thiết thống kê là một trong những phương pháp tốt trong việc lựa chọn loại mô hình. Đây là một dẫn chứng về mối quan hệ mật thiết giữa khai phá dữ liệu với thống kê toán học như đã nêu tại Chương 1. Kết quả đánh giá về tính khớp của loại mô hình với tập dữ liệu quan sát được là một cơ sở quan trọng để chọn lựa loại mô hình phù hợp với bài toán.

– Sau khi đã chọn được loại mô hình phù hợp với bài toán ứng dụng, cần lựa chọn từ loại mô hình đó ra mô hình thích hợp nhất với bài toán ứng dụng. Bài toán con này thường đưa đến việc giải quyết một bài toán tối ưu là chọn ra bộ tham số phù hợp nhất với miền tri thức đã có, mà thường là phù hợp nhất với tập dữ liệu quan sát được. Phương pháp bình phương tối thiểu, entropy cực đại, thuật toán Viterbi,... là các phương pháp thường được lựa chọn.

Trong một số trường hợp khác, đường như không có sự phân định rõ ràng vi hai bài toán con lựa chọn loại mô hình và xác định mô hình cụ thể được tiến hành pha trộn nhau.

Trong [BFS03], Pierre Baldi và cộng sự cho rằng, một bài toán cơ bản trong định hướng ứng dụng lý thuyết xác suất vào các mô hình khai phá dữ liệu là bài toán ước lượng tham số θ theo các giá thiết của một dạng hàm cụ

thể đối với một mô hình M. Chẳng hạn như, nếu mô hình theo phân bố Gauxo thì cần phải ước lượng giá trị hai tham số mô hình Gauxo chính là giá trị trung bình và độ lệch chuẩn.

Như đã được trình bày về mô hình xác suất Bayes, ta nhận được mối liên quan giữa xác suất tiên nghiệm, giá trị likelihood và xác suất hậu nghiệm với các tham số mô hình θ . Kiểu diễn hình tham số θ là một tập các số thực đặc trưng cho tập dữ liệu đã quan sát (thu nhận) được. Như vậy, cả xác suất tiên nghiệm $p(\theta)$ và xác suất hậu nghiệm $P(\theta|D)$ được xác định thông qua các hàm mật độ xác suất theo tập các số thực hiện có. Từ quan điểm của xác suất Bayes, các xác suất tiên nghiệm thể hiện độ tin cậy theo ngũ canh của θ .

Mục tiêu của công việc ước lượng tham số từ dữ liệu là tìm ra hay xấp xỉ tập "tối đa" các tham số cho mô hình, có nghĩa là, tìm tập tham số θ làm cực đại giá trị xác suất hậu nghiệm $P(\theta|D)$ hoặc $\log P(\theta|D)$. Điều đó được gọi là ước lượng cực đại hậu nghiệm (maximum a posteriori: MAP). Thực hiện công việc này tương đương với đi tìm cực tiểu của $-\log P(\theta|D)$:

$$\epsilon(\theta) = -\log P(\theta|D) = -\log P(D|\theta) - \log P(\theta) + \log P(D) \quad (3.18)$$

Xác suất $P(D)$ không phụ thuộc tham số, vì vậy nhận được [BFS03]:

$$\epsilon(\theta) = -\log P(D|\theta) - \log P(\theta) \quad (3.19)$$

hoặc theo thủ tục ước lượng kỳ vọng hậu nghiệm (the mean posteriori: MP) là đơn giản hơn. Tìm giá trị tham số θ làm cực tiểu hàm:

$$\epsilon(\theta) = -\log P(D|\theta) \quad (3.20)$$

Trong nhiều trường hợp, hàm số cần được tối ưu rất phức tạp, không tính toán được theo giải tích. Khi đó một số phương pháp thống kê có thể được áp dụng như giảm gradient, cực đại kỳ vọng, tinh luyện mô phỏng. Cũng có thể chỉ cần tìm lời giải xấp xỉ hoặc lời giải tựa tối ưu về việc tìm ra lời giải tối ưu không thực hiện được về tính toán (thời gian giải quá dài, không chấp nhận được). Trong nhiều trường hợp, ước lượng kỳ vọng hậu nghiệm có hiệu lực tốt trong tính toán xác suất hậu nghiệm $P(\theta|D)$.

3.3. Thuật toán Viterbi

Thuật toán Viterbi được ứng dụng khá phổ biến để giải quyết bài toán giải mã. Khi sử dụng phương pháp máy trạng thái hữu hạn, đặc biệt đối với bài toán trích chọn thông tin trên Web. Nội dung thuật toán có sự kết hợp các nội dung của đồ thị và xác suất.

Thuật toán Viterbi mang tên tác giả Andrew Viterbi, là một thuật toán quy hoạch động nhằm tìm dãy tương tự nhất của các trạng thái ẩn, được gọi là đường đi Viterbi – cho ra kết quả là dãy các sự kiện gắn kết. Thuật toán

được ứng dụng rộng rãi trong các phương pháp máy trạng thái hữu hạn, đặc biệt đối với bài toán trích chọn thông tin. Thuật toán tiền "forward" là thuật toán đi kèm với thuật toán Viterbi thực hiện việc tính toán xác suất một dãy các sự kiện gắn kết.

Thuật toán Viterbi được đánh giá như một sơ đồ chỉnh sửa lỗi trong các đường truyền thông số. Thuật toán cũng rất phổ dụng trong lý thuyết thông tin, nhận dạng tiếng nói, ngôn ngữ tính toán, tin - sinh học,... Chẳng hạn, trong bài toán chuyển bài nói sang văn bản của lĩnh vực nhận dạng tiếng nói, tín hiệu âm thanh như dây gắn kết các sự kiện, và một xâu văn bản được xem xét như một "nguồn ẩn" của tín hiệu âm thanh. Thuật toán Viterbi tìm ra xâu văn bản giống nhất được cho với tín hiệu âm thanh.

Thuật toán cung cấp một cách thức hiệu quả để tìm ra dây trạng thái giống nhất theo nghĩa làm cực đại xác suất hậu nghiệm từ một quá trình được giả thiết là một quá trình Markov thời gian rời rạc trạng thái hữu hạn (*finite-state discrete-time Markov*). Những quá trình như vậy được gộp vào khung thống kê của lý thuyết quyết định hỗn hợp. Dưới đây là một số nội dung về lý thuyết này trong ngữ cảnh nhận dạng văn bản.

Lý thuyết quyết định hỗn hợp: Giả sử ta có một văn bản gồm n ký tự. Mỗi ký tự mang tới một vector đặc trưng z_i , $i = 1, 2, \dots, n$.

Ký hiệu $p(Z|C)$ là hàm mật độ xác suất của dây vector $Z = z_1, z_2, \dots, z_n$ phụ thuộc có điều kiện dây các định danh $C = c_1, c_2, \dots, c_n$, trong đó z_k là vector đặc trưng đối với ký tự thứ k , còn c_k lấy giá trị trên M giá trị (số các chữ cái trong bảng chữ) với $k = 1, 2, \dots, n$.

Ký hiệu $P(C)$ là xác suất tiên định (prior probability) của dây các giá trị C , nói khác đi, $P(C)$ là phân bố xác suất tiên định của tất cả các dây có n ký tự. Xác suất phân lớp chính xác văn bản được cực đại hoá nhờ chọn dây ký tự mà có xác suất hậu nghiệm cực đại hay còn được gọi là cực đại hậu nghiệm (MAP) xác suất $P(C|Z)$.

Từ quy tắc xác suất có điều kiện Bayes, ta có

$$P(C|Z) = \frac{p(Z|C)p(C)}{p(Z)} \quad (3.21)$$

Do $p(Z)$ là độc lập đối với dây C ($p(Z)$ chỉ là một thừa số vô hướng), nên chỉ cần lâm cực đại hàm phân biệt:

$$g_c = p(Z|C)p(C) \quad (3.22)$$

Dung lượng lưu trữ đối với các xác suất này là không lồ về thực tế, vì vậy cần có các giả thiết nhằm rút gọn bài toán tới kích thước có thể quản lý được. Các giả thiết sau đây được đặt ra:

- Độ dài của dãy các quan sát (observation – thể hiện) không lớn. Gọi n là độ dài của một từ khoá. Do đó, $P(C)$ là tần suất xuất hiện của các từ khoá.

- Tính độc lập có điều kiện giữa các vector đặc trưng. Hình dạng của ký tự sinh ra vector đặc trưng đã cho là độc lập với hình dạng của các ký tự kế cận, và như vậy, chỉ phụ thuộc ký tự trong truy vấn.

Do các giả thiết này và các điều đã được giải thích, công thức (3.22) được rút gọn thành:

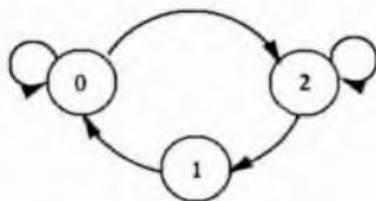
$$g_c(Z) = \sum_{i=1}^n \log p(z_i | c_i) + \log P(c_1, \dots, c_n) \quad (3.23)$$

Trong trường hợp thuật toán Viterbi, nếu bổ sung thêm giả thiết quá trình là Markov bậc 1 thì công thức (3.23) được rút gọn thành:

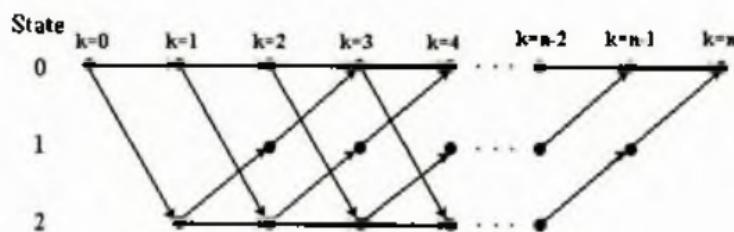
$$g_c(Z) = \sum_{i=1}^n \log p(z_i | c_i) + \log [P(c_1 | c_0) + P(c_2 | c_1) + \dots + P(c_n | c_{n-1})] \quad (3.24)$$

Bài toán định giá trị dãy cực đại hậu nghiệm MAP cũng có thể được xem xét như bài toán tìm đường đi ngắn nhất đọc theo đồ thị. Sự tương tự này cho thấy tính dễ quy luật tự nhiên của thuật toán Viterbi.

Nhằm minh họa cho cách thức thuật toán Viterbi đạt được đường đi ngắn nhất này, cần trình bày quá trình Markov theo cách dễ hiểu nhất. Một sơ đồ trạng thái, chẳng hạn như trong Hình 3.7 thường được sử dụng. Trong sơ đồ trạng thái này, các nút (hình tròn) là các trạng thái, các cung có hướng là các chuyển; và theo thời gian, quá trình phát hiện ra đường đi đọc theo các trạng thái (tuyến suốt sơ đồ).



Hình 3.7. Sơ đồ chuyển trạng thái của quá trình ba trạng thái



Hình 3.8. Dản đồ với quá trình ba trạng thái tại Hình 3.7

Mô tả quá trình trên được chỉ dẫn trong Hình 3.8, được gọi là *dàn* (*trellis*). Trong một dàn, mỗi nút tương ứng với một trạng thái riêng biệt tại thời điểm t cho và mỗi cung có hướng biểu diễn một chuyển từ trạng thái cũ tại thời điểm mới. Dàn bắt đầu và kết thúc tại các trạng thái đã biết c_0 và c_n . Tính chất quan trọng nhất của nó là với mọi dây trạng thái có thể có C luôn tương ứng một đường đi duy nhất đọc theo dàn và ngược lại.

Giá sử gần tới mọi đường đi một độ dài tỷ lệ với $-\log[p(Z|C) + P(C)]$. Do hàm $\log()$ là đơn điệu tăng và sự tương ứng một - một giữa các đường đi - dây. Vì vậy, chỉ cần tìm đường đi mà $-\log[p(Z|C) + P(C)]$ là cực tiểu, vì nó sẽ cho dây trạng thái mà $p(Z|C)P(C)$ là cực đại, hay nói khác đi, dây trạng thái với xác suất hậu nghiệm cực đại, và đáp ứng vấn đề cần được giải quyết. Độ dài tổng cộng của đường đi tương ứng với dây trạng thái C là

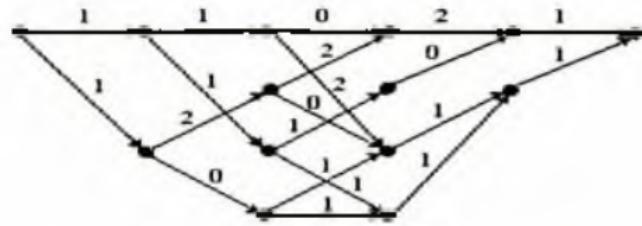
$$-\log[p(Z|C)P(C)] = \sum_{k=1}^n l(t_k).$$

Trong đó $l(t_k)$ là độ dài tương ứng với mỗi bộ chuyển t_k từ c_k tới c_{k+1} . Đoạn đường đi ngắn nhất như vậy được gọi là *vết* (*survivor*) tương ứng với nút c_k và được ký hiệu là $S(c_k)$. Tại mọi thời điểm $k > 0$, có M vết tắt cả, mỗi vết cho một c_k . Quan sát điều này cho thấy đường đi đầy đủ ngắn nhất S phải bắt đầu từ một trong các vết này. Như vậy, tại thời điểm k chỉ cần nhớ M vết $S(c_k)$ và độ dài tương ứng của chúng. Để di tới thời điểm $k+1$, chỉ cần mở rộng mọi vết thời điểm k trong một đơn vị thời gian, tính toán độ dài của các đoạn đường đi mở rộng, và đổi với nút c_{k+1} là vết $k+1$ tương ứng. Đệ quy tiếp tục không hạn định, bỏ qua số vết vượt quá M. Thuật toán này là dạng đơn giản của quy hoạch động tiến.

Mình họa điều này qua một ví dụ liên quan đến một dàn có 4 trạng thái qua 5 đơn vị thời gian. Dàn hoàn thiện với mỗi nhánh được gắn nhãn bằng độ dài được chỉ ra trong Hình 3.9.

Thuật toán Viterbi được coi như tìm đường đi ngắn nhất đọc theo một đồ thị là:

Input $Z = z_1, z_2, \dots, z_n$ //dây quan sát đầu vào
Khởi tạo



Hình 3.9. Dàn được gắn nhãn theo độ dài nhánh, $M = 4$, $n = 5$

```

k ← 1           // chỉ số lặp
S(c1) ← c1
L(c1) ← 0   // biến chứa tổng độ dài, khởi tạo là 0

```

Đệ quy:

repeat

For ∀ bộ chuyển $t_k = (c_k, c_{k+1})$

$L(c_k, c_{k+1}) \leftarrow L(c_k) + l[t_k = (c_k, c_{k+1})]$ theo ∀c_k

Tìm $L(c_{k+1}) = \min L(c_k, c_{k+1})$

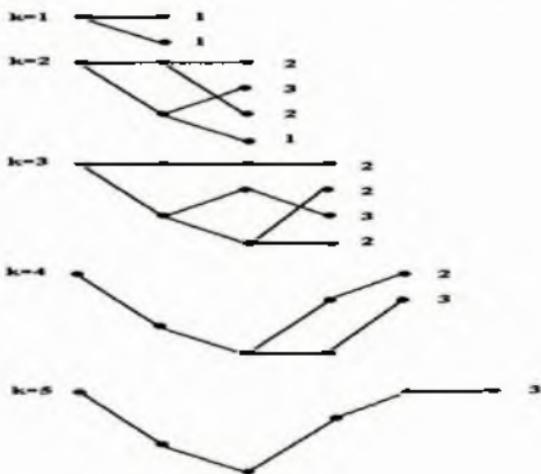
For mỗi c_{k+1}

lưu $L(c_{k+1})$ và vết S(c_{k+1}) tương ứng

$k \leftarrow k + 1$

until $k = n$

5 bước đệ quy khi thực hiện thuật toán xác định đường đi ngắn nhất từ nút khởi động tới nút kết thúc được cho trong Hình 3.10. Tại mỗi bước chỉ có 4 (hoặc ít hơn) các vết được trình bày cùng với độ dài của chúng.



Hình 3.10. Xác định đệ quy đường đi ngắn nhất theo thuật toán Viterbi

Với dãy trạng thái hữu hạn C, thuật toán xác định tại thời điểm n đường đi hoàn thiện ngắn nhất được lưu tại vết S(c_k).

Độ phức tạp tính toán của thuật toán được ước lượng như sau:

– Về mặt bộ nhớ: Thuật toán đòi hỏi M chỗ lưu trữ, trong đó mỗi chỗ cần đủ khả năng bảo quản độ dài L(m) và một vết được cắt tia S(m) các ký tự.

– *Về mặt thời gian tính toán:* Mỗi bước thuật toán thực hiện nhiều nhất M^2 phép cộng và M phép so sánh trong số M^2 kết quả.

Nhu vậy, dung lượng bộ nhớ tỷ lệ với số trạng thái, còn khối lượng tính toán tỷ lệ với số lượng bộ chuyên.

Câu hỏi và bài tập

1. Trình bày mô hình đồ thị, thử tìm ví dụ trong thực tế và bài toán trong khai phá dữ liệu Web mà có thể ứng dụng mô hình này.
2. Trình bày về mạng xã hội. Liệu mạng xã hội có thể biểu diễn bằng mô hình đồ thị hay không?
3. Trình bày mô hình học máy Bayes và tìm một ví dụ về bài toán khai phá Web có thể ứng dụng mô hình này.
4. Trình bày về thuật toán Viterbi và tìm ví dụ trong thực tế có thể áp dụng thuật toán này để giải quyết.

Chương 4

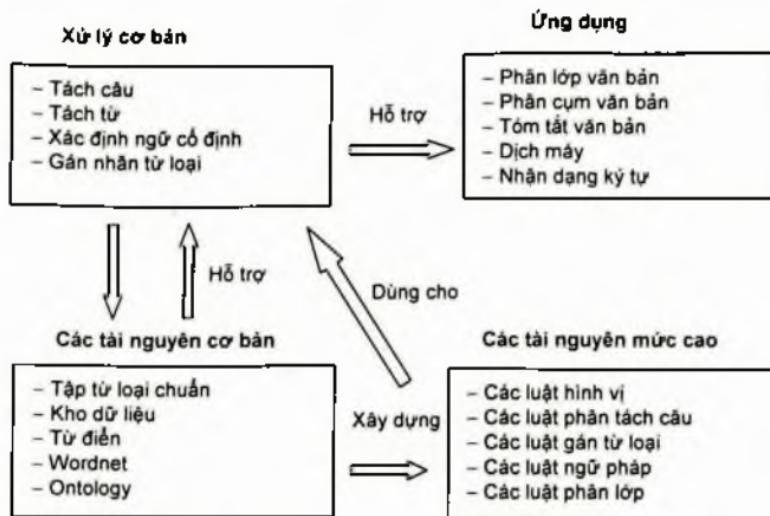
MỘT SỐ VẤN ĐỀ VỀ XỬ LÝ NGÔN NGỮ TIẾNG VIỆT CHO KHAI PHÁ VĂN BẢN

4.1. Giới thiệu

Chương này giới thiệu một số vấn đề cơ bản trong xử lý ngôn ngữ tự nhiên phục vụ cho quá trình khai phá dữ liệu văn bản. Vì văn bản là sự diễn giải ngôn ngữ tự nhiên dưới dạng văn viết và được cấu thành từ những yếu tố cơ bản trong ngôn ngữ như từ, câu, cú pháp; xử lý ngôn ngữ tự nhiên là cơ sở cho các bước khai phá văn bản mức cao hơn.

Xử lý ngôn ngữ tự nhiên [Wi09] kết hợp hai lĩnh vực chính, gồm khoa học máy tính và ngôn ngữ học, nhằm mô tả sự tương tác giữa máy tính và con người thông qua ngôn ngữ. Đây là vấn đề sử dụng máy tính nhằm xác định các đặc trưng cấu trúc ngôn ngữ trong giao tiếp. Dựa trên các "mức ngôn ngữ", có thể phân chia xử lý ngôn ngữ tự nhiên thành xử lý mức thấp và xử lý mức cao. Mức xử lý thấp (xử lý nồng) tập trung phân tích cấu trúc tạo thành từ (hình thái học) hay chức năng ngữ pháp của từ (từ vựng học). Mức xử lý cao phân tích cấu trúc ngữ pháp (cây cú pháp) hay ngữ nghĩa trong câu (phân biệt các từ đồng âm khác nghĩa,...).

Các phương pháp xử lý ngôn ngữ tự nhiên có thể được phân chia thành hai hướng chính: (1) các phương pháp dựa trên luật; (2) các phương pháp thống kê. Hướng tiếp cận dựa trên luật tiến hành phân tích các hiện tượng ngôn ngữ và biểu diễn chúng dưới dạng cấu trúc luật, logic trên cơ sở hiểu biết sâu về ngôn ngữ. Sau đó, các luật này được xử lý dựa trên các thủ tục lập luận logic. Lập luận là quá trình lặp lại gồm có lựa chọn các luật, trong đó phản điêu kiện được thỏa mãn và thực thi luật. Khác với hướng tiếp cận dựa trên luật, hướng tiếp cận dựa trên thống kê không yêu cầu những hiểu biết sâu về ngôn ngữ mà dựa trên những ví dụ quan sát được từ tập dữ liệu để tìm ra các mẫu cơ bản trong ngôn ngữ. Các mô hình thống kê thường được dùng như mô hình Markov ẩn (HMM), mô hình Conditional Random Fields....



Hình 4.1. Vai trò của xử lý ngôn ngữ tự nhiên và các tài nguyên ngôn ngữ cho các ứng dụng trong khai phá dữ liệu văn bản

Cùng với sự phát triển gần đây của các nền văn hóa châu Á, ngôn ngữ của các quốc gia châu Á cũng chia sẻ nhiều đặc điểm chung. Một ví dụ điển hình là nhiều ngôn ngữ châu Á cần phải giải quyết bài toán tách từ (trong tiếng Trung, tiếng Nhật, tiếng Việt,...) do không tồn tại ranh giới giữa các từ hoặc dấu cách không phải là ký tự phân tách từ.

Sự hợp tác nghiên cứu đối với lĩnh vực xử lý ngôn ngữ tự nhiên, khai phá dữ liệu văn bản của các nước châu Á còn được thể hiện trong việc tổ chức các khoá huấn luyện về lĩnh vực này. Khoá huấn luyện ADD (Asian Applied Natural Language Processing for Linguistics Diversity and Language Resource Development)⁽¹⁾ được tổ chức thường niên trong bốn năm (2006 – 2009) không chỉ cung cấp những nội dung đang được quan tâm hiện nay về lĩnh vực xử lý ngôn ngữ tự nhiên mà còn chứng tỏ mối liên hệ chặt chẽ giữa lĩnh vực xử lý ngôn ngữ tự nhiên với khai phá dữ liệu văn bản. Đáng chú ý là, vấn đề bản địa hoá khai phá dữ liệu văn bản là một nội dung quan trọng trong chương trình của các khoá huấn luyện ADD.

Chuỗi các khoá học về xử lý ngôn ngữ châu Á (khoá huấn luyện ADD) có mục tiêu chính là chia sẻ kinh nghiệm giữa các chuyên gia trong lĩnh vực xử lý ngôn ngữ tự nhiên. Các vấn đề được thảo luận bao gồm vấn đề về chuyên giao công nghệ, chia sẻ tài nguyên phát triển và nghiên cứu ngôn ngữ thông qua một loạt các bài giảng và các tài nguyên ngôn ngữ.

⁽¹⁾ <http://203.144.225.124/add/index.php>

4.2. Kho dữ liệu

Kho dữ liệu văn bản [Wi09] là một tập hợp văn bản lớn có cấu trúc. Chúng thường được sử dụng với mục đích phân tích thống kê, kiểm thử giả thiết, hay kiểm tra sự xuất hiện, hoặc đánh giá các luật ngôn ngữ trong một tập hợp xác định.

Một kho dữ liệu có thể lưu trữ văn bản trong một ngôn ngữ (kho dữ liệu đơn ngôn ngữ) hay dữ liệu nhiều ngôn ngữ (kho dữ liệu đa ngôn ngữ). Các kho dữ liệu đa ngôn ngữ được định dạng bằng cách so sánh các thành phần tương ứng giữa các ngôn ngữ này và được gọi là "kho dữ liệu đồng hàng" (ví dụ, kho dữ liệu cho dịch máy).

Các kho dữ liệu thường được gán nhãn để phục vụ cho các nghiên cứu về ngôn ngữ. Một ví dụ về kho dữ liệu được gán nhãn là dữ liệu cho bài toán gán từ loại, trong đó thông tin về các từ loại (động từ, danh từ, tính từ,...) được thêm vào kho dữ liệu dưới dạng thẻ.

| | |
|---|--|
| 1 | 00 : 37 : 00/E↓ |
| 2 | Sương/N dán/V vỗ_dán/V .../... /:↓ |
| 3 | Zg/N 2/E /:↓ Quả_dăm/N và/C nhồng/L giác/N xo/V đổi_dời/V .../↓ |
| 4 | T./Nz /-- Với/E một/E bô_phán/N lớn/A thanh_thiếu_niên/N nóng_thòn/N Thái_Lan/Np /., Muay_Thái/Np có_thể/A là/V lối_thoát/N duy_nhất/A giúp/V hò/P thoát/V khỏi/V cảnh/K nghec/A .../. |

Hình 4.2. Ví dụ về dữ liệu có gán thẻ (VietTreebank)

Một số kho dữ liệu có cấu trúc phức tạp hơn cho những mức phân tích sâu hơn. Đặc biệt, một số kho dữ liệu được xây dựng cho việc phân tích câu, và thường được biểu diễn dưới dạng cây cú pháp, ví dụ: Penn TreeBanks. Các kho dữ liệu có ý nghĩa hết sức quan trọng trong vấn đề học thông kê. Tuy nhiên, việc xây dựng một kho dữ liệu tốt không hề đơn giản, trong đó một trong những vấn đề khó khăn nhất là làm sao đảm bảo được tính nhất quán cho dữ liệu gán nhãn trên toàn bộ kho dữ liệu.

4.3. Quan hệ ngữ nghĩa trong văn bản

Mục này trình bày một nội dung trong xử lý ngôn ngữ tự nhiên có liên quan mật thiết tới nhiều bài toán ứng dụng trong khai phá văn bản, là mối quan hệ ngữ nghĩa trong văn bản. Theo nghĩa hẹp, Birger Hjorland đã định nghĩa *quan hệ ngữ nghĩa* [BIGER]: *Là mối quan hệ về mặt ngữ nghĩa giữa hai hay nhiều khái niệm. Trong đó, khái niệm được biểu diễn dưới dạng từ hay cụm từ.* Ví dụ, có một câu: "Hội Lim được tổ chức ở Bắc Ninh", thì mối

quan hệ ngữ nghĩa giữa hai khái niệm "Hội Lim" và "Bắc Ninh" là "được tổ chức".

Theo Girju [Girju02], một số mối quan hệ ngữ nghĩa quan trọng thường được dùng để thể hiện mối quan hệ giữa các khái niệm như: hyponymy/hypernymy (is - a), meronymy/holonymy (part - whole), synonymy và antonymy.

• **Hyponymy:** Mỗi quan hệ giữa hai từ có tính quan hệ trên - dưới, mà ở đó, một từ luôn bao gồm ngữ nghĩa của từ kia, nhưng không ngược lại. Đây là mối quan hệ ngữ nghĩa cơ bản, được sử dụng với mục đích phân loại những thực thể khác nhau nhằm tạo ra các ontology có phân cấp. Ví dụ, "Động vật" bao gồm "con chó" hoặc "chó" là một loại "động vật".

• **Meronymy:** Là mối quan hệ ngữ nghĩa thể hiện mối quan hệ bộ phận - toàn phần giữa hai khái niệm. Mỗi quan hệ ngược lại được gọi là holonymy. Ví dụ: "tay" là một phần của "cơ thể người" và "cơ thể người" có một phần là "tay" ("human body" is a holonymy of "hand").

• **Synonymy:** Là mối quan hệ đồng nghĩa, trong đó hai từ được xem là đồng nghĩa nếu chúng cùng đề cập tới một khái niệm ngữ nghĩa, hoặc chúng đồng nghĩa với nhau. Ví dụ, "hoa hồng" và "phản trám" đều chỉ về tiền trả cho người làm trung gian, môi giới trong việc giao dịch, mua bán.

• **Antonyms:** Là mối quan hệ đối nghĩa (hai khái niệm trái ngược nhau). Ví dụ, "lạnh - ấm", "mua - bán", "thành công - thất bại"....

Trong WordNet⁽¹⁾, một tập các quan hệ giữa các khái niệm trong CSDL các từ tiếng Anh cũng được xác định. Trong thực tiễn, quan hệ ngữ nghĩa có thể xảy ra giữa nội dung các câu hoặc các đoạn văn bản; chẳng hạn, một đoạn mô tả một vấn đề nào đó, còn đoạn văn khác lại đề cập đến nguyên nhân gây ra vấn đề đó.

Bài toán trích rút quan hệ là một trong những bài toán quan trọng trong lĩnh vực khai phá tri thức mang tầm vóc quốc tế như ACE (Automatic Content Extraction)⁽²⁾, DARPA EELD (Evidence Extraction and Link Discovery)⁽³⁾, ARDA-AQUAINT (Question Answering for Intelligence), ARDA NIMD (Novel Intelligence from Massive Data) vì ứng dụng của nó rất đa dạng. Ngoài việc làm giàu thêm lượng thông tin, nó được xem là một phương pháp hiệu quả để đưa ra phương pháp xử lý cho các hệ thống như: hệ thống hỏi đáp (Question Answering) [Etzioni05, Girju08, Kamb04], xây dựng cơ sở tri thức (KB construction) [MG01], phát hiện ánh qua đoạn văn

⁽¹⁾ <http://wordnet.princeton.edu>

⁽²⁾ <http://www.itl.nist.gov/iaid/894.01/tests/ace/>.

⁽³⁾ <http://w2.eff.org/Privacy/TIA/eeld.php>

bản (text-to-image generation) [CP01], tìm mối quan hệ bệnh tật – Genes (gene-disease relationships) [HYJ06], ảnh hưởng qua lại giữa protein – protein (Protein – Protein interaction) [HZH04],...

Các phương pháp trích rút quan hệ ngữ nghĩa đã được đề xuất khá phong phú, từ phương pháp học thống kê, có giám sát dựa trên dữ liệu đã gán nhãn, hay các phương pháp dựa trên luật học bản giám sát [Bri98, AG00]. Trong các phương pháp đó, học bản giám sát được xem như là một phương pháp tối ưu để giám thiểu chi phí xây dựng tài nguyên. Hướng tiếp cận chính được sử dụng cho việc học hiện nay thường sử dụng kỹ thuật bootstrapping. Kỹ thuật này nhận đầu vào là một tập nhỏ các hạt giống (seed) của một mối quan hệ cụ thể đã được xác định trước, từ đó tiến hành cho học để trích xuất ra một tập các mẫu quan hệ ngữ nghĩa và tiến hành sinh thêm tập seed mới. Kết quả thu được là một tập dữ liệu lớn biểu diễn mối quan hệ được quan tâm. Dưới đây là một số phương pháp tiêu biểu.

4.3.1. Phương pháp DIPRE

Năm 1998, Sergey Brin⁽¹⁾ đã giới thiệu một phương pháp học bản giám sát, đặt tên là DIPRE (Dual Iterative Pattern Relation Expansion) cho việc trích rút mẫu quan hệ ngữ nghĩa [Bri98]. Phương pháp được thử nghiệm để trích rút mối quan hệ "tác giả" của "cuốn sách", gọi tắt là quan hệ "author – book" với tập dữ liệu ban đầu khoảng 5 ví dụ cho mỗi quan hệ mẫu. Hệ thống DIRPE mở rộng tập ban đầu thành một danh sách khoảng 15.000 cuốn sách.

Phương pháp DIPRE có thể được mô tả như sau:

Đầu vào: Tập các quan hệ mẫu seed S = {<author, book>}}, tập dữ liệu D;

Đầu ra: Tập R các quan hệ trích rút được;

1. $R \leftarrow S$

Tập đích được khởi tạo từ tập mẫu S. Tập quan hệ mẫu S có thể là rất nhỏ;

2. $O \leftarrow \text{FindOccurrences}(R^1, D)$.

Tìm tất cả các xuất hiện của các quan hệ mẫu seed trong tập dữ liệu D;

3. $P \leftarrow \text{GenPatterns}(O)$

Dựa vào tập câu đã tìm được, tiến hành tìm các mẫu quan hệ giữa các thành phần của seed ban đầu. Sergey Brin định nghĩa mẫu ban đầu rất đơn giản, bằng việc giữ lại m ký tự trước thành phần seed đầu tiên gọi là prefix, giữ lại phía sau thành phần thứ hai n ký tự gọi là suffix, và k ký tự nằm giữa hai thành phần này là middle. Bằng một phương pháp đơn giản để giữ lại các prefix, suffix và middle phổ biến (xuất hiện nhiều lần trong dữ liệu). Kết quả nhận được sau bước này là các mẫu dùng để trích rút các

⁽¹⁾ <http://infolab.stanford.edu/~sergey/>

quan hệ mới trong dữ liệu. Mẫu quan hệ được biểu diễn dưới dạng: [order, prefix, suffix, middle], trong đó order thể hiện thứ tự xuất hiện của author và book trong một câu. order = 1 thi author đứng trước book, trường hợp còn lại thì book đứng trước author:

4. $R' \leftarrow M_0(O)$.

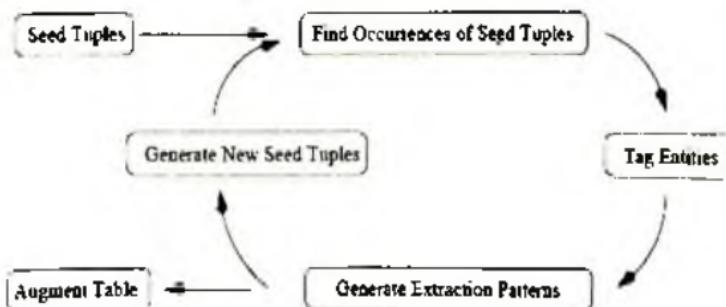
Từng mẫu mới được thu nhận được dùng để trích rút các cặp quan hệ <author, book> mới trong tập dữ liệu D. Ví dụ, với một mẫu [order, prefix, suffix, middle] có order = 1, thi một cặp quan hệ <author, book> được trích rút nếu có một câu khớp với mẫu: prefix * middle * suffix, trong đó chuỗi ký tự khớp với dấu * đầu tiên đóng vai trò là author, chuỗi ký tự khớp với dấu * thứ hai đóng vai trò là book. Trường hợp order = 0 thi cặp quan hệ mới <author, book> được trích rút, trong đó chuỗi ký tự khớp với dấu * đầu tiên đóng vai trò là book, chuỗi ký tự khớp với dấu * thứ hai đóng vai trò là author. Bổ sung các cặp quan hệ thu được vào R;

5. Khi R đủ lớn thi dừng lại, ngược lại quay lại bước 2 để tìm ra những cặp quan hệ và mẫu mới.

Phương pháp của Sergey Brin tuy rất đơn giản nhưng đã được cải tiến trong nhiều nghiên cứu để trích rút các cặp quan hệ khác trong dữ liệu văn bản.

4.3.2. Phương pháp Snowball

Cũng dựa trên ý tưởng bootstrapping của phương pháp DIPRE, Snowball là hệ thống trích rút mỗi quan hệ dựa trên một tập nhỏ dữ liệu quan hệ mẫu làm nhân. Sau đó trong quá trình thực hiện tập mẫu và tập quan hệ mới được sinh ra cần được đánh giá chất lượng [AG00]. Giải thuật được thực nghiệm trên mỗi quan hệ "tổ chức – địa điểm", thể hiện trụ sở chính của "tổ chức" có vị trí thuộc "địa điểm" với seed ban đầu như Microsoft – Redmond, IBM – Armonk, Boeing – Seattle, Intel – Santa Clara. Kiến trúc của Snowball được minh họa trong Hình 4.3.



Hình 4.3. Kiến trúc của hệ thống Snowball

Nội dung cơ bản của phương pháp Snowball được trình bày như sau:

Đầu vào: Một tập văn bản D (tập huấn luyện); tập nhán seed ban đầu S gồm các cặp quan hệ mẫu nào đó. Ví dụ, về cặp quan hệ là <ORGANIZATION, LOCATION> như đề cập ở trên. Mỗi cặp quan hệ bao gồm hai thực thể A, B có mối quan hệ với nhau theo dạng <A, B> hay <thực thể 1, thực thể 2>, như vậy S = {<A_i, B_j>}.

Đầu ra: Tập R các quan hệ trích rút được;

Bước 1: *Tìm sự xuất hiện của các cặp quan hệ trong dữ liệu*

$$R = S;$$

Với mỗi cặp quan hệ <A, B> trong S, tìm trong D tất cả các câu có chứa cả A và B. Tiến hành phân tích, chọn lọc và rút trích các mẫu. Tương tự như DIPRE, một câu khớp với biểu thức ' A " B ' thì cụm từ đứng trước A gọi là *left*, cụm từ đứng giữa A và B gọi là *middle* và cụm từ đứng sau B gọi là *right*. Một ví dụ về mẫu Snowball là bộ 5 <{<the, 0.2>}, LOCATION, {<-, 0.5>, <based, 0.5>}, ORGANIZATION, {}> trong đó <{<the, 0.2>}> là *left*, <-, 0.5>, <based, 0.5> là *middle* và {} là *right*. Các hệ số 0.2, 0.5, 0.5 là trọng số ngữ cảnh có giá trị bằng tần số xuất hiện của các thành phần left, middle và right của mẫu trong tập dữ liệu D. Các giá trị này được sử dụng để tính độ tương đồng giữa các mẫu.

Bước 2: *Tìm sự xuất hiện của các thực thể trong dữ liệu*

Tiến hành phân cụm tập mẫu. Snowball sử dụng hàm Match để tính độ tương đồng giữa các mẫu và xác định ngưỡng tương đồng *t_{th}* cho việc gom cụm nhằm làm giảm số lượng các mẫu cũng như làm cho mẫu có tính khái quát cao hơn. Độ tương đồng giữa 2 mẫu được biểu diễn bằng hàm Match(mẫu1, mẫu2):

$$\begin{aligned} \text{Match}(\text{mẫu1}, \text{mẫu2}) = & (\text{wleft1} \cdot \text{wleft2}) + (\text{wmiddle1} \cdot \text{wmiddle2}) \\ & + (\text{wright1} \cdot \text{wright2}) \end{aligned}$$

trong đó (wleft1, wmiddle1, wright1) và (wleft2, wmiddle2, wright2) là hệ số ngữ cảnh tương ứng của mẫu1 và mẫu2.

Các mẫu sau khi tìm thấy được dùng để trích rút các cặp quan hệ mới trong D, sau đó đổi chiều từng cặp quan hệ mới tìm thấy <A', B'> với tập R để kiểm tra chất lượng của mẫu. Từ đó chọn ra các mẫu mới có độ chính xác cao. Cặp quan hệ mới <A', B'> thuộc một trong các trường hợp sau:

Positive: Nếu <A', B'> đã nằm trong tập R;

Negative: Nếu <A', B'> chỉ có đúng một trong hai (A' hoặc B') xuất hiện trong tập R;

Unknown: Nếu <A', B'>, cả A', B' đều không xuất hiện trong tập R. Tập Unknown được xem là tập các quan hệ mới cho vòng lặp sau.

Bước 3: Sinh mẫu mới

Thủ tục sinh mẫu mới được trình bày tại Hình 4.4.

Snowball tính độ chính xác của từng mẫu dựa trên số Positive và Negative của nó và chọn ra N mẫu có điểm số cao nhất. Độ tin tưởng của mẫu được tính theo công thức:

$$\text{belief}(P) = \frac{P_{\text{positive}}}{(P_{\text{positive}} + P_{\text{negative}})}$$

Bước 4: Tìm các seed mới

Với mỗi mẫu trong danh sách top N thu nhận được lại được bổ sung vào tập mẫu để trích rút tập R' các cặp quan hệ mới trong tập dữ liệu D. Bổ sung tập các quan hệ vào tập R: $R = R \cup R'$. Để làm tăng tính chính xác cho hệ thống, các cặp quan hệ trong tập R' được đánh giá để lựa chọn ra M cặp được đánh giá tốt nhất và M cặp này được dùng làm seed cho quá trình rút mẫu kế tiếp. Hệ thống tiếp tục được quay lại bước 1. Quá trình trên tiếp tục lặp cho đến khi hệ thống không tìm được cặp mới hoặc lặp theo số lần mà ta xác định trước.

```

sub GenerateTuples(Patterns)
    foreach text_segment in corpus
        (1) {< o, t >, < l1, l2, m1, m2, t1, t2, r1 >} =
            - CreateOccurrence(text_segment);
            Tc = < o, t >;
            SimBest = 0;
            foreach p in Patterns
                sim = Match(< l1, l2, m1, m2, t1, t2, r1 >, p);
                if (sim ≥ τsim)
                    (2) UpdatePatternSelectivity(p, Tc);
                    if (sim ≥ SimBest)
                        SimBest = sim;
                        Pbest = p;
                if (SimBest ≥ τsim)
                    CandidateTuples[Tc].Patterns[Pbest] =
                        SimBest;
    return CandidateTuples;

```

Hình 4.4. Thủ tục sinh mẫu mới của Snowball [AG00]

4.3.3. Trích rút quan hệ ngữ nghĩa trong Tiếng Việt

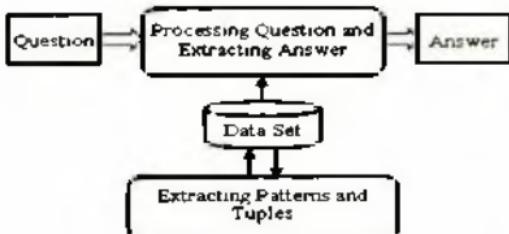
Trong hệ thống hoi – đáp [VVU09], Vũ Trần Mai và cộng sự trình bày nghiên cứu thử nghiệm một hệ thống hoi – đáp tự động tiếng Việt (Hình 4.5a) với pha cần quan tâm đầu tiên là pha trích chọn mẫu và seed (quan hệ ngữ nghĩa, Hình 4.5b). Các tác giả sử dụng phương pháp kết hợp giữa phương pháp Snowball [AG00] và phương pháp sử dụng máy tìm kiếm [Hov02].

Một số ví dụ về seed cho trong bảng dưới đây:

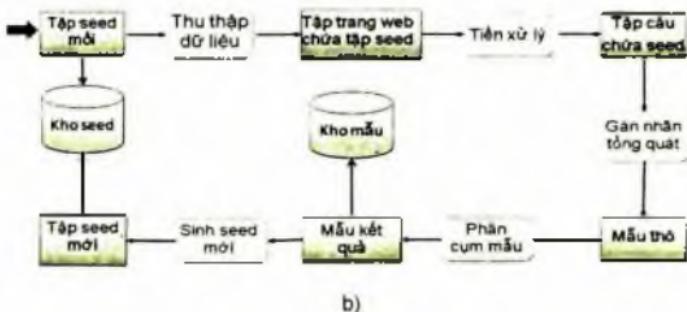
| Mối quan hệ | Phân đầu của seed | Phân cuối của seed |
|----------------------|-------------------|--------------------|
| <Bãi biển, Địa điểm> | Quảng Bình | Nam Định |
| <Bãi biển, Địa điểm> | Hải Phòng | Nam Định |
| <Lễ hội, Địa điểm> | Hội Phù Giầy | Nam Định |
| | | |

Tương tự, dưới đây là một số mẫu tổng quát:

| Mô hình quan hệ | Mẫu tổng quát |
|----------------------|--|
| <Bãi biển, Địa điểm> | <BÃI BIỂN> bãi – biển thuộc <ĐỊA ĐIỂM> |
| <Bãi biển, Địa điểm> | <ĐỊA ĐIỂM> có bãi – biển <BÃI BIỂN> |
| <Bãi biển, Địa điểm> | |
| <Lễ hội, Địa điểm> | <LỄ HỘI> khai_mạc tại <ĐỊA ĐIỂM> |
| <Lễ hội, Địa điểm> | <ĐỊA ĐIỂM> tổ_chức LỄ_HỘI <LỄ HỘI> |
| <Lễ hội, Địa điểm> | |



a)



Hình 4.5. (a) Mô hình hệ thống hỏi – đáp tiếng Việt (VVU09):

(b) Pha trich rút quan hệ ngữ nghĩa (các mẫu và seed)

Quy trình thi hành cần thực hiện theo mô tả:

Đầu vào: Tập các quan hệ nhân ban đầu, các quan hệ gồm hai thành phần <thực thể 1, thực thể 2>.

Đầu ra: Tập các cặp quan hệ trích rút được và tập các mẫu trích rút.

Bước 1. Thu thập dữ liệu

Nhằm tận dụng miền tri thức nên lớn từ các máy tìm kiếm như Google, Yahoo, Altavista,... Với đầu vào là một tập seed ban đầu được xây dựng bằng tay, thông qua máy tìm kiếm ta tìm được một tập các trang Web có chứa đầy đủ hai thành phần của tập seed này.

Bước 2: Tiền xử lý

- Loại bỏ thẻ HTML, lấy nội dung chính của từng trang Web.
- Tách câu trên tập dữ liệu thu được và giữ lại những câu chứa cả hai thành phần của seed.
- Tách từ, loại bỏ từ dừng cho lập câu này.
- Áp dụng phương pháp sinh tự động tập thực thể để mở rộng tập thực thể từ những thực thể ban đầu cho từng mối quan hệ đã được xác định trước các nhãn thực thể. Phương pháp này được trình bày ở phần tiếp theo

Bước 3: Gán nhãn tổng quát

- Dưa vào tập thực thể mở rộng, tiến hành tìm và xác định nhãn cho các thực thể có chứa trong tập câu thu được ở bước trên.
- Sau khi các thực thể được gán nhãn, xác định các thành phần trái, thành phần phải, thành phần giữa cho các thực thể có chứa trong tập seed dựa vào tập câu thu được
- Biểu diễn các thành phần trái, thành phần phải và thành phần giữa dưới dạng các vector, ta thu được một tập các mẫu thô.

Bước 4: Phân cụm mẫu.

- Tiến hành so khớp các thành phần trái, thành phần phải và thành phần giữa cho các mẫu thô để loại bỏ các mẫu thô trùng.
- Dựa theo phương pháp Snowball, xác định các mẫu quan hệ được thực hiện bằng việc phân cụm mẫu thô. Mỗi cụm đại diện bởi một mẫu và quá trình phân cụm mẫu được thực hiện như sau: Với những mẫu thô mới được sinh ra, tiến hành tính độ tương đồng với các mẫu đại diện theo công thức sau

$$\text{Match } (mẫu1, mẫu2) = (\text{prefix1 prefix2}) + (\text{suffix1 suffix2}) \\ + (\text{middle1 middle2})$$

- Nếu độ tương đồng vượt qua một ngưỡng xác định, thì mẫu thô đó thuộc vào nhóm có độ tương đồng với nó cao nhất. Ngược lại, mẫu đó là đại diện cho một nhóm mới được sinh ra

Bước 5: Sinh seed mới

- Những mẫu tổng quát đã thu nhận được là đầu vào cho máy tìm kiếm để tìm ra tập các câu có chứa các mẫu đó.
- Nhận dạng các thực thể có chứa trong tập câu dựa vào tập các thực thể mở rộng.
- Kiểm tra độ tin cậy của các seed mới được sinh ra. Những seed vượt qua được giá trị ngưỡng thì giữ chúng lại.
- Sau đó quay lại bước 1, sử dụng tập seed mới thu được cùng với tập seed ban đầu đưa vào máy tìm kiếm để tiến hành sinh tập seed mới và tìm thêm tập mẫu quan hệ mới cho mỗi quan hệ đó. Vòng lặp dừng khi seed mới hoặc mẫu mới không còn được tiếp tục sinh ra

Với tập seed và mẫu mới được sinh ra sau mỗi vòng lặp, việc đánh giá độ chính xác của chúng được sử dụng theo phương pháp Snowball.

Đề xuất giải thuật này được các tác giả thử nghiệm để trích rút cắp quan hệ "sự kiện" được tổ chức tại "địa điểm" (ký hiệu là <"sự kiện", "địa điểm">) trên Internet và đã thu được kết quả rất kha quan. Ngoài ra, các cắp quan hệ trích rút được cũng được tác giả ứng dụng vào hệ thống hỏi đáp tiếng Việt.

4.4. Xử lý ngôn ngữ tiếng Việt

4.4.1. Một số đặc trưng tiếng Việt

a) Cấu tạo của âm tiết

Âm tiết là đơn vị phát âm tự nhiên nhỏ nhất của lời nói, được cấu tạo bằng một hoặc một loạt động tác cầu âm, làm thành một thể thống nhất chặt chẽ về mặt ngữ âm. Biên giới âm tiết có thể trùng hợp với biên giới của các đơn vị có ý nghĩa của ngôn ngữ, nhưng cũng không phải nhất thiết như vậy. Âm tiết còn là một tiêu chí xác định mặt ngữ âm của từ (đơn tiết, song tiết, đa tiết). Đối với tiếng Việt, mỗi âm tiết phát thành một tiếng. Ở chữ Nôm, chữ Quốc ngữ mỗi âm tiết viết rời thành một chữ. Âm tiết tiếng Việt gồm âm đầu, âm đệm, âm chính, âm cuối và thanh diệu; cấu trúc tối thiểu có một âm chính và một thanh diệu. Ví dụ: "à", "hà", "hoà", "hoàn".

Có thể hình dung về cấu tạo âm tiết tiếng Việt trong một mô hình như sau:

Thanh diệu: không (zero), huyền (˘), sắc (ˊ), hỏi (՞), ngã (՞), nặng (՞)

| t Âm đầu | Vần | | |
|-------------|--------|----------|---------|
| | o | a | u |
| | Âm đệm | Âm chính | Âm cuối |
| | | | |

Quan sát ví dụ trên ta thấy, âm tiết tiếng Việt có 3 bộ phận mà người ban ngữ nào cũng nhận ra: thanh diệu, phần đầu và phần sau. Phần đầu của âm tiết được xác định là Âm đầu, vì ở vị trí này chỉ có một âm vị tham gia cấu tạo. Phần sau của âm tiết được gọi là phần Vần. Người Việt chưa biết chữ không cam nhận được cấu tạo của phần vần. Vào lớp 1, trẻ em bắt đầu "danh vần", tức là phân tích, tò mò hợp các yếu tố tạo nên vần, rồi ghép với âm đầu để nhận ra âm tiết. Ví dụ:

$$U + Â + N = UÂN, X + UÂN = XUÂN$$

Các âm đầu vần, giữa vần và cuối vần (U, Â, N) được gọi là Âm đệm, Âm chính và Âm cuối.

O phương diện ngữ pháp từ là đơn vị nhỏ nhất có nghĩa và có thể hoạt động tự do trong câu. Từ tiếng Việt không có hiện tượng biến hình bằng những phụ tố mang ý nghĩa ngữ pháp bên trong từ như các ngôn ngữ Án - Âu.

b) Đặc điểm từ trong tiếng Việt

Đơn vị cơ sở cấu tạo từ là tiếng, tức là những âm tiết được sử dụng trong thực tiễn ngôn ngữ Việt. Tiếng có thể có nghĩa dù rõ, có thể mang nghĩa đã bị phai mờ và tiếng có thể tự mình không có nghĩa. Hơn nữa, 3 hiện tượng này có thể chuyển hoá lẫn nhau.

Tính chất âm tiết (tiếng) là một trong những đặc điểm chỉ phái đặc tính loại hình của ngôn ngữ Việt. Xét ở mặt số lượng tiếng ta có:

- Từ chỉ chứa một tiếng, gọi là từ đơn, như: nhà, đã,...
- Từ gồm nhiều tiếng, phần lớn là 2 tiếng, gọi là từ phức, như: nhà cửa, sạch sẽ....

Nếu xét ở số lượng từ lô (yếu tố nhỏ nhất tham gia cấu tạo từ) tham gia cấu tạo từ thì có sự phân chia như sau:

- Từ chỉ chứa một từ lô, gọi là đơn lô, như: nhà, đúng đinh, ra đi ô,...
- Từ đơn lô gồm nhiều tiếng và có hiện tượng hoà âm tạo nghĩa, gọi là từ láy. Nếu không thi nó thuộc loại ngẫu kết.
- Từ chứa nhiều tự lô, gọi là từ đa lô, như: nhà cửa, xe đạp, sạch sẽ,...
- Từ đa lô nếu có hiện tượng hoà phôi ngữ âm tạo nghĩa thì thuộc kiêu láy. Nếu không thi thuộc loại từ ghép.

4.4.2. Unicode

Unicode [Wi09] là bảng mã chuẩn hoá để biểu diễn hầu hết các văn bản trong các hệ thống chữ viết trên thế giới. Được phát triển và công bố dưới dạng sách như là chuẩn Unicode. Unicode bao gồm hơn 100.000 ký tự, cùng với một tập hợp các bảng mã tham chiếu về ký tự ở dạng trực quan, một phương pháp luận để mã hoá ký tự và một tập các cách mã hoá chuẩn,... Một phần bảng mã Unicode được cho trong Hình 4.6.

UTF-8:

Nếu Unicode là bảng mã quy định giá trị (code point) tương ứng đối với các ký tự, thi UTF-8 là một cách mã hoá các giá trị đó dưới dạng chuỗi byte trong máy tính. UTF-8 được thiết kế để tương thích với chuẩn ASCII. UTF-8 có thể sử dụng từ một (cho những ký tự trong ASCII) cho đến 6 byte để biểu diễn một ký tự.

Chính vì tương thích với ASCII, UTF-8 có lợi thế khi được sử dụng để bổ sung hỗ trợ Unicode cho các phần mềm có sẵn.Thêm vào đó, các nhà

phát triển phần mềm vẫn có thể sử dụng các hàm thư viện có sẵn của ngôn ngữ lập trình C để so sánh (*comparisons*) và xếp thứ tự. Ngược lại, để hỗ trợ các cách mã hoá 16 bit hay 32 bit như ở trên, một số lớn phần mềm buộc phải viết lại, do đó tốn rất nhiều công sức. Một điểm mạnh nữa của UTF-8 là với các văn bản chỉ có một số ít các ký tự ngoài ASCII, hay thậm chí cho các ngôn ngữ dùng bảng chữ cái Latinh như tiếng Việt, tiếng Pháp, tiếng Tây Ban Nha..., cách mã hoá kiểu này cực kỳ tiết kiệm không gian lưu trữ.

| | | | | |
|----------|----------|----------|----------|----------|
| À 256 | ã 257 | À 258 | ã 259 | À 260 |
| Č 266 | ć 267 | Č 268 | ć 269 | Đ 270 |
| È 276 | ë 277 | È 278 | ë 279 | Ę 280 |
| Ğ 286 | ğ 287 | Ğ 288 | ğ 289 | Ğ 290 |

Hình 4.6. Ký tự ở dạng trực quan và mã tương ứng trong bảng mã Unicode

UTF-8 được thiết kế đảm bảo không có chuỗi byte của ký tự nào lại nằm trong một chuỗi của ký tự khác dài hơn. Điều này khiến cho việc tìm kiếm ký tự theo byte trong một văn bản là rất dễ dàng. Một số dạng mã hoá khác (như Shift-JIS) không có tính chất này, khiến cho việc xử lý chuỗi ký tự trở nên phức tạp hơn nhiều. Mặc dù để thực hiện điều này đòi hỏi phải có độ dư (văn bản sẽ dài thêm), nhưng những ưu điểm mà nó mang lại vẫn nhiều hơn. Việc nền dữ liệu không phải là mục đích hướng tới của Unicode và việc này cần được tiến hành một cách độc lập.

Các quy định chính xác của UTF-8 như sau (các số bắt đầu bằng 0x là các số biểu diễn trong hệ thập lục phân):

- Các ký tự có giá trị nhỏ hơn 0x80 sử dụng 1 byte có cùng giá trị.
- Các ký tự có giá trị nhỏ hơn 0x800 sử dụng 2 byte: byte thứ nhất có giá trị 0xC0 cộng với 5 bit từ thứ 7 tới thứ 11 (7th-11th least significant bits); byte thứ hai có giá trị 0x80 cộng với các bit từ thứ 1 tới thứ 6 (1st-6th least significant bits).
- Các ký tự có giá trị nhỏ hơn 0x10000 sử dụng 3 byte: byte thứ nhất có giá trị 0xE0 cộng với 4 bit từ thứ 13 tới thứ 16; byte thứ hai có giá trị 0x80 cộng với 6 bit từ thứ 7 tới thứ 12; byte thứ ba có giá trị 0x80 cộng với 6 bit từ thứ 1 tới thứ 6.
- Các ký tự có giá trị nhỏ hơn 0x200000 sử dụng 4 byte: byte thứ nhất có giá trị 0xF0 cộng với 3 bit từ thứ 19 tới thứ 21; byte thứ hai có giá trị 0x80 cộng với 6 bit từ thứ 13 tới thứ 18; byte thứ ba có giá trị 0x80 cộng với

6 bit từ thứ 7 tới thứ 12; byte thứ tư có giá trị 0x80 cộng với 6 bit từ thứ 1 tới thứ 6.

Hiện nay, các giá trị khác ngoài các giá trị trên đều chưa được sử dụng. Tuy nhiên, các chuỗi ký tự dài tới 6 byte có thể được dùng trong tương lai.

- Chuỗi 5 byte dùng để lưu trữ được mã ký tự chứa đến 26 bit: byte thứ nhất có giá trị 0xF8 cộng với 2 bit thứ 25 và 26, các byte tiếp theo lưu giá trị 0x80 cộng với 6 bit có ý nghĩa tiếp theo.

- Chuỗi 6 byte dùng để lưu trữ được mã ký tự chứa đến 31 bit: byte thứ nhất có giá trị 0xFC cộng với bit thứ 31, các byte tiếp theo lưu giá trị 0x80 cộng với 6 bit có ý nghĩa tiếp theo.

4.4.3. Các bài toán cơ bản trong xử lý tiếng Việt

a) Tách câu

• Khái niệm câu

Quan niệm câu là một chuỗi ký tự kết thúc bởi một dấu chấm (.), (?) hay (!) không thể loại trừ các nhập nhằng, trong đó dấu chấm câu không chỉ là ký hiệu kết thúc câu: một số dùng trong các từ viết tắt, hoặc trong chuỗi số. Tuy nhiên, phương pháp dựa trên kinh nghiệm cơ bản này cho kết quả không tồi: nhìn chung, khoảng 90% các dấu chấm là ký hiệu kết thúc câu [MS99]. Tuy nhiên, cũng cần lưu ý các trường hợp: trong đó các ký hiệu khác có thể được coi là dấu hiệu kết thúc câu. Ví dụ, các dấu câu như hai chấm, chấm phẩy và dấu ngang (‘;’ ‘,’ và ‘—’) có thể theo sau bởi một câu hoàn chỉnh.

• Tách câu dựa trên kinh nghiệm

Những lưu ý về câu cho phép chúng ta tiến hành phân tách câu dựa trên kinh nghiệm như sau:

Bảng 4.1. Thuật toán tách câu dựa trên kinh nghiệm

- Đặt các điểm phân cách câu giả định sau sự xuất hiện của. ? ! (và có thể là ‘;’ –).
- Đặt điểm phân cách câu sau dấu đóng ngoặc kép (nếu có).
- Loại ra một điểm phân cách câu giả định (là dấu chấm) trong các trường hợp sau:
 - + Nếu nó đi sau một từ viết tắt thường không xuất hiện ở cuối câu, nhưng thường đi trước một danh từ riêng, ví dụ như Prof. hay vs.
 - + Nếu nó đi sau một từ viết tắt đã biết và không đi trước một từ viết hoa. Trường hợp này có thể giải quyết đúng hầu hết các trường hợp viết tắt như etc. hoặc Jr. (những từ có thể xuất hiện ở giữa hoặc cuối câu).
- Loại một điểm phân cách câu giả định với ? hay ! nếu nó đi trước một từ không viết hoa.
- Xem xét tất cả các điểm phân cách câu giả định còn lại như các điểm phân cách câu thực sự

• Tách câu dựa trên Maximum Entropy

Phuong H.L. và Vinh H.T. [PV08] đề xuất phương pháp tách câu trong tiếng Việt dựa trên hướng tiếp cận cực đại Entropy. Với một kho dữ liệu được đánh dấu kết thúc câu, ta học ra một mô hình để phân loại các điểm kết thúc câu tiềm năng xem đó có phải thực sự là điểm kết thúc câu hay không. Kết quả tách câu được dẫn trong bài báo vào khoảng 95%.

– Mô hình Maximum Entropy cho tách câu:

Phuong H.L. và Vinh H.T. [PV08] mô hình hoá bài toán tách câu dưới dạng bài toán phân lớp dựa trên Maximum Entropy. Với mỗi ký tự có thể là điểm phân tách câu (".", "?" hay "!"), chúng ta ước lượng xác suất đồng thời của ký tự đó cùng với ngữ cảnh xung quanh (biểu diễn bởi biến ngẫu nhiên c) và biến ngẫu nhiên thê hiện đó có thực sự là điểm phân tách câu hay không (bằng {no, yes}). Xác suất của mô hình được định nghĩa như sau:

$$p(b, c) = \pi \prod_{j=1}^{t-1} \alpha_j^{f_j(b, c)}$$

Ở đây: α_j là các tham số chưa biết của mô hình, mỗi α_j tương ứng với một hàm đặc trưng f_j . Gọi $B = \{\text{no}, \text{yes}\}$ là tập các lớp và C là tập của các ngữ cảnh. Các đặc trưng là các hàm nhị phân $f_j: B \times C \rightarrow \{0, 1\}$ dùng để mã hoá các thông tin ngữ cảnh cần thiết. Xác suất để quan sát được điểm phân tách câu trong ngữ cảnh c được đặc trưng bởi xác suất $p(\text{yes}, c)$. Tham số α_j được chọn là giá trị làm cực đại hàm likelihood của dữ liệu huấn luyện với các thuật toán GIS và HS....

Để phân lớp một ký tự tách câu tiềm năng vào một trong hai lớp {yes, no} – lớp yes nghĩa là đó thực sự là một ký tự phân tách câu, còn no thì là ngược lại, ta dựa vào luật phân lớp như sau:

$$p(\text{yes}|c) = p(\text{yes}, c)/p(c) = p(\text{yes}, c)/(p(\text{yes}, c) + p(\text{no}, c))$$

Ở đây c là ngữ cảnh xung quanh ký tự tách câu tiềm năng đó và bao gồm cả ký tự đang xem xét. Sau đây là những lựa chọn hàm tiềm năng f_j để phân tách câu trong tiếng Việt.

– Lựa chọn đặc trưng:

Các đặc trưng trong maximum entropy mã hoá các thông tin hữu ích cho bài toán tách câu. Nếu đặc trưng xuất hiện trong tập đặc trưng, trọng số tương ứng của nó dùng để hỗ trợ cho tính toán xác suất $p(b|c)$.

Các ký tự tách câu tiềm năng được xác định bằng cách duyệt qua văn bản, xác định các chuỗi ký tự được phân cách bởi dấu cách (còn gọi là token) và chứa một trong các ký tự ".", "?" hay "!" . Thông tin về token và

thông tin ngữ cảnh về token liền trái, phải của token hiện tại được xác định xác suất phân lớn.

Gọi các token chứa các ký tự kết thúc câu tiềm năng là "ứng viên". Phần ký tự đi trước ký tự kết thúc câu tiềm năng được gọi là "tiền tố", phần đi sau gọi là "hậu tố". Vị trí của ký tự kết thúc câu tiềm năng cũng được mô tả trong tập đặc trưng. Tập các ngữ cảnh được xem xét từ chuỗi ký tự được mô tả như dưới đây:

1. Có/không có 1 ký tự trắng trước ký tự kết thúc câu tiềm năng.
2. Có/không có 1 ký tự trắng sau ký tự kết thúc câu tiềm năng.
3. Ký tự kết thúc câu tiềm năng.
4. Đặc trưng tiền tố.
5. Độ dài tiền tố nếu nó có độ dài lớn hơn 0.
6. Ký tự đầu tiên của tiền tố là ký tự.
7. Tiền tố nằm trong danh sách các từ viết tắt.
8. Đặc trưng hậu tố.
9. Token đi trước token hiện tại.
10. Ký tự đầu tiên của token liền trước viết hoa/không viết hoa.
11. Token liền trước nằm trong danh sách các từ viết tắt.
12. Token liền sau.
13. Token ứng viên được viết hoa/không viết hoa.

Từ các mẫu ngữ cảnh trên, có thể rút ra tập ngữ cảnh từ dữ liệu (tập C). Tập ngữ cảnh cùng với nhãn từ dữ liệu tạo ra một tập đặc trưng tương ứng. Xét ví dụ sau để làm rõ sự mối quan hệ giữa ngữ cảnh, đặc trưng:

"Những hacker máy tính sẽ có cơ hội chiếm giải thưởng trị giá 10.000USD và 10.000 đôla Singapore (5.882 USD) trong một cuộc tranh tài quốc tế mang tên "Hackers Zone" được tổ chức vào ngày 13/5/1999 tại Singapore."

Xem xét ký tự kết thúc câu tiềm năng ":" trong token "10.000UDS", từ vị trí này, ta có thể rút ra một số ngữ cảnh sau:

1. Không có ký tự trắng trước ký tự ứng viên.
2. Không có ký tự trắng sau ký tự ứng viên.
3. Ký tự ứng viên là ":"
4. Tiền tố: 10

Từ dữ liệu học này, có thể trích rút ra các đặc trưng như ví dụ dưới đây:

$\{ \text{ký tự trắng trước ứng viên, no} \} = 1$. Ý nghĩa của đặc trưng này là phát biểu: "token không có ký tự trắng trước ứng viên và nhãn là no" là đúng (đặc trưng nhận giá trị 1).

Sau khi ước lượng trọng số đặc trưng (bước huân luyện từ dữ liệu học), ta dựa vào các tham số đó để tính giá trị $p(\text{yes}|\mathbf{c})$. Nếu giá trị này $> 50\%$, nhãn tương ứng với ký tự ứng viên được ghi nhận là "yes" hay ký tự ứng viên thực sự là ký tự phân tách câu.

b) Tách từ

Có thể hiểu đơn giản bài toán tách từ tiếng Việt là cho trước một văn bản tiếng Việt, cần xác định trong văn bản đó ranh giới giữa các từ trong câu. Tuy nhiên, khác với tiếng Anh, tiếng Việt dấu cách trông không phải là ranh giới giữa các từ. Ví dụ, trong câu "tách từ tiếng Việt là một bài toán quan trọng", có thể thấy dấu cách trông không phải là dấu hiệu để nhận ra ranh giới của các từ. Kết quả của quá trình tách từ là thu được một câu, trong đó các âm tiết được kết nối với nhau để tạo thành từ "tách từ tiếng Việt là một bài toán quan trọng".

• Phương pháp khớp tối đa

Tư tưởng của phương pháp khớp tối đa (Maximum Matching) là duyệt một câu từ trái qua phải và chọn từ có nhiều tiếng nhất mà có mặt trong từ điển tiếng Việt. Nội dung thuật toán này dựa trên thuật toán đã được Chih-Hao Tsai giới thiệu năm 1996 [Tsa96]. Thuật toán có 2 dạng sau:

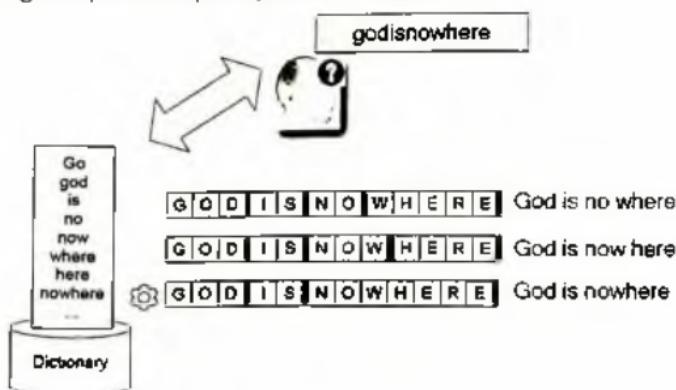
– *Dạng đơn giản*: Giả sử có một chuỗi các tiếng trong câu là t_1, t_2, \dots, t_n . Thuật toán kiểm tra xem t_1 có mặt trong từ điển hay không, sau đó kiểm tra tiếp t_1-t_2 có trong từ điển hay không. Tiếp tục như vậy cho đến khi tìm được từ có nhiều tiếng nhất có mặt trong từ điển và đánh dấu từ đó. Sau đó tiếp tục quá trình trên với tất cả các tiếng còn lại trong câu và trong toàn bộ văn bản. Dạng này khá đơn giản, nhưng nó gặp phải rất nhiều nhập nhằng trong tiếng Việt. Ví dụ, nó bị gặp phải lỗi khi phân đoạn từ câu sau: "học sinh | học sinh | học", câu đúng phải là "học sinh| học| sinh học".

– *Dạng phức tạp*: Dạng này có thể tránh được một số nhập nhằng gặp phải trong dạng đơn giản. Đầu tiên thuật toán kiểm tra xem t_1 có mặt trong từ điển không, sau đó kiểm tra tiếp t_1-t_2 có mặt trong từ điển không. Nếu t_1-t_2 đều có mặt trong từ điển, thì thuật toán thực hiện chiêu thuật chọn 3-từ tốt nhất. Tiêu chuẩn 3-từ tốt nhất được Chen và Liu đề xuất vào năm 1992 như sau [Tsa96]:

Độ dài trung bình của 3 từ là lớn nhất. Ví dụ, chuỗi "cơ quan tài chính" được phân đoạn đúng thành "cơ quan | tài chính", tránh được việc phân đoạn sai thành "cơ | quan tài | chính" vì cách phân đúng phải có độ dài trung bình lớn nhất.

Sự chênh lệch độ dài của 3 từ là ít nhất. Ví dụ, chuỗi "công nghiệp hoá chất phát triển" được phân đoạn đúng thành "công nghiệp | hoá chất | phát triển", thay vì phân đoạn sai thành "công nghiệp hoá | chất | phát triển". Cá

2 cách phân đoạn này đều có độ dài trung bình bằng nhau, nhưng cách phân đoạn đúng có sự chênh lệch độ dài 3 từ ít hơn.



Hình 4.7. Ví dụ lách lử với phương pháp khớp tối đa

Nhận xét: Tuy 2 tiêu chuẩn trên có thể hạn chế được một số nhập nhằng, nhưng không phải tất cả. Ví dụ, với câu "ông già đi nhanh" thì cả 2 cách phân đoạn sau đều có cùng độ dài trung bình và độ chênh lệch giữa các từ: "ông | già | đi | nhanh" và "ông già | đi | nhanh", do đó thuật toán không thể chỉ ra cách phân đoạn đúng được.

Ưu điểm của phương pháp trên có thể thấy rõ là đơn giản, dễ hiểu và chạy nhanh. Hơn nữa, chỉ cần một tập từ điển đầy đủ là có thể tiến hành phân đoạn các văn bản, hoàn toàn không phải trải qua huấn luyện như các phương pháp sẽ được trình bày tiếp theo.

Nhược điểm của phương pháp này là nó không giải quyết được 2 vấn đề quan trọng nhất của bài toán phân đoạn từ tiếng Việt: thuật toán gặp phải nhiều nhập nhằng, hơn nữa nó hoàn toàn không có chiến lược gì với những từ chưa biết.

• Phương pháp WFST

Phương pháp WFST (Weighted Finite – State Transducer) [DK01] còn gọi là phương pháp chuyên dịch trạng thái hữu hạn có trọng số. Ý tưởng chính của phương pháp này áp dụng cho phân đoạn từ tiếng Việt là các từ được gán trọng số bằng xác suất xuất hiện của từ đó trong dữ liệu. Sau đó duyệt qua các câu, cách duyệt có trọng số lớn nhất được chọn là cách dùng để phân đoạn từ. Hoạt động của WFST có thể chia thành ba bước sau:

– Xây dựng từ điển trọng số: Từ điển trọng số D được xây dựng như là một đồ thị biến đổi trạng thái hữu hạn có trọng số. Giả sử:

+ H là tập các tiếng trong tiếng Việt.

+ P là tập các loại từ trong tiếng Việt.

+ Mỗi cung của D có thể là:

* Từ một phần tử của H tới một phần tử của H;

* Từ phần tử ε (xâu rỗng) đến một phần tử của P.

Mỗi từ trong D được biểu diễn bởi một chuỗi các cung bắt đầu bởi một cung tương ứng với một phần tử của H, kết thúc bởi một cung có trọng số tương ứng với một phần tử của ε × P. Trọng số biểu diễn một chi phí ước lượng (estimated cost) cho bước công thức:

$$C = -\log \left(\frac{f}{N} \right) \quad (4.1)$$

Trong đó, f là lần số xuất hiện của từ; N là kích thước tập mẫu.

- Xây dựng các khả năng tách từ: Bước này thông kê tất cả các khả năng phân đoạn của một câu. Giả sử câu có n tiếng, thì có tối 2^{n-1} cách phân đoạn khác nhau. Để giám sự bùng nổ các cách phân đoạn, thuật toán loại bỏ ngay những nhánh phân đoạn mà chứa từ không xuất hiện trong từ điển.

- Lựa chọn khả năng tách tối ưu: Sau khi liệt kê tất cả các khả năng phân đoạn từ, thuật toán chọn cách tách từ tối nhất, đó là cách tách từ có trọng số bé nhất.

Ví dụ: Câu "Tốc độ truyền thông tin sẽ tăng cao" (theo [DK01])

Từ điển trọng số:

| | |
|----------------|-------|
| "tốc độ" | 8,68 |
| "truyền" | 12,31 |
| "truyền thông" | 12,31 |
| "thông tin" | 7,24 |
| "tin" | 7,33 |
| "sẽ" | 6,09 |
| "tăng" | 7,43 |
| "cao" | 6,95 |

Trọng số theo mỗi cách tách từ được tính là:

"Tốc độ # truyền thông # tin # sẽ # tăng # cao." = 8.68 + 12.31 + 7.24

$$+ 6.09 + 7.43 + 6.95 = 48.79$$

"Tốc độ # truyền # thông tin # sẽ # tăng # cao." = 8.68 + 12.31 + 7.24

$$+ 6.09 + 7.43 + 6.95 = 48.79$$

Do đó, ta có được tách từ tối ưu là cách phân đoạn: "Tốc độ # truyền # thông tin # sẽ # tăng # cao."

* Phương pháp trường ngẫu nhiên cho tách từ

Fương pháp trường ngẫu nhiên có điều kiện (Conditional Random Fields – CRFs) [LCP01] chứng tỏ hiệu lực tốt trong nhiều bài toán xử lý văn bản, đặc biệt trong các bài toán trích chọn thông tin trên Web. Trong [TKH06], C.T. Nguyen và cộng sự tiếp cận bài toán tách từ trong tiếng Việt dựa trên phương pháp CRFs. Kết quả tách từ có thể đạt được khoảng 94% theo những thử nghiệm trong bài báo. Dữ liệu và công cụ tách từ có thể được truy nhập từ <http://jvnsegmenter.sourceforge.net/>. Dưới đây trình bày một số nội dung về CRFs; ngoài ra, trong Chương 9 có trình bày một số nội dung bổ sung về phương pháp này.

- Định nghĩa CRFs:

Ký hiệu X là biến ngẫu nhiên có tương ứng với chuỗi dữ liệu cần gán nhãn và Y là biến ngẫu nhiên tương ứng với chuỗi nhãn. Mỗi thành phần Y_v của Y là một biến ngẫu nhiên nhãn giá trị trong tập hữu hạn các trạng thái S . Ví dụ, trong bài toán phân đoạn từ, X nhận giá trị là các câu trong ngôn ngữ tự nhiên, còn Y là chuỗi nhãn tương ứng với các câu này. Mỗi thành phần Y_v của Y là một nhãn xác định phạm vi của một từ trong câu (bắt đầu một từ, ở trong một từ và kết thúc một từ).

Cho một đồ thị vô hướng không có chu trình $G = (V, E)$, trong đó E là tập các cạnh vô hướng của đồ thị. V là tập các đỉnh của đồ thị sao cho $Y = \{Y_v \mid v \in V\}$. Nói cách khác, tồn tại ánh xạ một – một giữa một đỉnh đồ thị và một thành phần Y_v của Y . Nếu mỗi biến ngẫu nhiên Y_v tuân theo tính chất Markov đối với đồ thị G , tức là xác suất của biến ngẫu nhiên Y_v cho bởi X và tất cả các biến ngẫu nhiên khác $Y_{v'}$ ($v, v' \in V$): $p(Y_v \mid X, Y_{v'}, v' \neq v, \{u, v\} \in E)$ bằng xác suất của biến ngẫu nhiên Y_v cho bởi X và các biến ngẫu nhiên khác tương ứng với các đỉnh kề với đỉnh v trong đồ thị: $p(Y_v \mid X, Y_u, u, v \in E)$, thì ta gọi (X, Y) là một trường ngẫu nhiên điều kiện (Conditional Random Field).

Như vậy, một CRFS là một trường ngẫu nhiên phụ thuộc toàn cục vào chuỗi quan sát X . Trong bài toán phân đoạn từ nói riêng và các bài toán xử lý dữ liệu dạng chuỗi nói chung, thì đồ thị G đơn giản chỉ là dạng chuỗi, $V = \{1, 2, \dots, m\}$, $E = \{(i, i+1)\}$.

Ký hiệu $X = (X_1, X_2, \dots, X_n)$ và $Y = (Y_1, Y_2, \dots, Y_n)$ thì mô hình đồ thị G có dạng như Hình 4.8.

Gọi C là tập các đồ thị con đầy đủ của G . Vì G có dạng chuỗi, nên đồ thị con đầy đủ thực ra chỉ là một đỉnh hoặc một cạnh của đồ thị G . Áp dụng kết quả của Hammerley – Clifford [LCP01] cho các trường ngẫu nhiên Markov, thì phân phối của chuỗi nhãn Y với chuỗi quan sát X cho trước có dạng

$$P(y|x) = \prod_{A \in C} \psi_A(A|x) \quad (4.2)$$

Trong đó, Ψ_A gọi là hàm tiềm năng, nhận giá trị thực, dương. Lafferty xác định hàm tiềm năng này dựa trên nguyên lý cực đại entropy. Việc xác định một phân phối theo nguyên lý cực đại entropy có thể hiểu là phải xác định một phân phối sao cho "phân phối đó tuân theo mọi giả thiết suy ra từ thực nghiệm, ngoài ra không đưa thêm bất kỳ giả thiết nào khác" và gần nhất với phân phối đều.

Như đã giới thiệu ở Chương 3, entropy là độ đo thể hiện tính không chắc chắn, hay độ không đồng đều của phân phối xác suất. Độ đo entropy điều kiện $H(Y|X)$ được cho bởi công thức

$$H(Y|X) = - \sum_{x,y} p(x,y) \log q(y|x) \quad (4.3)$$

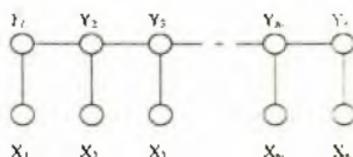
Với $p(x,y)$ là phân phối thực nghiệm của dữ liệu. Theo cách trên, Lafferty đã chỉ ra hàm tiềm năng của mô hình CRFs có dạng

$$\psi_A(A|x) = \exp \sum_k \lambda_k f_k(A|x) \quad (4.4)$$

Trong đó λ_k là thừa số Lagrangian ứng với thuộc tính f_k . Cũng có thể xem λ_k như là trọng số xác định độ quan trọng của thuộc tính f_k trong chuỗi dữ liệu. Có hai loại thuộc tính là thuộc tính chuyên (ký hiệu là f) và thuộc tính trạng thái (ký hiệu là g) tùy thuộc vào A là một định hay một cảnh của đồ thị. Thay công thức hàm tiềm năng vào công thức (4.2) và thêm thừa số chuẩn hóa để đảm bảo thỏa mãn điều kiện xác suất ta được

$$p(y|x) = \frac{1}{Z(x)} \exp \left(\sum_i \sum_k \lambda_k f_k(y_{i-1}, y_i, x) + \sum_i \sum_k \mu_k g_k(y_i, x) \right) \quad (4.5)$$

Ở đây, x là chuỗi dữ liệu, y là chuỗi trạng thái tương ứng. $f_k(y_{i-1}, y_i, x)$ là thuộc tính của chuỗi quan sát và các trạng thái ứng với vị trí thứ i và $i-1$ trong chuỗi trạng thái. $g_k(y_i, x)$ là thuộc tính của chuỗi quan sát và trạng thái ứng với vị trí thứ i trong chuỗi trạng thái.



Hình 4.8. Mô hình đồ thị của trường ngẫu nhiên

Các thuộc tính này được rút ra từ tập dữ liệu và có giá trị cố định.
Ví dụ:

$$f_i = \begin{cases} 1 & \text{nếu } x_{i-1} = \text{"Học"}, x_i = \text{"sinh"} \text{ và } y_{i+1} = B_W, y_i = L_W; \\ 0 & \text{nếu ngược lại.} \end{cases}$$

$$g_i = \begin{cases} 1 & \text{nếu } x_i = \text{"Học" và } y_i = B_W; \\ 0 & \text{nếu ngược lại} \end{cases}$$

Vấn đề bây giờ là phải ước lượng được các tham số $(\lambda_1, \lambda_2, \dots; \mu_1, \mu_2, \dots)$ từ tập dữ liệu huấn luyện.

- Huấn luyện CRFs:

Việc huấn luyện mô hình CRFs thực chất là tìm tập tham số của mô hình. Kỹ thuật được sử dụng là làm cực đại độ đo likelihood giữa phân phối mô hình và phân phối thực nghiệm. Vì thế, việc huấn luyện mô hình CRFs trở thành bài toán tìm cực đại của hàm logarit của hàm likelihood.

Giả sử dữ liệu huấn luyện gồm một tập N cặp, mỗi cặp gồm một chuỗi quan sát và một chuỗi trạng thái tương ứng, $D = \{(x^{(i)}, y^{(i)})\}$ với $\forall i = 1, \dots, N$. Hàm log-likelihood có dạng sau

$$l(\theta) = \sum_{x,y} \bar{p}(x,y) \log(p(y|x, \theta)) \quad (4.6)$$

Ở đây $\theta(\lambda_1, \lambda_2, \dots, \mu_1, \mu_2, \dots)$ là các tham số của mô hình và $\bar{p}(x,y)$ là phân phối thực nghiệm đồng thời của x, y trong tập huấn luyện. Thay $p(y|x)$ của CRFs trong công thức (4.5) vào (4.6) ta được:

$$l(\theta) = \sum_{x,y} \bar{p}(x,y) \left[\sum_{i=1}^{n+1} \lambda_i f_i + \sum_{i=1}^n \mu_i g_i \right] - \sum_x \bar{p}(x) \log Z \quad (4.7)$$

Ở đây, $\lambda(\lambda_1, \lambda_2, \dots, \lambda_n)$ và $\mu(\mu_1, \mu_2, \dots, \mu_m)$ là các vector tham số của mô hình, f là vector các thuộc tính chuyển, g là vector các thuộc tính trạng thái.

Người ta đã chứng minh được hàm log-likelihood là một hàm lõm và liên tục trong toàn bộ không gian của tham số. Vì vậy, có thể tìm cực đại hàm log-likelihood bằng phương pháp vector gradient. Mỗi thành phần trong vector gradient được gán bằng 0:

$$\frac{\partial l(\theta)}{\partial \lambda_k} = E_{\bar{p}(x,y)} [f_k] - E_{\bar{p}(y|x,0)} [f_k] \quad (4.8)$$

Việc thiết lập phương trình trên bằng 0 tương đương với việc đưa ra ràng buộc với mô hình là: giá trị kỳ vọng của thuộc tính f_k đối với phân phối

mô hình phải bằng giá trị kỳ vọng của thuộc tính f_k đối với phân phối thực nghiệm.

Hiện nay có khá nhiều phương pháp để giải quyết bài toán cực đại hàm log-likelihood, ví dụ như các phương pháp lặp (IIS và GIS), các phương pháp tối ưu số (Conjugate Gradient, phương pháp Newton,...). Theo đánh giá của Malouf vào năm 2002 [LCP03] thì phương pháp được coi là hiệu quả nhất hiện nay đó là phương pháp tối ưu số bậc hai L-BFGS (limited memory BFGS).

- *Lập luận với CRFs:*

Sau khi tìm được mô hình CRFs từ tập dữ liệu huấn luyện, nhiệm vụ lúc này là làm sao dựa vào mô hình đó để gán nhãn cho chuỗi dữ liệu quan sát mới, điều này tương đương với việc làm cực đại phân phối xác suất giữa chuỗi trạng thái y và dữ liệu quan sát x . Quá trình này còn được gọi là quá trình lập luận dựa trên mô hình. Chuỗi trạng thái y^* mà ta tìm nhất chuỗi dữ liệu quan sát x là nghiệm của phương trình $y^* = \operatorname{argmax}\{p(y|x)\}$.

Chuỗi y^* có thể xác định được bằng thuật toán Viterbi. Gọi S là tập tất cả trạng thái có thể, ta có $|S| = m$. Xét một tập hợp các ma trận cỡ $m \times m$, ký hiệu $\{M_i(x) \mid i = 0, 2, \dots, n - 1\}$ được định nghĩa trên từng cặp trạng thái $y, y' \in S$ như sau

$$M_i(y', y | x) = \exp \left(\sum_k \lambda_k f_k(y', y, x) + \sum_k \mu_k g_k(y, x) \right) \quad (4.9)$$

Bằng việc đưa thêm hai trạng thái y_0 và y_n vào trước và sau chuỗi trạng thái. Coi như chúng ứng với trạng thái "start" và "end", phân phối xác suất có thể viết là

$$p(y | x, \lambda) = \frac{1}{Z(x)} \prod_{i=0}^n M_i(y', y | x) \quad (4.10)$$

Ở đây $Z(x)$ là thừa số chuẩn hóa được đưa thêm vào và có thể tính được dựa vào các M_i , nhưng vẫn đề quan tâm là cực đại hóa $p(y|x)$, nên không cần thiết phải tính $Z(x)$. Như vậy, chỉ cần cực đại hóa tích $(n - 1)$ phần tử trên. Tứ trường chính của thuật toán Viterbi là tăng dần chuỗi trạng thái tối ưu bằng việc quét các ma trận từ vị trí 0 cho đến vị trí n . Tại mỗi bước i ghi lại tất cả các chuỗi tối ưu kết thúc bởi trạng thái y với $\forall y \in S$ (ký hiệu là $y^i(y)$) và tích tương ứng $P_i(y)$.

Chuỗi $y_{t+1}^*(y)$ chính là chuỗi có xác suất $p(y^*|x)$ lớn nhất, đó cũng chính là chuỗi nhãn phù hợp nhất với chuỗi dữ liệu quan sát x cho trước.

Bước 1. $P_0(y) = M_0(\text{start}, y | x)$ và $y_0^*(y) = y$

Bước lặp: Cho i chạy từ 1 đến n tính:

$$P_i(y) = \max_{y' \in S} P_{i-1}(y) \times M_i(y', y | x)$$

$$y_i^*(y) = y_{i-1}^*(y) (y).$$

trong đó $\hat{y} = \operatorname{argmax}_{y' \in S} P_{i-1}(y) \times M_i(y', y | x)$ và " \cdot " là toán tử cộng chuỗi

Hình 4.9. Thuật toán tìm chuỗi đầu ra tối ưu

- Lựa chọn đặc trưng cho tách từ:

Đặc trưng của CRFs cũng giống như đặc trưng trong Maximum Entropy theo khía cạnh nó cũng là một hàm nhị phân của ngữ cảnh và nhãn. Trong bài toán tách từ, xem xét 3 loại nhãn {B_W, I_W, O}, trong đó B_W có nghĩa là "bắt đầu một từ", I_W nghĩa là "trong từ" và O nghĩa là những trường hợp còn lại. Một chuỗi bắt đầu bởi B_W và một loạt I_W liên tiếp tạo thành một từ duy nhất trong tiếng Việt. Vì các đặc trưng cạnh được trích rút trực tiếp từ dữ liệu học (dựa vào nhãn của hai tiếng liên tiếp). Các đặc trưng định được lựa chọn từ dữ liệu theo một loạt các mẫu ngữ cảnh.

Bảng 4.2. Mẫu ngữ cảnh cho tách từ với CRFs

| Syllable_Conj (SC) | Syllable_Conjunction (-2,2) |
|-------------------------------------|--|
| Dictionary (Dict) | Is_Lac_Viet_Dictionary (-2,2) |
| External Resources (ERS.) | Is_Personal_Name_List(0,0), Is_Family_Name_List(0,0), Is_Middle_Name_List(-2,2), Is_Location_List(-2,2) |
| Miscellaneous (Misc) | Is-Regular_Expression(0,0), Is_Initial_Capitalization(0,0), Is_All_Capitalization(0,0), Is_First_Observation(0,0), Is_Marks(0,0) |
| Vietnamese_Syllable_Detection (VSD) | Is_Valid_Vietnamese_Syllable(0,0) |

Ở đây một từ tiếng Việt phải tuân theo các cấu trúc chuẩn về từ trong tiếng Việt được mô tả trong phần trên; Is_Marks kiểm tra xem quan sát hiện tại có phải là dấu câu không; Is-Regular Expression mô tả các công thức thường dùng như thời gian, số lượng.

Các mẫu ngữ cảnh hầu hết được lấy trong cửa sổ (-2, 2) xung quanh tiếng hiện tại (2 tiếng trước đến 2 tiếng sau tiếng hiện tại). Syllable_Conj(-2, 2) kết hợp các tiếng trong cửa sổ trượt hiện tại. Xét ví dụ sau:

Kỹ thuật môi trường, khoa học môi trường, công nghệ

B_WI_WB_WI_WOB_WI_WB_WI_WOB_WI_W

môi trường có phải là một ngành.

B_WI_WB_WB_WB_WB_WB_WO

Nếu tiếng hiện tại là tiếng thứ hai "thuật", một số ngữ cảnh có thể được sinh ra từ mẫu Syllable_Conj(-2, 2) là kỹ_thuật, thuật_môi_. môi_trường_.
thuật_môi_trường...

Kết quả tách từ tốt nhất với CRFs được mô tả trong bài báo lên tới 94.05%.

c) Gắn nhãn từ loại

Gắn nhãn từ loại là việc xác định các chức năng ngữ pháp của từ trong câu. Đây là bước cơ bản trước khi phân tích xâu văn phạm hay các vấn đề xử lý ngôn ngữ phức tạp khác. Thông thường, một từ có thể có nhiều chức năng ngữ pháp. Ví dụ, trong câu "con ngựa đá đá con ngựa đá", cùng một từ "đá", nhưng từ thứ nhất và thứ ba giữ chức năng ngữ pháp là danh từ, nhưng từ thứ hai lại là động từ trong câu.

Trong [DK03], một số hướng tiếp cận chính trong gắn nhãn từ loại tiếng Anh đã được trình bày là gắn nhãn dựa trên mô hình Markov ẩn (HMM); các mô hình dựa trên bộ nhớ (Daelemans đề xuất năm 1996); mô hình dựa trên luật (Transformation Based Learning) [Bri95]; mô hình cực đại Entropy (Maximum Entropy); cây quyết định và mạng nơron (được Schmid đề cập vào năm 1994)... Trong các hướng tiếp cận đó, phương pháp dựa trên học máy được đánh giá tốt.

Hai nhóm nghiên cứu trong nước khảo sát và nghiên cứu về bài toán gắn nhãn từ loại tiếng Việt điển hình là nhóm Đinh Diên và Hoàng Kiếm (Đại học Quốc gia Thành phố Hồ Chí Minh) [DK01, DK03] và nhóm Nguyễn Thị Minh Huyền (Đại học Quốc gia Hà Nội) và Vũ Xuân Lương (Trung tâm Từ điển học – Vietlex) [HRV03, HRR07].

Nhóm thứ nhất [DK03] xây dựng hệ thống gắn nhãn từ loại cho tiếng Việt dựa trên việc chuyển đổi và ảnh xạ từ thông tin từ loại từ tiếng Anh. Cơ sở của hướng tiếp cận này nằm ở hai ý: (1) gắn nhãn từ loại trong tiếng Anh đã đạt độ chính xác cao (trên 97% cho độ chính xác ở mức từ) và (2) những thành công gán dây của các phương pháp giống hàng từ (word alignment methods) giữa các cặp ngôn ngữ. Cụ thể, nhóm này đã xây dựng một tập ngữ liệu song ngữ Anh – Việt lên đến 5 triệu từ (cả Anh lẫn Việt). Sau đó thực hiện gắn nhãn từ loại cho bên tiếng Anh (dựa trên Transformation-

based Learning – TBL [Bri95]) và thực hiện giống hảng giữa hai ngôn ngữ (độ chính xác khoảng 87%) để chuyển thông tin về nhãn từ loại từ tiếng Anh sang tiếng Việt. Cuối cùng, dữ liệu tiếng Việt với thông tin từ loại mới thu nhận được cần được hiệu chỉnh bằng tay để làm dữ liệu huấn luyện cho bộ gắn nhãn từ loại tiếng Việt. Ưu điểm của phương pháp này là tránh được việc gắn nhãn từ loại bằng tay nhờ tận dụng thông tin từ loại ở một ngôn ngữ khác.

Nhóm thứ hai [HRV03, HRR07] tiếp cận vấn đề này dựa trên nền tảng và tính chất ngôn ngữ của tiếng Việt. Nhóm này để xuất xây dựng tập từ loại (tagset) cho tiếng Việt dựa trên chuẩn mô tả khá tổng quát của các ngôn ngữ Tây Âu, MULTTEXT, nhằm môđun hóa tập nhãn ở hai mức: (1) mức cơ bản/cốt lõi (kernel layer) và (2) mức tính chất riêng (private layer). Mức cơ bản nhằm đặc tả chung nhất cho các ngôn ngữ, trong khi mức thứ hai mở rộng và chi tiết hoá cho một ngôn ngữ cụ thể dựa trên tính chất của ngôn ngữ đó. Cụ thể, mức cơ bản của từ loại do nhóm này để xuất bao gồm: danh từ (noun – N), động từ (verb – V), tính từ (adjective – A), đại từ (pronoun – P), mạo từ (determine – D), trạng từ (adverb – R), tiền – hậu giới từ (adposition – S), liên từ (conjunction – C), số từ (numeral – M), tình thái từ (interjection – I) và từ ngoại Việt (residual – X, như foreign words,...). Mức thứ hai được triển khai tuỳ theo các dạng từ loại trên như danh từ đếm được/không đếm được đối với danh từ, giống đực/cái đối với đại từ,... Với cách phân loại này, có thể co giãn hệ phân loại từ ở mức chung (cơ bản) hoặc cụ thể (chi tiết hoá) tương đối dễ dàng.

4.5. Giới thiệu một số nghiên cứu xử lý tiếng Việt

Như đã biết, dù đã trải qua thời gian 60 năm phát triển, xử lý ngôn ngữ tự nhiên ngày càng nhận được sự quan tâm rộng rãi của khối khoa học và công nghiệp, đặc biệt là trong giai đoạn phát triển của Internet. Chính vì lý do đó, Xử lý tiếng Việt luôn là một nội dung quan trọng trong công cuộc phát triển và ứng dụng CNTT của nước ta. Mục 4.3 đã đề cập tới một số nghiên cứu của các nhóm nghiên cứu về các giải pháp đối với các bài toán cơ bản trong xử lý tiếng Việt. Cộng đồng các nhà khoa học nghiên cứu về xử lý tiếng Việt trải ra ở hầu hết các khoa CNTT trọng điểm, các viện nghiên cứu chuyên ngành (diễn hình là Viện CNTT – Viện KH&CN Quốc gia, Trung tâm Từ điển học, Viện Ngôn ngữ – Viện KH&NV Quốc gia), một số công ty có triển khai các ứng dụng liên quan (diễn hình là Lạc Việt, Tinh Vân). Một số nhóm nghiên cứu đã công bố các nghiên cứu, trao đổi trên trang Web.

Đề tài KC01.01/06-10 "Nghiên cứu phát triển một số sản phẩm thiết yếu về xử lý tiếng nói và văn bản tiếng Việt" (VLSP) do Viện CNTT – Viện KH&CN Quốc gia chủ trì phối hợp với 7 đơn vị khoa học – công nghệ trên toàn quốc thực hiện. Một số sản phẩm chính về xử lý văn bản được thực hiện trong đề tài là:

- Từ điển dùng cho xử lý ngôn ngữ với sản phẩm VCL dựa trên mô hình LMF do tiêu ban ISO/TC 37/SC 4 phát triển.
- Kho ngữ liệu câu tiếng Việt có chủ giải (VieTreeBank) tương tự như TreeBank với một số tài liệu hướng dẫn tiếng Việt.
- Hai kho ngữ liệu câu Anh – Việt phổ quát và chuyên ngành có sử dụng công cụ khai thác ngữ liệu song ngữ trên Internet, hiệu chỉnh: công cụ quản lý kho ngữ liệu.
- Hệ phân tách từ tiếng Việt với sản phẩm điện hình là công cụ VnTokenizer sử dụng kỹ thuật so khớp tối đa, thống kê unigram và bigram.
- Hệ phân loại từ tiếng Việt dựa trên học máy MaxEnt và CRFs.
- Hệ phân cụm từ tiếng Việt theo mô hình học máy có cấu trúc (CRF, SVM, Online Learning,...).
- Hệ phân tích cú pháp tiếng Việt theo hướng phân tích cú pháp dựa trên luật và theo thống kê.

Câu hỏi và bài tập

1. Trình bày các đặc điểm của ngôn ngữ tiếng Việt so với tiếng Anh, từ đó đưa ra những thuận lợi và khó khăn cho các bài toán xử lý ngôn ngữ tự nhiên cho tiếng Việt nói chung và bài toán trích rút quan hệ nói riêng.
2. Đề xuất một quan hệ ngữ nghĩa trong tiếng Việt có ý nghĩa nào đó và dùng giải thuật DIPRE để trích rút cặp quan hệ đó. Nếu giải thuật DIPRE hoạt động không tốt, hãy đề xuất giải pháp cải tiến.

Chương 5

CÁC PHƯƠNG PHÁP BIỂU DIỄN VĂN BẢN

Trong chương này trình bày một số khái niệm cơ bản, thuật ngữ liên quan đến phân tích văn bản. Tiếp theo giới thiệu các phương pháp chung được dùng để đánh trọng số trong văn bản. Sau đó giới thiệu các mô hình chung được dùng trong biểu diễn văn bản. Cuối cùng là các phương pháp và độ đo khi lựa chọn các từ biểu diễn văn bản.

Trong [Mla98], Dunja Mladenic cung cấp thông tin tổng hợp về một số công trình nghiên cứu về biểu diễn Text cho đến năm 1998. Ở đó chúng tôi rằng, vẫn để biểu diễn Text thu hút được sự quan tâm của nhiều nhà khoa học. Các nội dung liên quan tới biểu diễn Text là lựa chọn đặc trưng và các thuật toán học máy được sử dụng.

5.1. Phân tích văn bản

Từ khi phát minh ra giấy viết và mục in, có thể nói rằng, dữ liệu văn bản trở thành một trong những dạng dữ liệu truyền thống và quan trọng nhất được dùng để lưu trữ thông tin của con người. Một trong những vấn đề khó khăn nhất của máy tính là làm thế nào để biểu diễn văn bản phản ánh đúng nội dung. Công việc này còn gọi là đánh chỉ số văn bản. Trước đây, quá trình này được làm thủ công với sự giúp đỡ của người dùng hoặc các chuyên gia trong một số lĩnh vực chuyên ngành. Tuy nhiên, với số lượng lớn các văn bản ngày càng tăng, thi việc đánh chỉ số thủ công là không khả thi, do vậy, việc đánh chỉ số một cách tự động là cần thiết.

Lựa chọn các từ để đánh chỉ số cũng là một công việc không dễ dàng. Tuỳ theo lĩnh vực hoặc mục đích của người dùng, có thể có nhiều cách đánh chỉ số khác nhau. Những nghiên cứu đầu tiên của Keen đã chỉ ra rằng, việc đánh chỉ số thủ công lẩn tịt động là thoa mẩn ba mục đích sau [Kee77]:

1. Cho phép vị trí của từ đó liên quan tới chủ đề người dùng quan tâm.
2. Gắn kết các từ và các chủ đề liên quan với nhau bằng cách phân biệt được các từ riêng biệt đối với các lĩnh vực.
3. Dự đoán được mức độ liên quan của từ đó tới thông tin yêu cầu của người dùng, với lĩnh vực và chuyên ngành cụ thể.

Phương pháp sử dụng để đạt được ba mục tiêu kể trên phụ thuộc vào môi trường đánh chỉ số. Có thể phân biệt các môi trường đó như sau:

Thứ nhất là sự khác biệt giữa đánh chỉ số thu công và tự động. Như đề cập ở trên, trước đây việc đánh chỉ số được thực hiện thu công bởi các chuyên gia hoặc người có kiến thức trong một hoặc nhiều lĩnh vực cụ thể. Ngày nay, phương pháp này vẫn được sử dụng đối với một số lĩnh vực chuyên ngành hẹp với sự giúp đỡ của các công cụ tự động điều khiển quá trình đánh chỉ số, bao gồm danh sách các thuật ngữ, các bảng biểu có cấu trúc để ghi lại quá trình đánh chỉ số.

Sự khác biệt *thứ hai* là, việc đánh chỉ số có điều khiển và không điều khiển. Đánh chỉ số không điều khiển bao gồm đánh chỉ số tất cả các từ trong văn bản, điều này có thể dẫn tới việc đánh chỉ số tất cả các dạng biểu diễn của từ trong ngôn ngữ tự nhiên, từ đó có thể gây nhầm lẫn hoặc lỗi. Để hạn chế điều này, một số phương pháp đánh chỉ số có điều khiển như lược bỏ các dạng khác nhau của từ, lược bỏ các từ đồng nghĩa,... được áp dụng.

Vẫn đề *thứ ba* liên quan đến loại từ diễn được sử dụng cho mục đích đánh chỉ số. Có thể sử dụng các từ đơn lẻ, hoặc sử dụng các từ có liên quan đến tập hợp các từ khác trong văn bản để biểu diễn nội dung văn bản.

Sau đây sẽ khảo sát sự phân bố của các từ có trong văn bản, từ đó sẽ giúp xác định nên lựa chọn các từ nào để đánh chỉ số. Sự phân bố này dựa trên quan sát thực nghiệm và được gọi là luật Zipf.

5.1.1. Phân bố các từ trong văn bản – Luật Zipf

Hầu hết các phương pháp đánh chỉ số đều bắt đầu bằng lập luận rằng, tần số của các từ đóng vai trò quan trọng trong biểu diễn văn bản. Để giải thích điều này, H.P. Luhn (một trong những người tiên phong trong việc đánh chỉ số văn bản) đã viết [Luh58]: "Việc do độ quan trọng của các từ bằng tần số sử dụng dựa trên thực tế rằng, người viết thường lặp lại các từ nhất định khi người đó phát triển ý tưởng, hoặc trình bày các lập luận và khi phân tích các khía cạnh của chủ đề. Dấu hiệu nhấn mạnh này được coi như là một biểu hiện của độ quan trọng..."

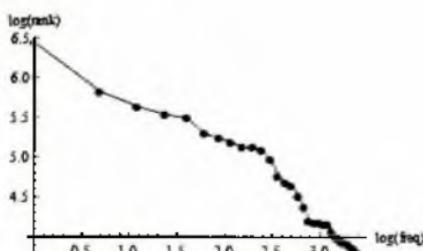
Mỗi liên hệ giữa tần số và độ quan trọng của từ trong văn bản được đặc trưng bởi quan sát thực nghiệm: những từ có tần suất xuất hiện ít trong văn bản thì có ảnh hưởng ít đến văn bản đó, hay có thể phát biểu rằng: tần số của một từ trong văn bản tỷ lệ nghịch với độ quan trọng của nó trong văn bản.

Quan sát này được đưa ra bởi Zipf năm 1949 dưới dạng phát biểu như một quan sát [Zip49], nhưng ở thời điểm đó đã được gọi là "định luật Zipf". Dung hơn, đây không phải thực sự là một định luật mà là một hiện tượng xấp xỉ toán học.

Để mô tả định luật Zipf, gọi tổng số tần số xuất hiện của từ khoá 1 trong tập hợp D là f_1 . Sau đó sắp xếp tất cả các từ khoá trong tập hợp theo chiều giảm dần của tần số xuất hiện f và gọi thứ hạng của mỗi từ khoá 1 là r_1 . Định luật Zipf được phát biểu dưới dạng công thức: $r_1 \cdot f_1 \approx K$.

Trong đó K là một hằng số. Trong tiếng Anh, người ta thấy rằng số $K \approx N/10$, trong đó N là tổng số các từ trong tập hợp.

| | | |
|-----------|-----------|----------|
| the 632 | and 338 | a 278 |
| to 252 | she 242 | of 199 |
| it 189 | in 178 | was 167 |
| alice 167 | an 163 | said 144 |
| you 118 | her 108 | that 105 |
| as 91 | at 79 | with 67 |
| is 66 | had 65 | all 64 |
| on 64 | little 59 | out 54 |
| down 52 | thus 51 | t 50 |
| for 48 | but 47 | they 45 |



Hình 5.1. Mô tả tần xuất của các từ tiếng Anh trong truyện "Alice ở xứ sở diệu kỳ".

Tỷ lệ x, y lấy theo log của các giá trị, trong đó x là tần số, y là thứ hạng của từ
(Nguồn <http://mathworld.wolfram.com>).

Bây giờ viết lại định luật Zipf như sau:

$$r_i \approx \frac{K}{f_i}$$

Giả sử từ khoá a là từ được sắp ở vị trí thấp nhất với một tần số xuất hiện là β nào đấy, từ khoá b cũng được sắp ở vị trí thấp nhất với một tần số xuất hiện là $\beta + 1$. Ta có thể thu được thứ hạng xấp xỉ của các từ khoá này là $r_a \approx \frac{K}{\beta}$ và $r_b \approx \frac{K}{\beta + 1}$. Trừ hai biểu thức này cho nhau, ta có một xấp xỉ đối với các từ riêng biệt có tần số xuất hiện là β .

$$r_a - r_b \approx \frac{K}{\beta} - \frac{K}{\beta + 1} = \frac{K}{\beta(\beta + 1)}$$

Bây giờ xấp xỉ giá trị của từ trong tập hợp có thứ hạng cao nhất. Tóm quát hơn, đó là một từ chỉ xuất hiện một lần trong tập hợp, ta có:

$$r_{\max} \approx \frac{K}{1} = K$$

Xét phân bố của các từ đơn xuất hiện β lần trong tập hợp, chia hai về cho nhau, ta được $\frac{1}{\beta(\beta + 1)}$. Do đó, định luật Zipf cho thấy một sự phân bố

đáng chú ý của các từ riêng biệt trong một tập hợp được hình thành bởi các từ khoá xuất hiện ít nhất trong tập hợp.

Một câu hỏi được đặt ra là: Tần số có phải là yếu tố quan trọng trong văn bản hay không? Để dễ dàng thấy rằng, các từ có tần số xuất hiện cao nhất (trong tiếng Anh) thông thường là các giới từ như "the", "a", "an" và hiện

nhiên là những từ này không đóng góp nhiều trong việc phản ánh nội dung văn bản. Mặt khác, những từ chỉ xuất hiện một hoặc hai lần cũng không đóng vai trò quan trọng. Những từ đóng vai trò quan trọng là những từ có tần số xuất hiện trung bình. Những nghiên cứu đầu tiên của Luhn đưa ra một phương pháp đơn giản cho việc lựa chọn các từ để biểu diễn văn bản như sau:

1. Cho một tập gồm n văn bản, tính tần số của mỗi từ duy nhất (xuất hiện một lần) trong mỗi văn bản.

2. Tính tần số xuất hiện của mỗi từ trong toàn bộ tập n văn bản.

3. Sắp xếp tần số các từ giảm dần. Chọn một giá trị ngưỡng trên để loại bỏ các từ có tần số cao hơn ngưỡng đó. Việc này sẽ loại bỏ các từ có tần số cao.

4. Cũng như vậy, chọn một giá trị ngưỡng dưới để loại bỏ các từ có tần số thấp.

5. Các từ còn lại là các từ được dùng cho quá trình đánh chỉ số văn bản.

Việc quyết định từ nào được chọn để đánh chỉ số là vấn đề quan trọng, trong học máy nói chung, việc này còn gọi là lựa chọn đặc trưng. Đây cũng là một chủ đề nghiên cứu trong học máy cũng như trong khai phá dữ liệu.

5.1.2. Các phương pháp đánh trọng số

Phản trước chỉ ra mối liên hệ giữa tần số và độ quan trọng của các từ trong tập văn bản thuộc miền ứng dụng. Sử dụng các phương pháp lựa chọn từ (chẳng hạn, phương pháp Luhn), ta nhận được tập từ vựng T , được dùng để biểu diễn văn bản thuộc miền ứng dụng. Đặc trưng cho độ quan trọng của từ thuộc tập T trong một văn bản bất kỳ là một giá trị số được gán cho từ đó trong văn bản đã cho. Công việc tính trọng số của từ còn được gọi là đánh trọng số các từ trong văn bản. Bài toán đánh trọng số có thể được phát biểu như sau:

Input: Cho một từ $t_i \in T$ và một văn bản d_j thuộc miền ứng dụng

Output: Giá trị w_{ij} là trọng số (độ quan trọng) của từ t_i trong văn bản d_j .

Dưới đây là một số phương pháp (mô hình) đánh trọng số từ điển hình.

• Phương pháp Boolean

Gia sư có một tập gồm m văn bản $D = \{d_1, d_2, \dots, d_m\}$. Tập từ vựng T gồm có n từ khóa $T = \{t_1, t_2, \dots, t_n\}$. Gọi $W = (w_{ij})$ là ma trận trọng số, trong đó w_{ij} là trọng số của từ khóa t_i trong văn bản d_j .

Phương pháp Boolean là phương pháp đánh trọng số đơn giản nhất, giá trị trọng số w_{ij} được xác định như sau:

$$w_{ij} = \begin{cases} 1 & t_i \in d_j \\ 0 & t_i \notin d_j \end{cases}$$

• Phương pháp dựa trên tần số

Phương pháp dựa trên tần số xác định giá trị các số trong ma trận $W = (w_{ij})$, dựa vào tần số xuất hiện của các từ khoá trong văn bản và tần số xuất hiện của văn bản trong tập D gồm m văn bản đang được xem xét. Dưới đây là ba phương pháp đánh trọng số dựa trên tần số phổ biến.

- Phương pháp dựa trên tần số từ khoá (TF – Term Frequency)

Giá trị của một từ khoá được tính dựa trên số lần xuất hiện của từ khoá trong văn bản. Gọi tf_{ij} là số lần xuất hiện của từ khoá t_i trong văn bản d_j , khi đó có thể chọn cách tính w_{ij} theo một từ ba công thức dưới đây:

$$w_{ij} = \sqrt{tf_{ij}} \text{ hoặc } w_{ij} = 1 + \log(tf_{ij}) \text{ hoặc } w_{ij} = tf_{ij}$$

Phương pháp này được lý giải từ lập luận rằng, trong một văn bản thì một từ xuất hiện nhiều thường quan trọng hơn một từ xuất hiện ít.

- Phương pháp dựa trên nghịch đảo tần số văn bản (IDF – Inverse Document Frequency)

Gọi df_i là số lượng văn bản có chứa từ khoá t_i trong tập m văn bản đang xét, thì giá trị trọng số từ được tính bởi công thức:

$$w_{ij} = \log\left(\frac{m}{df_i}\right) = \log(m) - \log(df_i)$$

Phương pháp này được lý giải từ lập luận rằng, một từ xuất hiện trong nhiều văn bản thuộc tập văn bản D thì không quan trọng bằng một từ xuất hiện trong ít văn bản thuộc tập D, nghĩa là một từ quá thông dụng (xuất hiện trong nhiều văn bản) sẽ có độ quan trọng kém hơn từ chỉ xuất hiện trong một văn bản hoặc một tập nhỏ các văn bản.

- Phương pháp TFIDF

Phương pháp này là tổng hợp của hai phương pháp TF và IDF, giá trị của ma trận trọng số được tính như sau:

$$w_{ij} = \begin{cases} \left[1 + \log(tf_{ij})\right] \log\left(\frac{m}{df_i}\right) & \text{nếu } tf_{ij} \geq 1, \\ 0 & \text{nếu } tf_{ij} = 0. \end{cases}$$

5.2. Các mô hình biểu diễn văn bản

5.2.1. Mô hình Boolean

Giả sử có một tập gồm m văn bản $D = \{d_1, d_2, \dots, d_m\}$. Mỗi văn bản gồm n từ khoá $T = \{t_1, t_2, \dots, t_n\}$. Gọi $W = (w_{ij})$ là ma trận trọng số, trong đó w_{ij} là trọng số của từ khoá t_i trong văn bản d_j .

Mô hình Boolean là mô hình đơn giản nhất, trong đó trọng số các từ trong văn bản là 0 hoặc 1 (như cách đánh trọng số Boolean). Khi đó, mỗi văn bản sẽ được biểu diễn dưới dạng tập hợp như sau:

$$d_i = \{t_{ij}\}, \text{trong đó } t_{ij} \text{ là từ } t_i \text{ có trọng số } w_{ij} \text{ trong văn bản } d_i \text{ là } 1.$$

Ví dụ: Giả sử có một văn bản đơn giản gồm các từ như sau:

$$d = "Hello world ! Hello Vietnam !"$$

Khi đó văn bản d được biểu diễn như sau:

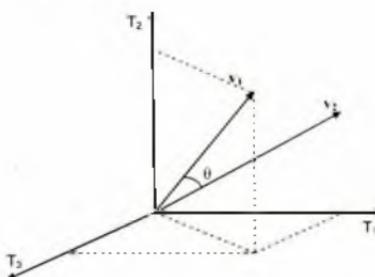
$$d = \{"Hello", "world", "Vietnam", "?"\}.$$

Giải thuật phân lớp Naïve Bayesian sử dụng mô hình Boolean này còn được gọi là mô hình Bernoulli.

5.2.2. Mô hình không gian vector

Mô hình không gian là một trong những mô hình toán học được sử dụng rộng rãi nhất trong biểu diễn văn bản bởi tính chất dễ hiểu của nó. Mô hình này được đề xuất bởi Salton và cộng sự năm 1975 [SWY75] khi giải quyết bài toán truy vấn thông tin. Theo cách biểu diễn này, mỗi văn bản được biểu diễn trong một không gian nhiều chiều, trong đó mỗi chiều tương ứng với một từ trong văn bản. Một từ với độ quan trọng của nó được xác định bằng một phương pháp đánh giá số trong văn bản và giá trị trong số được chuẩn hóa trong đoạn [0, 1]. Hình 5.2 mô tả hai văn bản d_1 và d_2 được biểu diễn bởi các vector ký hiệu là v_1 và v_2 , gồm ba chiều là T_1 , T_2 , T_3 , trong đó T_i là từ khác nhau đặc trưng cho độ quan trọng của từ đó trong văn bản.

Tổng quát hơn, một văn bản d trong không gian vector, ký hiệu là v_d sẽ được biểu diễn trong một không gian vector gồm N chiều, trong đó N là số lượng từ có trong tập văn bản.



Hình 5.2. Biểu diễn văn bản theo không gian vector
 v_1 , v_2 là hai văn bản trong không gian vector ba chiều
 T_1 , T_2 , T_3 , trong đó T_i là từ

$$v_d = [w_{1,d}, w_{2,d}, \dots, w_{N,d}]^T$$

Khi đó độ giống nhau giữa hai văn bản sẽ được tính bằng độ đo cos giữa hai vector: $\cos \theta = \frac{(\mathbf{v}_1, \mathbf{v}_2)}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|}$

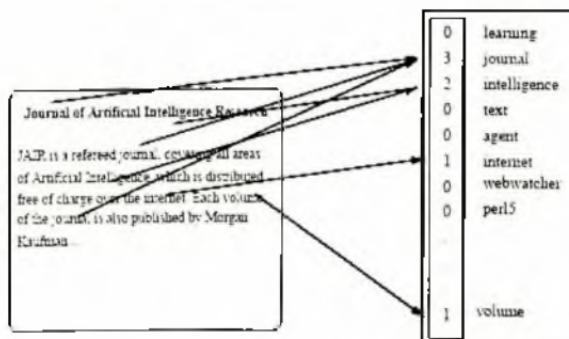
Mô hình không gian vector là mô hình toán học hết sức quan trọng trong biểu diễn văn bản, đặc biệt là trong lĩnh vực truy vấn thông tin.

5.2.3. Mô hình xác suất

Mô hình xác suất là mô hình toán học làm việc với các biến ngẫu nhiên và phân bố xác suất của nó. Theo thuật ngữ toán học, một mô hình xác suất có thể được coi như một cặp (Y, P) , trong đó Y là tập các quan sát (biến ngẫu nhiên) và P là tập các phân bố xác suất trên Y . Khi đó, sử dụng suy diễn xác suất sẽ cho ta kết luận về các phân tử của tập Y . Các phương pháp suy diễn có thể là các phương pháp hồi quy hoặc suy diễn Bayes.

Văn bản trong mô hình xác suất được coi như như một quan sát trong tập Y , trong đó các từ trong văn bản được giả thiết là độc lập, không phụ thuộc vào vị trí cũng như ngữ pháp trong văn bản. Khi đó văn bản sẽ gồm các từ mà nó chứa trong đó, chính vì vậy mà phương pháp này được gọi là biểu diễn túi – các – từ (bag – of – word). Để đơn giản, người ta còn gọi là mô hình biểu diễn theo túi – các – từ. Mô hình này được sử dụng nhiều trong phân lớp văn bản khi áp dụng suy diễn Bayes trong bài toán phân lớp.

Ví dụ, văn bản đưa tin về Tạp chí Nghiên cứu về trí tuệ nhân tạo được biểu diễn theo mô hình túi – các – từ một cách đơn giản theo bảng với trọng số là tần số từ có trong văn bản (Hình 5.3).



Hình 5.3. Biểu diễn văn bản theo túi-các-từ (mô hình xác suất) [Mla98]

5.2.4. Một số mô hình khác

Ngoài ba mô hình toán học cơ bản trên để biểu diễn văn bản còn có các phương pháp khác biểu diễn văn bản. Tuy nhiên, về cơ bản các cách biểu diễn sau này đều dựa trên ba mô hình toán học cơ bản trên. Ở đây chúng tôi giới thiệu sơ lược hai phương pháp chính là LSI và biểu diễn theo cụm các từ.

• Phương pháp LSI (*Latent Semantic Indexing*)

LSI (Latent Semantic Indexing) đánh chỉ số ngữ nghĩa tiềm năng, là phương pháp được Deerwester và các đồng nghiệp đề xuất năm 1988 và được áp dụng nhiều trong truy vấn văn bản hoặc bài toán phân lớp [BC98]. Ý tưởng chính của phương pháp này là, ánh xạ mỗi văn bản vào một tập không gian ít chiều hơn, trong đó mỗi chiều được gắn với một khái niệm. Như vậy, bản chất của phương pháp này là chuyên từ không gian từ khóa sang không gian các khái niệm. Một số nội dung liên quan trực tiếp tới phương pháp này được giới thiệu tại một số phần trong giáo trình này, đặc biệt là Chương 9 và Chương 4.

Các bước để chuyển từ không gian các từ sang không gian khái niệm tương đối phức tạp. Trước tiên, LSI lập ma trận từ – văn bản với trọng số là một phương pháp đánh chỉ số nào đó (thông thường là tfidf). Sau đó, LSI tìm một xấp xỉ hạng của ma trận thấp hơn của ma trận này. Việc xấp xỉ này sử dụng các phép toán biến đổi ma trận tương đối phức tạp, chi tiết của các phép biến đổi có thể tham khảo tại [DFD90].

Lấy một ví dụ đơn giản như sau: Giả sử văn bản gồm các từ "car", "auto", "house" được biểu diễn trong không gian ba chiều. sau khi biểu diễn bằng LSI có thể giảm còn hai chiều, chẳng hạn:

$$\{(car), (auto), (house)\} \rightarrow \{(2,05 * car + 0,45 * auto), (house)\}$$

Trong đó "car" và "auto" được chuyển sang thành một khái niệm với mức độ quan trọng khác nhau thể hiện qua các trọng số gắn với mỗi từ.

Không gian khái niệm có thể được sử dụng trong các ứng dụng sau:

– So sánh các văn bản trong không gian khái niệm (phân cụm và phân lớp văn bản).

– Tìm các văn bản giống nhau giữa các ngôn ngữ, điều này có thể thực hiện sau khi phân tích các văn bản được dịch sang cùng một ngôn ngữ.

– Tìm mối liên hệ giữa các từ (từ đồng nghĩa).

– Tìm các văn bản liên quan tới câu hỏi (trong truy vấn thông tin).

Các điểm hạn chế của phương pháp LSI có thể kể đến như sau:

– Việc chuyển sang không gian khái niệm tương đối khó diễn tia sang ngôn ngữ tự nhiên. Chẳng hạn, trong ví dụ trước:

$\{(car), (auto), (house)\} \rightarrow \{(2,05 * car + 0,45 * auto), (house)\}$
có thể hiểu rằng: $2,05 * car + 0,45 * auto$ là chiều chỉ khái niệm "vehicle".
Tuy nhiên cũng có thể xảy ra trường hợp sau:

$\{(car), (guitar), (house)\} \rightarrow \{(2,05 * car + 0,45 * guitar), (house)\}.$

Trường hợp này thì "car" và "guitar" là hai khái niệm không giống nhau và khó có thể diễn tả theo ngôn ngữ tự nhiên khi cùng đưa về một khái niệm chung.

- LSI không làm việc được với tính đa hình của từ. Một từ có thể có nhiều nghĩa, tức là về nguyên tắc là có thể làm tăng số chiều biểu diễn. Tuy nhiên, LSI làm giảm số chiều, như vậy LSI không làm việc được với từ mang nhiều nghĩa khác nhau.

• Biểu diễn theo cụm các từ

Ý tưởng biểu diễn theo cụm các từ cũng xuất phát từ việc làm giảm số chiều biểu diễn như phương pháp LSI, nhưng tiếp cận theo phương pháp khác. Trước hết là gom các từ lại thành các cụm (clusters), sau đó sử dụng các cụm này như các đầu vào để giải quyết các bài toán như phân cụm, phân lớp. Các nghiên cứu chi tiết hơn có thể được tham khảo tại [BC98, KN0, PTL93].

5.3. Các phương pháp lựa chọn các từ trong biểu diễn văn bản

Trong phần này chúng ta sẽ xem xét các phương pháp lựa chọn các từ biểu diễn trong văn bản. Như đã trình bày trong luật Zipf, những từ có tần số xuất hiện cao nhất (trong tiếng Anh) như "a", "an", "the" không phải là những từ quan trọng nhất. Cách đơn giản nhất là lập một danh sách các từ như vậy và loại bỏ nó khi biểu diễn văn bản. Đối với từ có tần số thấp nhất (xuất hiện từ 1 đến 2 lần) thì thông thường người ta vẫn giữ lại khi biểu diễn, bởi vì các từ này đóng vai trò quan trọng trong nội dung văn bản.

5.3.1. Loại bỏ các từ dừng trong biểu diễn văn bản

Trước hết có thể quan sát thấy rằng, trong tiếng Anh có nhiều từ chỉ dùng để phục vụ cho biểu diễn cấu trúc câu, chứ không biểu đạt nội dung của nó, chẳng hạn như các giới từ, từ nối,... Những từ như vậy xuất hiện nhiều trong văn bản mà không có liên quan gì tới chủ đề, hoặc nội dung nào đó của văn bản. Do đó, có thể loại bỏ những từ như vậy, những từ đó được xem như là những từ dừng (stop words). Ví dụ, danh sách các từ dừng có thể xem trong Bảng 5.1.

Bảng 5.1. Ví dụ danh sách các từ dừng [Sal71]

| | | |
|---------|----------|--------|
| a | been | do |
| able | before | does |
| about | below | during |
| after | best | each |
| again | but | else |
| all | by | enough |
| almost | came | ever |
| also | can | except |
| am | cannot | few |
| and | clearly | for |
| are | come | former |
| as | consider | from |
| at | could | get |
| be | despite | goes |
| because | did | going |
| ... | | |

5.3.2. Chọn từ gốc (word stemming)

Trong tiếng Anh hay trong nhiều ngôn ngữ khác, nhiều từ có chung một từ gốc, hoặc là biến thể sang từ một từ gốc nào đó. Chẳng hạn, các từ "computer", "computers" hoặc "computing" đều có chung một từ gốc là "comput". Ý tưởng chọn từ gốc để biểu diễn là biểu diễn các từ trong văn bản thông qua từ gốc. Có nhiều phương pháp để chọn từ gốc, trong đó phương pháp thường được dùng nhất là thuật toán Porter [Por90].

5.3.3. Biểu diễn thông qua mô hình n-gram

Một phương pháp khác biểu diễn theo cụm các từ là biểu diễn theo n-gram. n-gram là mô hình xác suất cho việc dự đoán phần tử tiếp theo trong một chuỗi. Trong biểu diễn văn bản, gram được coi là một đơn vị từ, trong đó có thể gồm 1 từ (unigram), 2 từ (bigram) hoặc 3 từ (trigram). Chẳng hạn, cụm 2 từ như "personal computer", "information retrieval", "computer science" có thể là 2-gram, biểu diễn nội dung của cụm hai từ. Việc sử dụng n-gram trong biểu diễn là để làm giàu thêm tính ngữ nghĩa của văn bản, tăng độ chính xác trong biểu diễn. Các chi tiết về mô hình n-gram và các ứng dụng của nó có thể tham khảo tại [MS99].

5.3.4. Độ đo trong lựa chọn đặc trưng

Một trong những phương pháp lựa chọn từ để biểu diễn là dựa vào độ đo đặc trưng. Đặc trưng (feature) là khái niệm chung trong học máy và khai

phá dữ liệu. Trong dữ liệu văn bản, có thể chia độ đo lựa chọn đặc trưng thành hai loại sau: độ đo dựa trên tần số và độ đo dựa trên lý thuyết thông tin. Độ đo thứ nhất dựa trên tần số thường đơn giản lược bỏ những từ có tần số xuất hiện thấp (từ 1 đến 3 hoặc 5 lần) trong văn bản tùy theo ứng dụng. Độ đo thứ hai được nghiên cứu nhiều hơn là độ đo sử dụng lý thuyết thông tin, độ đo này thường dựa trên thông tin của chủ đề của văn bản. Giả sử tập các chủ đề là $C = \{c_i\}_{i=1}^m$, ký hiệu $f(t_k, c_i)$ là độ đo của từ t_k trong chủ đề c_i . Có thể kể tới các độ đo sau:

1. DIA (Darmstadt Indexing Approach – Tiếp cận đánh chỉ số Darmstadt): Được đề xuất bởi Fuhn và đồng nghiệp [FHK91]:

$$f(t_k, c_i) = z(t_k, c_i) = p(c_i | t_k)$$

2. Độ đo IG (Information Gain):

$$f(t_k, c_i) = IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} p(t, c) \log \frac{p(t, c)}{p(t)p(c)}$$

3. Độ đo thông tin tương hỗ (mutual information):

$$f(t_k, c_i) = \log \frac{p(t_k, c_i)}{p(t_k)p(c_i)}$$

4. Độ đo Khi-bình phương (Chi-square):

$$f(t_k, c_i) = \chi^2(t_k, c_i) = \frac{|Tr|[(p(t_k, c_i).p(\bar{t}_k, \bar{c}_i)) - p(t_k, \bar{c}_i)p(\bar{t}_k, c_i)]^2}{p(t_k)p(\bar{t}_k)p(c_i)p(\bar{c}_i)}$$

5. Độ đo liên quan (Relevancy score):

$$f(t_k, c_i) = RS(t_k, c_i) = \log \frac{p(t_k | \bar{c}_i) + d}{p(\bar{t}_k, \bar{c}_i) + d}$$

6. Tỷ lệ dư (Odd Ratio):

$$f(t_k, c_i) = OR(t_k, c_i) = \frac{p(t_k | c_i)(1 - p(t_k | \bar{c}_i))}{(1 - p(t_k | c_i))p(t_k | \bar{c}_i)}$$

Trong đó, $p(t_k, c_i)$ ký hiệu là xác suất của từ t_k có trong chủ đề c_i và $p(t_k, \bar{c}_i)$ là xác suất của từ t_k không có trong chủ đề c_i .

Các độ đo trên là tính cho từng lớp. Độ đo cho toàn bộ các lớp trong tập hợp có thể được tính theo nhiều cách khác nhau:

$$f(t_k) = \sum_{i=1}^{|C|} f(t_k, c_i) \text{ hoặc } f(t_k) = \sum_{i=1}^{|C|} p(c_i)f(t_k, c_i) \text{ hoặc } f(t_k) = \max_{i=1}^{|C|} f(t_k, c_i)$$

Trong thực tế, hai cách tính đầu tiên được sử dụng nhiều nhất. Việc nghiên cứu các độ đo lựa chọn đặc trưng trong học máy và khai phá dữ liệu nói chung có thể tham khảo tại [KJ97, LM98], trong văn bản nói riêng có thể tham khảo tại [DPH98, Lew91].

5.4. Thu gọn đặc trưng biểu diễn

Với các tài liệu văn bản, mỗi một từ khoá duy nhất sẽ biểu diễn một chiều trong không gian biểu diễn. Do đó, kích thước của không gian biểu diễn văn bản thường rất lớn, việc tính toán sẽ tốn nhiều thời gian. Thêm nữa, một tài liệu văn bản khi được biểu diễn dưới dạng một vector, thì số lượng các phần tử trong vector đó có giá trị 0 là rất lớn, điều này cũng có thể là một nguyên nhân làm cho việc tính toán phân lớp phức tạp và khó khăn hơn. Một trong những giải pháp để khắc phục những vấn đề trên là thu gọn số lượng các đặc trưng (feature) bằng cách lựa chọn các đặc trưng có khả năng ảnh hưởng đến chất lượng phân lớp của các giải thuật phân lớp, còn các đặc trưng khác có thể bỏ qua. Việc thu gọn này cần đảm bảo sao cho các đặc trưng còn lại vẫn có khả năng "đại diện" cho toàn bộ văn bản, không làm giảm chất lượng phân lớp.

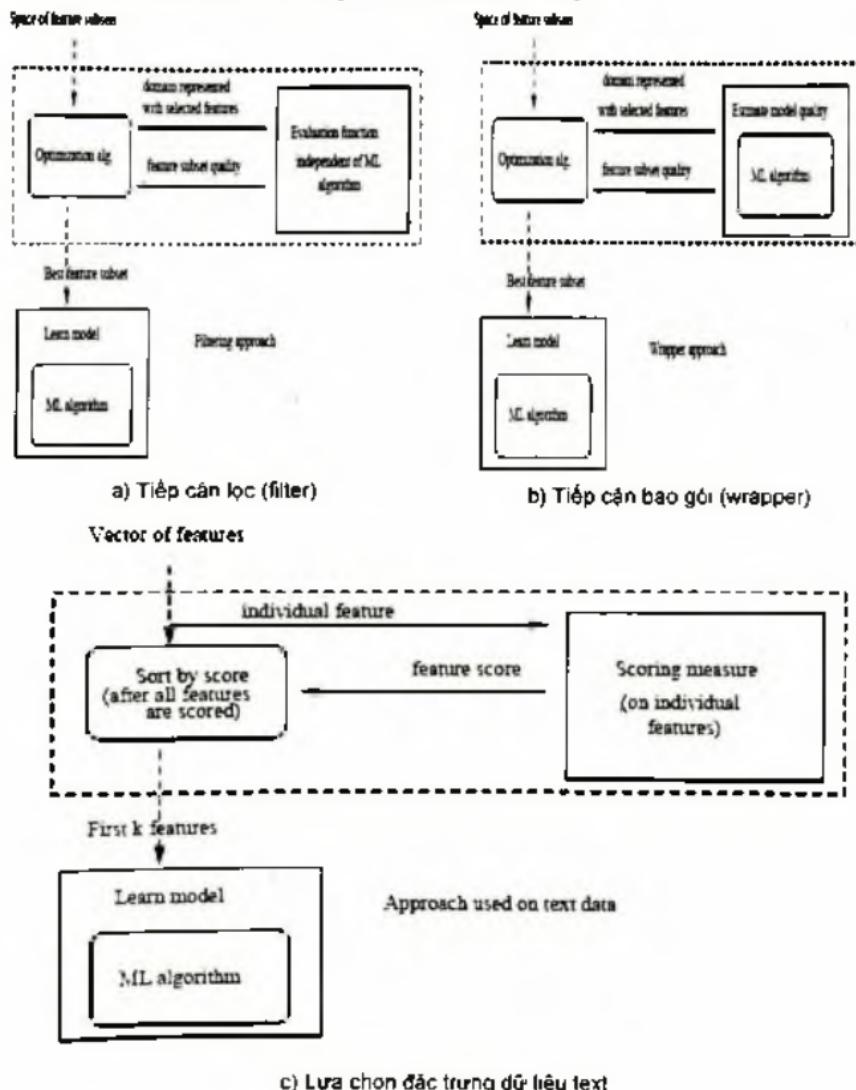
Lựa chọn đặc trưng là tiến trình lựa chọn một tập các đặc trưng xuất hiện trong tập đào tạo và chỉ sử dụng các tập này như là các đặc trưng để biểu diễn văn bản. Lựa chọn đặc trưng bao gồm 2 mục đích chính: *Thứ nhất*, nó làm cho quá trình huấn luyện các bộ phân lớp hiệu quả hơn bằng cách giảm kích thước của không gian các đặc trưng, điều này đặc biệt quan trọng đối với các giải thuật có chi phí huấn luyện là đắt. *Thứ hai*, lựa chọn các đặc trưng thường tăng tính đúng đắn cho quá trình phân lớp, vì nó có thể giúp loại bỏ các đặc trưng nhiễu. Một đặc trưng nhiễu là một đặc trưng mà khi thêm vào biểu diễn tài liệu, nó sẽ làm tăng các lỗi phân loại trên dữ liệu mới. Ví dụ, trong miền dữ liệu huấn luyện của một tập dữ liệu nào đó có một thuật ngữ hiếm gặp, chẳng hạn từ *arachnocentric*. Từ này không có thông tin nào để cập đến lớp China, nhưng tất cả các tài liệu chứa từ arachocentric đều xuất hiện trong các tài liệu huấn luyện cho lớp China. Khi đó, các giải thuật phân lớp thường gán một trọng số nào đó cho từ arachnocentric ảnh hưởng đến việc phân cho lớp China đối với dữ liệu mới.

Chúng ta có thể xem lựa chọn đặc trưng như một phương pháp để thay thế một bộ phân lớp phức tạp (sử dụng tất cả các đặc trưng) bằng một bộ phân lớp đơn giản hơn (do nó chỉ sử dụng một tập con của các đặc trưng).

Theo Dunja Mladenic' [Mla98], bài toán lựa chọn (thu gọn) đặc trưng là một bài toán tối ưu chọn ra từ một tập F các đặc trưng một tập con F^* , sao cho đại diện tốt nhất cho F trong bài toán phân lớp. Như đã biết, tập tất cả các tập con của F có lực lượng $2^{|F|}$ là rất lớn, vì vậy, chỉ có thể thi hành các thuật toán tìm kiếm trong không gian gồm có $2^{|F|}$ phần tử nói trên. Một số phương pháp tìm kiếm tập con F^* điển hình là:

- Lựa chọn "tiến": Xuất phát từ tập con rỗng, bổ sung dần các đặc trưng tốt nhất vào.

- Loại bỏ "lùi": Xuất phát từ tập F, loại bỏ dần các đặc trưng kém giá trị ra.
- Lựa chọn "tiến bậc thang": Xuất phát từ tập con rỗng, trong mỗi bước dùng chiến thuật tham lam bổ sung và loại bỏ đặc trưng.
- Loại bỏ "lùi bậc thang": Xuất phát từ tập F, trong mỗi bước dùng chiến thuật tham lam bổ sung và loại bỏ đặc trưng.



Hình 5.4. Sơ đồ lựa chọn đặc trưng dữ liệu [Mla98]

- Biến đổi ngẫu nhiên: Xuất phát từ tập con các đặc trưng được lựa chọn ngẫu nhiên, từng bước thêm, bớt các đặc trưng một cách ngẫu nhiên. Thuật toán dừng sau một số lần thao tác.

Hai cách tiếp cận để lựa chọn đặc trưng là lọc (filter) và bao gói (wrapper) được trình bày tại Hình 5.4a,b.

Dunja Mladenic cũng giới thiệu một sơ đồ lựa chọn đặc trưng cho dữ liệu văn bản (Hình 5.4c). Đầu vào của sơ đồ này là tập đặc trưng biểu diễn văn bản ban đầu, đầu ra là k đặc trưng thu gọn tối đa được dùng để biểu diễn văn bản trong bài toán phân lớp.

Thuật toán lựa chọn đặc trưng cơ bản được mô tả ở hình 5.5. Cho một lớp c, tính toán một hàm tiện ích A(t, c) cho mỗi thuật ngữ trong tập từ vựng, sau đó lựa chọn k thuật ngữ có giá trị A(t, c) là cao nhất. Tất cả các thuật ngữ còn lại sẽ bị loại bỏ và không được sử dụng phân lớp. Ở đây giới thiệu ba hàm tiện ích khác nhau trong phân lớp: thông tin tương hỗ (mutual information) $A(t, c) = I(Ut, Cc)$; hàm χ^2 , $A(t, c) = \chi^2(t, c)$ và hàm tần số $A(t, c) = N(t, c)$.

```

SELECTFEATURES(ID, c, k)
1   V ← EXTRACTVOCABULARY(ID)
2   L ← {}
3   for each t ∈ V
4     do A(t, c) ← COMPUTEFEATUREUTILITY(ID, t, c)
5     APPEND(L, (A(t, c), t))
6   return FEATURESWITHLARGESTVALUES(L, k)

```

Hình 5.5. Thuật toán lựa chọn đặc trưng cơ bản
cho việc lựa chọn k đặc trưng tối đa

5.4.1. Thông tin tương hỗ (Mutual Information)

Một phương pháp lựa chọn đặc trưng phổ biến để tính toán $A(t, c)$ là thông tin tương hỗ MI (Mutual Information) của thuật ngữ t với lớp c. MI đo mức độ thông tin (xuất hiện/không xuất hiện) của thuật ngữ t góp phần làm cho quyết định cho quá trình phân lớp đúng đắn trên lớp c. Công thức của MI là: $I(U, C) = \sum_{e_i \in \{1, 0\}} \sum_{e_c \in \{1, 0\}} p(U = e_i, C = e_c) \log \frac{p(U = e_i, C = e_c)}{p(U = e_i)p(C = e_c)}$

Với U là biến ngẫu nhiên, nó có giá trị là $e_i = 1$ (tài liệu hiện tại chứa thuật ngữ t) và $e_i = 0$ (tài liệu không chứa thuật ngữ t). Và C là biến ngẫu nhiên, nó có giá trị $e_c = 1$ (tài liệu có trong lớp c) and $e_c = 0$ (tài liệu không có trong lớp c).

Công thức trên được tính theo phương pháp ước lượng maximum-likelihood estimation (MLE), được biểu diễn bằng công thức:

$$I(U, C) = \frac{N_{11}}{N} \log_2 \frac{NN_{11}}{N_1 N_1} + \frac{N_{01}}{N} \log_2 \frac{NN_{01}}{N_0 N_1} + \frac{N_{10}}{N} \log_2 \frac{NN_{10}}{N_1 N_0} + \frac{N_{00}}{N} \log_2 \frac{NN_{00}}{N_0 N_0}$$

Trong đó các biến N được tính thông qua các giá trị e_l và e_c (được ký hiệu bằng các con số trong phần chỉ số dưới). Ví dụ. N_{10} là số tài liệu mà có chứa t ($e_l = 1$) và không có trong c ($e_c = 0$). $N_{11} = N_{10} + N_{11}$ là số tài liệu mà có chứa t ($e_l = 1$). Các biến N_{ij} khác được giải thích tương tự. Và N là tổng các tài liệu: $N = N_{11} + N_{01} + N_{10} + N_{11}$.

Ví dụ, xem xét lớp poultry và thuật ngữ export trong tập dữ liệu Reuters-RCV1. Giá trị đếm cho các tài liệu với 4 khả năng kết hợp các giá trị chỉ của chỉ số dưới như sau:

| | $e_l = e_{\text{export}} = 1$ | $e_l = e_{\text{export}} = 0$ |
|--------------------------------|-------------------------------|-------------------------------|
| $e_c = e_{\text{poultry}} = 1$ | $N_{11} = 49$ | $N_{01} = 141$ |
| $e_c = e_{\text{poultry}} = 0$ | $N_{10} = 27,652$ | $N_{00} = 774,106$ |

Sau khi đưa các giá trị đó vào trong biểu thức MLE, ta có:

$$\begin{aligned} I(U, C) = & \frac{49}{801,948} \log_2 \frac{801,948.49}{(49+27,652)(49+141)} \\ & + \frac{141}{801,948} \log_2 \frac{801,948.141}{(141+774,106)(49+141)} \\ & + \frac{27,652}{801,948} \log_2 \frac{801,948.27,652}{(49+27,652)(27,652+774,106)} \\ & + \frac{774,106}{801,948} \log_2 \frac{801,948.774,106}{(141+774,106)(27,652+774,106)} \\ & \approx 0,000105. \end{aligned}$$

Để lựa chọn k các thuật ngữ: t_1, \dots, t_k cho bởi lớp nào đó, sử dụng thuật toán lựa chọn đặc trưng trong hình 5.4, tính toán các hàm tiện ích cho tất cả các thuật ngữ $A(t, c) = I(U_t, C_t)$ và sau đó lựa chọn k thuật ngữ có các giá trị lớn nhất.

Thông tin tương hỗ xác định bao nhiêu thông tin (theo lý thuyết thông tin) mà một thuật ngữ có trong một lớp nào đó. Nếu một thuật ngữ có phân bố trên 1 lớp giống phân bố trên toàn tập dữ liệu thì $I(U; C) = 0$. M1 đạt đến giá trị lớn nhất của nó nếu thuật ngữ là một đại diện hoàn hảo cho một lớp: thuật ngữ xuất hiện trong tài liệu nếu và chỉ nếu tài liệu thuộc một lớp nào đó.

Hình 5.6 liệt kê các thuật ngữ với giá trị thông tin tương hỗ cao cho 6 lớp. Các thuật ngữ được lựa chọn (như london, uk, british cho lớp UK) là các minh chứng rõ ràng cho việc đưa ra quyết định phân lớp các tài liệu thuộc vào lớp tương ứng. Cuối danh sách của lớp UK, có thể tìm thấy các thuật ngữ như peripherals và tonight (không chỉ ra trong hình), các thuật ngữ này rõ ràng không hữu ích trong việc quyết định phân lớp các tài liệu tương ứng với lớp này.

| UK | China | poultry | |
|------------|-----------|-------------|--------|
| london | 0.1925 | chicken | 0.0097 |
| uk | 0.0755 | meat | 0.0008 |
| british | 0.0596 | agriculture | 0.0005 |
| stg | 0.0555 | avian | 0.0004 |
| britain | 0.0469 | broiler | 0.0003 |
| pic | 0.0357 | veterinary | 0.0003 |
| england | 0.0238 | bird | 0.0003 |
| pence | 0.0212 | inspection | 0.0003 |
| pounds | 0.0149 | pathogenic | 0.0003 |
| english | 0.0126 | | |
| coffee | elections | sports | |
| coffee | 0.0117 | soccer | 0.0681 |
| bags | 0.0042 | cup | 0.0515 |
| growers | 0.0025 | match | 0.0441 |
| kg | 0.0019 | matches | 0.0408 |
| colombia | 0.0018 | played | 0.0388 |
| brazil | 0.0016 | league | 0.0386 |
| export | 0.0014 | beat | 0.0301 |
| exporters | 0.0013 | game | 0.0299 |
| exports | 0.0013 | games | 0.0284 |
| crop | 0.0012 | team | 0.0264 |
| democratic | 0.0198 | | |

Hình 5.6. Các đặc trưng với điểm thông tin tương hỗ cao cho 6 lớp Reuters-RCV1 trong các mô hình đa thức và Bernoulli

Như mong muốn, việc giữ lại các thuật ngữ có thông tin và loại bỏ đi các thuật ngữ mà không có thông tin để giảm nhiễu và cải thiện độ chính xác của các bộ phân lớp. Minh chứng cho việc làm tăng độ chính xác khi thực hiện việc thu gọn các đặc trưng này có thể xem Hình 5.6, trong đó mô tả giá trị của F1 thay đổi tương ứng với số lượng các đặc trưng, thực nghiệm trên tập dữ liệu RCV1. So sánh giá trị F1 khi sử dụng toàn bộ 132.766 đặc trưng với giá trị F1 khi sử dụng 50 đặc trưng, ta thấy rằng, việc lựa chọn đặc trưng MI làm tăng F1 khoảng 0,1 với mô hình đa thức và nhiều hơn 0,2 cho mô hình Bernoulli.

Với mô hình Bernoulli, giá trị F1 tăng khá sớm tại giá trị 10 đặc trưng, và mô hình Bernoulli tốt hơn mô hình đa thức. Như vậy, khi số lượng các đặc trưng là ít, ta nên quan tâm đến mô hình Boolean. Với mô hình đa thức, giá trị F1 tăng tại vị trí 30 đặc trưng và hiệu quả của nó cũng tăng tại vị trí 1000 đặc trưng trở đi. Nguyên nhân là do mô hình đa thức dựa vào tần suất xuất hiện của các thuật ngữ trong công thức ước lượng tham số và phân lớp. Do đó, nó khai thác tốt một số lượng lớn các đặc trưng hơn khi so sánh với mô hình Bernoulli. Cho dù sử dụng phương pháp nào đi chăng nữa, thì việc lựa chọn một số lượng các đặc trưng thích hợp sẽ cho kết quả tốt hơn khi dùng toàn bộ tập các đặc trưng.

5.4.2. Lựa chọn đặc trưng χ^2

Phương pháp lựa chọn đặc trưng thông dụng khác là χ^2 . Theo thống kê, kiểm tra χ^2 được áp dụng để kiểm tra tính độc lập của hai biến ngẫu nhiên, trong đó hai sự kiện A và B được định nghĩa tính độc lập nếu

$p(AB) = p(A)p(B)$, hay $p(A|B) = P(A)$ và $p(B|A) = P(B)$. Trong lựa chọn đặc trưng, hai biến ngẫu nhiên là sự xuất hiện của thuật ngữ U và sự xuất hiện của lớp c. Công thức tính giá trị χ^2 được diễn giải như sau:

$$\chi^2(D, t, c) = \sum_{e_t \in \{0, 1\}} \sum_{e_c \in \{0, 1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

Các giá trị e_t và e_c có ý nghĩa như đã đề cập ở trên. N là tần số quan sát được từ dữ liệu D, còn E là tần số kỳ vọng. Ví dụ, E_{11} là tần số kỳ vọng của t và c xuất hiện đồng thời trong một tài liệu với giả sử rằng thuật ngữ và lớp là độc lập với nhau.

Ví dụ, tính toán giá trị của E_{11} cho một tập dữ liệu nhận được:

$$E_{11} = N \times p(t) \times p(c) = N \times \frac{N_{11} + N_{01}}{N} \times \frac{N_{11} + N_{10}}{N}$$

$$= N \times \frac{49 + 41}{N} \times \frac{49 + 27652}{N} \approx 6,6$$

Trong đó N là tổng số tài liệu như mô tả ở trên. Tính các giá trị E_{ij} khác theo cùng cách:

| | $e_{\text{export}} = 1$ | $e_{\text{export}} = 0$ |
|--------------------------------|--|--|
| $e_c = e_{\text{poultry}} = 1$ | $N_{11} = 49$ $E_{11} \approx 6.6$ | $N_{01} = 141$ $E_{10} \approx 183.4$ |
| $e_c = e_{\text{poultry}} = 0$ | $N_{10} = 27,652$ $E_{01} \approx 27,694.4$ | $N_{00} = 774,106$ $E_{00} \approx 774,063.6$ |

Đưa các giá trị tìm được vào công thức tính χ^2 , chúng ta lấy giá trị của χ^2 là 284:

$$\chi^2(D, t, c) = \sum_{e_t \in \{0, 1\}} \sum_{e_c \in \{0, 1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \approx 284$$

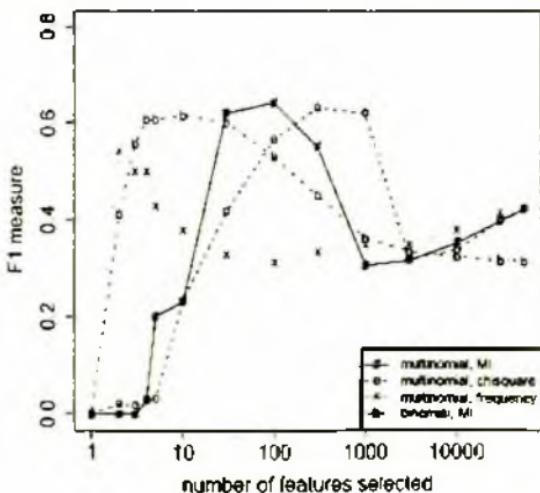
χ^2 là cách đo độ chênh lệch nhau giữa giá trị kỳ vọng E và giá trị quan sát N. Giá trị của χ^2 mà cao thì nó chỉ ra rằng, giả thuyết độc lập (hay giá trị kỳ vọng E và giá trị quan sát được là giống nhau) là sai. Trong ví dụ này, $\chi^2 \approx 284$ là có giá trị cao, ta có thể loại bỏ giả thuyết rằng, đặc trưng poultry và export là độc lập với nhau.001 cơ hội sai. Nếu có hai sự kiện phụ thuộc, thì sự xuất hiện của các thuật ngữ tạo ra sự xuất hiện của lớp là cao hay thấp. Do đó, nó cũng hữu ích như một đặc trưng. Đây là ý nghĩa cơ bản của phương pháp lựa chọn đặc trưng χ^2 . Công thức đơn giản hơn dùng để tính toán giá trị χ^2 là:

$$\chi^2(D, t, c) = \frac{(N_{11} + N_{10} + N_{01} + N_{00})(N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11} + N_{01})(N_{11} + N_{10})(N_{10} + N_{00})(N_{01} + N_{00})}$$

5.4.3. Lựa chọn đặc trưng dựa trên tần suất

Phương pháp lựa chọn thứ ba là lựa chọn đặc trưng dựa trên tần suất, ví dụ, lựa chọn các thuật ngữ là thông dụng nhất trong lớp. Tần suất có thể được định nghĩa là tần xuất của tài liệu (số tài liệu trong lớp c mà chứa thuật ngữ t), hoặc như tần suất toàn cục (số lần xuất hiện của t trong lớp tài liệu c). Tần suất của tài liệu phù hợp cho mô hình Bernoulli, tần suất toàn cục thì phù hợp với mô hình đa thức.

Phương pháp lựa chọn đặc trưng dựa trên tần suất có thể sẽ lựa chọn một vài thuật ngữ không có thông tin cụ thể liên quan đến lớp. Ví dụ, tên các thứ trong tuần (Monday, Tuesday,...) có thể xuất hiện thường xuyên trong tất cả các lớp trong miền dữ liệu các trang Web tin tức trên mạng. Nhưng khi có hàng nghìn đặc trưng được lựa chọn, thì lựa chọn đặc trưng dựa trên tần suất thường xử lý tốt (quá trình được thực hiện với tốc độ rất cao do không cần tính toán phức tạp). Do vậy, trong một số trường hợp nhất định, thì lựa chọn đặc trưng dựa trên tần xuất có thể là sự lựa chọn tối ưu so với các phương pháp lựa chọn đặc trưng khác. Tuy nhiên, trong kết quả thực nghiệm được trên Hình 5.7 cho thấy lựa chọn đặc trưng dựa trên tần xuất thực hiện kém lại kết quả phân lớp kém hơn so với phương pháp M1 và χ^2 . Do đó trong trường hợp này không nên lựa chọn phương pháp lựa chọn đặc trưng dựa trên tần suất.



Hình 5.7. Ảnh hưởng của kích thước tập đặc trưng lên tính chính xác

5.4.4. Lựa chọn đặc trưng cho nhiều bộ phân lớp

Trong trường hợp hệ thống có nhiều bộ phân lớp khác nhau hay trong tập dữ liệu có nhiều hơn 2 lớp, việc tạo ra các tập đặc trưng riêng lẻ cho từng bộ phân lớp sẽ làm cho quá trình phân lớp phức tạp. Để khắc phục điều này, có thể chọn ra một tập đặc trưng chung cho tất cả các bộ phân lớp. Một cách để thực hiện việc này là tính toán thông kê χ^2 cho bảng $n \times 2$, trong đó các cột thể hiện sự xuất hiện/không xuất hiện của các thuật ngữ và mỗi dòng tương ứng với một lớp. Ta có thể chọn thuật ngữ k với giá trị thông kê χ^2 cao nhất như giải thuật đề cập ở trước.

Một phương pháp thông dụng hơn là, tính toán để lựa chọn các đặc trưng cho từng lớp, sau đó thực hiện thao tác kết hợp. Phương pháp kết hợp tính toán giá trị thông tin kết hợp của từng đặc trưng. Ví dụ, tính giá trị trung bình A(t, c) cho từng đặc trưng t, sau đó lựa chọn k đặc trưng cao nhất. Một phương pháp phổ biến khác là lựa chọn đặc trưng cho từng lớp, với mỗi lớp lấy ra k đặc trưng để kết hợp lại và tạo ra một tập đặc trưng toàn cục cho toàn bộ các lớp.

5.5. Phương pháp biểu diễn trang Web

Khác với một trang văn bản thông thường, trong nội dung một trang Web còn có các chi dẫn (liên kết) ngoài tới các trang Web khác với ý nghĩa là nội dung đang được nói tại trang Web hiện thời cũng là một nội dung được quan tâm của trang Web được chỉ dẫn tới. Trong nhiều trường hợp, nội dung tại trang Web được chỉ dẫn tới còn là một "giải thích" cho nội dung đang được quan tâm. Điều đó có nghĩa là, một chủ đề trong tập chủ đề của một trang Web cũng là một chủ đề trong tập chủ đề của các trang Web mà nó chỉ dẫn tới. Quan hệ "chủ đề cùng quan tâm" là đối xứng giữa hai trang Web tồn tại một liên kết giữa chúng. Chính vì lý do đó, biểu diễn trang Web có những điểm mở rộng so với biểu diễn văn bản thông thường. Những khía cạnh mở rộng đáng kể nhất của biểu diễn trang Web so với biểu diễn văn bản thông thường gồm có việc mở rộng nội dung trang Web từ các trang Web kề cận nó và khai thác kiến trúc trang Web vào biểu diễn nó.

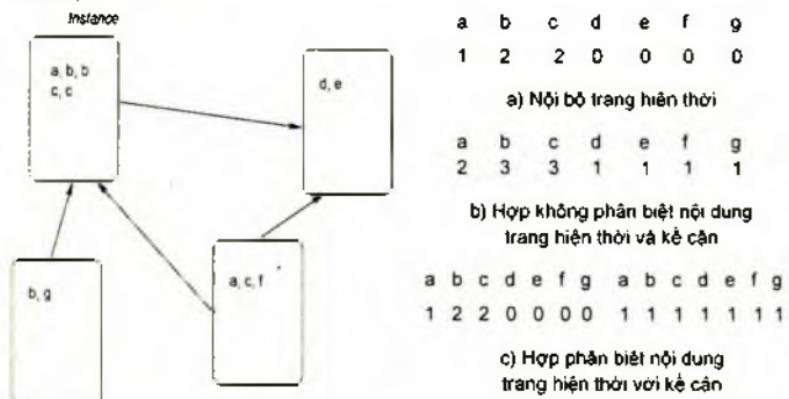
5.5.1. Mở rộng nội dung văn bản trang Web bằng nội dung văn bản trang Web kề cận

Khái niệm "kề cận" của hai trang Web được hiểu theo nghĩa tồn tại ít nhất một liên kết giữa chúng. Việc mở rộng nội dung văn bản từ các văn bản kề cận nó xuất phát từ nhận định cho rằng, việc sử dụng các siêu liên kết có nguồn gốc từ sự liên quan về nội dung giữa chúng.

Trong [Sla02], Sen Slattery giới thiệu bốn phương án xây dựng biểu diễn một trang Web (Hình 5.8), trong đó:

- Phương án đầu tiên chỉ sử dụng nội dung trang Web hiện thời.
- Phương án thứ hai sử dụng trộn nội dung trang Web hiện thời với các kè cận của nó.
- Trong phương án thứ ba, biểu diễn trang Web gồm hai phần: phần đầu sử dụng nội dung trang Web hiện thời, phần thứ hai sử dụng nội dung của các trang Web kè cận với nó.

- Phương án cuối cùng được coi là tổng quát hóa của phương án thứ ba theo hướng số lượng mức được tăng từ 2 lên k. Trong biểu diễn loại này, cho trước một mức k và một kho dữ liệu trang Web. Biểu diễn trang Web sẽ bao gồm k thành phần (chẳng hạn, k = 4 như Hình 5.8d) tương ứng với k tập trang Web liên quan tới Web hiện thời (mỗi tập trang Web đó được gọi là section). Section 1 gồm chính trang hiện thời. Section 2, 3,..., k là tập trang Web kè cận với các trang Web ở section i và không thuộc các section 1, 2,..., i - 1.



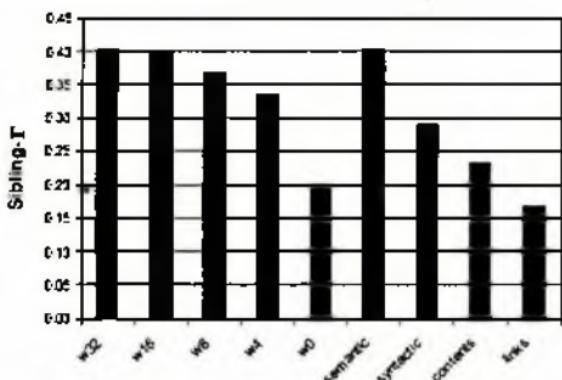
Hình 5.8. Bốn phương án biểu diễn trang Web cho sơ đồ Web (*) [Sla02]

Sen Slattery cùng trình bày thực nghiệm so sánh ba thuật toán học phân lớp FOIL, kNN, kNN mở rộng với cách thức biểu diễn trang Web theo phương án thứ 3. Kết quả thực nghiệm cho thấy, việc mở rộng nội dung trang Web là có ý nghĩa.

Trong [HGK02], Taher H. Haveliwala và cộng sự đề xuất giải pháp khai thác nội dung các trang Web có siêu liên kết tới trang Web w để mở rộng nội dung của w theo hai nguồn chính:

– Nội dung các văn bản "neo" (anchor text) đính kèm với siêu liên kết trả tới w.

– Đoạn văn bản lân cận về hai phía của siêu liên kết trả tới w trong trang Web chứa siêu liên kết này.



Hình 5.9. Hiệu quả của biểu diễn theo lân cận siêu liên kết [HGK02]

Giải pháp do Taher H. Haveliwala và cộng sự có cơ sở lý luận từ tính cục bộ ngữ nghĩa của siêu liên kết, những từ lân cận siêu liên kết (bao gồm cả trong văn bản neo) là những từ liên quan nhất tới trang Web w mà siêu liên kết mong muốn chuyển tới w. Thực nghiệm mà các tác giả tiến hành đã minh chứng cho lý luận nói trên (Hình 5.9). Trong mô tả kết quả thực nghiệm:

– *w_i*: kích thước của số liên kết, chính là số lượng từ khoá mở rộng theo hai phía của siêu liên kết. Ở đây có kết hợp giữa nội dung của trang Web với nội dung văn bản anchor và lân cận.

– *Link*: biểu diễn theo nội dung các siêu liên kết tới w.

– *Content*: biểu diễn theo nội dung của chính trang w.

– *Syntactic*: phân tích biến của số động theo cú pháp.

– *Sematic*: phân tích biến của số động theo ngữ nghĩa.

Kết quả mô tả trên Hình 5.9 cho thấy độ hiệu quả của thuật toán phân lớp đơn diệu tăng theo kích thước của số liên kết.

5.5.2. Khai thác các yếu tố trong trang Web được bổ sung từ ngôn ngữ tạo trang Web

Khai thác các yếu tố trong trang Web được bổ sung từ ngôn ngữ tạo trang Web (chẳng hạn, các thẻ tạo trang Web) vào việc xác định các giá trị trọng số tương ứng với các từ. Vì lý do các thẻ HTML trong một trang Web thường được gán một ý nghĩa nhất định. Do đó, việc khai thác các thẻ này sẽ làm cho biểu diễn của văn bản được làm giàu hơn.

Ví dụ, cặp thẻ `<title> ... </title>` được quy định là biểu diễn tiêu đề của nội dung trang Web để cặp đến. Cặp thẻ để mục `<h1> ... </h1>`, `<h2> ... </h2>`,... được quy định để hiển thị các đề mục to cho đến các đề mục nhỏ hơn trong nội dung trang Web.

Tiêu đề và các đề mục thường nêu các ý chính, quan trọng của trang Web. Do đó, có thể gán cho nội dung (các từ khoá xuất hiện) trong các cặp thẻ này có trọng số cao hơn so với các nội dung khác (các từ khoá ở các vị trí khác).

Câu hỏi và bài tập

1. Tạo một tập dữ liệu các trang Web lấy từ một trang tin tức tiếng Việt nào đó (chẳng hạn <http://vnexpress.net>), chia làm 2 tập: một tập dùng để huấn luyện, tập còn lại dùng để kiểm tra (đánh giá) các bộ phân lớp. Tao danh sách các từ dùng cho tiếng Việt (những từ được coi là không có tác dụng trong quá trình phân lớp). Biểu diễn các trang Web thu được bằng mô hình Boolean và mô hình đa thức.
2. Thực hiện quá trình thu gọn các đặc trưng biểu diễn bằng các phương pháp đã được đề cập trong bài. Thực hiện việc phân lớp (Chương 8) tập dữ liệu này và so sánh chất lượng phân lớp khi sử dụng toàn bộ các đặc trưng với trường hợp phân lớp chỉ sử dụng các đặc trưng đã được lựa chọn

Chương 6

HỆ THỐNG TÌM KIẾM

6.1. Tìm kiếm trên Web

World Wide Web là một kho thông tin khổng lồ, giàu tài nguyên, song chứa đựng nhiều thách thức đối với khai phá dữ liệu. Thách thức thứ nhất là khối lượng dữ liệu trên Web tăng trưởng rất nhanh và ngày càng khổng lồ. Theo Gregory Piatetsky – Shapiro [GPS06], với tốc độ tăng trưởng nhanh chóng, Google lưu được khoảng 8 tỷ trang Web, Yahoo lưu khoảng 20 tỷ trang Web. Theo nguồn tin trên Internet⁽¹⁾ vào tháng 9/2009, Google đã đánh chỉ số được hơn 1 trillion (1 nghìn tỷ) địa chỉ Web khác nhau. Con người đã thừa nhận hiện tượng bùng nổ thông tin trên Web. Thách thức thứ hai là thông tin trên các trang Web rất đa dạng và phong phú về nội dung cũng như hình thức. Cùng với sự thay đổi và phát triển hàng ngày, hằng giờ về nội dung cũng như về số lượng các trang Web, thi vẫn để tìm kiếm thông tin đối với người sử dụng trên Internet lại ngày càng trở nên khó khăn hơn. Chính vì lẽ đó, chỉ một phần thông tin rất nhỏ từ khối lượng thông tin khổng lồ, đa dạng và biến động như thế là hữu ích đối với mỗi người dùng.

Một nhu cầu cấp bách được đặt ra là, cần phải xây dựng các công cụ tìm kiếm thông tin thực hiện việc quản lý nội dung các trang Web, tiếp nhận yêu cầu tìm kiếm của người dùng và cung cấp cho họ các trang Web có nội dung đáp ứng yêu cầu tìm kiếm.

Theo Sergey Brin và Lawrence Page [SL98], hai kiểu công cụ thông tin trên Internet điển hình tìm kiếm là máy tìm kiếm (search engine) và thư mục phân lớp (classified directory).

Máy tìm kiếm là công cụ tìm kiếm cho phép tìm kiếm các trang Web dựa theo các từ khóa trong một tập rất lớn các tài liệu Web. Kết quả tìm kiếm là danh sách các trang Web (tài liệu) có chứa từ khóa nói trên và được liệt kê theo thứ tự về độ quan trọng hay "hạng" của chúng. Thông thường, một số khó khăn thường gặp trong quá trình tìm kiếm hoặc là danh sách kết quả quá dài, hoặc là số lượng từ khóa trong câu hỏi ít (thường nhỏ hơn 4) và

⁽¹⁾ <http://www.news.com.au/technology/story/0,28348,25857420-5018992,00.html>

đặc biệt là có ngữ nghĩa không rõ ràng. Trong số các máy tìm kiếm điển hình thì AltaVista, Hotbot, Infoseek,... thuộc loại *máy tìm kiếm chung*, thường có độ chính xác tìm kiếm thấp; còn Inktomi, Excite, www.netpart.com, Cora,... thuộc loại cung cấp công nghệ tìm kiếm theo kiểu *dịch vụ tìm kiếm* trong miền thu hẹp cho nên có độ chính xác cao.

Thư mục phân lớp là công cụ tìm kiếm làm việc trên tập tài liệu Web với số lượng không nhiều, các tài liệu được tổ chức dưới dạng thư mục. Công việc tìm kiếm được tiến hành đọc theo các thư mục và kết quả danh sách theo thư mục. Lycos, Yahoo, CiteSeer,... là các thư mục phân lớp điển hình.

Khuynh hướng hiện nay đổi với công cụ tìm kiếm là *tích hợp các chức năng* tìm kiếm theo từ khóa và thư mục phân lớp. Hiện nay, AltaVista cũng cung cấp các dịch vụ catalog, Lycos trộn dịch vụ vào chức năng thu nhận được từ HotBot, còn Northern Light cung cấp *tổ hợp* các dịch vụ tìm kiếm *tổ chức động* kết quả tìm theo từ khóa thành nhóm, hoặc theo chủ đề tương tự, hoặc theo nguồn, kiểu.

Máy tìm kiếm còn được phân loại theo miền ứng dụng của nó. Theo cách phân loại này, máy tìm kiếm được phân loại thành loại phổ dụng và loại chuyên lĩnh vực. Máy tìm kiếm có miền ứng dụng cho mọi lĩnh vực và vì vậy, miền dữ liệu của nó là tập toàn bộ trang Web. Máy tìm kiếm chuyên lĩnh vực có miền ứng dụng là một lĩnh vực hay nhóm lĩnh vực có liên quan nhau.

6.1.1. Bài toán tìm kiếm thông tin

Định nghĩa 6.1: Bài toán tìm kiếm thông tin (Information Retrieval - IR) được mô tả như sau:

Cho D là một tập hợp gồm hữu hạn các văn bản, $D = \{d\}$ là nguồn tài nguyên chứa thông tin của hệ thống phục vụ nhu cầu người dùng. Đối với câu hỏi người dùng q (mà thường được cho dưới dạng một từ, hoặc một cụm các từ), hãy tìm ra tập con $R(q)$ trong D gồm các văn bản có liên quan tới câu hỏi q .

Các hệ thống tìm kiếm thông tin được xây dựng nhằm giải quyết bài toán tìm kiếm thông tin (IR) nói trên. Các hệ thống tìm kiếm thực hiện việc tìm kiếm các tài liệu, hoặc dựa trên phương pháp lựa chọn tài liệu, hoặc dựa trên phương pháp tính hạng liên quan. Thực tế, đối với câu hỏi q , kết quả nhận được từ hệ thống tìm kiếm không chính xác được tập $R(q)$ mà là tập $R'(q)$, được coi là một xấp xỉ nào đó của tập $R(q)$.

Theo phương pháp lựa chọn tài liệu (Document selection), hệ thống tìm kiếm thông tin có thể được biểu diễn như một hàm tìm kiếm:

$$f(d, q): D \times D \rightarrow \{0, 1\}$$

theo đó, với mỗi câu hỏi q thì hệ thống cần cung cấp tập:

$$R'(q) = \{d \in D \mid f(d, q) = 1\}$$

Công việc xây dựng hệ thống tìm kiếm chính là học hàm $f(d, q)$ nói trên.

Phương pháp lựa chọn là rất đơn giản, tuy nhiên, lại gặp một số hạn chế. Trong trường hợp "câu hỏi q quá phổ dụng" (chẳng hạn, câu hỏi q là từ "thể thao"), thì tập $R'(q)$ thường là quá lớn và không chính xác. Trong trường hợp ngược lại, khi mà "câu hỏi q quá chuyên biệt" thì tập $R'(q)$ có thể lại quá ít (chẳng hạn, câu hỏi q là cụm từ "thể thao và cuộc sống số"), thậm chí là rỗng vì không tìm được tài liệu liên quan nào.

Phương pháp tính hạng tài liệu (Document Ranking) khắc phục được hạn chế của phương pháp lựa chọn tài liệu. Theo phương pháp này, hệ thống tìm kiếm thông tin tiến hành việc học hàm tính hạng $f(d, q)$: $D \times D \rightarrow [0, 1]$ và với câu hỏi q, hệ thống cung cấp tập $R'(q) = \{d \in D \mid f(d, q) \geq \alpha\}$ với α là một số dương cho trước}. Tiêu chuẩn đặt ra đối với hàm $f(d, q)$ là nó cần thỏa mãn tính chất (một số tác giả gọi là tính *đơn điệu*) là nếu như văn bản d_1 liên quan với q nhiều hơn văn bản d_2 thì hàm f phải đảm bảo tính chất $f(d_1, q) \geq f(d_2, q)$.

Các công cụ tìm kiếm trên Internet, chẳng hạn như máy tìm kiếm sẽ được giới thiệu trong phần tiếp theo, thường kết hợp hai phương pháp này, theo đó tiến hành phương pháp lựa chọn tài liệu để chọn các trang Web có chứa các từ khoá trong câu hỏi, còn phương pháp tính hạng tài liệu được dùng để hiển thị danh sách các trang Web kết quả theo thứ tự giảm dần về hạng để trả lời người dùng.

6.1.2. Đánh giá chất lượng tìm kiếm thông tin

Tương tự như việc đánh giá chất lượng các bộ phân lớp, trích chọn thông tin, chất lượng của hệ thống tìm kiếm thông tin được đo bằng hai độ đo điển hình là độ hồi tưởng ρ (Recall) và độ chính xác π (Precision). Các độ đo này được tính toán theo các công thức sau:

$$\rho = \frac{|R \cap R'|}{|R|} \quad \text{và} \quad \pi = \frac{|R \cap R'|}{|R'|} \quad (6.1)$$

Điển giải bằng lời về độ hồi tưởng ρ là "tỷ lệ số lượng tài liệu tìm kiếm được liên quan tới q so với số lượng tài liệu liên quan tới q có trong nguồn tài nguyên" và độ chính xác π là "tỷ lệ số lượng tài liệu tìm kiếm được liên quan tới q so với số lượng tài liệu tìm kiếm được".

Trong nhiều trường hợp, thường sử dụng độ đo f_β được tổ hợp từ hai độ đo nói trên theo công thức tính toán sau:

$$f_\beta = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho}$$

Trường hợp đặc biệt, khi $\beta = 1$, nhận được một độ đo được sử dụng khá phổ biến:

$$f_1 = \frac{2\pi\rho}{\pi + \rho} \quad (6.2)$$

6.2. Máy tìm kiếm

6.2.1. Khái niệm

Máy tìm kiếm là một hệ thống phần mềm được xây dựng nhằm tiếp nhận yêu cầu tìm kiếm của người dùng, sau đó phân tích yêu cầu này và tìm kiếm thông tin trong CSDL tài liệu được tải xuống từ Internet và đưa ra kết quả là danh sách các trang Web có liên quan với yêu cầu cho người dùng.

Người dùng gửi một truy vấn dạng đơn giản nhất là một danh sách các từ khoá và máy tìm kiếm làm việc để trả lại một danh sách các trang Web có liên quan, hoặc có chứa các từ khoá đó. Phức tạp hơn, truy vấn có thể là cả một văn bản, hoặc một đoạn văn bản, hoặc nội dung tóm tắt của văn bản. Trong một số máy tìm kiếm, truy vấn được biểu diễn theo một ngôn ngữ hỏi (AltaVista, ASPseek,...).

6.2.2. Sơ lược về quá trình phát triển máy tìm kiếm

Theo Sergey Brin và Lawrence Page [SL98], máy tìm kiếm đầu tiên xuất hiện vào năm 1994 để bắt đầu một quá trình phát triển nhanh chóng của máy tìm kiếm.

Máy tìm kiếm đầu tiên WWW (WWW Worm) được McBryan phát triển vào năm 1994. Hệ thống cho phép chỉ số hoá được chừng 110.000 trang Web truy nhập được. Theo tổng hợp, giai đoạn từ tháng 3/1994 đến tháng 4/1994, máy tìm kiếm này nhận khoảng 1.500 câu hỏi hàng ngày.

Tới năm 1997 đã xuất hiện một số máy tìm kiếm chỉ số hoá từ 2 triệu (như WebCrawler) tới 100 triệu trang Web (như Watch). Vào tháng 11/1997, Alta Vista tiếp nhận 20 triệu câu hỏi hàng ngày.

Kể từ năm 2000 cho tới nay, đã xuất hiện các máy tìm kiếm chỉ số hoá được hàng tỷ trang Web và hàng ngày đáp ứng hàng trăm triệu câu hỏi.

Tại trang Web <http://www.searchengineshowdown.com/>, Greg R. Notess giới thiệu về một số nghiên cứu chung có tính cập nhật về các hệ thống tìm kiếm trên thế giới. Trong phần sau sẽ trình bày các kết quả nghiên cứu của tác giả về đặc trưng của các máy tìm kiếm. Khảo sát sự thay đổi theo nhiều phương diện tìm kiếm Web trong giai đoạn 1997 – 2003, Spink A. và Jansen B.J. [SJ04] cung cấp các kết quả nghiên cứu về ứng xử theo tương tác người – máy và ở mức độ câu hỏi được các máy tìm kiếm ghi nhận được. Qua khảo sát hoạt động tìm kiếm đa phương tiện của 20 máy tìm kiếm Trung Quốc diên hình (Hình 6.1), Chang Y.K. và Spink A. [CS07] trình bày kết quả khảo sát về hoạt động tìm kiếm ảnh, video và nội dung trộn đa phương tiện qua các máy tìm kiếm Trung Quốc. Các tác giả nhận định rằng, tại thời điểm khảo sát thì các máy tìm kiếm chung hỗ trợ nhiều lựa chọn cho người dùng hơn so với các máy tìm kiếm chuyên dụng.

| | |
|-----------------|---|
| 163 | www.163.com |
| Baidu | www.baidu.com |
| Chinasite | www.chinasite.com |
| google | www.google.cn |
| iask | www.iask.com |
| MSN | cn.msn.com |
| ODP | http://dmoz.org/World/Chinese_Simplified |
| PKU | http://ie.pku.edu.cn/ |
| Sina | www.sina.com |
| sogou | www.sogou.com |
| Sogou | www.sogou.com |
| Sohu | www.sohu.com |
| SoSo | www.soso.com |
| souyo | www.souyo.com |
| Sowang | www.sowang.com |
| Timway graphics | http://graphics.timway.com |
| Tom | http://hi.tom.com |
| Yanh | http://cn.yahoo.com |
| Zhongsou | http://www.zhongsou.com |
| 21CN | http://search.21cn.com/index.html |

Hình 6.1. 20 máy tìm kiếm Trung Quốc được khảo sát
về tìm kiếm đa phương tiện [CS07]

Một vài năm gần đây đã chứng kiến xu hướng phát triển các thế hệ máy tìm kiếm thông minh. Máy tìm kiếm thực thể, chẳng hạn Cazoodle⁽¹⁾, Arnetminer⁽²⁾ là một loại máy tìm kiếm thông minh, hoạt động của chúng có thể được coi như một hệ thống quản trị thực thể trên Web. Trích chọn thông tin trên Web (Chương 9) là một bài toán cơ bản trong các máy tìm kiếm thực thể. Một số nội dung về máy tìm kiếm thực thể sẽ được giới thiệu ở cuối chương này.

Dưới đây là một số nội dung sơ bộ về một số máy tìm kiếm điển hình; chúng xuất hiện ngay từ thời kỳ sơ khai của máy tìm kiếm, tồn tại và phát triển không ngừng cho tới ngày nay.

• Máy tìm kiếm AltaVista:

Theo [SHM98, SH99], AltaVista là một hệ thống bao gồm hai thành phần là máy tìm kiếm (<http://www.altavista.com>) và log câu hỏi (Query Log).

① Máy tìm kiếm của AltaVista lưu trữ văn bản theo mô hình boolean có trọng số. Hệ thống cho phép hai dạng câu hỏi là dạng đơn giản và dạng mở rộng.

- Dạng câu hỏi đơn giản:

Trong dạng đơn giản, câu hỏi thuộc vào một trong các dạng như sau:

(1) <http://www.cazoodle.com/>

(2) <http://www.arnetminer.org/>

- * từ khoá;
- * dãy từ khoá (hoặc phép toán OR);
- * -word: các tài liệu không chứa word (phép toán NOT);
- * +word: các tài liệu chứa cả word;
- * "dãy từ": các tài liệu chứa dãy từ theo đúng thứ tự chặt chẽ như câu hỏi; số hạng hỏi: hoặc từ khoá, hoặc dãy từ khoá trong cặp dấu " và ".

- Dạng câu hỏi mở rộng:

Trong dạng câu hỏi mở rộng, các phép toán logic rõ ràng (hiên) *and*, *or*, *not* thực sự là phép toán boolean (kết quả), không phải là phép toán theo từ khoá tìm kiếm. Bộ sung phép toán *near* các từ lân cận không bắt buộc thứ tự chặt chẽ như " " của mode đơn giản. Hơn nữa, cho phép người dùng đặt câu hỏi theo *vết*, dùng menu hạn chế trang kết quả theo ngôn ngữ riêng. Mode mở rộng còn cho phép hạn chế cập nhật theo ngày, hoặc một khoảng ngày.

Kết quả tìm kiếm là danh sách các bộ 10URL kết quả một (mỗi URL có thông tin như tiêu đề) theo thứ tự "độ liên quan" (hàm "liên quan" trong AltaVista) tới câu hỏi.

② Log câu hỏi tạo cơ chế để AltaVista định hướng người dùng, chẳng hạn sử dụng các phương pháp khai phá sử dụng Web (Chương 2). Log câu hỏi gồm nhiều thành phần, trong đó có file text là đây các yêu cầu. Yêu cầu hoặc là câu hỏi mới, hoặc là màn hình kết quả mới từ yêu cầu đã gửi. Mỗi yêu cầu gồm các trường:

- Thời gian yêu cầu gửi đi, tính theo đơn vị mili giây, từ 1/1/1970;
- Cookie: phải chăng hai câu hỏi xuất phát từ cùng một người dùng;
- Số lượng (như đã được gửi đi);
- Màn hình kết quả;
- Các biến thuộc về kiểu người dùng: ngày, khoảng ngày;
- Thông tin đệ trình: câu hỏi đơn giản/mở rộng;
- Thông tin về sự duyệt: trình duyệt, địa chỉ IP, ...

Ngoài ra còn có các khái niệm liên quan như phiên truy nhập, tập dữ liệu log được đặt ra trong nội dung Log câu hỏi của AltaVista.

• *Máy tìm kiếm Google*:

Hình 6.12 trình bày kiến trúc máy tìm kiếm Google ở mức cao. Theo Sergey Brin và Lawrence Page, hai đồng sáng lập hãng Google [PBM198, BP98], tên gọi của máy tìm kiếm Google có nguồn gốc chính là từ chữ "Googol" (có nghĩa như 10^{100}), mang nghĩa về một máy tìm kiếm rất lớn. Google cũng định hướng người dùng, vì vậy, yêu cầu cần có log câu hỏi và

định hướng thiết kế cần đạt mục tiêu đáp ứng nhu cầu (số lượng câu hỏi) hàng ngày tăng không ngừng. Để đáp ứng được mục tiêu như vậy, một số yêu cầu thiết kế được đặt ra:

- Công nghệ Crawling cần có tốc độ cao khi thu thập tài liệu Web và cập nhật chúng;
- Hệ thống lưu trữ hiệu quả cho phép lưu không chỉ chỉ số mà còn toàn văn nội dung tài liệu;
- Hệ thống đánh chỉ số hiệu quả được thi hành trên hàng trăm gigabyte dữ liệu;
- Câu hỏi cần được nắm bắt và đáp ứng nhanh theo cỡ hàng trăm nghìn câu hỏi trong một giây.

Khi thiết kế các phiên bản đầu tiên của Google, Sergey Brin và Lawrence Page quan tâm tới thách thức từ sự tăng trưởng quá nhanh của Web. Quan hệ theo thời gian giữa độ tăng trưởng Web với độ tăng cường năng lực hệ thống máy tính được phân tích. Các giải pháp tận dụng hiệu năng và giá thành phần cứng, bao gồm yêu cầu hệ thống cần trong suốt theo tốc độ di chuyển đầu đọc đĩa và độ mạnh hệ điều hành cũng đã được xem xét. Kết quả phân tích, thiết kế hệ thống cho thấy, hệ thống chỉ hoạt động hiệu quả khi mà các thành phần lưu chỉ số cần được thiết kế với các cấu trúc dữ liệu tối để tối ưu hoá được truy nhập. Google làm việc theo một hệ thống file riêng (được gọi là *BigFiles*), có nhiều thành phần tổ chức dữ liệu tối, bao gồm kho chứa trang Web, tài liệu trang Web được chỉ số hoá theo ISAM (Index Sequential Access Mode), có hệ thống từ điển, danh sách HIT, hệ thống chỉ số thuận và hệ thống chỉ số ngược.

Năm 1997, luận điểm "*chỉ với chỉ số tìm kiếm đầy đủ là có thể dễ dàng tìm được mọi thứ*" là không còn đúng và đánh chỉ số đầy đủ không đảm bảo được "chất lượng" kết quả tìm kiếm. Chang hạn, các thử nghiệm được tiến hành vào năm 1997 phát hiện rằng, "*chỉ có 1/4 máy tìm kiếm thương mại hàng đầu tìm được chính mình trong số mười máy tìm kiếm thương mại hàng đầu*". Tuy nhiên, chỉ số hoá tối lại giúp tăng tốc độ đáp ứng nhanh câu hỏi người dùng, vì vậy, các phương án chỉ số hoá nhận được sự quan tâm đặc biệt của các tác giả thiết kế Google.

Số lượng trang Web tăng theo tốc độ bùng nổ, khả năng nhìn của con người thường chỉ liên quan đến 10 trang đầu tiên, vì vậy, giải pháp hiển thị nhanh các trang kết quả "*có độ chính xác cao*" được thi hành với thuật toán PageRank tính hạng trang hiệu quả.

Vào năm 1997, máy tìm kiếm Google được thiết kế thành công, có năng lực lưu trữ 100 triệu trang Web và chạy trên các hệ điều hành phổ dụng như Solaris và Linux. Tại thời điểm đó, hệ thống chỉ số của Google được tập trung nhằm tạo thuận lợi cho tìm kiếm.

Qua hơn mươi năm phát triển vừa qua, Google không ngừng tăng trưởng cả về quy mô và chất lượng; hằng Google đã trở thành một người không lồ về lĩnh vực tìm kiếm.

- **Một số đặc trưng của máy tìm kiếm:**

Như đã được giới thiệu, Greg R. Notess tiến hành các nghiên cứu có tính cập nhật về các máy tìm kiếm, bao gồm các kết quả khảo sát về đặc trưng cơ bản của chúng (<http://www.searchengineshowdown.com/features/>). Hình 6.2 trình bày một số thông tin đối sánh theo đặc trưng của một số máy tìm kiếm điển hình tại thời điểm cuối năm 2007.

Các hệ thống được xem xét gồm có Google (<http://www.google.com/>), Yahoo! (<http://search.yahoo.com/>), Ask (<http://ask.com/>), Live Search (<http://live.com/>), Gigablast (<http://gigablast.com/>) và Exalead (<http://exalead.com/>). Đã có sự thay đổi về danh sách máy tìm kiếm được xem xét so với phiên bản tháng 10/2003, có các máy tìm kiếm được xem xét là Google, AllTheWeb, Lycos, AltaVista Simple, AltaVista Advanture, HotBot, MSN Search, Teoma, WiseNut và Gigablast.

Các đặc trưng được đưa ra đối sánh là Boolean, Default, Proximity, Truncation, Fields, Limits, Stop, Sorting; không xem xét hai đặc trưng Truncation và Case nữa.

Dưới đây là danh sách và ý nghĩa của các đặc trưng so sánh điển hình nhất có trong phiên bản 2007 của Greg R. Notess:

- Boolean: Việc cho phép ngầm định các phép toán logic (and, or, not, (), +, -) trong câu hỏi tìm kiếm và thực hiện. Chỉ số này gặp ở hầu hết các máy tìm kiếm điển hình.
- Default: Phép toán logic được thi hành ngầm định. Hầu hết các máy tìm kiếm ngầm định phép toán and.
- Proximity: Thực hiện tìm theo cụm từ ở hầu hết các máy tìm kiếm. Một số máy tìm kiếm còn cho phép tìm gần đúng kè cận.
- Truncation: Tiến hành tìm kiếm theo từ gốc; Google, Yahoo cho phép có ký hiệu đại diện trong câu hỏi.
- Fields: Cho phép đặt tham số tìm kiếm theo một số trường như tiêu đề, địa chỉ URL, liên kết, miền/site, kiểu file,...
- Limits: Cho phép đưa ra một số hạn chế về thời gian, về lĩnh vực, nội dung đa phương tiện, độ sâu của trang Web,...
- Stop (stop word): Cho phép loại bỏ từ dừng; một số trường hợp không tiến hành tìm kiếm từ quá thông dụng.
- Sorting: Sắp xếp kết quả tìm kiếm theo độ liên quan, phân cụm theo site, sắp theo thứ tự thời gian, kích thước.

Search Engine Features Chart

- * See also Search Engines by Search Features
- * Search engines grouped by size; all words link to more detailed reviews.

Last updated Oct. 01, 2007
by Greg R. Atwood

| SEARCH ENGINES | BOOLEAN | BOOLEAN PHRASE | FRACTIONAL | PROXIMITY | MINUS | BLANK | BLURB | ZOOMING |
|---------------------|---------------------------------|----------------|---------------------------------|-----------------------------------|--|---------------------------------------|--|-----------------|
| Google Review | -, OR and | Phrase | No (terms) word in phrase | title, url, intext, site, more | language, file type, date, domain | Few, + Relevance, site searches | | |
| Yahoo! Review | AND, OR, NOT, (), - | and | Phrase | No word in phrase | intitle, inturl, intext, site, more | language, file type, date, domain | No | Relevance, site |
| AOL Review | -, OR | and | Phrase | No | intitle, inturl, site | language, site, date | Yes, + Relevance, searches metasearch | |
| Lycos Search Review | AND, OR, NOT, (), - | and | Phrase | No | intitle, inturl, loc, | language, site | Varies, + Relevance, site, searches sidebar | |
| EggleSearch Review | AND, OR, AND NOT, and (), - | | Phrase | No | title, site, file, more | domain, type | Varies, + Relevance searches | |
| ExciteSearch Review | AND, OR, NOT, (), - | and | Phrase, NEAR | Yes and terms | intitle, inturl, intext, site | language, file type, date, domain | Varies, + Relevance, date searches | |

Hình 6.2. Một số đặc trưng của một số máy tìm kiếm
Nguồn: <http://www.searchengineshowdown.com/features/>

6.3. Cấu trúc và hoạt động của một máy tìm kiếm

Máy tìm kiếm được xem là hệ thống tìm kiếm thông tin diển hình [LCH02]. Hệ thống tìm kiếm thông tin thường tập trung vào việc cài thiện hiệu quả thông tin được lấy ra bằng cách đánh chỉ số dựa trên từ khoá và kỹ thuật cấu trúc lại câu truy vấn [KHM03]. Quá trình xử lý các văn bản dựa trên từ khoá thực hiện việc trích các từ khoá trong văn bản, sử dụng một từ diển được xây dựng trước, một tập các từ dừng và các quy tắc để chuyển các hình thái của từ về dạng từ gốc. Sau khi các từ được lấy ra, các hệ thống tìm kiếm thường sử dụng phương pháp TF-IDF (hoặc biến thể của nó) để biểu diễn trang Web. Mức độ tương tự do được giữa một câu truy vấn và một văn bản thường được tính theo độ đo nào đó, trong đó cosin của góc giữa hai vector biểu diễn là một độ đo phổ biến.

Kiến trúc mức cao của máy tìm kiếm Google (Hình 6.11) được chọn để minh họa cho các nội dung được trình bày tiếp theo đây. Mặc dù trong thực tiễn, mỗi máy tìm kiếm có cách thực thi riêng mà theo đó các thành phần được trình bày như dưới đây có thể được nhập lại hoặc tách ra. Tuy nhiên, những nội dung được trình bày dưới đây mang tính bản chất về hoạt động của các thành phần chức năng trong máy tìm kiếm.

- **Thành phần crawling (Crawler):** Đây là thành phần có chức năng thu thập tài nguyên trang Web cho máy tìm kiếm. Thành phần này thực hiện việc duyệt không gian Web, đi dọc theo các liên kết trên các trang Web để

thu thập nội dung các trang Web. Crawler nhận tập các địa chỉ URL xuất phát từ dòng xếp hàng các trang Web chưa được thăm (dưới đây gọi là *frontier* theo thuật ngữ tiếng Anh thông dụng của nó) thực hiện tải các trang Web tương ứng về. Trong nhiều trường hợp, thành phần crawling còn bao gồm bộ phân tích cú pháp (parser), bộ điều khiển crawler. Bộ phân tích cú pháp thi hành đối với trang Web, cung cấp các địa chỉ URL chưa được thăm vào dòng xếp hàng. Bộ điều khiển crawler quyết định xem URL nào được duyệt tiếp theo và gửi kết quả cho crawler. Nội dung các trang Web đã được tải về sẽ được store server lưu vào kho trang Web (page repository). Quá trình này được lặp lại cho tới khi đạt tới điều kiện kết thúc.

• **Thành phần đánh chỉ mục (indexer):** Đây là thành phần có nhiệm vụ tiếp nhận kết quả phân tích cú pháp trang Web đã được tải về và đánh chỉ mục cho nội dung trang Web.

Kết quả của việc đánh chỉ mục sinh ra một tập bảng chỉ mục rất lớn. Nhờ có tập bảng chỉ mục này, máy tìm kiếm nhanh chóng cung cấp được tất cả các địa chỉ URL của các trang Web đáp ứng truy vấn người dùng. Thông thường, bộ tạo chỉ mục tạo ra chỉ mục nội dung (content index) và chỉ mục cấu trúc (structure index). Chỉ mục nội dung chứa thông tin về các từ khóa xuất hiện trong các trang Web. Chỉ mục cấu trúc thể hiện mối liên kết giữa các trang Web, tận dụng được đặc tính quan trọng của dữ liệu Web là có các liên kết. Nó chính là một dạng đồ thị Web (Chương 3). Cách thức index ngược (invert index) theo từ khoá thường được sử dụng để làm tăng tốc độ tìm kiếm theo từ khoá.

• **Thành phần phân tích tập (Collection Analysis Module):** Hoạt động dựa vào đặc trưng của thành phần truy vấn. Chẳng hạn, nếu thành phần truy vấn chỉ đòi hỏi việc tìm kiếm hạn chế trong một số Web site đặc biệt, hoặc giới hạn trong một tên miền, thì công việc sẽ nhanh và hiệu quả hơn. Thành phần này sử dụng thông tin từ hai loại chỉ mục cơ bản (chỉ mục nội dung và chỉ mục cấu trúc) do thành phần đánh chỉ mục cung cấp cùng với thông tin các từ khoá trong trang Web và các thông tin tinh hạng để tạo ra các chỉ mục tiện ích.

• **Thành phần truy vấn (query engine):** Thành phần này chịu trách nhiệm nhận các yêu cầu tìm kiếm của người sử dụng. Nó thường xuyên truy vấn CSDL, đặc biệt là các bảng chỉ mục để trả về danh sách các tài liệu thỏa mãn yêu cầu của người dùng. Do số lượng các trang Web là rất lớn và thông thường người dùng chỉ đưa vào một vài từ khoá trong câu truy vấn nên tập kết quả thường rất lớn. Bộ xếp hạng (ranking) có nhiệm vụ sắp xếp các tài liệu này theo mức độ phù hợp với yêu cầu tìm kiếm để hiển thị kết quả cho người sử dụng. Khi muốn tìm kiếm các trang Web về một chủ đề nào đó, người sử dụng đưa vào một số từ khoá liên quan. Thành phần truy vấn dựa

theo các từ khoá này để tìm trong bảng chỉ mục nội dung các địa chỉ URL mà nội dung có chứa từ khoá này. Sau đó, thành phần truy vấn sẽ chuyên các trang Web cho bộ xếp hạng để sắp xếp các kết quả giảm dần về độ liên quan giữa trang Web với truy vấn, rồi hiển thị kết quả cho người sử dụng.

6.4. Crawling trang Web

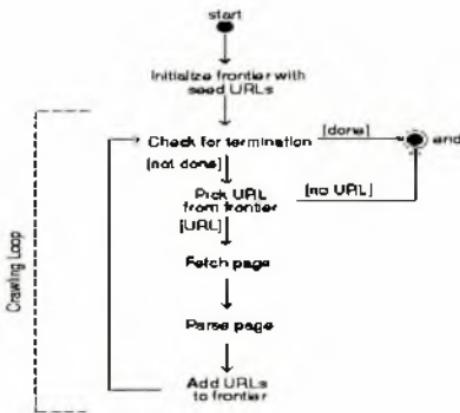
Kích thước quá lớn và bản chất thay đổi không ngừng của Web đặt ra một nhu cầu mang tính nguyên tắc là, cần phải cập nhật không ngừng tài nguyên cho các hệ thống trích chọn thông tin trên Web. Thành phần crawler đáp ứng được nhu cầu này bằng cách dí theo các siêu liên kết trên các trang Web để tải về một cách tự động nội dung các trang Web. Web crawler khai thác sơ đồ cấu trúc của Web để duyệt không gian

Web bằng cách chuyên từ trang Web này sang trang Web khác [BP98, PSM04, BCS04, JO02, HM99].

Hình 6.3 biểu diễn sơ đồ khái niệm của một crawler đơn luồng. Chương trình crawler yêu cầu một danh sách các URL chưa được thăm (frontier). Ban đầu frontier chứa các URL hạt nhân do người dùng hoặc các chương trình khác cung cấp. Mỗi vòng lặp crawling bao gồm: lấy ra URL tiếp theo cần được tải về từ frontier, nạp trang Web tương ứng với URL đó bằng giao thức HTTP, chuyên nội dung trang Web vừa được tải về cho phục vụ kho chứa trang Web. Quá trình crawling được kết thúc theo hai tình huống:

- Đạt được điều kiện dừng cho trước, chẳng hạn như số lượng các trang Web được tải về đã đáp ứng được yêu cầu đặt ra.

- Danh sách URL tại frontier rỗng, không còn trang Web yêu cầu crawler phải tải về. Lưu ý rằng, điều kiện frontier rỗng được tính với một độ trễ nào đó, bởi có một số trường hợp, bộ điều khiển crawling chưa chuyên kịp các danh sách URL sẽ tới thăm.



Hình 6.3. Sơ đồ cơ bản của một crawler đơn luồng
[PSM03]

Trong quá trình nói trên, danh sách các URL tiếp theo tại frontier được bổ sung bằng các thành phần khác của máy tìm kiếm, đặc biệt là module phân tích cú pháp (parser) và thành phần đánh chỉ mục (indexer).

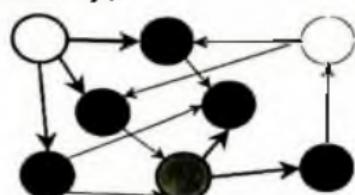
Hoạt động của thành phần crawler có thể được xem như một bài toán duyệt đồ thị. Toàn bộ thế giới Web được xem như một đồ thị lớn với các đỉnh là các trang Web và các cung là các siêu liên kết. Quá trình tải một trang Web và di tới một trang Web mới tương tự như quá trình mở rộng một đỉnh trong bài toán tìm kiếm trên đồ thị.

6.4.1. Dòng đợi URL

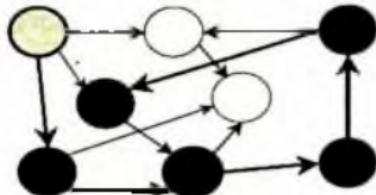
Frontier là dòng xếp hàng các URL cần tiếp tục được tải về, chứa URL của các trang Web cần được thăm. Trong thuật ngữ tìm kiếm đồ thị, frontier gọi là danh sách mở các nút chưa được thăm. Mặc dù có thể frontier phải lưu trên đĩa đối với các crawler lớn, ở đây đơn giản hoá quan niệm rằng, frontier như là một cấu trúc dữ liệu hiện ở bộ nhớ trong. Dựa trên dung lượng bộ nhớ cho phép, có thể quyết định kích thước cực đại của frontier. Dựa vào dung lượng lớn của bộ nhớ máy tính ngày nay, kích thước một frontier vào khoảng 100.000 URL không phải là hiếm. Do các frontier chỉ có kích thước giới hạn, cho nên cần có một cơ chế để quyết định URL nào cần bị bỏ qua khi số lượng URL trên frontier đạt tới giới hạn đó. Cần lưu ý rằng, frontier có thể bị đầy nhanh hơn nhiều so với số lượng trang Web được duyệt. Hệ thống có thể đạt tới 60.000 URL trong frontier khi mới duyệt được khoảng 10.000 trang Web do trung bình có khoảng 7 liên kết trong một trang Web [PSM04].

Frontier được thực thi có thể như một hàng đợi FIFO nếu muốn xây dựng một crawler theo duyệt chiều rộng (breadth – first) để duyệt Web theo chiến lược mù (blindly) (Hình 6.4). URL cần được duyệt tiếp theo được lấy từ đỉnh của hàng đợi và các URL mới được thêm vào cuối hàng đợi. Do kích thước hạn chế của frontier, cần phải đảm bảo là không thêm các URL lặp lại vào hàng đợi. Do vậy, một cơ chế tìm kiếm tuyển tính để tìm ra một URL mới được trích ra từ nội dung trang Web có trên frontier hay chưa là rất cần thiết. Một giải pháp được đưa ra là định vị một lượng bộ nhớ cần thiết để duy trì một bảng băm riêng (với khoá là URL) để lưu giữ các URL tại frontier để thuận lợi cho việc tìm kiếm. Bảng băm này phải được giữ nhất quán với frontier thực sự. Một giải pháp khác tồn tại lâu hơn là duy trì bản thân hàng đợi đó như một bảng băm (cũng với khoá là URL). Điều này cung cấp một cách tìm kiếm nhanh chóng để tránh việc lưu, lặp lại các URL. Tuy nhiên, mỗi lần crawler cần một URL để duyệt, nó cần phải tìm kiếm và lấy ra URL mới được đưa vào frontier gần đây nhất. Nếu bộ nhớ không phải là bài toán khắt khe như với tốc độ, thì giải pháp đầu tiên là tốt hơn. Một khi frontier đạt tới kích thước tối đa, thì crawler theo chiều

rộng chỉ có thể thêm duy nhất một URL chưa được thăm từ mỗi trang Web đã được duyệt.



a) Theo chiều rộng



b) Theo chiều sâu

Hình 6.4. Hai chiến lược crawling duyệt Web
(<http://oak.cs.ucla.edu/~choftalks/2001/Defense.ppt>)

Nếu frontier được thi hành như một hàng đợi có ưu tiên, thì có nghĩa là chúng ta đã xây dựng một crawler ưu tiên hay còn gọi là crawler cái tốt nhất (best – first crawler). Hàng đợi ưu tiên có thể là một mảng động, luôn được sắp xếp theo độ đo đánh giá các URL chưa được thăm. Tại mỗi bước, URL tốt nhất được lấy ra ở đầu hàng đợi. Một khi trang Web tương ứng được nạp, các URL được trích chọn ra và được đánh giá theo một độ đo. Sau đó, các URL này được thêm vào frontier tại các vị trí phụ thuộc vào độ đo đó. Khi frontier đạt tới kích thước tối đa là MAX, thì chỉ có MAX URL tốt nhất được giữ lại trong frontier.

Nếu chương trình crawler thấy frontier rỗng, trong khi nó cần URL tiếp theo để duyệt, quá trình crawling sẽ ngừng lại và sau một lượng từ thời gian thì crawler ngừng hẳn. Với một giá trị MAX lớn và một vài URL hạt nhân thì thông thường frontier rất hiếm khi đạt tới trạng thái rỗng.

Tại một số thời điểm, crawler có thể gặp bẫy nhện (spider trap) dẫn nó tới một số lượng lớn các URL khác nhau cùng trỏ tới một trang Web. Có thể hạn chế điều này bằng cách hạn chế số lượng các trang Web mà crawler truy cập tới từ một miền xác định. Đoạn mã lệnh liên quan tới frontier có thể đảm bảo rằng, mọi chuỗi k URL liên tiếp (thường k = 100) được lấy ra bởi crawler chỉ chứa duy nhất một địa chỉ URL chuẩn hoá. Điều này tránh được việc crawler phải truy cập cùng một Web site quá nhiều lần đồng thời và nội dung các trang Web tải được sẽ có xu hướng khác biệt nhiều hơn.

6.4.2. Danh sách URL đã thăm

Danh sách URL đã thăm (dưới đây dùng thuật ngữ *history* như tiếng Anh) của crawler là một danh sách động các URL mà nội dung của trang Web đã được crawler tải về. Nó chứa các URL mà crawler đã đi qua bắt đầu từ các trang hạt nhân. History được sử dụng để phân tích và đánh giá các trang Web. Chẳng hạn, có thể gắn cho mỗi trang Web một giá trị trên đường dẫn và xác định các sự kiện có ý nghĩa (ví dụ, khám phá ra một nguồn tài

nguyên quan trọng mới). Trong một số trường hợp, danh sách URL thuộc history được lưu trữ ở bộ nhớ ngoài, nhưng nó cũng được duy trì như một cấu trúc dữ liệu tại bộ nhớ trong. Điều này cho phép tìm kiếm nhanh để kiểm tra xem liệu một trang Web đã được duyệt hay chưa. Việc kiểm tra như vậy là rất quan trọng để tránh đi thăm lặp các trang Web, và do đó tránh việc thêm các URL đã được duyệt vào trong frontier. Vì vậy, việc chuẩn hóa các URL trước khi thêm chúng vào history là rất quan trọng.

6.4.3. Tài trang Web

Để nạp một trang Web, hệ thống cần một HTTP khách gửi một yêu cầu HTTP về trang Web nói trên và đọc các trả lời từ phía phục vụ. Phía khách cần đưa ra thời gian quá hạn (timeout) để đảm bảo rằng, nó không bị lãng phí quá nhiều thời gian để giao tiếp với một phục vụ chủ quá chậm, hoặc đọc một trang Web quá lớn. Trên thực tế, người ta thường giới hạn chương trình khách chỉ tài về các trang Web có kích thước nhỏ hơn 20KB (đại đa số là nhỏ hơn 10KB). Phía khách cần duyệt các trả lời phần đầu (header) để lấy các mã trạng thái và các sự định hướng lại.

Việc kiểm tra lỗi và ngoại lệ là rất quan trọng trong máy tìm kiếm khi tài trang Web do phải liên hệ tới hàng triệu phục vụ ở xa bằng cùng một đoạn mã lệnh.Thêm vào đó, việc thu thập các thông kê về quá hạn và các mã trạng thái cũng rất hữu ích trong việc xác định các vấn đề này sinh hoặc để thay đổi tự động giá trị quá hạn. Các ngôn ngữ lập trình hiện đại như Java hoặc Perl cung cấp các cơ chế đơn giản cùng nhiều giao diện lập trình để tài các trang Web. Tuy nhiên, cần thận trọng khi sử dụng các giao diện bậc cao do khó tìm ra được các lỗi ở bậc thấp mà nhiều khả năng có thể xảy ra.

Khi thi hành crawling cần quan tâm tới giao thức loại trừ robot (Robot Exclusion Protocol). Giao thức này cho phép quản trị phục vụ Web thi hành cơ chế thông báo quyền truy nhập file trên phục vụ. Việc chỉ định các file mà crawler không được phép truy cập được đặt tại file robots.txt ở thư mục chủ của phục vụ Web (<http://www.biz.uiowa.edu/robots.txt>). Chi tiết về giao thức loại trừ này có tại <http://www.robotstxt.org/wc/norobots.html>. Dưới đây là một số ví dụ về danh sách các file bị cấm crawling.

| Ví dụ | Ý nghĩa |
|---|--|
| User-agent: * Disallow: /cgi-bin/ Disallow: /tmp/ | Tất cả các máy tìm kiếm có thể thăm tất cả các thư mục ngoại trừ hai thư mục đề cập ở đây |
| User-agent: BadBot Disallow: / | Máy tìm kiếm BadBot không được phép thăm bất cứ thư mục nào. |
| User-agent: BadBot Disallow: / User-agent: * Disallow: /private/ | Riêng máy tìm kiếm BadBot không được phép thăm bất cứ thư mục nào còn tất cả các máy tìm kiếm còn lại đều có quyền thăm tất cả các thư mục ngoại trừ thư mục "private" |

Xây dựng cây thẻ HTML:

Một số phương pháp biểu diễn trang Web đòi hỏi gán trọng số cho URL/từ khoá có trong nội dung trang Web phụ thuộc vào ngữ cảnh (vị trí) URL/từ khoá xuất hiện. Ngữ cảnh này thường được chọn từ ngữ cảnh của thẻ HTML chứa URL/từ khoá đó. Để làm được điều này, crawler cần sử dụng cây các thẻ hoặc cấu trúc DOM của trang HTML [BP98, PSM04, BCS04, JO02, HM99]. Một cấu trúc cây các thẻ tương ứng với văn bản HTML nguồn sẽ được hình thành, trong đó thẻ <html> được lấy làm gốc của cây và văn bản tạo thành nội dung các nút của cây. Tuy nhiên, nhiều trang Web lại có cấu trúc HTML không chuẩn, chẳng hạn như có "thẻ mờ" nhưng lại không có "thẻ đóng", hoặc các thẻ lồng nhau sai quy cách, hoặc (trong không ít trường hợp) thẻ <html> hoặc <body> có thể bị thiếu. Do đó, cần thực hiện bước tiền xử lý để chuẩn hoá cấu trúc trang HTML. Tiền xử lý trang HTML bao gồm cả việc chèn thêm các thẻ bị thiếu lẫn sắp xếp lại thứ tự các thẻ cho chính quy. Đây là việc làm rất cần thiết nhằm ánh xạ toàn vẹn nội dung của một trang vào cấu trúc cây tương ứng.

Chú ý rằng, việc phân tích cấu trúc DOM chỉ cần thiết nếu crawler theo chủ đề có ý định sử dụng cấu trúc trang HTML cho những phân tích phức tạp, chẳng hạn như, có định hướng khai phá cấu trúc Web. Nếu crawler chỉ cần đưa ra các liên kết, các từ khoá và vị trí xuất hiện trong trang Web, thì chỉ cần sử dụng các bộ duyệt HTML thông thường và các bộ duyệt này có sẵn trong nhiều ngôn ngữ lập trình.

6.4.4. Một số thuật toán crawling

Phần này trình bày sơ bộ một số thuật toán crawling điển hình [PSM03, MPR01, PR05]. Tồn tại nhiều thuật toán crawling, không ít trong số đó là biến thể của thuật toán chọn cái tốt nhất (best – first hoặc Naive Best – First). Sự khác biệt giữa các biến thể này chính là cách tính điểm cho các URL chưa được thăm. Một số thuật toán còn cho phép chỉnh lại giá trị các tham số trước hoặc trong quá trình crawling.

• Thuật toán Naive Best – First

Theo Menczer và cộng sự [MPR01], thuật toán crawling Naive Best – First do họ đề xuất (giả mã thuật toán được trình bày tại Hình 6.5) thi hành crawling theo chiều sâu không gian Web. Đầu vào của thuật toán là một chủ đề và một danh sách các URL (danh sách các URL nhận). Việc đầu vào có chứa tham số "topic" cho thấy thuật toán Naive Best – First thuộc loại hướng chủ đề.

Theo [PSM03], thuật toán Best – First sử dụng cách biểu diễn vector trọng số từ khoá đối với trang Web được tải về. Chương trình crawler sau đó

tính toán mức độ tương tự của trang Web với chủ đề đầu vào và tính điểm cho các URL chưa được thăm trong trang Web đó theo độ đo tương tự. Sau đó các URL được bổ sung vào frontier sẽ được quản lý dưới dạng một hàng đợi ưu tiên dựa trên các điểm được tính nói trên. Trong vòng lặp tiếp theo, mỗi luồng của crawler lấy ra URL tốt nhất trong frontier để crawling, trả về các URL chưa được thăm mới và chúng lại tiếp tục được thêm vào hàng đợi ưu tiên như đã diễn giải trên. Mức độ tương tự (được gọi là điểm – score) giữa trang p và câu truy vấn q được tính bởi:

$$\text{sim}(p, q) = \frac{(v_p, v_q)}{\|v_p\| \|v_q\|} = \frac{\sum_i v_{p_i} v_{q_i}}{\sqrt{\sum v_{p_i}^2} \sqrt{\sum v_{q_i}^2}} \quad (6.3)$$

Trong đó, v_p và v_q tương ứng là các vector biểu diễn TF của câu truy vấn và trang Web, (v_p, v_q) là tích vô hướng của hai vector và $\|v\|$ là chuẩn Euclit của vector v. Các biểu diễn vector phức tạp hơn của trang Web như mô hình TF-IDF thường được sử dụng trong lĩnh vực tìm kiếm thông tin lại rất khó áp dụng trong các crawler do nó đòi hỏi hiểu biết trước về việc phân phối các từ khoá trong toàn bộ trang Web được tải về. Trong thi hành đa luồng, các crawler thường hoạt động theo hướng tìm kiếm N trang tốt nhất đầu tiên, trong đó N là số lượng luồng crawler chạy đồng thời. Crawler đa luồng N tốt nhất đầu tiên là phiên bản tổng quát hoá của crawler tốt nhất đầu tiên, nó lấy ra N URL tốt nhất để duyệt trong một thời điểm. Chú ý rằng, crawler tốt nhất đầu tiên giữ cho kích thước của frontier trong giới hạn bằng cách chỉ giữ lại các URL tốt nhất dựa vào điểm số mà chúng được gán.

```

BestFirst(topic, starting_urls) {
    foreach link (starting_urls) {
        enqueue(frontier, link);
    }
    while (#frontier > 0 and visited < MAX_PAGES) {
        link := dequeue_link_with_max_score(frontier);
        doc := fetch_new_document(link);
        score := sim(topic, doc);
        foreach outlink (extract_links(doc)) {
            if (#frontier >= MAX_BUFFER) {
                dequeue_link_with_min_score(frontier);
            }
            enqueue(frontier, outlink, score);
        }
    }
}

```

Hình 6.5. Giả mã của thuật toán crawling Naive Best-First [MPR01]

• Thuật toán PageRank

Thuật toán crawler PageRank được căn cứ vào thuật toán tính hạng trang Web PageRank để sắp xếp các trang Web chưa được thăm vào frontier theo hạng của chúng. Giả mã của thuật toán được trình bày tại Hình 6.6. Hạng trang được tính toán dựa trên nhận định tính quan trọng của một trang Web phụ thuộc vào số trang Web chỉ dẫn tới nó, càng nhiều chỉ dẫn thì càng quan trọng. Mô tả chi tiết về thuật toán tính hạng trang PageRank có trong mục 6.6.

```
PageRank(topic, starting_urls) {
    foreach link (starting_urls) {
        enqueue(frontier, link);
    }
    while (#frontier > 0 and visited < MAX_PAGES) {
        if (multiplies_25(visited)) {
            foreach link (frontier) {
                PR(link) := recompute_PR;
            }
        }
        link := dequeue_link_with_max_PR(frontier);
        doc := fetch_new_document(link);
        score := sim(topic, doc);
        if (#buffered_pages >= MAX_BUFFER) {
            dequeue_page_with_min_score(buffered_pages);
        }
        enqueue(buffered_pages, doc);
        foreach outlink (extract_new_links(doc)) {
            if (#frontier >= MAX_BUFFER) {
                dequeue_link_with_min_PR(frontier);
            }
            enqueue(frontier, outlink);
        }
    }
}
```

Hình 6.6. Giả mã của thuật toán PageRank [MPR01]

• Thuật toán SharkSearch

Theo [BHK94, SW90], Paul De Bra và cộng sự đã đề xuất thuật toán crawling FishSearch. Tư tưởng của thuật toán rất tự nhiên với hai nội dung như sau:

- Có được điểm xuất phát tốt để khởi đầu tìm kiếm;
 - Có được một thứ tự "tối ưu" để duyệt các nút trên đồ thị Web.
- Thuật toán được đề xuất dựa trên 4 giả thiết về cấu trúc siêu liên kết, 3 giả thiết về hệ thống phân tán và mạng. Giả mã của FishSearch có trong [HHM98].

SharkSearch [HHM98] là một phiên bản cải tiến của FishSearch, tập trung vào bước 3.1 (Hình 6.7). Nó sử dụng một độ đo mức độ tương tự như trong crawler FishSearch để tính điểm của URL chưa được viếng thăm.

SharkSearch có cơ chế tính điểm tinh tế hơn FishSearch cho các liên kết trong frontier. Các từ thẻ hiện liên kết (anchor text), các từ xung quanh liên kết hoặc ngữ cảnh của liên kết và điểm được thừa kế từ URL cha đều ảnh hưởng tới việc tính điểm của liên kết. Các URL phía trước của một URL là các trang Web mà xuất hiện trong đường dẫn crawling để tới URL đó. SharkSearch giống như phiên bản xuất phát FishSearch và lưu giữ một giới hạn độ sâu. Nghĩa là, nếu chương trình crawler tìm thấy các trang Web không quan trọng trên đường dẫn khi crawling, nó dừng việc duyệt xa hơn trên đường dẫn đó. Để có thể theo dõi tất cả các thông tin, mỗi URL trong frontier được liên kết với một độ sâu và một điểm số. Giới hạn độ sâu (d) được cung cấp bởi người dùng, trong khi điểm số của một URL chưa được viếng thăm được tính bởi công thức:

$$\text{score(url)} = \gamma \cdot \text{inherited(url)} + (1 - \gamma) \cdot \text{neighborhood(url)}$$

Trong đó $\gamma < 1$ là một tham số, điểm số lân cận neighborhood score biểu thị các dấu hiệu ngữ cảnh tìm thấy trong trang Web chưa liên kết tới URL đó, và điểm số được thừa nhận được từ điểm số của các URL cha của URL hiện tại. Một cách chính xác hơn, điểm số được thừa kế được tính bởi:

$$\text{inherited(url)} = \begin{cases} \delta \cdot \text{sim}(q, p) & \text{sim}(q, p) > 0; \\ \delta \cdot \text{inherited}(p) & \text{otherwise}. \end{cases}$$

Trong đó, $\delta < 1$ là một tham số khác, q là câu truy vấn và p là trang Web mà từ đó URL được trích ra.

1. Compute the inherited score of child_node , $\text{inherited_score}(\text{child_node})$, as follows
 - o If $\text{relevance}(\text{current_node}) > 0$ (the current node is relevant)
 - Then $\text{inherited_score}(\text{child_node}) = \delta * \text{sim}(q, \text{current_node})$
 - where δ is a predefined decay factor
 - Else $\text{inherited_score}(\text{child_node}) = \delta * \text{inherited_score}(\text{current_node})$
2. Let anchor_text be the textual contents of the anchor pointing to child_node . and $\text{anchor_text_context}$ the textual context of the anchor (up to given predefined boundaries)
3. Compute the relevance score of the anchor text as $\text{anchor_score} = \text{sim}(q, \text{anchor_text})$
4. Compute the relevance score of the anchor textual context as follows:
 - o If $\text{anchor_score} > 0$.
 - Then $\text{anchor_context_score} = 1$
 - Else $\text{anchor_context_score} = \text{sim}(q, \text{anchor_text_context})$
5. Compute the score of the anchor, that we denote $\text{neighborhood_score}$ as follows

$$\text{neighborhood_score} = \beta * \text{anchor_score} + (1 - \beta) * \text{anchor_context_score}$$
 where β is a predefined constant
6. Compute the potential score of the child as

$$\text{potential_score}(\text{child_node}) = \gamma * \text{inherited_score}(\text{child_node}) + (1 - \gamma) * \text{neighborhood_score}(\text{child_node})$$
 where γ is a predefined constant

Hình 6.7. Đoạn mã của SharkSearch thay thế bước 3.1
trong thuật toán FishSearch [HJM97]

Một điểm số lân cận sử dụng các từ biểu diễn liên kết và các từ lân cận của liên kết nhằm cải tiến tổng điểm của một URL bằng cách lưu ý đến sự

khác nhau giữa các liên kết được tìm thấy trong cùng một trang. Để phục vụ mục đích này, các crawler áp dụng SharkSearch gán một điểm liên kết (anchor score) và một điểm ngữ cảnh (context score) cho mỗi URL. Trong khi điểm liên kết chỉ đơn giản là mức độ tương tự giữa các từ khoá dùng để biểu diễn liên kết tới URL với câu truy vấn q , ví dụ như, $\text{sim}(q, \text{anchor_text})$, thì điểm số ngữ cảnh mở rộng ngữ cảnh của liên kết tới cả các từ khoá gần đó. Kết quả là một ngữ cảnh mở rộng aug_context được sử dụng để tính giá trị của điểm ngữ cảnh như sau:

$$\text{context(url)} = \begin{cases} 1 & \text{anchor(url)} > 0; \\ \text{sim}(q, \text{aug_context}) & \text{otherwise.} \end{cases}$$

Cuối cùng, nhận được điểm lân cận từ điểm liên kết và điểm ngữ cảnh theo công thức:

$$\text{neighborhood(url)} = \beta \cdot \text{anchor(url)} + (1 - \beta) \cdot \text{context(url)}$$

trong đó, $\beta < 1$ là một tham số khác. Cần chú ý rằng, để thực thi được thuật toán SharkSearch cần phải đặt trước 4 tham số khác nhau: d , γ , δ và β .

• Crawler theo chủ đề

Charkrabarti [CBD99, CPS02, Cha03] đã phát triển một crawler theo chủ đề (*topic crawler* hay *focused crawler*) dựa trên một bộ phân lớp siêu liên kết. Ý tưởng cơ bản của crawler này là phân lớp các trang Web được tải về vào các lớp theo cấu trúc lớp chủ đề có sẵn. Để bắt đầu, crawler cần có một cấu trúc cây các chủ đề để phân lớp như trong Yahoo hoặc ODP (Open Directory Project). Thêm vào đó, người dùng cần cung cấp các URL mẫu để phân lớp. Các URL mẫu được phân lớp một cách tự động vào các lớp khác nhau trong cây chủ đề. Thông qua một quá trình tương tác, người dùng có thể sửa chữa việc phân lớp tự động đó, thêm chủ đề mới và đánh dấu một số lớp/chủ đề là tốt (dựa theo mức độ quan tâm của người dùng). Bộ crawler sử dụng các URL mẫu để xây dựng một bộ phân lớp Bayesian để tìm ra xác suất ($\text{Pr}(c|p)$) mà một trang Web đã được duyệt p thuộc vào lớp c trong cây chủ đề. Chú ý rằng, theo định nghĩa thì $\text{Pr}(r|p) = 1$ với r là lớp gốc của cây chủ đề. Một độ đo tính hợp lệ được gán cho mỗi trang Web được tải về theo công thức:

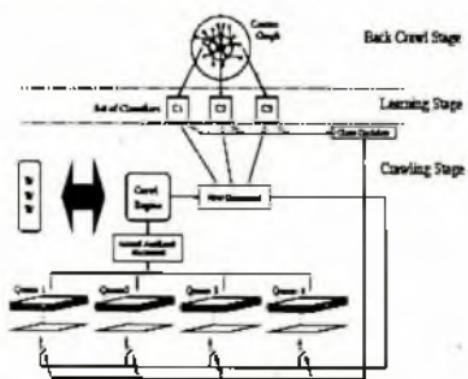
$$R(p) = \sum_{c \in \text{good}} \text{Pr}(c|p) \quad (6.4)$$

Crawler theo chủ đề được hoạt động theo cơ chế mềm và cứng. Crawler theo chủ đề mềm (soft focused) sử dụng độ đo tính hợp lệ của trang Web được tải để tính điểm cho các URL chưa được thăm trích ra từ trang đó. Các URL đã được tính điểm, sau đó sẽ được thêm vào frontier. Tiếp theo, chương trình crawler lây ra các URL tốt nhất tiếp theo theo cách tương tự như trong thuật toán Naïve tốt nhất đầu tiên. Trong crawler theo chủ đề cứng, đối với mỗi trang Web đã được tải p , đầu tiên bộ phân lớp sẽ tìm các

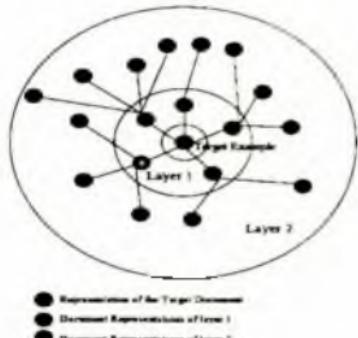
nút lá c^* trong cấu trúc lớp các chủ đề có xác suất trang p thuộc vào lớp đó là lớn nhất. Nếu bất kỳ chủ đề cha nào của c^* được người dùng đánh dấu là tốt, thì các URL được liên kết tới trong trang p sẽ được trích ra và thêm vào frontier.

• Crawler theo ngữ cảnh

Crawler theo ngữ cảnh (context focused crawler) [DCL00] sử dụng bộ phân lớp Bayesian để hướng dẫn quá trình crawling (Hình 6.8). Tuy nhiên, không giống crawler theo chủ đề, bộ phân lớp này được huấn luyện để đánh giá khoảng cách liên kết giữa trang được tải với trang Web liên quan chủ đề. Cách làm này tương tự như việc con người sử dụng kinh nghiệm khi duyệt Web. Chẳng hạn, để tìm ra các trang Web về "phân tích số học", đầu tiên con người có thể tới các trang chủ của khoa Toán (hoặc khoa Khoa học máy tính) và sau đó chuyên tới các phân trang nhỏ hơn để dẫn tới các trang liên quan về "phân tích số học". Một Web site của khoa Toán thường không chứa cụm từ "phân tích số học" tại trang chủ. Trong trường hợp này, một crawler theo thuật toán Naive Best – First có thể gán một độ ưu tiên thấp cho trang Web "phân tích số học" thuộc khoa Toán và có thể không bao giờ thăm được trang Web đó. Nếu crawler có thể ước lượng được khoảng cách giữa một trang liên quan tới chủ đề "phân tích số học" với trang đang được duyệt, thì đưa đến cách thức đặt trang chủ của khoa Toán có độ ưu tiên cao hơn trang chủ của một trường Luật.



Hình 6.8. Sơ đồ minh họa Crawler theo ngữ cảnh [DCL00]



Hình 6.9. Một đồ thị ngữ cảnh về cách thức trang Web đích được truy nhập từ Web [DCL00]

Crawler theo ngữ cảnh được huấn luyện nhờ đồ thị ngữ cảnh (context graph) có L tầng tương ứng với mỗi trang Web hạt nhân (Hình 6.9). Trang hạt nhân là tầng thứ 0 của đồ thị. Tập trang Web, chứa liên kết tới trang hạt nhân tạo thành tầng 1, tập trang chứa liên kết tới các trang thuộc tầng 1 tạo thành tầng thứ 2 và cứ như thế. Có thể đi theo các liên kết vào trang nhân

link để tới các trang thuộc tầng bất kỳ bằng cách sử dụng một bộ tìm kiếm. Khi sơ đồ ngữ cảnh của tất cả các trang hạt nhân đã được thiết lập, các tầng cùng mức của sơ đồ ngữ cảnh thành phần được kết hợp lại thành một tầng chung với mức tương ứng. Như vậy, sơ đồ ngữ cảnh tổng hợp (merged context graph) đã được hoàn thành.

Sau khi hoàn thành sơ đồ ngữ cảnh tổng thể, tiến hành bước lựa chọn đặc trưng, theo đó mọi trang Web nhân (có thể bao gồm cả các trang Web ở tầng 1) được tích hợp lại thành một văn bản chung. Dùng cách thức biểu diễn TFIDF đối với văn bản chung, một số từ có trọng số cao nhất sẽ được dùng để xây dựng nên bộ từ điển (không gian các đặc trưng) cho phân lớp.

Một tập các bộ phân lớp Naïve Bayes được xây dựng, mỗi tầng trong sơ đồ ngữ cảnh tổng hợp có bộ phân lớp riêng (Hình 6.8). Tất cả các trang trong một tầng được sử dụng để tính giá trị $Pr(t|c_i)$, là xác suất xuất hiện từ khoá t trong lớp c_i tương ứng với tầng thứ i . Xác suất tiên nghiệm $Pr(c_i) = 1/L$ được gán cho mỗi lớp, trong đó L là số lượng các tầng. Xác suất của một trang Web p thuộc vào một lớp c_i được tính bởi $Pr(c_i|p)$. Xác suất này được tính cho tất cả các lớp. Lớp có xác suất lớn nhất được coi là lớp trội. Nếu xác suất của lớp trội không nhỏ hơn giá trị ngưỡng, thì trang Web đó được phân lớp vào lớp trội, ngược lại, nó được phân vào lớp "other". Lớp "other" chứa các trang Web không phù hợp với bất kỳ lớp nào trong sơ đồ ngữ cảnh.

Tập các bộ phân lớp tương ứng với sơ đồ ngữ cảnh cung cấp một cơ chế đánh giá khoảng cách liên quan giữa trang Web hiện thời và một trang Web liên quan. Khi sử dụng cơ chế này, trang chủ của khoa Toán có thể được phân lớp vào tầng thứ 2 trong khi trang chủ của Luật lại được phân vào lớp "other". Chương trình crawler cần lưu một hàng đợi cho mỗi lớp, hàng đợi này chứa các trang Web đã được duyệt và phân vào lớp đó. Danh sách các URL trong mỗi hàng đợi được sắp xếp theo giá trị ($Pr(c_i|p)$). Khi chương trình crawler cần một URL để tải, nó lấy ra trang Web ở đỉnh của một hàng đợi không rỗng có giá trị I là nhỏ nhất, do đó, nó có khuynh hướng trước tiên lấy các trang có khoảng cách gần với các trang liên quan nhất. Các URL là liên kết ra từ các trang này sẽ được crawling trước.

6.4.5. Đánh giá crawler

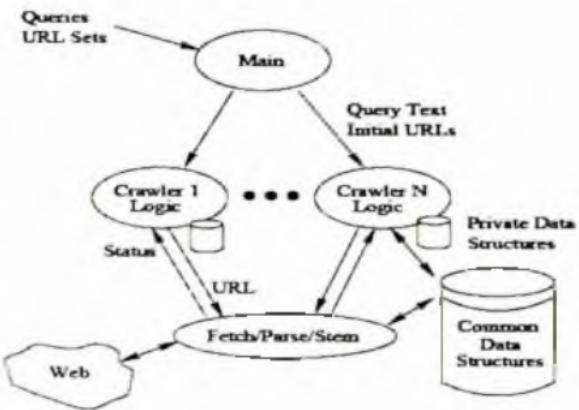
Thứ nhất, tiêu chí đầu tiên để đánh giá crawler là chất lượng lấy trang Web về. Như vậy, một crawler (đặc biệt là crawler hướng chủ đề) thường được đánh giá dựa theo khả năng lấy được các trang Web "tốt". Tuy nhiên, vẫn đề mấu chốt là làm thế nào để nhận ra một trang Web là "tốt"? Trong môi trường tương tác, một người dùng thực có thể xác định được sự phù hợp của các trang được tải về và thông qua đó, cho phép xác định được crawling có thành công hay không? Đáng tiếc, việc thi hành các thực

nghiệm hiệu quả với sự tham gia của nhiều người dùng thực tế vào việc đánh giá chất lượng của một crawler là cực kỳ khó khăn. Do kích thước không lồ của Web, thì để có thể nhận được các đánh giá phù hợp về mức độ hiệu quả của crawling cần phải xem xét rất nhiều lần việc thực hiện crawling, đòi hỏi số lượng rất lớn người dùng tham gia.

Thứ hai, việc crawling phải đáp ứng các ràng buộc nghiêm ngặt về thời gian. Do đó, nếu quá trình crawling không được thực hiện trong thời gian ngắn là không phù hợp với người dùng. Nếu giảm thời gian tải thì lại làm hạn chế quy mô của quá trình crawling và việc đánh giá lại không chuẩn xác.

Hiện tồn tại các crawler với sự trợ giúp của các tác tử Web (Web agent) đại diện cho người dùng hoặc các Web agent khác. Do đó, việc khảo sát các crawler là khá hợp lý trong một ngữ cảnh khi mà các tham số về thời gian crawling và khoảng cách crawling có thể khắc phục hạn chế bị áp đặt khi thử nghiệm với người dùng thực.

Nhiều nghiên cứu về đánh giá các crawler đã được tiến hành, chẳng hạn như [MPR01, MRS03]. Về mặt nguyên lý, để đánh giá so sánh các crawler thì cần một *độ đo* về *độ quan trọng* của các trang Web và một *phương pháp* để *tổng hợp* các *hiệu năng* crawler thông qua một tập các trang Web được crawling. Trong [MPR01], Filippo Menczer và cộng sự trình bày một phương pháp đánh giá các crawler hướng chủ đề và kiến trúc hệ thống được trình bày trên Hình 6.10.



Hình 6.10. Kiến trúc đánh giá so sánh các crawler [MPR01]

6.4.6. Một số nội dung cơ bản của bài toán crawling

- *Quy mô hệ thống crawler*

Nội dung cơ bản đầu tiên là cách thức hoạt động của crawler. Ban đầu, động cơ chủ yếu thúc đẩy việc thiết kế các Web crawler là việc lấy ra nội

dung các trang Web và thêm chúng (hoặc biểu diễn của chúng) vào các kho chứa cục bộ. Các kho chứa này sau đó sẽ đáp ứng phục vụ các ứng dụng cụ thể, chẳng hạn một hệ thống tìm kiếm trên Web. Ở dạng đơn giản nhất (Hình 6.4), một chương trình crawler bắt đầu từ một địa chỉ nguồn khởi đầu nào đó và sử dụng các liên kết ngoài trong trang Web đó để mở rộng ra các trang tiếp theo. Quá trình này tiếp tục với các trang Web mới, các trang này lại cung cấp các liên kết ngoài khác để đi theo. Cứ như vậy cho tới khi đạt tới một số lượng trang Web xác định, hoặc một mục tiêu nào đó đạt được. Phía sau sự mô tả một cách đơn giản này là một mảng các vấn đề phức tạp có liên quan như việc kết nối mạng, các tiêu chuẩn về một URL, việc duyệt các trang HTML và cách thức để giao tiếp với các Server ở xa. Trên thực tế, các thê hệ Web crawler gần đây có thể được coi là một trong những phần phức tạp nhất của hệ thống tích hợp nó. Hình 6.10 mô tả kiến trúc của một hệ thống crawler lớn chứa đựng rất nhiều thành phần chức năng và hoạt động rất phức tạp.

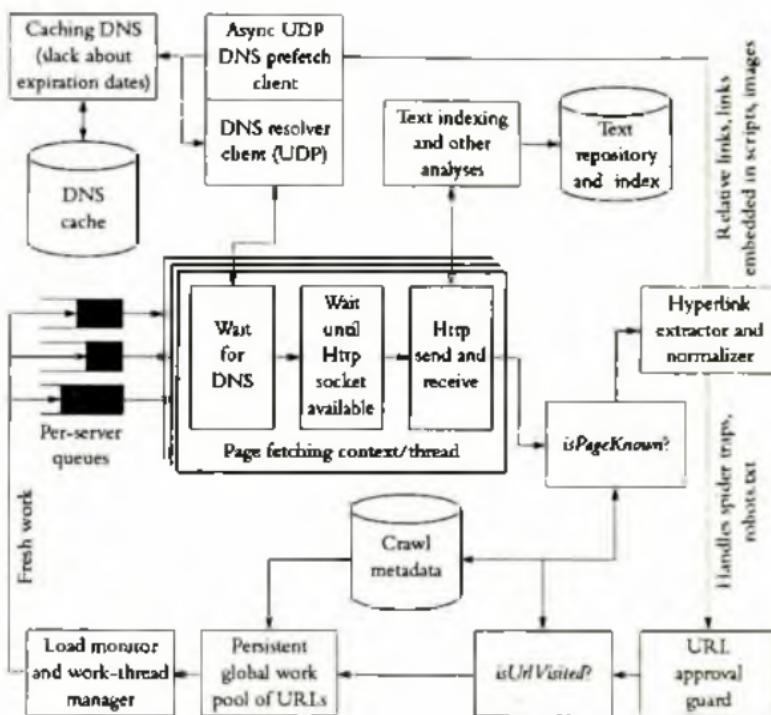
• Tài nguyên được lưu trong hệ thống

Nội dung thứ hai là tài nguyên trên Web cần được lưu trong hệ thống. Các hệ thống tìm kiếm thường cố gắng thu thập được càng nhiều trang Web càng tốt. Các hệ thống này thường sử dụng Web crawler để bảo trì CSDL được đánh chỉ mục của chúng, cân bằng cái giá của quá trình crawling và đánh chỉ mục với hàng triệu truy vấn mà hệ thống nhận được. Thành phần crawler của các hệ thống này thường có xu hướng và mục tiêu chính là tài về cho bằng hết các trang Web được gặp. Ngược lại, các crawler khác lại chỉ chọn một số trang Web để tải và duyệt trong số rất nhiều các trang Web nó gặp, các crawler này được gọi là các crawler có lựa chọn *preferential crawler* hoặc crawler dựa trên kinh nghiệm. Chúng được sử dụng để xây dựng các kho dữ liệu có chủ đề, tự động hóa các nguồn lực khai phá và đáp ứng cho các đại lý phần mềm. Như đã biết, crawler hướng chủ đề được lựa chọn xây dựng để lấy ra các trang Web theo một chủ đề xác định.

• Cách thức duyệt Web

Như đã được giới thiệu, duyệt Web được tiến hành theo chiều rộng và chiều sâu. Một nội dung quan trọng cần xem xét khi nghiên cứu các crawler, đặc biệt là các crawler hướng chủ đề, đó là bản chất của nhiệm vụ duyệt Web. Các tính chất của việc crawling như là các truy vấn, hay là các từ khoá được cung cấp như là các đầu vào cho các crawler, các hồ sơ người dùng user – profile, hay các thuộc tính của trang Web cần tải (các trang tương tự, các trang Web phổ biến,...) có thể dẫn tới các thay đổi đáng kể trong việc thiết kế và thực thi các crawler. Các tác vụ có thể bị ràng buộc bởi các tham số như số lượng cực đại các trang Web cần nạp, hay dung lượng bộ nhớ có thể... Do đó, một nhiệm vụ crawling có thể được xem như một bài toán tìm

kiểm bị ràng buộc bởi nhiều mục tiêu. Tuy nhiên, sự đa dạng của các hàm mục tiêu cộng với việc thiếu hiểu biết chính xác về không gian tìm kiếm làm cho vấn đề càng trở nên phức tạp. Hơn nữa, một chương trình crawler có thể sẽ phải giải quyết các vấn đề về tối ưu hoá như tối ưu toàn cục và tối ưu cục bộ.



Hình 6.11. Kiến trúc điển hình của một crawler lớn [Cha03]

• Lập lịch làm tươi trang Web

Tính "tươi" của tài nguyên trong hệ thống là một nội dung cơ bản trong thành phần crawler. Môi trường Web là một thực thể động, với các không gian con thay đổi theo các xu hướng khác nhau và thường là với tốc độ rất nhanh. Theo Jaysethuraman và Leyla Ozsen [IO02] thì trung bình 23% trang Web thay đổi hàng ngày (tỷ lệ này là 40% với trang Web thương mại) và chu kỳ phản rã một trang Web là 20 ngày. Do đó, máy tìm kiếm cần có cơ chế "làm tươi" để tài nguyên dữ liệu được nó lưu trữ được cập nhật so với thế giới thực của hệ thống Web. Như vậy, máy tìm kiếm cần "thường xuyên thăm" các trang Web đã được đánh chỉ số để đảm bảo tính tươi mới trên. Theo Jaysethuraman và Leyla Ozsen, chiến lược thăm tối ưu trang

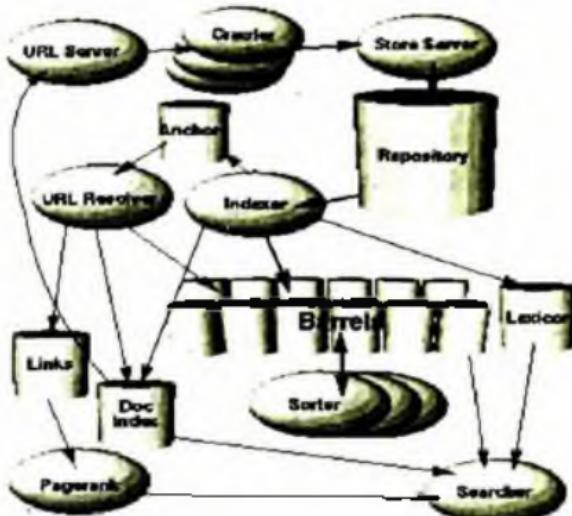
Web liên quan tới hai bài toán về tính thường xuyên và lập lịch crawling. Mục tiêu bài toán về tính thường xuyên nhằm tối ưu số lượng lần crawling mỗi trang (tối ưu hóa thời điểm thăm trang). Một số yếu tố về ràng buộc trong đời sống thực và khối lượng tính toán đã được nêu ra. Đầu ra của bài toán về tính thường xuyên là dâu vào của bài toán lập lịch crawling, theo đó với tập hợp thời điểm tối ưu đến thăm và crawling mỗi trang Web thì tìm ra một lịch tối ưu cho toàn bộ các trang Web, sao cho giá phải trả là nhỏ nhất. Bài toán được đưa về bài toán vận tải.

Khi giải bài toán về tính thường xuyên, cần trả lời câu hỏi nội dung hiện có của trang Web có "mới hơn" nội dung đã được lưu trữ trong máy tìm kiếm hay không? Giải pháp hàm bấm được dùng để đưa ra câu trả lời. Khi lưu trang Web vào kho chứa, máy tìm kiếm thường lưu một giá trị bấm nội dung trang Web cùng với kết quả nén nội dung trang Web. Theo một chiến lược được chọn, khi trang Web đó được tải về theo chế độ lâm tưối, nội dung hiện thời của nó bấm so sánh với giá trị bấm đã có, nếu khác nhau thì nội dung mới sẽ thay thế cho nội dung cũ.

6.5. Phân tích và đánh chỉ số

Trong [PR98], Sergey Brin và Lawrence Page đã trình bày cụ thể về máy tìm kiếm Google theo quan điểm của những người thiết kế. Hình 6.12 trình bày kiến trúc mức cao của Google. Theo [PR98], quy trình hoạt động của Google có thể được diễn tả như sau (cho phép khai thác năng lực song song và đa luồng của hệ thống):

- URLserver gửi danh sách URL Webpage sẽ được đưa về cho crawler (hoặc các crawler phân tán).
- Các crawler tải nội dung trang Web về gửi cho StoreServer.
- StoreServer nén và lưu Webpage vào kho chứa lên đĩa. Việc lưu trữ trang Web trước khi đưa trang Web ra phân tích có tác dụng: (i) lưu nhanh được nội dung trang Web tải về; (ii) tăng tốc độ hoạt động của crawling; (iii) tận dụng được năng lực thực hiện đồng thời của các hệ thống.
- Indexer tiến hành các hoạt động theo chức năng là: (i) đọc nội dung trang Web từ kho chứa ra; (ii) giải nén nội dung trang Web; (iii) gọi Parser để phân tích cú pháp dựa trang Web.
- Indexer cùng Sorter gán DocID cho Web page. DocID được gán mỗi khi Parser phát hiện một URL mới trong nội dung trang Web đang được phân tích.
- Mỗi nội dung Web được chuyển đổi thành tập các xuất hiện của các từ khóa (gọi là hit) trong nó. Chi tiết về hit được giới thiệu ở mục sau.



Hình 6.12. Kiến trúc Google ở mức cao [PB98]

- Indexer phân tích các siêu liên kết, lưu các thông tin quan trọng vào file "anchor" cho phép xác định: (i) nguồn, đích của siêu liên kết; (ii) nội dung văn bản trong siêu liên kết.
- URLsolver đọc file anchor, chuyển đổi URL tương đối thành URL tuyệt đối.
- URLsolver cập nhật lại theo chi số DocID.
- URLsolver đưa text anchor vào *index thuận* (hướng trỏ anchor).
- URLsolver sinh CSDL liên kết chứa các cặp liên kết (DocID₁, DocID₂) được dùng cho việc tính PageRank.
- Sorter đọc các Barrel (được xếp theo DocID) để sắp lại theo WordID tạo ra các *index ngược*. Bổ sung danh sách các wordID và giá số trong index ngược.
- DumpLexicon lấy từ các từ điển lexicon và danh sách wordID để sinh ra lexicon mới.
- Searcher chạy theo yêu cầu của Webserver để đưa ra trả lời câu hỏi dựa trên lexicon mới, giá trị hạng PageRank và CSDL index ngược.

6.5.1. Các cơ sở dữ liệu của máy tìm kiếm

* Kho chứa các trang Web

Theo quy trình hoạt động của máy tìm kiếm Google, sau khi được crawler tải về, trang Web được kho chứa lưu vào kho chứa nội dung để sau đó được đánh chỉ số để phục vụ cho thành phần khác trong hệ thống.

Ở dạng đơn giản nhất, kho chứa các trang Web có thể lưu các trang Web đã được crawling dưới dạng các file riêng biệt. Khi đó, mỗi trang phải được ánh xạ tới một tên file duy nhất. Một cách thi hành điều này là ánh xạ URL của mỗi trang tới một chuỗi nén bằng cách sử dụng một dạng hàm băm với xác suất xung đột thấp (để đảm bảo tính duy nhất của tên file). Các giá trị băm được sử dụng làm các tên file. Sử dụng hàm băm một chiều MD5 để cung cấp mã băm 128 bit cho mỗi URL. Giá trị băm 128 bit sau đó được chuyển thành 32 ký tự ở dạng cơ số 16 tương ứng. Theo cách này, các tên file có chiều dài cố định cho các URL có độ dài bất kỳ. Các kho chứa nội dung trang Web có thể được sử dụng để kiểm tra liệu một URL đã được crawling trước đó hay chưa bằng cách chuyển URL đó sang 32 ký tự thập lục phân và kiểm tra sự tồn tại của nó trong kho chứa. Trong một số trường hợp, điều này có thể dẫn tới sự không cần thiết của cấu trúc dữ liệu history trong bộ nhớ trong.

Repository: 53.5 GB = 147.8 GB uncompressed

| | | |
|------|--------|-------------------|
| sync | length | compressed packet |
| sync | length | compressed packet |

Packet (stored compressed in repository)

| docid | ecode | urllen | pagelen | url | page |
|-------|-------|--------|---------|-----|------|
| | | | | | |

a) Kho chứa trang Web

Hit: 2 bytes

| | | | | |
|---------|-------|---------|--------------|---------------|
| plain: | cap:1 | imp:3 | position: 12 | |
| fancy: | cap:1 | imp = 7 | type: 4 | position: 8 |
| anchor: | cap:1 | imp = 7 | type: 4 | hash:4 pos: 4 |

b) Các kiểu hit

Hình 6.13. Một số cấu trúc dữ liệu của Google [BP98]

Để lưu trữ số lượng lớn trang Web, cách thức lưu trữ trang Web được tải về như trên là không thích hợp. Việc sử dụng hàm băm một chiều MD5 vẫn được tiến hành, song tập các trang Web tải về được tổ chức trong hệ thống CSDL đáp ứng yêu cầu đảm bảo số lượng lớn và tốc độ cập nhật cao. Hình 6.13a cung cấp một số nội dung về kho chứa trong máy tìm kiếm Google [BO98], trong đó:

- Kho chứa được tổ chức và hoạt động theo phương thức ISAM (Index Sequel Access Mode).

- Mỗi trang Web được tải về tương ứng với với một bản ghi trong file ISAM với ba trường là sync, length và compressed packet, trong đó độ dài

Forward Barrels: total 43 GB

| | | | |
|-------|-------------|--------|-----------------|
| docid | wordid 24 | nhts 2 | hit hit hit hit |
| | wordid 24 | nhts 2 | hit hit hit hit |
| | null wordid | | |
| docid | wordid 24 | nhts 2 | hit hit hit hit |
| | wordid 24 | nhts 2 | hit hit hit hit |
| | wordid 24 | nhts 2 | hit hit hit hit |
| | null wordid | | |

c) Chỉ số thuận

Lexicon: 293 MB Inverted Barrels: 41 GB

| | | | |
|--------|------|---|---------------------------------|
| wordid | nhts | → | docid 27 nhts 5 hit hit hit hit |
| wordid | nhts | → | docid 27 nhts 5 hit hit hit hit |
| wordid | nhts | → | docid 27 nhts 5 hit hit hit hit |
| wordid | nhts | → | docid 27 nhts 5 hit hit |

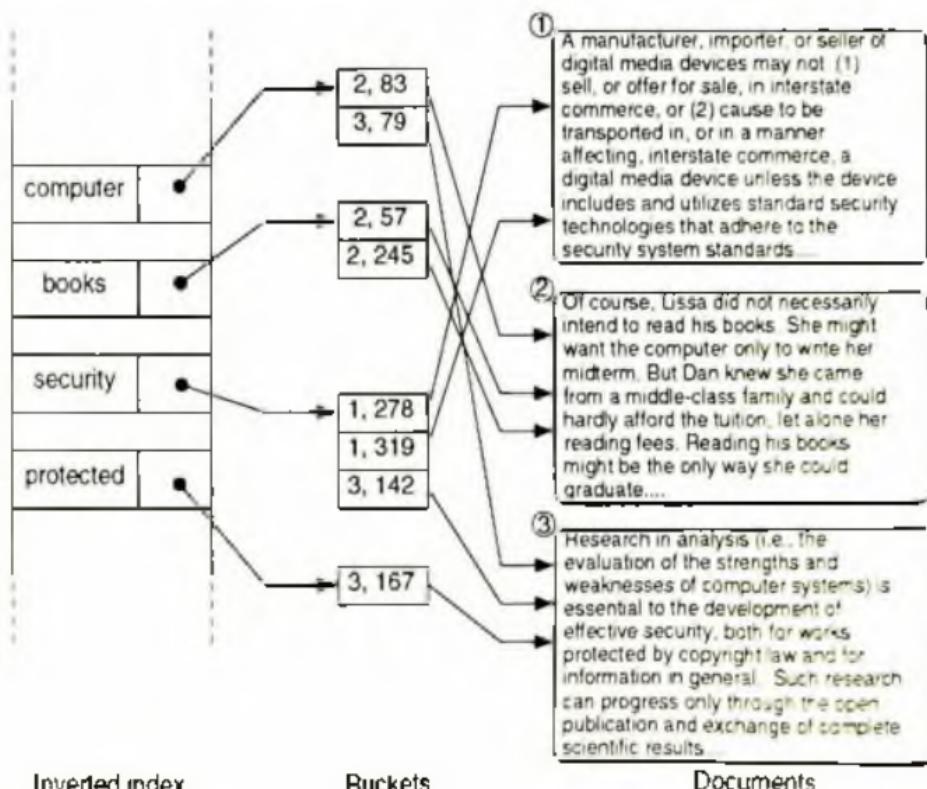
d) Chỉ số ngược

của trường compressed packet thay đổi theo độ dài của trang Web. Trường compressed packet là kết quả nén của một thành phần liên quan tới trang Web. DocID cũng là chỉ số về bàn ghi nói trên theo tổ chức của ISAM. Google sử dụng thuật toán nén/giai nén zlib khi cân nhắc kết hợp giữa tỷ lệ nén với thời gian nén/giai nén.

- **Các cấu trúc dữ liệu chỉ số quan trọng**

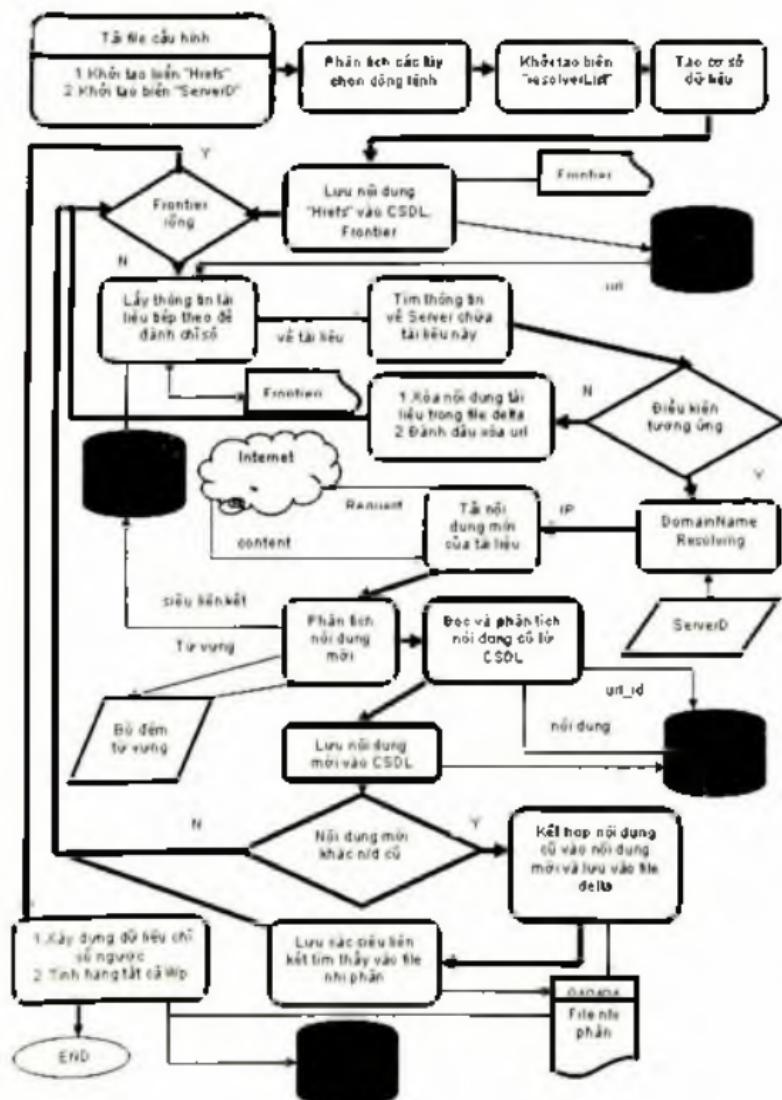
Kho từ vựng (lexicon), hệ thống chỉ số thuận (forward barrels), hệ thống chỉ số ngược (inverted barrels) là ba cấu trúc dữ liệu chỉ số quan trọng trong các máy tìm kiếm.

Kho từ vựng lưu trữ các từ khoá đã được máy tìm kiếm đánh chỉ số. Các trường thông tin quan trọng trong kho từ vựng là chỉ số của từ khoá, nội dung từ khoá và số lượng các tài liệu chứa từ khoá nói trên. Hình 6.13d cho một phần nội dung của kho từ vựng, trong đó trường ndocs chứa số lượng các tài liệu liên quan. Điều đó giải thích tại sao khi người dùng đưa ra một từ khoá hoặc một cụm từ khoá, máy tìm kiếm đưa ra thông tin rất nhanh về số lượng tài liệu chứa từ khoá đó.



Hình 6.14. Cấu trúc chỉ số ngược điển hình [BFS03]

Hạt nhân của các hệ thống chỉ số thuận và ngược là hit. Hit được lưu trong 2 byte. Hit có hai kiểu là plain và fancy. Hit plain tương ứng với word trong nội dung (có phân biệt chữ hoa, chữ thường). Hit fancy lại gồm hai loại con là hit fancy thường và hit fancy anchor. Hit fancy anchor có 4 bit vị trí trong anchor và 4 bit hash cho DocID chứa anchor. Indexer phân bổ lưu trữ các hit thành các tập "thùng chứa" (barrel) lưu trữ các chỉ số đã được sắp xếp.



Hình 6.15. Mô hình hoạt động mức thấp của thành phần indexer trong ASPSeek
(<http://www.aspseek.com/>)

Hệ thống chỉ số thuận cho mô tả ánh xạ từ tài liệu (DocID) tới từ khoá (WordID), bao gồm các thông tin là tài liệu chứa bao nhiêu từ khoá, đó là các từ khoá nào và vị trí cụ thể của từng từ khoá trong tài liệu. Trong một số trường hợp, dùng cách thức ghi nhận giá số kế tiếp để liệt kê các vị trí xuất hiện của từ khoá trong tài liệu. Hình 6.13c mô tả cơ bản về chỉ số thuận.

Hệ thống chỉ số ngược cùng với kho từ vựng cho ánh xạ ngược lại với chỉ số thuận, tức là ánh xạ từ từ khoá (WordID) sang tài liệu (DocID), bao gồm các thông tin là từ khoá xuất hiện trong bao nhiêu tài liệu, đó là các tài liệu nào và vị trí cụ thể xuất hiện của từ khoá trong tài liệu. Tương tự, liệt kê các vị trí xuất hiện của từ khoá trong tài liệu bằng ghi nhận giá số kế tiếp. Hình 6.13d mô tả cơ bản về chỉ số ngược còn Hình 6.14 cho một mô tả cụ thể về chỉ số ngược. Hệ thống chỉ số ngược tuy làm tăng thêm dung lượng nhớ, nhưng lại làm nền tảng cho việc tìm kiếm nhanh trong máy tìm kiếm. Hình 6.15 cung cấp mô hình hoạt động ở mức thấp của Indexer trong máy tìm kiếm ASPseek.

6.5.2. Tách và chuẩn hoá URL

Bộ duyệt HTML đã được xây dựng sẵn trong rất nhiều ngôn ngữ. Chúng cung cấp các tính năng để dễ dàng xác định các thẻ HTML và liên kết giá trị các cặp thuộc tính trong một văn bản HTML cho trước. Để lấy ra được các URL hyperlink từ một trang Web, có thể sử dụng các bộ duyệt ở trên để tìm các thẻ anchor và lấy ra các giá trị của thuộc tính href tương ứng. Tuy nhiên, hệ thống cần chuyển các URL tương đối sang các địa chỉ URL tuyệt đối sử dụng URL cơ sở của trang Web nơi chúng được trích ra.

Các URL khác nhau tương ứng với cùng một trang Web có thể được ánh xạ vào một dạng chuẩn nhất. Điều này rất quan trọng, nhằm tránh được việc nạp cùng một trang Web nhiều lần. Sau đây là các bước được sử dụng trong các hàm chuẩn hoá thông dụng:

- Chuyển giao thức và tên máy chủ sang dạng chữ thường.
- Loại bỏ phần anchor hoặc reference của URL.
- Thực hiện việc mã hoá URL bằng các ký tự thông dụng như ‘~’.
- Đổi với một số URL, thêm vào dấu ‘/’, <http://dollar.biz.uiowa.edu> và <http://dollar.biz.uiowa.edu/> phải cùng ánh xạ vào cùng một dạng chuẩn.
- Sử dụng các kinh nghiệm để nhận ra các trang Web mặc định.
- Loại bỏ ký tự ‘..’ và thư mục cha khỏi đường dẫn URL.
- Đẽ lại các số hiệu công trong URL, ngoại trừ đó là công 80. Một cách khác là đẽ lại các số hiệu công trong URL và thêm công 80 nếu số hiệu công không được chỉ định.

Loại bỏ các từ dừng và chuyển các dạng thức của từ sang dạng gốc.

Như đã trình bày về biểu diễn văn bản, khi duyệt một trang Web để trích ra các thông tin nội dung hoặc để tính điểm các URL mà trang đó trỏ tới, thông thường cần loại bỏ các từ dừng và đưa các từ về từ gốc.

6.6. Tính hạng trang Web

Việc giải quyết bài toán tìm kiếm và cung cấp thông tin được người dùng thực sự quan tâm trong giới hạn thời gian cho phép đã trở thành công việc hết sức cần thiết khi xây dựng các công cụ tìm kiếm thông tin. Công nghệ xây dựng công cụ tìm kiếm thông tin trên Internet (diễn hình là máy tìm kiếm) cần được không ngừng cải tiến, nhằm bảo đảm thỏa mãn yêu cầu người dùng cả theo khía cạnh thời gian tìm kiếm nhanh lẫn tính sự phù hợp cao giữa các trang thông tin kết quả tìm được với yêu cầu tìm kiếm của người dùng.

Khi người dùng nhập vào một nhóm từ khoá tìm kiếm (hoặc một câu hỏi), máy tìm kiếm sẽ thực hiện nhiệm vụ tìm kiếm và trả lại một số trang Web theo yêu cầu người dùng. Nhưng số các trang Web liên quan đến từ khoá tìm kiếm có thể lên tới hàng vạn trang, trong khi người dùng chỉ quan tâm đến một số ít trang trong đó. Vậy, việc tìm ra các trang đáp ứng nhiều nhất yêu cầu người dùng để đưa lên đầu là cần thiết. Đó chính là công việc tính hạng trang Web của máy tìm kiếm – sắp xếp các trang Web theo độ phù hợp với câu hỏi. Khi hiển thị kết quả tìm kiếm, tập các trang Web kết quả được xuất hiện theo thứ tự giảm dần về độ quan trọng.

Cần thiết phải xác định phép đo về "độ phù hợp" của một trang Web với yêu cầu người dùng [BP98, KHM03]. Liên quan tới việc xác định phép đo như vậy, người ta quan tâm tới hai hướng giải quyết. Hướng thứ nhất sử dụng độ quan trọng (được xác định qua một đại lượng được gọi là hạng trang – page rank) của trang Web làm độ phù hợp với yêu cầu người dùng. Hầu hết các nghiên cứu đều thừa nhận một giả thiết là, nếu một trang Web mà có nhiều trang Web khác hướng (link) tới thì trang Web đó là trang Web quan trọng. Trong trường hợp này, hạng trang được tính toán chỉ dựa trên mối liên kết giữa các trang Web với nhau. Hầu hết các máy tìm kiếm sử dụng hạng trang làm độ phù hợp của kết quả tìm kiếm với các thuật toán diễn hình là PageRank, Modified Adaptive PageRank [KHM03]. Hướng thứ hai coi độ phù hợp của trang Web với câu hỏi của người dùng không chỉ dựa trên giá trị hạng trang Web như trên, mà còn phải tính đến mối liên quan giữa nội dung trang Web đó với nội dung câu hỏi theo yêu cầu của người dùng. Thuật toán diễn hình theo hướng này là Topic-sensitive PageRank [Hav02]. Một số nghiên cứu khai thác khía cạnh nội dung của trang Web phục vụ việc tính toán độ phù hợp của trang Web tìm kiếm với câu hỏi người dùng cũng được đề cập trong một số công trình [NMT04, Hav99].

6.6.1. Độ quan trọng của trang Web

Một số yếu tố dưới đây được sử dụng để đo độ quan trọng của các trang Web, bao gồm cả tính phù hợp của trang Web đối với truy vấn.

1. Các từ khoá trong văn bản: Một trang Web được coi là phù hợp nếu nó có chứa một số, hoặc tất cả các từ khoá trong câu truy vấn. Ngoài ra, tần số xuất hiện của từ khoá trong trang Web cũng được xem xét.

2. Mức độ tương tự với câu truy vấn: Người dùng chỉ định thông tin cần tìm bằng một câu truy vấn ngắn, hay bằng các cụm từ dài. Mức độ tương tự giữa mô tả ngắn (hay dài) truy vấn người dùng với nội dung mỗi trang Web được tải về có thể được sử dụng để xác định tính phù hợp của trang Web.

3. Mức độ tương tự với trang hạt nhân: Hệ thống định ra một số URL hạt nhân và các trang tương ứng với các URL hạt nhân được sử dụng để đo mức độ phù hợp của mỗi trang được tải. Các trang hạt nhân được kết hợp với nhau thành một văn bản lớn duy nhất. Mức độ gần nhau của văn bản này với trang Web hiện thời được coi là điểm số của trang hiện thời đó.

4. Điểm số qua phân lớp: Có thể dùng bộ phân lớp trang Web sau khi được huấn luyện để xác định tính phù hợp của một trang Web với thông tin hoặc yêu cầu cần giải quyết. Các trang hạt nhân (hoặc các trang Web được chỉ định trước) được coi là các ví dụ dương để tiến hành huấn luyện bộ phân lớp. Sau khi được huấn luyện, bộ phân lớp thực hiện việc gán các điểm số nhị phân (0, 1) hoặc có thứ tự cho các trang Web được duyệt.

5. Đánh giá độ quan trọng dựa trên liên kết: Một crawler (crawler) có thể sử dụng các thuật toán như PageRank hoặc HITS để cung cấp cách thức đánh giá độ quan trọng của mỗi trang Web được duyệt. Đơn giản hơn, chỉ cần sử dụng số lượng các liên kết tới trang Web đó để xác định thông tin về độ quan trọng của nó.

6.6.2. Thuật toán tính hạng trang

Trong [BP98, PBM98], Sergey Brin, Lawrence Page và cộng sự đưa ra một phương pháp nhằm trợ giúp việc tính toán hạng trang. Phương pháp này dựa trên ý tưởng rằng: "Nếu có liên kết từ trang A đến trang B thì độ quan trọng của trang A cũng ảnh hưởng đến độ quan trọng của trang B". Ví dụ, một cách trực quan có thể kiểm định nhận định này là nếu trang Web bất kỳ được trang Yahoo chỉ dẫn tới, chắc chắn sẽ quan trọng hơn trang Web được một trang Web vô danh nào khác chỉ tới. Giả sử, đã có một tập hợp các trang Web cùng các liên kết giữa chúng, thì khi đó, có thể lập một đồ thị với các đỉnh là các trang Web còn các cạnh là các liên kết giữa chúng. Như đã biết, người ta gọi đồ thị như vậy là đồ thị Web.

• Thuật toán PageRank

Giả sử rằng, các trang Web tạo thành một đồ thị liên thông, nghĩa là từ một trang có thể có đường liên kết tới một trang Web khác trong đồ thị đó.

Các trang Web được đánh số từ 1, 2, ..., m. Gọi $N(i)$ là số trang Web có liên kết từ trang i và $B(i)$ là số các trang Web có liên kết đến trang i .

Khi đó, giá trị PageRank $r(i)$ ứng với trang i được tính theo công thức:

$$r(i) = \sum_{j \in B(i)} r(j)/N(j)$$

Đặt $r = [r(1), r(2), \dots, r(n)]$ là vector PageRank, trong đó thành phần thứ i là hạng tương ứng của trang Web i . Các phương trình trên được viết dưới dạng ma trận $r = A^T r$ (vector PageRank r chính là vector riêng của ma trận A^T), trong đó A là ma trận có kích thước $n \times n$ với giá trị các phần tử được tính toán theo:

$$a_{ij} = \begin{cases} \frac{1}{N_j} & \text{Có liên kết từ } i \text{ tới } j: i \rightarrow j; \\ 0 & \text{Không có liên kết từ } i \text{ tới } j. \end{cases} \quad (6.5)$$

Như đã thấy, việc tính toán mức độ quan trọng (hay hạng trang) của trang Web theo phương pháp PageRank có thể được thực hiện thông qua việc phân tích các liên kết tới trang Web đó. Nếu có các liên kết quan trọng trả tới, thì rất có thể trang đó là trang quan trọng. Tuy nhiên, việc tính toán hạng trang lại phụ thuộc vào việc biết được hạng của các trang Web có liên kết tới nó, và như vậy, muốn tính hạng trang này cần phải biết được hạng của trang liên kết, điều này có thể gây ra việc lặp vô hạn rất tốn kém. Có thể tính toán được các hạng trang thông qua việc tính toán vector riêng của ma trận A^T . Tồn tại một số phương pháp tính vector riêng của ma trận, tuy nhiên, phương pháp lặp là khá thuận tiện và có thể được áp dụng vào việc tính toán vector PageRank. Quy trình tính toán như sau:

1. $s \leftarrow$ vector bất kỳ;
2. $r \leftarrow A^T s$;
3. Nếu $\|r - s\| < \epsilon$ thì kết thúc (ϵ là số dương rất bé, được gọi là sai số lặp), nhận được r là vector PageRank; nếu không thì $s \leftarrow r$ và quay lại bước 2.

Trong nhiều trường hợp, giả thiết về tính liên thông của đồ thị trang Web đang được xem xét là không thực tế. Trên WWW có rất nhiều các trang Web mà chúng không hề có trang nào liên kết tới (Web leak) hay không liên kết tới trang Web nào khác (Web sink). Đối với một số trường hợp các trang Web không có liên kết tới, khi áp dụng thuật toán PageRank nguyên thuỷ sẽ nhận được kết quả hạng trang = 0 đối với trang 4 và trang 5. Điều này không phù hợp với thực tế, vì bất kỳ trang Web nào được xây dựng cũng mang một ngữ nghĩa nào đó, tức là có tính quan trọng, hay độ quan trọng của nó phải dương. Do vậy, cần điều chỉnh công thức tính PageRank nhờ thêm vào một hệ số α để bao hàm được nội dung này. Công thức PageRank được sửa đổi có dạng như sau:

$$r(i) = d * \sum_{j \in B} r(j)/N(j) + (1-d)/n$$

Việc thêm "hệ số hâm" d (theo thực nghiệm, $d = 0.85$) có ý nghĩa như việc bổ sung thêm giá trị PageRank cho nhóm trang không có link ra ngoài nhóm. Khi $d = 1$ sẽ quay lại trường hợp PageRank nguyên thuỷ.

Sergey Brin và Lawrence Page [BP98] chỉ ra rằng, các giá trị này có thể hội tụ khá nhanh, sau khoảng 100 vòng lặp có thể nhận được kết quả với sai số cho phép.

• Một số thuật toán khác

Một số phương pháp tăng hiệu năng tính toán của thuật toán PageRank nhờ tăng tốc độ tính toán đã được đề xuất. Một trong các phương pháp tăng tốc độ tính toán được phổ biến hiện nay là Modified Adaptive PageRank [Hav02, Hav03, NMT04].

PageRank là một trong những phương pháp lịnh hành nhất và có hiệu quả nhất trong công việc tìm kiếm các thông tin trên Internet. Như đã xét ở trên, PageRank tìm cách đánh giá hạng các trang thông qua các liên kết giữa các trang Web. Điều này tiến hành thông qua việc tính toán vector riêng của ma trận kè biêu diễn các trang Web. Nhưng với kích cỡ không lồ của WWW, công việc tính toán này tốn thời gian nhiều ngày. Cần phải tăng được tốc độ tính toán hạng trang.

Mặt khác, cần sớm có kết quả để đưa những thông tin hạng trang sang các thành phần khác của máy tìm kiếm, vì vậy tính toán nhanh vector PageRank có thể giúp giảm thiểu thời gian chết của những thành phần liên quan đó.

Hiện nay, các phương pháp nghiên cứu mới đều tập trung vào việc đánh giá dựa trên những tiêu chí có tính đến sự quan tâm của người dùng, do vậy cần phải tính toán nhiều vector PageRank, mỗi vector hướng tới một tiêu đề khác nhau. Việc tính toán nhiều vector này đòi hỏi mỗi vector thành phần cần được tính toán nhanh chóng.

Như đã nói, việc tăng cường tốc độ tính toán có thể vẫn phải khó khăn do kích thước của WWW. Sepandar Kamvar và cộng sự [KHM03] giới thiệu một giải pháp nhằm tăng tốc quá trình tính toán được nhanh hơn. Phương pháp này xuất phát từ ý tưởng sau: khi cài đặt chương trình và chạy, độ quan trọng các trang Web có tốc độ hội tụ nhanh chậm khác nhau. Điều này dẫn đến ý tưởng tận dụng những trang hội tụ sớm và kết quả độ quan trọng của các trang đã hội tụ có thể không cần phải tính tiếp. Điều này cho phép giam lược những tính toán dư thừa, và do đó làm tăng được hiệu suất tính toán của hệ thống.

Một số kết quả nghiên cứu cài tiến giải pháp tính hạng trang trong một số máy tìm kiếm tiếng Việt thử nghiệm đã được công bố [NNT05, TNT06]. Phát triển phương pháp của Sepandar D. Kamvar và cộng sự [KHM03b] về

việc tính toán hạng trang theo khôi. Thuy Q.H. và cộng sự [TNT06] đề xuất giải pháp tính hạng các trang Web theo các thành phần liên thông, sau đó tích hợp các kết quả. Trong các thực nghiệm tính toán cho máy tìm kiếm tiếng Việt thử nghiệm, thời gian tính hạng giảm khoảng 30% so với thuật toán PageRank nguyên thuỷ với cùng một kết quả xếp hạng.

• Thuật toán Adaptive PageRank

Giả sử việc tính toán vector PageRank đã được thực hiện đến vòng lặp thứ k. Cần tính toán

$$x^{(k+1)} = Ax^{(k)} \quad (*)$$

Gọi C là tập hợp các trang Web đã hội tụ đến mức ε và N là tập hợp các trang Web chưa hội tụ. Khi đó, chia ma trận A ra làm hai ma trận con, A_N có $m \times n$ là ma trận kè đại diện cho những liên kết của m trang chưa hội tụ, còn A_C có $(n - m) \times n$ là ma trận kè đại diện cho những liên kết của $(n - m)$ trang đã hội tụ.

Tương tự, nên chia vector $x^{(k)}$ ra thành 2 vector $X_N^{(k)}$ tương ứng với những thành phần của $x^{(k)}$ đã hội tụ còn $X_C^{(k)}$ tương ứng với những thành phần của $x^{(k)}$ chưa hội tụ. Ma trận A và $x^{(k)}$ được viết lại dưới dạng sau :

$$X^{(k)} = \begin{pmatrix} X_N^{(k)} \\ X_C^{(k)} \end{pmatrix} \text{ và } A = \begin{pmatrix} A_N \\ A_C \end{pmatrix}$$

Và phương trình (*) được viết lại như sau:

$$\begin{pmatrix} X_N^{(k+1)} \\ X_C^{(k+1)} \end{pmatrix} = \begin{pmatrix} A_N \\ A_C \end{pmatrix} \cdot \begin{pmatrix} X_N^{(k)} \\ X_C^{(k)} \end{pmatrix}$$

Do các thành phần của $X_C^{(k)}$ đã hội tụ, do vậy không cần tính $X_C^{(k+1)}$ nữa; việc tính toán được giảm đi do không phải tính toán $A_C X^{(k)}$ mà chỉ cần thực hiện $X_N^{(k+1)} = A_N X^{(k)}$.

• Topic sensitive PageRank

Như đã biết, PageRank là phương pháp tìm kiếm hiện đang được áp dụng trên máy tìm kiếm Google. Tuy nhiên, phương pháp này chỉ quan tâm đến các liên kết mà không quan tâm đến nội dung của trang Web có chứa liên kết đó. Do vậy, có thể dẫn tới những sai lạc trong thông tin tìm kiếm được. Yêu cầu đặt ra là, cần phải đưa ra một phương pháp có tốc độ nhanh như phương pháp PageRank và lại có quan tâm đến nội dung của trang Web thông qua "chủ đề" của nó. Hơn nữa, nếu khai thác được mối quan tâm của người dùng đối với các trang Web trong việc tính độ phù hợp của trang Web với câu hỏi người dùng, thì điều đó hết sức có ý nghĩa. Taher H. Haveliwala [Hav02] đề xuất phương pháp mới nhằm đáp ứng yêu cầu trên, đó là phương pháp PageRank theo chủ đề (Topic sensitive PageRank; TSP). Tác

giả sử dụng khái niệm "phạm vi ngữ cảnh" để biểu thị miền quan tâm của người dùng. Trong [NMT04], thuật toán tìm kiếm trang Web có nội dung tương tự cho một cách tiếp cận khác khi đề cập tới xem xét khía cạnh nội dung trang Web trong bài toán tìm kiếm.

Thuật toán TSP gồm hai bước. Tại bước đầu tiên, các trang Web trong CSDL được phân thành các lớp theo các chủ đề c_1, c_2, \dots, c_n ; gọi T_j là tập hợp những trang Web theo chủ đề c_j . Mỗi lớp tương ứng với một vector PageRank của chủ đề mà mỗi thành phần là giá trị PageRank của mỗi trang trong lớp.

Bước thứ hai được thực hiện trong thời gian hỏi – đáp. Giả sử có truy vấn q , gọi q' là phạm vi ngữ cảnh của q . Mô tả sơ bộ khái niệm phạm vi ngữ cảnh như sau: Với truy vấn thông thường (từ hộp thoại) thì q' chính là q ; trường hợp truy vấn q được đặt bằng cách tô sáng từ khoá q trong trang Web u thì q' sẽ chứa các từ khoá trong u bao gồm cả q . Sau đó tính xác suất để q' thuộc về các chủ đề khác nhau. Sử dụng thuật toán phân lớp Bayes với (i) tập huấn luyện gồm những trang được liệt kê trong các chủ đề; (ii) đầu vào là câu truy vấn hoặc phạm vi ngữ cảnh của câu truy vấn; (iii) đầu ra là xác suất để q' thuộc mỗi chủ đề.

6.6.3. Spam trang Web

Nhu đã được trình bày, máy tìm kiếm hiển thị danh sách các trang Web kết quả theo thứ tự giảm dần về hạng liên quan của chúng với câu hỏi người dùng. Thông thường, người dùng chỉ tập trung vào trang hiển thị đầu tiên và một vài trang hiển thị tiếp theo. Điều đó có nghĩa là, chỉ một phần nhỏ các trang Web kết quả tìm kiếm được người dùng duyệt nội dung.

Các trang thương mại điện tử và quảng cáo được xây dựng để quảng bá tới người dùng. Vì vậy một nhu cầu đặt ra là, chúng phải được truy nhập nhiều. Máy tìm kiếm là một công cụ quảng bá tối thông qua quảng cáo trên nó hoặc hạng trang Web cao. Số lần truy nhập trang Web được một vài giải pháp tính hạng trang quan tâm, song yếu tố này chỉ có được khi trang Web đã có một thời gian sống đủ dài. Người xây dựng trang Web mong muốn trang Web của mình được máy tìm kiếm xếp hạng để được hiển thị tới người dùng. Người tạo trang Web cố gắng áp dụng các kỹ thuật tự cải thiện thứ hạng trang Web của mình. Hiện tượng đó được Monika Henzinger và cộng sự [HMS02] gọi là "spam máy tìm kiếm" hay "web spam" và trang Web sử dụng các kỹ thuật đó cũng được gọi là Web spam. Đồng thời, các dịch vụ tối ưu hạng trang Web xuất hiện và tương ứng là một lĩnh vực mới ra đời, đó là lĩnh vực tối ưu máy tìm kiếm SEOs (Search Engines Optimizers).

Vấn đề Web spam còn phải nói đến các trang Web với thông tin không đúng, mang những nội dung sai trái. Ở đây chỉ đề cập đến vấn đề spam đối

với máy tìm kiếm. Trong ngữ cảnh giới hạn như vậy, công nghệ spam là các kỹ thuật nhằm mục đích nâng cao hạng của các trang Web.

Ngày nay, spam đã trở thành phổ biến và được thương mại hóa, nên một trong những vấn đề đặt ra cho máy tìm kiếm là đưa ra độ đo để xác định, loại bỏ spam, nhằm đảm bảo sự chính xác và phù hợp của hạng trang.

Một số công nghệ để nhận diện và loại bỏ spam đã được phát triển và cài tiến. Nhưng khi công nghệ tìm kiếm càng phát triển, thì các kỹ thuật spam mới cũng được phát triển tương ứng. Do vậy, công nghệ chống spam ở các máy tìm kiếm thường không công khai để hạn chế thông tin nhằm ngăn chặn sự phá hoại của những người tạo spam.

Năm bối được cách thức tạo spam được coi là tiền đề cần thiết để nhận diện spam và phát triển các thuật toán tính hạng trang có loại trừ ảnh hưởng của nó luôn là vấn đề cần được nghiên cứu trong lĩnh vực máy tìm kiếm.

• Các công nghệ tạo spam

Theo Monika Henzinger và cộng sự [HMS02], công nghệ spam có thể được chia thành 3 loại chính là spam văn bản (text spam), spam liên kết (link spam) và giấu dạng (cloaking).

- Spam văn bản:

Tất cả các máy tìm kiếm đều dựa vào nội dung văn bản để quyết định độ phù hợp của từng trang theo câu truy vấn (độ đo TFIDF). Từ đó, công nghệ spam văn bản hướng vào việc thay đổi nội dung văn bản, nhằm nâng cao hạng trang theo một số cách sau đây:

Dựa vào các đặc điểm của máy tìm kiếm, tập trung vào một tập nhỏ các từ khoá và cố gắng nâng cao chất lượng của tập từ khoá đó trong văn bản:

- + Lặp các từ khoá ở cuối trang để không ảnh hưởng nhiều tới người dùng, nhưng lại có ý nghĩa đối với máy tìm kiếm. Để không ảnh hưởng nhiều tới người dùng, phần văn bản được lặp đó có thể được tạo với phông chữ nhòe, hay được ẩn đi bằng cách sử dụng màu chữ cùng màu nền.

- + Đưa từ khoá vào phần tiêu đề của trang hay các mục lớn của trang Web. Vì các máy tìm kiếm thường đánh giá cao các từ khoá ở tiêu đề.

- + Thêm các từ khoá vào phần nội dung thẻ META (một thẻ hay tag của ngôn ngữ HTML), nội dung trong đó được máy tìm kiếm đánh giá cao do ngầm định ở đó chứa các thông tin quan trọng của trang Web. Do vậy, những người tạo spam có thể lạm dụng thẻ này. Ví dụ, máy ảnh, máy quay, máy in, Sony, Canon, Epson, Xerox.

Ngoài ra có thể thêm từ khoá vào nội dung của các liên kết (anchor text). Một ví dụ đơn giản: máy tính, máy in, PC, Laptop, ổ cứng, HDD, thiết bị giá rẻ, miễn phí, bảo hành, tiết kiệm.

- + Cố gắng tăng số lượng từ khoá của văn bản được đánh giá: (i) cách đơn giản nhất là thêm một tập lớn từ (có thể là cả từ điển) ở cuối trang Web

để tăng khả năng được hiển thị cho nhiều truy vấn khác nhau, khả năng trang Web đặc biệt với các câu truy vấn không rõ nghĩa; (ii) thậm chí có thể lặp nội dung của cả văn bản và đồng thời lặp các từ khoá ở nhiều vị trí trong văn bản.

- *Spam liên kết:*

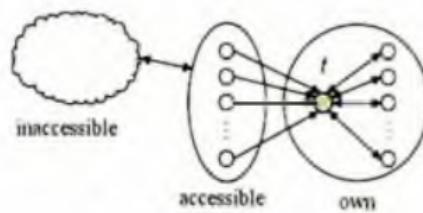
Giải thích được thừa nhận là độ quan trọng của trang phụ thuộc vào số lượng liên kết trỏ tới trang đó là nền tảng của các phương pháp tính hạng trang dựa vào liên kết. Đối với các phương pháp tính hạng trang như vậy, máy tìm kiếm có khả năng xác định hạng của trang Web độc lập với yêu cầu của người dùng vì chỉ căn cứ vào liên kết trong đồ thị Web. Tuy nhiên, điều đó cũng được những người tạo spam lợi dụng để nâng cao hạng trang theo cách thay đổi cấu trúc đồ thị Web. Đó là công nghệ link spam hay spam liên kết.

Mục đích nhắm vào các hệ thống dùng phương pháp tính hạng thô dựa trên số liên kết vào để quyết định độ quan trọng của trang Web như các thuật toán PageRank, HITS. Mô hình cấu trúc liên kết nhắm nâng cao hạng trang PageRank của Z. Gyongyi và H. Garcia-Molina [GG05] được trình bày trên Hình 6.16. Trong mô hình có:

- + Các trang *inaccessible* là các trang mà người tạo spam không có quyền thay đổi, thêm nội dung mới.
- + Các trang *own* là các trang do người tạo spam làm chủ, có toàn quyền sửa đổi, tạo mới.
- + Các trang *accessible* là các trang không phải own nhưng cho phép viết thêm nội dung (như viết bài trong các blog).

Mục tiêu của người tạo spam là tạo các liên kết có lợi để tăng hạng của một hay nhiều trang trong nhóm *own*, nhóm các trang *own* đó được gọi là spam farm. Như trong mô hình trên là cấu trúc liên kết nhắm nâng cao độ quan trọng của trang. Z. Gyongyi và H. Garcia-Molina đưa ra một số kỹ thuật tạo link spam nhằm tăng số liên kết đến và liên kết ra của các trang spam: Những người tạo spam có thể dễ dàng thêm các liên kết ra từ các trang Web của họ tới các trang tốt, với hy vọng tăng trọng số hub (một độ đo tính theo thuật toán HITS) của trang.

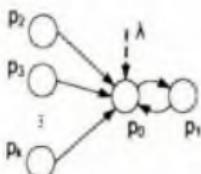
Tren các site *dmoz.org*, *Yahoo!* có danh sách địa chỉ các Web site được phân theo các chủ đề từ lớn đến nhỏ rất cụ thể. Do vậy, những người tạo



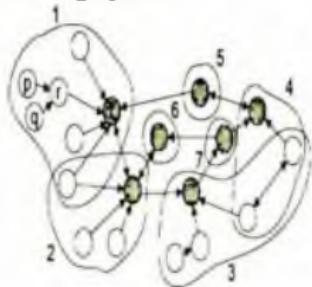
Hình 6.16. Một cấu trúc tối ưu nhằm tăng hạng trang (GG05)

spam dễ dàng lấy thông tin đó đưa vào trang Web của mình, từ đó tạo ra một cấu trúc liên kết ngoài rất lớn. Việc tăng số liên kết đến của một trang Web không đơn giản như việc thêm các liên kết ra, những người tạo spam có thể dựa vào một số kỹ thuật:

+ Tạo một nhóm các trang Web cung cấp các thông tin hữu ích (như các tài liệu hướng dẫn lập trình Java bằng tiếng Việt) mà chúng tôi gọi là trang gốc, và từ các trang đó tạo các liên kết đến các trang spam. Hình 6.17 mô tả một dạng spam với p_0 là trang gốc, p_1 là trang spam.



Hình 6.17. Một dạng spam với trang gốc p_0 .



Hình 6.18. Một cấu trúc liên kết một số spam farm không theo quy luật

Các trang gốc chứa thông tin hữu ích, nên có khả năng sẽ được nhiều trang khác trỏ tới và sẽ có hạng cao. Những trang gốc này không nhất thiết trùng chủ đề với các trang spam do mục tiêu nhằm có được các trang có hạng cao và phân chia hạng đó cho các trang spam qua các liên kết ra.

Tạo các bài viết chứa các liên kết tới trang muốn spam tại các trang cho phép viết bài như các trang blog, wikiVdots. Để tránh việc kiểm soát của những người quản lý, những người tạo spam có thể sử dụng các kỹ thuật để che dấu các liên kết đó với người xem, nhưng vẫn được xử lý bởi các máy tìm kiếm (như việc sử dụng linh hoạt màu sắc).

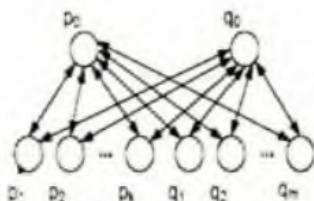
+ Mua các tên miền đã hết hạn và tận dụng các liên kết sẵn có tới các trang Web trong đó.

+ Một kỹ thuật quan trọng đó là việc tạo spam farm (nhóm các trang Web spam có liên kết với nhau). Những người tạo spam có thể nắm giữ một số lượng lớn các site, vì vậy họ dễ dàng tạo cấu trúc liên kết tùy ý giữa các trang trong các site của họ nhằm nâng cao hạng của các trang đó. Hình 6.18 chỉ dẫn một cấu trúc liên kết một số spam farm với các nút màu xám là các trang spam.

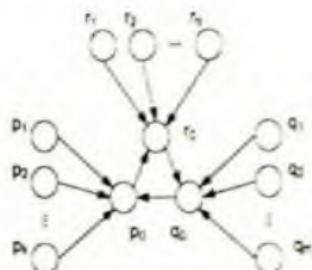
+ Một nhóm những người tạo spam liên kết lại với nhau và tạo các liên kết tới các site của nhau. Hình 6.19 là ví dụ với các trang p , q thuộc hai spam farm.

Một phương pháp cơ bản tạo link spam là người tạo spam đặt link farm, một tập hợp các liên kết trỏ tới tất cả các trang trong cùng site nào đó mà họ

muôn, ở cuối mọi trang Web. Đây là trường hợp đơn giản của spam farm, do vậy dễ dàng được máy tìm kiếm nhận ra, nhưng còn có những kỹ thuật khác tinh vi hơn, như việc tạo các Web vòng (web-ring) như Hình 6.20 với các trang spam r_0, p_0, q_0 có liên kết tạo vòng, hay tạo nhóm các trang Web có mật độ liên kết lớn.



Hình 6.19. Hai spam farm chia sẻ liên kết với nhau



Hình 6.20. Ba spam farm chia sẻ liên kết theo dạng vòng

- Công nghệ giả dạng:

Bên cạnh hai kỹ thuật tạo spam trên, giả dạng (cloaking) là kỹ thuật tạo ra nội dung hoàn toàn khác giữa những gì máy tìm kiếm crawling về với những gì sẽ được hiển thị cho người dùng. Hơn nữa, kỹ thuật này cũng hướng tới sự khác nhau giữa các lần crawling khác nhau của máy tìm kiếm. Việc kết hợp với các kỹ thuật spam ván bàn và spam liên kết cũng được áp dụng cho các trang Web trả về cho máy tìm kiếm để nâng cao hạng trang. Vì vậy, máy tìm kiếm bị đánh lừa về nội dung của trang Web và đưa ra đánh giá hạng trang không chính xác.

• Phát hiện Web spam

Nhiều công trình nghiên cứu về phương pháp xác định link spam đã được công bố [BCS05, WD05, MCL05, MD05, GG05, GGP04, GG05]. điển hình là các công bố tại *First International Workshop on Adversarial Information Retrieval on the Web*, WWW2005 (Chiba, Nhật Bản). Các phương pháp xác định link spam được chia theo các hướng tiếp cận:

Nhận dạng LinkSpam dựa vào cấu trúc liên kết: Với nhận định các trang Web có số liên kết đến và liên kết ra tuân theo phân phối luật số lớn và do đó các trang Web có trọng số PageRank tuân theo luật số lớn. Benetut và cộng sự [BCS05] cho rằng, trang Web được phân phối trọng số PageRank từ các trang liên kết tới tuân theo luật số lớn. Vì vậy, các trang có phân phối trọng số PageRank không tuân theo luật đó được nhận định là link spam. Đồng thời, D. Fetterly và cộng sự [BCS05] cũng phân tích các mô hình phân phối liên kết trên Web để xác định các trang spam. Phương pháp này chỉ thích hợp với những dạng link spam đơn giản với cấu trúc liên kết lớn

được sinh tự động. Với sự phát triển của công nghệ spam hiện nay thì các phương pháp đó không hiệu quả.

Tạo nhóm các trang có liên kết mạnh với nhau, còn gọi là các trang có cấu kết với nhau là một phương pháp nâng cao hạng trang rất hiệu quả và thường được sử dụng bởi những người tạo spam. Baoning Wu và cộng sự [WD05] đã đưa ra một thuật toán hiệu quả để nhận diện nhóm các trang đó và Z. Gyongyi, H. Garcia-Molina [GG05] cũng đưa ra các cải tiến trong công thức tính hạng trang tùy theo cấu trúc liên kết trong các nhóm đó.

Các phương pháp này tập trung vào phân tích các cấu trúc liên kết, tức các trang liên kết với nhau như thế nào để quyết định một trang là spam hay không và thay đổi giá trị hạng trang của chúng.

Xác định spam bằng cách đánh giá độ tốt của các trang, thay vì tìm các trang xấu và hạng các trang Web được phân phối từ hạng của các trang trong tập nhân các trang Web tốt cho trước. Z. Gyongyi và cộng sự [GGP04] đã đưa ra phương pháp này dựa trên quan niệm: các trang tốt thường chỉ liên kết đến các trang tốt, các trang xấu thường chỉ được các trang xấu liên kết đến.



Hình 6.21. Entity Search Engine Cazoodle Apartment Search (<http://www.cazoodle.com/>)

Ngoài ra, với sự ra đời của các trang Web cho phép viết bài phương pháp xác định các link spam, hay cụ thể là blog spam do Gilad Mishne và cộng sự [MCL05] đưa ra nhằm xác định các bài viết xấu chứa liên kết tới các trang Web có chủ đề không phù hợp với chủ đề của bài viết.

6.7. Máy tìm kiếm thực thể

Trong Chương 1, các khuynh hướng cơ bản phát triển ngành khoa học máy tính đã được giới thiệu [Hop07], trong đó phát triển các hệ thống lý thuyết, mô hình và giải pháp tìm kiếm là một khuynh hướng quan trọng. Khuynh hướng phát triển đó không chỉ đáp ứng sự bùng nổ của không gian

tìm kiếm mà còn đáp ứng sự thay đổi câu hỏi tìm kiếm từ dạng cụ thể, thống kê sang dạng tự vẫn và có tính tích hợp cao.

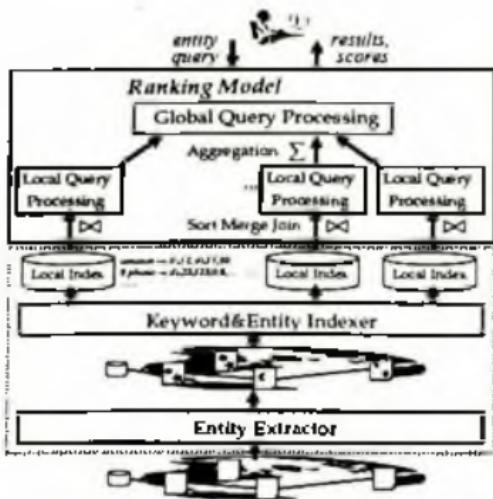
Hệ thống tìm kiếm thực thể (*entity search engine*) và hệ thống tìm kiếm đối tượng (*object search engine*) là thế hệ mới của hệ thống tìm kiếm thông minh, thực hiện việc tìm kiếm, quản lý, xử lý thông tin và phát hiện tri thức về các thực thể thuộc lĩnh vực ứng dụng, nhằm đáp ứng nhu cầu thông tin và tri thức của người sử dụng trên Internet (chẳng hạn, Kevin C. Chang, 2008 [Cha08]; Nigel Collier và cộng sự, 2007 [CKD07]). Khác với máy tìm kiếm thông thường có đối tượng tìm kiếm là trang Web, hệ thống tìm kiếm thực thể có đối tượng tìm kiếm là các thực thể được "trích chọn" ra từ nội dung của một hoặc nhiều trang Web liên quan. Hệ thống Cazoodle (Hình 6.21) do nhóm nghiên cứu của Kevin Chen-Chuan Chang (thuộc Phòng Thi nghiệm Cơ sở dữ liệu và Hệ thống thông tin DAIS thuộc University of Illinois at Urbana-Champaign - UIUC, USA) phát triển là một hệ thống tìm kiếm thực thể điển hình. Một ví dụ khác về hệ thống tìm kiếm thực thể là hệ thống tìm kiếm đối tượng Arnetminer của nhóm Knowledge Engineering Tsinghua University, China (Hình 6.22).



Hình 6.22. Object Search Engine Arnetminer (<http://www.ernetminer.org/>)

Lý thuyết đô thị và lý thuyết xác suất là nền tảng toán học tốt để đề xuất các mô hình và giải pháp trong các hệ thống tìm kiếm thực thể (Dragomir R. Radev, 2009 [Rad09]; Xiaojin Zhu, 2008 [Zhu08]; John E. Hopcroft, 2007 [Hop07]; Tao Cheng, Kevin Chen-Chuan Chang, 2007 [CC07]).

Hình 6.23 trình bày kiến trúc của một hệ thống thực thể [CYC07a]. Một bài toán quan trọng nhất trong hệ thống tìm kiếm thực thể là bài toán tính hạng thực thể. Tính hạng thực thể giải quyết bài toán chọn thông tin thực thể từ nội dung các trang Web.



Hình 6.23. Kiến trúc của một hệ thống tìm kiếm thực thể [CYC07a]

Khác với máy tìm kiếm thông dụng, chưa có một máy tìm kiếm thực thể cho một miền tìm kiếm phổ dụng mà thường hướng tới một hoặc một vài lĩnh vực có liên quan tới nhau.

Câu hỏi và bài tập

1. Cài đặt thủ tục tách và chuẩn hoá URL với đầu vào là một trang Web như đã được đề cập trong chương.
2. Cài đặt module crawler theo thuật toán Naïve tốt nhất đầu tiên.
3. Cài đặt module crawler theo thuật toán SharkSearch.
4. Cài đặt module crawler theo phương pháp crawler hướng tâm.
5. Cài đặt module crawler theo phương pháp crawler hướng ngũ cành.
6. Cài đặt thuật toán tính hạng trang PageRank nguyên thuỷ và thuật toán Adaptive PageRank.

Chương 7

PHÂN CỤM VĂN BẢN

7.1. Giới thiệu

7.1.1. Bài toán phân cụm Web

Một trong những bài toán quan trọng trong lĩnh vực khai phá Web là bài toán phân cụm Web. Phân cụm Web, nói một cách khái quát là việc tự động sinh ra các lớp (cụm) trang Web dựa vào sự tương tự của các trang Web. Các lớp trang Web ở đây là chưa biết trước, người dùng có thể chỉ yêu cầu số lượng các lớp cần phân loại, hệ thống sẽ đưa ra các trang Web theo từng tập hợp (tung cụm), mỗi tập hợp chứa các trang Web tương tự nhau. Bài toán phân cụm được xếp vào dạng học không giám sát (unsupervised learning). Phân cụm Web hiểu một cách đơn giản là phân cụm trên tập các trang Web được lấy từ World Wide Web.

7.1.2. Ứng dụng của phân cụm

Phân cụm có rất nhiều ứng dụng hữu ích. Trong tìm kiếm tương tự (similar search), nếu trước đó đã phân cụm dữ liệu, thì khi lọc các kết quả, ta chỉ tập trung vào các trang Web nằm trong cụm có liên quan nhiều đến câu truy vấn. Như vậy, chất lượng của kết quả tìm kiếm sẽ tốt hơn, không gian tìm kiếm sẽ được thu nhỏ và thời gian tìm kiếm sẽ nhanh hơn. Trong phân cụm phân cấp, có thể tạo ra một hệ thống cây phân cấp các chủ đề của các trang Web, làm cho người đọc có thể tìm các trang Web theo chủ đề người đó quan tâm một cách nhanh chóng. Phân cụm cũng có thể ứng dụng vào việc nhóm các kết quả trả về của một máy tìm kiếm thành các nhóm có chủ đề, và như vậy người dùng có thể tìm đến các trang Web thuộc chủ đề quan tâm một cách nhanh chóng mà không phải duyệt qua toàn bộ danh sách kết quả trả về của máy tìm kiếm. Ngoài ra, phân cụm đã và đang được khai thác trong các ứng dụng khác nhau.

7.1.3. Các phương pháp phân cụm

Có rất nhiều phương pháp phân cụm dựa trên cách các cụm được biểu diễn như thế nào (phẳng hay phân cấp), các thuộc tính của các cụm và các

kiểu giải thuật được sử dụng để phân cụm. Một phương pháp nhằm thực thi thuật toán phân cụm là phân hoạch tập trang Web S thành các cụm (các tập con) S_1, \dots, S_k sao cho đạt được cực tiêu khoáng cách nội tại cụm $\sum_i \sum_{d_1, d_2 \in S_i} \delta(d_1, d_2)$ hoặc làm đạt được cực đại độ tương tự nội tại cụm $\sum_i \sum_{d_1, d_2 \in S_i} \text{sim}(d_1, d_2)$. Trong một mô hình phân cụm, nếu một biểu diễn trang Web được chọn thì biểu diễn đó cũng được dùng để biểu diễn cụm. Chẳng hạn, nếu sử dụng mô hình không gian vector để biểu diễn các trang Web, thì một cụm các trang Web có thể được biểu diễn bằng trọng tâm của các vector trang Web thuộc cụm. Khi chọn biểu diễn một cụm bằng một vector đại diện, thì mục tiêu phân hoạch S thành S_1, \dots, S_k hướng tới việc cực tiêu hoá lòng khoáng cách nội tại $\sum_i \sum_{d \in S_i} \delta(d, c_i)$ hoặc cực đại hoá độ tương tự nội tại $\sum_i \sum_{d \in S_i} \text{sim}(d, c_i)$, trong đó c_i là vector biểu diễn của cụm i . Dưới đây liệt kê một số cách phân loại các phương pháp phân cụm:

- Phương pháp phân cụm dựa trên mô hình (model) và phân cụm phân vùng (partitioning): Phương pháp thứ nhất tạo ra các mô hình biểu diễn các cụm; phương pháp thứ hai chỉ đơn giản là tập hợp các phần tử dữ liệu (trang Web) vào các vùng (cụm).

- Phân cụm đơn định (deterministic) và phân cụm xác suất: Trong phân cụm đơn định, mỗi một phần tử dữ liệu (trang Web) chỉ phụ thuộc vào một cụm. Có thể xem xét việc gán trang Web d thuộc cụm i như là việc đặt một giá trị Boolean $z_{d,i}$ là 1. Trong phân cụm xác suất, mỗi phần tử dữ liệu sẽ có xác suất nào đó đối với mỗi cụm. Trong ngữ cảnh này, $z_{d,i}$ có giá trị là một số thực trong đoạn từ 0 đến 1. Trong bài cảnh như vậy, mong muốn tìm hàm $z: S \times S \rightarrow [0, 1]$ và các vector c_i làm cực tiêu hoá $\sum_i \sum_{d \in S_i} \delta(d, c_i)$ hoặc cực đại hoá $\sum_i \sum_{d \in S_i} \text{sim}(d, c_i)$.

- Phân cụm phẳng và phân cụm phân cấp: Phân cụm phẳng chỉ đơn giản là chia tập dữ liệu thành một số tập con, còn phân cụm phân cấp tạo ra một cây phân cấp của các cụm. Việc phân hoạch có thể thực hiện theo hai cách. Cách thứ nhất bắt đầu bằng việc coi mỗi trang Web vào một cụm của nó và tiến hành kết hợp các cụm trang Web lại với nhau cho đến khi số các cụm là phù hợp; cách này được gọi là phân cụm *từ dưới lên* (bottom – up). Cách thứ hai bắt đầu bằng việc khai báo các cụm nguyên thủy và sau đó gán các trang Web vào các cụm; cách này được gọi là phân cụm *từ trên xuống* (top – down). Tương ứng sẽ này sinh các kỹ thuật phân cụm. Như vậy, có thể xem xét kỹ thuật phân cụm bottom – up dựa vào quá trình lặp lại việc trộn các cụm trang Web tương tự nhau cho đến khi đạt được số cụm mong

muốn: kỹ thuật phân cụm top – down làm mịn dần bằng cách gán các trang Web vào các cụm được thiết đặt trước. Kỹ thuật bottom – up thường chậm hơn, nhưng có thể được dùng trên một tập nhỏ các mẫu có trước để khởi tạo các cụm nguyên thủy trước khi tiến hành kỹ thuật từ trên xuống. Chương này sẽ tập trung vào phương pháp phân cụm phẳng và phân cụm phân cấp.

– Phân cụm theo lô (batch) và phân cụm gia tăng: Trong phân theo lô, toàn bộ tập dữ liệu được sử dụng để tạo ra các cụm. Trong phân cụm gia tăng, giải thuật phân cụm lấy từng phần từ dữ liệu và cập nhật các cụm để phân vào cụm thích hợp.

7.1.4. Các chế độ thực hiện phân cụm

Có hai chế độ thực hiện phân cụm trang Web. Chế độ thứ nhất là phân cụm trên toàn bộ một tập có sẵn gồm rất nhiều trang Web. Thuật toán phân cụm cần tiến hành việc phân cụm toàn bộ tập dữ liệu đó. Tình huống này thường được gọi là phân cụm *ngoại tuyến* (off-line). Chế độ thứ hai thường được áp dụng trên một tập trang Web nhỏ là tập hợp các trang Web do máy tìm kiếm trả về theo một truy vấn của người dùng. Trong trường hợp này, giải pháp phân cụm được tiến hành kiểu *trực tuyến* (on-line) theo nghĩa việc phân cụm tiến hành theo từng bộ phận các trang Web nhận được. Khi đó, thuật toán phải có tính chất "*gia tăng*" (incremental) để tiến hành phân cụm ngay khi chưa có đủ trang Web và phân cụm tiếp theo không cần tiến hành với dữ liệu đã được phân cụm. Do tập trang Web trên Web là vô cùng lớn cho nên cách phân cụm trực tuyến là thích hợp hơn và phải đòi hỏi tính "*gia tăng*" của thuật toán phân cụm.

Quá trình xử lý truy vấn và kết quả phân hạng được phản hồi từ các máy tìm kiếm phụ thuộc vào việc tính toán độ tương tự giữa truy vấn và các trang Web. Mặc dù các truy vấn liên quan phần nào đến các trang Web cần tìm, nhưng nó thường quá ngắn và dễ xảy ra sự nhập nhằng. Như đã biết, trung bình các truy vấn trên Web chỉ gồm hai đến ba từ, do đó gây nên độ nhập nhằng. Chẳng hạn, truy vấn star dẫn đến sự nhập nhằng rất cao, các trang Web lấy được liên quan đến astronomy, plants, animals, popular media và sports figures.... Độ tương tự giữa các trang Web của một truy vấn từ đơn như vậy là khác nhau rất lớn. Vì lẽ đó, nếu máy tìm kiếm phân cụm các kết quả theo từng chủ đề thì người dùng có thể hiệu truy vấn nhanh chóng hoặc tìm vào một chủ đề xác định.

7.1.5. Đặc điểm của bài toán phân cụm Web

Việc phân cụm trực tuyến các trang Web kết quả trả về từ máy tìm kiếm là rất khác so với việc phân cụm các trang Web thông thường. Một đặc điểm là số lượng các trang Web là vô cùng lớn và luôn luôn thay đổi. Ngoài ra,

một vấn đề nữa là các hệ thống tìm kiếm thông tin là tương tác người dùng, cho nên thời gian đáp ứng của hệ thống phải đủ nhanh, cụ thể bài toán ở đây cần thời gian đáp ứng cần tính bằng giây. Mỗi trang Web không chỉ liên quan đến một khía cạnh cụ thể nào đó mà để cập đến nhiều khía cạnh khác nhau. Chẳng hạn như, trang Web nói về "Việt Nam" cũng có thể để cập đến cuộc đời và sự nghiệp của "Các danh nhân Việt Nam". Cho nên tồn tại sự trùng lặp thông tin giữa các trang Web, có nghĩa là một trang Web có thể liên quan đến nhiều nội dung khác nhau.

Bảng 7.1. Ví dụ phân cụm kết quả truy vấn "Việt Nam" (Số trang Web: 185)

| Các cụm | Số trang Web | Chủ đề liên quan |
|---------|--------------|------------------------|
| 1 | 54 | Lịch sử Việt Nam |
| 2 | 10 | Đất nước Việt Nam |
| 3 | 9 | Kinh tế Việt Nam |
| 4 | 7 | Các danh nhân Việt Nam |
| 5 | 7 | Đu lịch Việt Nam |
| ... | ... | ... |

Xuất phát từ những đặc điểm trên, nên việc phân cụm chỉ nên được thực hiện trên tập các trang Web của mỗi truy vấn trả về từ máy tìm kiếm. Sau đó kết quả sẽ được tổ chức lại cho người sử dụng. Thông thường, một máy tìm kiếm phục vụ hàng triệu truy vấn một ngày, cho nên việc phân phối CPU cũng như bộ nhớ cho mỗi truy vấn cần được rút ngắn tối đa. Vì vậy, việc phân cụm có thể được thực hiện trên một máy tách riêng, tại đó chỉ nhận các kết quả của máy tìm kiếm như đầu vào, tạo ra các cụm và biểu diễn chúng cho người sử dụng.

Bảng 7.1 trình bày ví dụ về phân cụm kết quả của truy vấn "Việt Nam", ở đây chỉ đưa ra 5 cụm đầu tiên.

7.1.6. Các yêu cầu đối với phân cụm Web

Để có thể phân các trang Web thành các cụm, việc đầu tiên là cần phải tính được độ tương tự (hay độ tương đồng) giữa các trang Web trên cơ sở biểu diễn trang Web và xem xét các do độ tương tự giữa chúng. Thuật toán phân cụm cần đưa ra các điều kiện dùng và gắn nhãn cho các cụm một cách thích hợp nhất. Căn cứ đặc điểm và yêu cầu của bài toán phân cụm Web, phương pháp phân cụm được lựa chọn cần đáp ứng được các yêu cầu sau:

- *Tính phù hợp*: Phương pháp phải tạo nên các cụm, trong đó nhóm trang Web phù hợp với truy vấn của người dùng tách riêng với các nhóm không phù hợp khác.

- *Tóm tắt phải đọc*: Tránh trường hợp thay vì người dùng không phải xem xét danh sách các trang Web được phân cụm lại phải xem xét danh sách trang Web trong một cụm. Do đó, phương pháp phải cung cấp mô tả ngắn gọn và chính xác của các cụm.

- *Tính đa hình (morphology)*: Vì các trang Web có nhiều chủ đề, nên tránh việc hạn chế một trang Web chỉ thuộc về một cụm.

- *Sử dụng các mẫu thông tin*: Phương pháp phải tạo ra các cụm tốt, thậm chí chỉ sử dụng các mẫu thông tin được trả về bởi máy tìm kiếm (thông thường các máy tìm kiếm chỉ trả về các mẫu thông tin mô tả về trang Web). Điều này tránh cho việc người dùng phải chờ đợi hệ thống tải toàn bộ trang Web gốc từ Web.

- *Tốc độ*: Một người sử dụng dù kiên nhẫn cũng chỉ có thể xem xét khoảng 100 trang Web trong danh sách các trang Web được phân hạng. Hệ thống cần cho phép người dùng có thể đọc qua một tập đủ lớn các trang Web trong một thời gian chấp nhận được. Vì vậy, cần một phương pháp phân cụm khoảng 1000 mẫu thông tin trong vài giây.

- *Tính gia tăng*: Để tiết kiệm thời gian, phương pháp nên xử lý từng mẫu thông tin ngay khi lấy được từ Web để có được kết quả tức thời ứng với mỗi thời điểm.

7.1.7. Độ đo tương tự giữa các trang Web

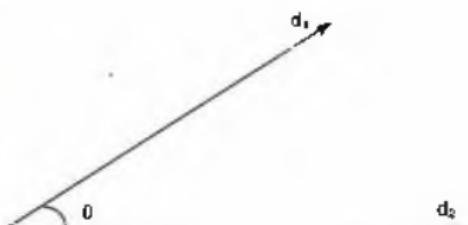
• *Biểu diễn các trang Web*: Các thuật toán phân cụm cũng sử dụng mô hình không gian vector để biểu diễn các trang Web. Giá trị tập từ khoá (từ vựng) là V và kích thước của tập V là N, thì mỗi tài liệu d sẽ được biểu diễn bằng một vector (a_1, a_2, \dots, a_N). Trong mô hình vector nhị phân, giá trị của a_i có thể là 1 hoặc 0 thể hiện rằng, từ khoá thứ i có xuất hiện trong trang Web d hay không. Trong trường hợp tổng quát, giá trị của a_i có thể là một số thực, ví dụ: $a_i = \text{TF} - \text{IDF}(t_i)$, trong đó t_i là từ khoá thứ i . Dựa trên mô hình không gian vector này, một số độ đo tương tự đã được định nghĩa.

• *Độ trùng lặp*: Độ trùng lặp dùng để đo độ tương tự của một trang Web này với trang Web khác hay với một truy vấn. Cách trực tiếp nhất là đo phần giao nhau của các đặc trưng tương ứng, ở đây là trùng lặp của các từ khoá. Đại lượng này cũng được gọi là mức kết hợp (coordination level). Gọi $T(d)$ là tập hợp các từ khoá có trong trang Web d. Độ trùng lặp của hai trang Web d_1 và d_2 là:

$$\text{CoordLevel}(d_1, d_2) = \frac{|T(d_1) \cap T(d_2)|}{|T(d_1) \cup T(d_2)|} \quad (7.1)$$

• **Dộ tương tự cosin:** Một phương pháp khác có thể được sử dụng để đo độ tương tự giữa các trang Web là độ tương tự cosin. Kỹ thuật cosin là một kỹ thuật (hay một phương pháp tính) được bắt nguồn từ tính toán vector. Trong thu nhận thông tin, công thức tính toán cosin được sử dụng để chỉ ra (đánh giá) mức độ tương tự giữa hai trang Web hoặc giữa trang Web và truy vấn, xem minh họa Hình 7.1.

Hai vector d_1 và d_2 càng gần nhau khi góc θ càng nhỏ, hay cosin của góc đó càng lớn. Có thể dùng giá trị cosin của góc θ làm độ tương tự của hai vector, trong đó cosin của góc giữa hai vector d_1 và d_2 được xác định như sau:



Hình 7.1. Độ đo tương tự cosin

$$\text{sim}(d_1, d_2) = \cos \theta = \frac{d_1 \cdot d_2}{\|d_1\| \|d_2\|} \quad (7.2)$$

7.2. Thuật toán phân cụm k-means

7.2.1. Thuật toán k-means với gán "cứng"

Giả sử đã biết trước số lượng các cụm cần phải tạo ra, giải pháp phân các trang Web vào các cụm là ý tưởng của thuật toán k-means. Thuật toán k-means có thể xếp vào lớp thuật toán phân cụm phẳng và đã được sử dụng trong nhiều thập kỷ. Ý tưởng chính là biểu diễn một cụm bằng trọng tâm của các trang Web nằm trong cụm đó. Việc quyết định phân một trang Web vào một cụm là dựa vào độ tương đồng của trang Web đó với trọng tâm của các cụm. Tồn tại hai dạng của thuật toán k-means là dạng cứng và dạng mềm. Dạng "cứng" ánh xạ trang Web tới các cụm theo một trong hai giá trị 0 hoặc 1, dạng "mềm" ánh xạ trang Web tới các cụm theo một giá trị trong khoảng 0 và 1.

Trong dạng tổng quát, thuật toán k-means sử dụng các biểu diễn nội tại cho các đối tượng được phân cụm và chính các cụm. Sử dụng phương pháp biểu diễn vector cho trang Web. Trong thuật toán này, dùng vector đại diện (thường chọn vector trọng tâm của tập các vector thuộc cụm) để thể hiện cho cụm, theo đó, cụm thứ i (ký hiệu là S_i) với vector đại diện d_i sẽ được mô tả như sau:

$$S_i = \{d \in S \mid \text{sim}(d, d_i) \leq \text{sim}(d, d_j) \forall j \neq i\} \quad (7.3)$$

trong đó $\text{sim}(u, v)$ là giá trị hàm khoảng cách giữa hai vector u và v . Nếu có yêu cầu về mỗi trang Web chỉ thuộc vào một cụm, thì trong trường hợp khoảng cách giữa vector trang Web tới vector đại diện một số cụm là như nhau, ta có thể gán trang Web vào cụm có chi số bé nhất. Đầu vào của giải thuật k-means là tập trang Web cần phân cụm S và số lượng các cụm cần phân chia k . Giải thuật hoạt động theo các bước sau:

Bước 1: Chọn ngẫu nhiên k trang Web trong S làm trọng tâm cho các cụm.

Bước 2: Phân các trang Web còn lại trong S vào các cụm dựa vào độ tương đồng của từng trang Web với trọng tâm của các cụm. trang Web sẽ được phân vào cụm i nếu độ tương đồng của nó với trọng tâm

$$c_i = \frac{1}{\|S_i\|} \sum_{d \in S_i} d \text{ của cụm thứ } i \text{ là lớn nhất, trong đó } S_i \text{ là tập hợp các}$$

trang Web có trong cụm i .

Bước 3: Tính lại trọng tâm của các cụm khi đã phân các trang Web mới.

Bước 4: Nhảy đến bước 2 cho đến khi quá trình hội tụ (không có sự phân chia lại các trang Web giữa các cụm, hay trọng tâm của các cụm là không đổi).

Trọng tâm c_i của cụm S_i là một vector $(c_{i1}, c_{i2}, \dots, c_{iN})$, trong đó $c_{ij} = \frac{1}{\|S_i\|} \sum_{d_j \in S_i} d_j$, và d_j là giá trị thứ j trong vector biểu diễn trang Web d .

Điểm mấu chốt của giải thuật là ở bước 2, các trang Web được di chuyển giữa các cụm để làm cực đại hóa độ tương tự giữa các trang Web bên trong một cụm (hay cực đại hóa độ tương tự trong nội tại một cụm, hay cực tiểu hóa khoảng cách giữa các trang Web trong nội tại một cụm). Độ đo tương tự trong nội tại một cụm được tính bằng công thức:

$$J = \sum_{i=1}^k \sum_{d_j \in S_i} \text{sim}(c_i, d_j) \quad (7.4)$$

Trong đó, S_i và c_i lần lượt là tập hợp các trang Web và trọng tâm của cụm i ; $\text{sim}(c_i, d_j)$ là độ cosin giữa c_i và d_j . Giải thuật k-means trả về số lượng biến thể các cụm là tối thiểu, nhưng nó không đảm bảo tìm được giá trị cực đại toàn cục của hàm J , nhưng có thể chạy thuật toán một số lần để thu được giá trị cực đại cục bộ. Kết quả cuối cùng của k-means phụ thuộc rất nhiều vào cách lựa chọn k trang Web ban đầu làm trọng tâm của k cụm. Bởi vì, sự lựa chọn k trang Web ban đầu là hoàn toàn ngẫu nhiên, nên kết quả thu được sau khi chạy k-means các lần khác nhau có thể khác nhau. Như vậy, có thể chạy thuật toán k-means một số lần và lấy kết quả của lần chạy có giá trị của hàm J là lớn nhất.

Trong thực tế, khi gặp trường hợp dữ liệu quá lớn, hoặc giải thuật không hội tụ (trọng tâm của các cụm cứ liên tục thay đổi) dẫn đến thời gian

chạy chương trình có thể rất lớn. Trong trường hợp này, người ta có thể sử dụng một số điều kiện dừng sau đây:

– Khi số lượng vòng lặp vượt qua một ngưỡng nào đó. Điều kiện này có thể làm cho chất lượng của giải thuật phân cụm không được tốt, vì nó chưa chạy đủ số vòng lặp cần thiết.

– Khi giá trị của J nhỏ hơn một ngưỡng nào đó (đảm bảo chất lượng của các cụm dù tốt, hay nó đã chạy được đủ số vòng lặp cần thiết). Trong thực tế điều kiện này thường được dùng kết hợp với điều kiện số vòng lặp ở trên.

Bảng 7.2. Dữ liệu mẫu dành cho phân cụm phẳng

| Tên trang Web | A1 | A2 | A3 | A4 | A5 | A6 |
|-------------------|-------|-------|-------|-------|-------|-------|
| Anthropology | 0 | 0.537 | 0.477 | 0 | 0.673 | 0.177 |
| Art | 0 | 0 | 0 | 0.961 | 0.195 | 0.196 |
| Biology | 0 | 0.347 | 0.924 | 0 | 0.111 | 0.112 |
| Chemistry | 0 | 0.975 | 0 | 0 | 0.155 | 0.158 |
| Communication | 0 | 0 | 0 | 0.78 | 0.626 | 0 |
| Computer Science | 0 | 0.989 | 0 | 0 | 0.13 | 0.067 |
| Criminal Justice | 0 | 0 | 0 | 0 | 1 | 0 |
| Economics | 0 | 0 | 1 | 0 | 0 | 0 |
| English | 0 | 0 | 0 | 0.98 | 0 | 0.199 |
| Geography | 0 | 0.849 | 0 | 0 | 0.528 | 0 |
| History | 0.991 | 0 | 0 | 0.135 | 0 | 0 |
| Mathematics | 0 | 0.616 | 0.549 | 0.49 | 0.198 | 0.201 |
| Modern Languages | 0 | 0 | 0 | 0.928 | 0 | 0.373 |
| Music | 0.97 | 0 | 0 | 0 | 0.17 | 0.172 |
| Philosophy | 0.741 | 0 | 0 | 0.658 | 0 | 0.136 |
| Physics | 0 | 0 | 0.894 | 0 | 0.315 | 0.318 |
| Political Science | 0 | 0.933 | 0.348 | 0 | 0.062 | 0.063 |
| Psychology | 0 | 0 | 0.852 | 0.387 | 0.313 | 0.162 |
| Sociology | 0 | 0 | 0.639 | 0.57 | 0.459 | 0.237 |
| Theatre | 0 | 0 | 0 | 0 | 0.967 | 0.254 |

– Khi hiệu của giá trị của J trong hai vòng lặp liên tiếp ($J_i - J_{i+1}$) nhỏ hơn một ngưỡng nào đó. Người ta cũng hay kết hợp điều kiện này với điều kiện vòng lặp để tránh chương trình bị chạy lặp.

Bảng 7.2 liệt kê tập dữ liệu dùng để minh họa cho thuật toán k-means, trong đó A1, A2, ..., A6 là giá trị của 6 đặc trưng (feature) của các trang Web này được tính theo TFIDF. Dùng tập dữ liệu này và dùng thuật toán k-means để phân thành 2 cụm ký hiệu là A và B. Vì chất lượng của giải thuật k-means phụ thuộc vào việc lựa chọn k trang Web làm trọng tâm của k cụm ban đầu, nên các lần chạy khác nhau của k-means sẽ cho kết quả khác nhau. Bảng 7.3 đưa ra kết quả của giải thuật k-means, trong đó việc lựa chọn k trọng tâm ban đầu là không được tốt (ta có thể xem giá trị của J ở vòng lặp đầu tiên). Ngược lại, Bảng 7.4 đưa ra kết quả của lần chạy khác của giải thuật k-means, trong đó sự lựa chọn k trọng tâm ban đầu cho kết quả tốt hơn so với kết quả ở trong Bảng 7.3.

Bảng 7.3. Kết quả của k-means với việc lựa chọn k trang Web ban đầu làm trọng tâm của các cụm không tốt

| Vòng lặp | Cụm A | Cụm B | Giá trị của J |
|----------|---|---|---|
| 1 | Computer Science, Political Science | Anthropology, Art, Biology, Chemistry, Communication, Criminal Justice, Economics, English, Geography, History, Mathematics, Modern Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre | 1 93554 (cụm A) + 4.54975 (cụm B) = 6.48529 |
| 2 | Chemistry, Computer Science, Geography, Political Science | Anthropology, Art, Biology, Communication, Criminal Justice, Economics, English, History, Mathematics, Modern Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre | 3 82736 (cụm A) + 10 073 (cụm B) = 13 9003 |
| 3 | Anthropology, Chemistry, Computer Science, Geography, Political Science | Art, Biology, Communication, Criminal Justice, Economics, English, History, Mathematics, Modern Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre | 4 60125 (cụm A) + 9 51446 (cụm B) = 14 1157 |

Bảng 7.4. Kết quả của k-means với việc lựa chọn k (trang Web ban đầu làm trọng tâm của các cụm tốt hơn)

| Vòng lặp | Cụm A | Cụm B | Giá trị của J |
|----------|--|---|---|
| 1 | Anthropology, Biology, Economics, Mathematics, Physics, Political Science, Psychology | Art, Chemistry, Communication, Computer Science, Criminal Justice, English, Geography, History, Modern Languages, Music, Philosophy, Sociology, Theatre | 5.04527 (cụm A) + 5.99025 (cụm B) = 11.0355 |
| 2 | Anthropology, Biology, Computer Science, Economics, Mathematics, Physics, Political Science, Psychology, Sociology | Art, Chemistry, Communication, Criminal Justice, English, Geography, History, Modern Languages, Music, Philosophy, Theatre | 7.23827 (cụm A) + 6.70864 (cụm B) = 13.9469 |
| 3 | Anthropology, Biology, Chemistry, Computer Science, Economics, Geography, Mathematics, Physics, Political Science, Psychology, Sociology | Art, Communication, Criminal Justice, English, History, Modern Languages, Music, Philosophy, Theatre | 8.53381 (cụm A) + 6.12743 (cụm B) = 14.6612 |

7.2.2. Thuật toán k-means với gán "mềm"

Thay vì chỉ rõ việc gán các trang Web cho các cụm, dạng "mềm" của k-means biểu diễn mỗi cụm c sử dụng một vector μ_c trong không gian. Do không có một sự rõ ràng trong việc gán các trang Web cho các cụm, μ_c không trực tiếp liên hệ với các trang Web, ví dụ nó không cần thiết là trọng tâm của các trang Web. Mục đích của k-means "mềm" là tìm một μ_c cho mỗi cụm c để tối thiểu hóa lỗi lượng từ $\sum_i \min_c |d - \mu_c|^2$. Một chiến lược đơn giản để giám lỗi là đưa ra các vector trung bình là khoảng cách từ các trang Web đến cụm gần nhất. Ta sẽ lặp lại việc quét qua các trang Web và với mỗi trang Web d, tích lũy một $\Delta\mu_c$ cho cụm μ_c gần d nhất:

$$\Delta\mu_c = \sum_j \begin{cases} \eta(d - \mu_c) & \text{nếu } \mu_c \text{ gần } d \text{ nhất;} \\ 0 & \text{các trường hợp khác.} \end{cases} \quad (7.5)$$

Sau khi quét một lần qua tất cả các trang Web, tất cả các μ_c được cập nhật đồng loạt bởi công thức $\mu_c \leftarrow \mu_c + \Delta\mu_c$, trong đó η được gọi là learning rate. Nó duy trì một số dữ liệu của quá khứ và làm ổn định hệ thống. Chú ý, mỗi trang Web d chỉ chuyên vào một μ_c trong mỗi đợt. Việc phân bổ trang

Web d không bị giới hạn đến chỉ một μ_c mà gần nó nhất. Việc phân bổ có thể được chia sẻ giữa nhiều trang Web, việc phân chia cho cụm c quan hệ trực tiếp đến độ tương tự hiện thời giữa μ_c và d. Ví dụ, có thể làm mềm công thức tính $\Delta\mu_c$ ở trên như sau:

$$\Delta\mu_c = \eta \frac{1/|d - \mu_c|^2}{\sum_i 1/|d - \mu_i|^2} (d - \mu_c) \quad (7.6)$$

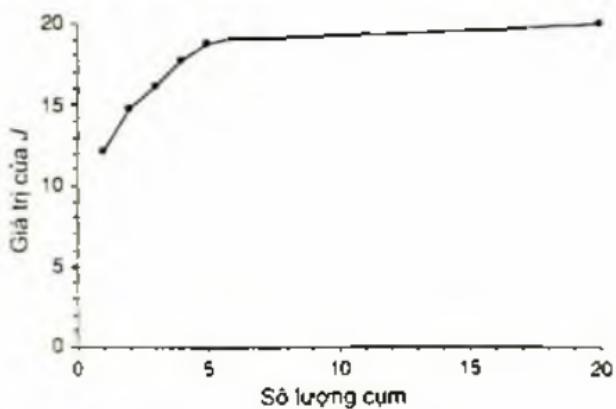
hoặc

$$\Delta\mu_c = \eta \frac{\exp(-|d - \mu_c|^2)}{\sum_i \exp(-|d - \mu_i|^2)} (d - \mu_c) \quad (7.7)$$

Tồn tại nhiều quy tắc cập nhật khác có thể được sử dụng. Gán "mềm" không làm mất đi liên kết chặt trong việc tạo nên phân bố các trang Web cho một cụm đơn đặt được một cách tý mè. Có rất nhiều loại k-means "mềm" là trường hợp đặc biệt của thuật toán Expectation Maximization - EM [DLR77], những thuật toán có thể được chứng minh là hội tụ cục bộ tối ưu.

7.2.3. Tìm số lượng cụm thích hợp

Giải thuật k-means trình bày ở trên cần xác định số lượng các cụm cố định từ trước. Tuy nhiên, trong nhiều trường hợp ta không thể biết trước được số lượng cụm như thế nào là cho chất lượng tốt nhất. Do vậy, rất hữu ích nếu giải thuật cung cấp cho số lượng các cụm như thế nào là tối ưu nhất. Dưới đây sẽ trình bày một giải pháp để tìm ra số cụm tối ưu chấp nhận được, dựa vào giá trị cực đại (có thể là cục bộ) của giá trị J.



Hình 7.2. Giải thuật k-means với các giá trị cụm khác nhau

Cho giải thuật k-means thực hiện với các tham số k (số lượng các cụm) khác nhau, giá trị k nào cho giá trị của J cao nhất thì đó là số cụm tối ưu.

Tuy nhiên, cũng phải cân đối với thời gian thực hiện của giải thuật. Hình 7.2 diễn tả giá trị của J tương ứng với số lượng các cụm khác nhau.

7.3. Thuật toán phân cụm phân cấp từ dưới lên

Mặc dù có nhiều dạng thức liên quan tới phương pháp phân cụm từ dưới lên, song một tự duy nhất tự nhiên để tìm ra các cụm là: (1) bắt đầu mỗi trang Web được coi như một cụm (số lượng cụm bằng số lượng trang Web); (2) sau đó từng bước kết hợp các cụm đã có thành các cụm lớn hơn với yêu cầu phải đảm bảo độ tương tự giữa trang Web nội bộ trong mỗi cụm cao (số lượng cụm giảm dần); (3) ngừng lại khi hoặc đã đạt được số lượng cụm mong muốn, hoặc chỉ còn một cụm duy nhất chứa tất cả các trang Web hay thỏa mãn một điều kiện dừng nào đó.

Thuật toán phân cụm tích tụ từ dưới lên (Hierarchical agglomerative clustering – HAC) là thuật toán phân cụm được sử dụng rất rộng rãi và được tích hợp vào các ứng dụng thu thập thông tin [ZDR07]. HAC chỉ yêu cầu định nghĩa hàm *khoảng cách* (hay *độ tương tự*) giữa các cụm/trang Web. Trong không gian vector, khoảng cách Euclidean là sự lựa chọn phù hợp nhất. Độ tương tự của 2 trang Web d_1, d_2 được định nghĩa là $\text{sim}(d_1, d_2) = \cos(d_1, d_2)$. Tiếp theo là tổng quát hóa độ đo này để tính độ tương tự giữa 2 cụm. Giả sử S_1 và S_2 là 2 cụm, có một số phương pháp tính độ tương tự giữa S_1 và S_2 là $\text{sim}(S_1, S_2)$ như sau:

– Độ tương tự giữa trọng tâm của S_1 và S_2 : $\text{sim}(S_1, S_2) = \text{sim}(c_1, c_2)$, trong đó c_1 và c_2 lần lượt là trọng tâm của hai cụm S_1 và S_2 .

– Độ tương tự cực đại giữa 2 tài liệu thuộc vào 2 cụm:

$$\text{sim}(S_1, S_2) = \max_{d_1 \in S_1, d_2 \in S_2} \text{sim}(d_1, d_2)$$

Giải thuật sử dụng độ đo này còn được gọi là phân cụm người láng giềng gần nhất, và độ đo này còn được gọi là single-link.

– Độ tương tự cực tiêu giữa 2 tài liệu thuộc vào 2 cụm:

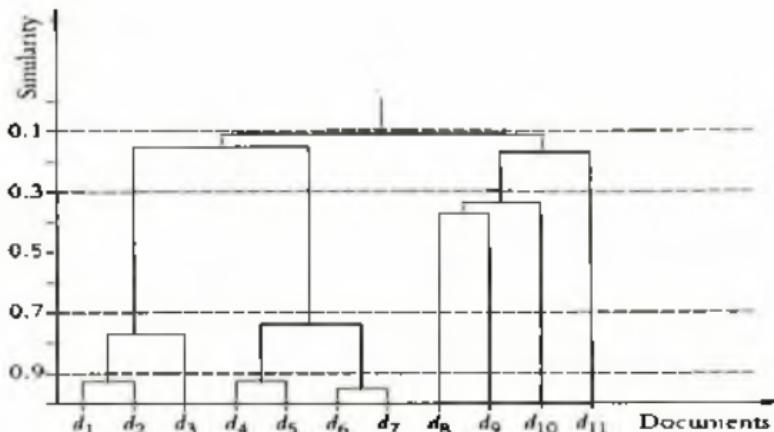
$$\text{sim}(S_1, S_2) = \min_{d_1 \in S_1, d_2 \in S_2} \text{sim}(d_1, d_2)$$

Giải thuật sử dụng độ đo này còn được gọi là phân cụm người láng giềng xa nhất, và độ đo này còn được gọi là complete-link.

– Độ tương tự trung bình giữa các tài liệu trong 2 cụm:

$$\text{sim}(S_1, S_2) = \frac{1}{|S_1||S_2|} \sum_{d_1 \in S_1, d_2 \in S_2} \text{sim}(d_1, d_2)$$

Độ đo này còn được gọi là group-average.



Hình 7.3. Một cây lược đồ phân cấp của thuật toán phân cụm HAC

Chú ý rằng, các độ đo trên cũng có thể áp dụng cho các trang Web, vì mỗi một trang Web d có thể coi là một cụm chỉ chứa mình nó ($\{d\}$). Tương tự như giải thuật phân cụm khác, mục đích của HAC là làm cực đại độ tương tự giữa các trang Web trong nội tại một cụm. Cũng giống như các thuật toán phân cụm khác, một trong các mục đích của giải thuật HAC là làm cực đại độ tương tự giữa các trang Web nội tại trong các cụm. Trong quá trình HAC hoạt động, các cụm được ghép lại với nhau tạo thành một cụm ở cấp cao hơn, độ tương tự nội tại của các cụm mới này sẽ giảm so với các cụm ở cấp thấp hơn trong cây phân cấp (Hình 7.3). Như vậy, để đạt được chất lượng phân cụm tổng thể tốt, có thể dừng quá trình ghép cụm ở một mức nào đó chứ không bắt buộc phải tạo ra một cụm duy nhất ở gốc của cây phân cấp. Để cài đặt ý tưởng này, có thể sử dụng các tham số điều khiển. Tham số thứ nhất k để dừng thuật toán là khi số lượng cụm mong muốn đã được tạo ra, tham số thứ hai q là dừng thuật toán khi độ tương tự giữa hai cụm được chọn để ghép nhau hơn một ngưỡng nào đó. Gọi G là tập các cụm, S là tập hợp các trang Web cần phân cụm, thuật toán HAC được thể hiện như sau:

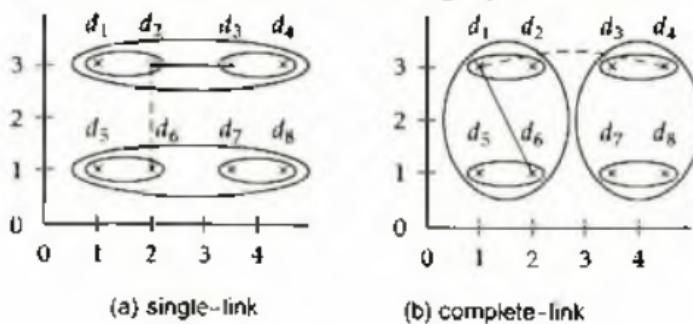
- 1 $G \leftarrow \{\{d\} \mid d \in S\}$ (khởi tạo G là tập các cụm chỉ gồm một trang Web trong tập S)
- 2 Nếu $|G| < k$ thì dừng thuật toán (đã đạt được số lượng cụm mong muốn)
- 3 Tìm 2 cụm $S_i, S_j \in G$ sao cho $(i, j) = \arg \max_{(i,j)} \text{sim}(S_i, S_j)$ (tim hai cụm có độ tương tự lớn nhất).
- 4 Nếu $\text{sim}(S_i, S_j) < q$ thì dừng thuật toán (độ tương tự của 2 cụm nhỏ hơn ngưỡng cho phép)
- 5 Loại bỏ S_i, S_j khỏi G
- 6 $G = G \cup \{S_i, S_j\}$ (ghép hai cụm S_i, S_j và đưa vào trong tập G)
- 7 Nhảy đến bước 2

Giải thuật có thể dừng tại bước 2 khi số lượng cụm k mong muốn đã thoả mãn, hay ở bước 4 khi độ tương tự lớn nhất giữa 2 cụm là nhỏ hơn ngưỡng q cho phép. Khi $k = 1$ và $q = 0$ thì G là cây phân cụm hoàn chỉnh có gốc là cụm duy nhất. Khi $k > 1$ thì có k cụm ở mức cao nhất. Một ví dụ về giải thuật phân cụm HAC là cây phân cấp ở Hình 7.3. Một điều đáng chú ý đối với thuật toán HAC là nó luôn tạo ra một cây nhị phân chứ không phải là một cây phân cấp tổng quát, vì khi ghép cụm nó chỉ ghép 2 cụm có độ tương tự nhau là lớn nhất.

Nhận xét về một số độ đo:

Với phân cụm dựa trên độ đo single-link, độ tương tự giữa 2 cụm được tính chính là độ tương tự lớn nhất giữa hai trang Web nằm trong 2 cụm (Hình 7.4a). Do đó, khi dùng độ đo này để quyết định ghép 2 cụm lại với nhau mang tính cục bộ. Vì khi ghép cụm, ta chỉ quan tâm đến những vùng dữ liệu mà ở đó có phần tử của 2 cụm gần nhau nhất, mà không cần quan tâm đến các phần tử khác trong cụm cũng như cấu trúc tổng thể của các cụm. Điều này sẽ làm cho chất lượng phân cụm của giải thuật có thể kém nếu có trường hợp chỉ có duy nhất 2 trang Web ở trong 2 cụm là gần nhau, còn các trang Web còn lại trong 2 cụm là ở rất xa nhau.

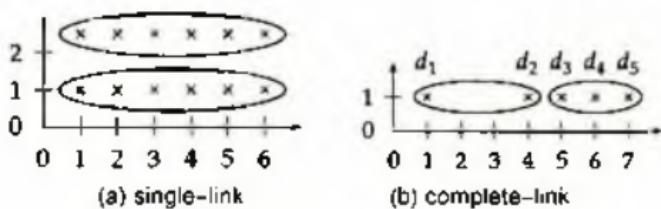
Với phân cụm dựa trên độ đo complete-link, độ tương tự của 2 cụm lại được lấy là độ tương tự của 2 trang Web nằm trong 2 cụm có giá trị nhỏ nhất (Hình 7.4b). Việc này tương đương với việc lựa chọn 2 cụm để ghép lại sẽ tạo ra cụm mới có đường kính nhỏ nhất. Điều kiện lựa chọn dùng để ghép 2 cụm này không mang tính cục bộ, vì cấu trúc toàn cục của các cụm được xem xét trong quá trình quyết định ghép cụm. Điều kiện này có ưu điểm là, luôn tạo ra các cụm "cô đọng" vì các cụm mới được tạo ra có bán kính nhỏ nhất. Cũng như phân cụm với single-link, giải thuật phân cụm với complete-link cũng có thể cho chất lượng kém khi có 2 trang Web trong 2 cụm ở rất xa nhau, trong khi trọng tâm của 2 cụm này lại rất gần nhau, khi đó 2 cụm này có thể không được lựa chọn để ghép lại với nhau.



Hình 7.4. Phân cụm với độ đo single-link và complete-link

Hình 7.4 minh họa phân cụm phân cấp HAC với độ đo single-link (a) và độ đo complete-link (b) trên 8 trang Web. Từ hình minh họa cho thấy 4 bước đầu tiên của cả 2 giải thuật đều tạo ra các cụm giống nhau. Bước thứ 5, giải thuật HAC với single-link sẽ ghép 2 cụm ở phía trên lại với nhau, và bước thứ 7 là ghép 2 cụm ở dưới lại. Trong khi đó, giải thuật HAC với complete-link lại ghép 2 cụm ở phía bên trái ở bước thứ 6 và ghép hai cụm phía bên phải lại ở bước thứ 7.

Cả hai độ đo single-link và complete-link đều đánh giá độ tương tự của 2 cụm dựa trên một cặp trang Web duy nhất, do đó giải thuật phân cụm sử dụng các độ đo này đều có khả năng tạo ra các cụm không mong muốn (không có chất lượng tốt). Hình 7.5a đưa ra ví dụ một trường hợp mà thuật toán HAC với độ đo single-link cho kết quả không mong muốn. Vì điều kiện ghép cụm của độ đo này là mang tính cục bộ mà không quan tâm đến hình dáng của cụm được tạo ra. Do đó, nó đã tạo ra một cụm có hình như một chuỗi (chain). Nếu để ý thì có thể nhận ra tình huống tạo chuỗi với độ đo single-link cũng xuất hiện ngay trong hình 7.4a. Nhưng giải thuật phân cụm HAC với độ đo complete-link với cùng tập dữ liệu này lại không tạo chuỗi (Hình 7.4b), do đó kết quả các cụm tạo ra trong trường hợp này là tốt hơn.



Hình 7.5. Trường hợp ghép cụm không lỗi của độ đo single-link vào complete-link

Còn giải thuật HAC với độ đo complete-link lại có nhược điểm khác, đó là khi ghép cụm lại với nhau nó lại quan tâm nhiều đến trường hợp ngoại lệ của 2 trang Web trong 2 cụm có độ tương tự nhau là thấp nhất mà không quan tâm đến các trang Web còn lại trong cụm, hay cấu trúc toàn cục của các cụm. Do đó, nó có thể tạo ra các cụm không mong muốn như minh họa trong hình 7.5b. Một cách trực quan, nếu ta quan tâm đến cấu trúc của dữ liệu thì kết quả phân cụm ở mức gần gốc nên là 2 cụm $\{d_1\}$ và $\{d_2, d_3, d_4, d_5\}$, thi tốt hơn nhiều so với 2 cụm $\{d_1, d_2\}$ và $\{d_3, d_4, d_5\}$.

Dộ đo group-average tính toán độ tương tự của 2 cụm dựa trên độ tương tự của toàn bộ các cặp trang Web trong 2 cụm, chứ không chỉ dựa trên một cặp trang Web duy nhất. Do đó, nó tránh được các trường hợp không mong muốn như 2 độ đo vừa thảo luận ở trên.

Dộ đo dựa vào trọng tâm cũng có đặc điểm là không dựa trên một cặp trang Web để quyết định độ tương tự của 2 cụm. Ở đây, giá trị của độ tương tự giữa 2 cụm chính là độ tương tự của trọng tâm của 2 cụm. Độ đo này tránh được một số nhược điểm của độ đo single-link và complete-link; tuy nhiên, nó cũng có nhược điểm là độ tương tự từ dưới lên trên cây phân cấp có thể là không giảm dần (do trọng tâm của các cụm ở mức cao có thể ở gần nhau hơn so với các cụm ở mức dưới). Điều này trái ngược với giả thiết cơ bản là các cụm nhỏ thường có độ kết định (coherent) cao hơn các cụm có kích thước lớn hơn.

7.4. Thuật toán phân hoạch từ trên xuống

Theo các nghiên cứu được công bố, kỹ thuật phân cụm *từ dưới lên* (bottom-up) được sử dụng trực tiếp tồn thời gian với độ phức tạp là $O(n^2)$ và không thích hợp cho các tập dữ liệu lớn. Nếu coi như đặt trước số cụm là k , kỹ thuật phân hoạch *từ trên xuống* (top-down) thường được sử dụng vì hiệu quả hơn. Một kỹ thuật đi theo hướng này là sử dụng thuật toán k-means. Thuật toán bắt đầu từ đỉnh của cây với chỉ có một cụm là toàn bộ các trang Web. Cụm này sẽ được phân chia ra thành các cụm nhỏ hơn bằng cách sử dụng thuật toán phân cụm phẳng (chẳng hạn như k-means). Với các cụm nhỏ ta lại áp dụng đệ quy thuật toán phân cụm phẳng. Về lý thuyết thì thuật toán phân cụm phân cấp từ trên xuống phức tạp hơn so với phương pháp phân cụm từ dưới lên vì ta gọi giải thuật phân cụm phẳng (như là một thủ tục) nhiều lần. Tuy nhiên, nó có ưu điểm trong trường hợp không cần thiết phải sinh ra một cây phân cấp hoàn chỉnh mà ở đó các cụm ở nút lá chỉ chứa đúng một trang Web. Khi giới hạn số lượng mức (level) của cây phân cấp và kết hợp sử dụng giải thuật phân cụm phẳng k-means, thuật toán phân cụm phân cấp từ trên xuống có độ phức tạp gần như là tuyến tính với số lượng các trang Web và số lượng các cụm. Do đó, thuật toán phân cụm từ trên xuống sẽ chạy nhanh hơn so với thuật toán phân cụm từ dưới lên.

Giải thuật phân cụm từ trên xuống được chứng minh là có độ chính xác cao hơn so với các giải thuật phân cụm từ dưới lên như HAC trong một số trường hợp. Lý do là giải thuật phân cụm từ dưới lên đưa ra quyết định ghép các cụm lại với nhau chỉ sử dụng các thông tin cục bộ (ở các cụm) mà không thể dựa trên thông tin toàn cục (tổng bộ tập dữ liệu). Các cụm sau khi ghép rồi thì không thể tách ra để ghép với các cụm khác. Ngược lại, các giải thuật phân cụm từ trên xuống ngay từ đầu đã khai thác được thông tin toàn cục (phân bố toàn cục của tập dữ liệu) khi quyết định phân dữ liệu đang xét thành các cụm nhỏ hơn.

7.5. Gán nhãn cho các cụm

Trong rất nhiều bài toán ứng dụng phân cụm (chẳng hạn như, ứng dụng phân cụm kết quả trả về của một máy tìm kiếm), ở đó người ta cần giao diện người sử dụng (hay tương tác với người dùng), việc gán nhãn cho các cụm là rất cần thiết để làm cho người sử dụng thuận tiện khi duyệt các cụm. Nói cách khác, người dùng sẽ có thông tin để quyết định sẽ xem kết quả ở cụm nào. Có rất nhiều phương pháp gán tên cho các cụm. Trước tiên sẽ tìm hiểu một số phương pháp gán nhãn cho các cụm trong phân cụm phẳng:

– *Gán nhãn dựa vào sự phân bố các từ khóa trong cụm này so với các cụm khác*: Phương pháp này có thể dựa trên phương pháp lựa chọn đặc trưng (feature). Dựa trên phân tích Mutual Information (MI), Information Gain, hay phân tích χ^2 -test [Christopher08], các từ khóa đặc trưng nhất cho các cụm sẽ được tìm ra sắp xếp theo thứ tự quan trọng giảm dần. Sau đó, có thể chọn một số từ khóa nằm ở đầu danh sách ra đặt tên cho cụm đó. Phương pháp này có đặc điểm là nhãn của cụm chỉ đơn thuần là danh sách một số từ khoá, chứ không phải là một cụm từ hay một câu có nghĩa. Và cũng có khả năng, một từ khoá tuy có trọng số cao được lựa chọn, nhưng nó không phản ánh dung nội dung của một cụm.

– *Gán nhãn dựa vào thông tin nội bộ cụm*: Phương pháp này gán nhãn cho một cụm chỉ dựa trên các trang Web trong nội bộ một cụm chứ không quan tâm đến các cụm khác. Nhãn của cụm sẽ là tiêu đề (title) của trang Web gần tâm của cụm nhất. Phương pháp này có ưu điểm là nhãn của một cụm thường là một cụm từ (hay một câu) có nghĩa, chứ không đơn thuần là danh sách một số từ khoá, do đó người dùng sẽ dễ đọc hơn. Ngoài việc lấy tiêu đề của trang Web ra làm nhãn cho một cụm, có thể lấy *nhãn liên kết* (anchor text) là đoạn văn bản nằm trong thẻ liên kết: "[nhãn liên kết](...)" đến trang Web gần trọng tâm của cụm nhất, vì nội dung của đoạn văn bản này cũng cung cấp nhiều thông tin quan trọng không kém gì tiêu đề của trang Web đó. Tuy nhiên, phương pháp gán nhãn này cũng có hạn chế, đó là việc chọn một tài liệu để đại diện cho một cụm có thể sẽ không chính xác trong một số trường hợp.

– *Chọn các từ khoá có trọng số (tần suất) cao ở gần trọng tâm của cụm làm nhãn của cụm*: Phương pháp này cũng không quan tâm đến nội dung của các cụm khác mà chỉ quan tâm đến bản thân nội dung của cụm. Người ta chọn một số từ khoá có trọng số cao nằm ở trọng tâm của nội tại một cụm làm nhãn cho cụm. Thậm chí có thể chọn đơn vị là các *cụm từ* chứ không phải là một từ đơn. Những từ khoá này sẽ phản ánh dung hơn nội dung của một cụm. Tuy nhiên, việc trích chọn các cụm từ sẽ tồn thời gian hơn so với việc trích chọn các từ khoá đơn. Phương pháp này rất hiệu quả, tuy nhiên

việc lựa chọn các từ khoá có tần suất cao trong một cụm mà không quan tâm đến nội dung của các cụm khác sẽ là việc lựa chọn mang tính cục bộ. Do đó, một số từ có tần suất cao có thể lại không mang nhiều thông tin đại diện cho cụm đó cũng giống như trường hợp sử dụng phương pháp gán nhãn thứ nhất. Chẳng hạn như, từ khoá "Tuesday" hay "year" có thể được lựa chọn đối với phương pháp này, tuy nhiên nó lại không thể hiện được nội dung của cụm đó.

Bảng 7.5 đưa ra kết quả phân cụm phẳng bằng giải thuật k-means và dùng các phương pháp khác nhau để gán nhãn cho ba trong số các cụm được tạo ra (các cụm số 4, 9 và 10). Trong bảng này cũng đưa ra một số ưu điểm cũng như nhược điểm của cả các phương pháp gán nhãn cho các cụm. Với phương pháp gán nhãn dựa vào Mutual Information chỉ liệt kê một danh sách các từ khoá, còn phương pháp gán nhãn dựa vào tiêu đề luôn cho một câu có nghĩa. Tuy nhiên, ở trường hợp cụm thứ 4, phương pháp lựa chọn tiêu đề cho kết quả không chính xác. Lý do là, cụm thứ 4 chủ yếu là nói về dầu mỏ, nhưng phương pháp gán nhãn dựa vào tiêu đề lại đề cập đến một nội dung hoàn toàn khác. Phương pháp gán nhãn dựa vào trọng tâm (phương pháp thứ 3) và dựa vào lựa chọn đặc trưng (phương pháp thứ nhất) đều đưa ra một số từ khoá không chính xác (tức là nó không phản ánh đúng nội dung của cụm). Và phương pháp thứ 3 sản sinh ra nhiều từ khoá không chính xác hơn.

Bảng 7.5. Gán nhãn cho cụm dùng các phương pháp khác nhau

| Cụm | Số trang Web trong cụm | Phương pháp dựa vào trọng tâm | Phương pháp Mutual information | Phương pháp chọn tiêu đề |
|-----|------------------------|--|--|---|
| 4 | 622 | oil plant mexico production crude power 000 refinery gas bpd | plant oil production barrels crude bpd mexico dolly capacity petroleum | MEXICO Hurricane Dolly heads for Mexico coast |
| 9 | 1017 | police security russian people military peace killed told grozny court | police killed military security peace told troops forces rebels people | RUSSIA Russia's Lebed meets rebel chief in Chechnya |
| 10 | 1259 | 00 000 tonnes traders futures wheat prices cents september tonne | delivery traders futures tonne tonnes desk wheal prices 000 00 | USA Export Business - Grain/oilseeds complex |

Đối với phân cụm phân cấp, việc gán nhãn cho cụm có phần phức tạp hơn. Lý do là, ngoài việc phân biệt nhãn cho các cụm con tại một nút trong cây phân cấp, còn phải phân biệt các nhãn đó với các nhãn ở các cụm cha và cụm con của nó. Các cụm ở nút con ngầm định là thành viên của cụm ở nút cha của nó, do đó không thể sử dụng các phương pháp đơn giản để tìm ra nhãn phân biệt nút cha với nút con.

7.6. Đánh giá thuật toán phân cụm

7.6.1. Nhận xét sơ bộ

Như đã được giới thiệu, thuật toán HAC thường chậm khi áp dụng cho các tập trang Web lớn. Các thuật toán khác theo hướng này như *Single-link* và *Group-average* có thời gian thực hiện là $O(n^2)$, đồng thời thời gian kết nối hoàn toàn (*complete-link*) là $O(n^3)$ [Christopher08]. Các thuật toán theo hướng này là quá chậm so với yêu cầu của bài toán phân cụm Web. Một điểm đáng chú ý nữa đối với các thuật toán HAC là điều kiện dừng. Đã có rất nhiều đề xuất về điều kiện dừng được đưa ra, nhưng chủ yếu là dựa trên việc điều kiện dừng đã được xác định trước (chẳng hạn, dừng khi chỉ còn 5 cụm). Điều kiện dừng đối với các thuật toán này (HAC) là cực kỳ quan trọng. Nếu như thuật toán ghép các cụm "tốt" với nhau có thể tạo ra kết quả không theo mong muốn của người dùng. Trên Web, với kết quả tra về theo truy vấn là vô cùng đa dạng (về số lượng, độ lớn, kiểu và sự phù hợp của các trang Web) thì điều kiện dừng không tốt sẽ làm cho kết quả trở nên nghèo nàn.

Thuật toán k-means thuộc vào lớp các thuật toán phân cụm thời gian tuyến tính và là những lựa chọn tốt nhất để đáp ứng yêu cầu về tốc độ của bài toán phân cụm on-line. Thời gian thực hiện của các thuật toán này là $O(nk)$, trong đó k là số các cụm mong muốn.Thêm một ưu điểm của thuật toán k-means so với HAC là việc đáp ứng các yêu cầu của bài toán phân cụm Web là nó có thể tạo ra các cụm có sự giao thoa. Điểm yếu chính của thuật toán này là nó chạy hiệu quả nhất chỉ khi các cụm mong muốn là các miền hình cầu đối với độ đo tương tự được dùng. Không có lý do gì để tin rằng, các trang Web sẽ thuộc vào các miền cầu. Vì vậy, thuật toán có thể làm mất đi các thông tin có giá trị.

Buckshot là thuật toán kết hợp giữa HAC và k-means, trong đó việc khai tạo các trọng tâm cụm cho k-means được thực hiện bởi thuật toán HAC trên một mẫu của tập trang Web [Cutting93].

Các thuật toán như HAC, k-means hay Buckshot đều không phải là các thuật toán có tính gia tăng. Một số thuật toán gia tăng đã được phát triển như

thuật toán phân cụm cây hậu tố (Suffix Tree Clustering – STC), với thời gian thực hiện là $O(n)$, trong đó n là kích thước của tập trang Web.

7.6.2. Đánh giá dựa trên độ tương tự

Các thuật toán phân cụm nhóm các trang Web lại với nhau thành cụm dựa vào độ tương tự tính toán trên không gian vector (dùng để biểu diễn các trang Web). Do đó, có thể dùng bất kỳ hàm đánh giá nào liên quan đến độ tương tự của các trang Web trong cùng một cụm. Trong một số trường hợp, muốn đánh giá sự khác nhau giữa các tài liệu trong 2 cụm khác nhau. Bàn thân các hàm dựa trên độ tương tự (hàm J) cũng đã được tích hợp vào trong một số giải thuật phân cụm. Dưới đây sẽ trình bày hàm đánh giá dựa trên độ tương tự là *tổng bình phương lỗi*.

Hàm tổng bình phương lỗi: Với mỗi cụm S_i , xác định trọng tâm c_i của cụm đó. Ý tưởng của hàm đánh giá này là dựa trên quan điểm trọng tâm của mỗi cụm sẽ biểu diễn tốt nhất cụm đó, với mỗi trang Web d trong cụm đó càng cách xa trọng tâm của cụm thì "lỗi" của trang Web đó càng cao. Giá trị lỗi của trang Web d trong cụm được đo bằng chiều dài của vector $d - c_i$. Hàm đánh giá chất lượng phân cụm này được tính bằng:

$$J_c = \sum_{i=1}^k \sum_{d \in S_i} \|d - c_i\|^2 \quad (7.8)$$

Trong đó k là tổng số cụm, c_i là trọng tâm của cụm S_i , $c_i = \frac{1}{|S_i|} \sum_{d \in S_i} d$.

Do đó, giá trị J_c của một giải thuật nào đó càng nhỏ thì chất lượng phân cụm của nó càng tối. Bằng cách biến đổi số học, công thức (7.8) trên có thể được viết lại thành tổng khoảng cách từng cặp trang Web trong cụm:

$$J_v = \frac{1}{2} \sum_{i=1}^k \frac{1}{|S_i|} \sum_{d_j, d_l \in S_i} \|d_j - d_l\|^2 \quad (7.9)$$

Trong các giải thuật phân cụm Web, ta sử dụng độ đo tương tự, do đó công thức tương đương dùng để đánh giá chất lượng của giải thuật phân cụm Web sẽ là:

$$J_s = \frac{1}{2} \sum_{i=1}^k \frac{1}{|S_i|} \sum_{d_j, d_l \in S_i} \text{sim}(d_j, d_l) \quad (7.10)$$

Biến đổi tương đương công thức trên có thể được viết lại thành:

$$J_s = \frac{1}{2} \sum_{i=1}^k \frac{1}{|S_i|} \sum_{d_j, d_l \in S_i} \text{sim}(d_j, d_l) = \frac{1}{2} \sum_{i=1}^k |S_i| \text{sim}(S_i) \quad (7.11)$$

Trong đó $\text{sim}(S_i)$ là độ tương tự trung bình giữa các cặp trang Web trong cụm S_i . Với công thức này, giá trị của J , càng lớn thì càng chứng tỏ giải thuật phân cụm cho chất lượng càng tốt.

Tóm lại, các hàm đánh giá dựa trên độ tương tự có 2 dạng tương đương nhau: độ tương tự so với trọng tâm của cụm và độ tương tự giữa các cặp trang Web trong cùng một cụm. Bảng 7.6 đưa ra một số giá trị của hàm J , (giá trị nằm ở trong cặp ngoặc vuông) cho giải thuật phân cụm k-means đối với tập dữ liệu liệt kê trong Bảng 7.2.

Bảng 7.6. Giá trị của hàm đánh giá dựa trên độ đo tương tự với giải thuật k-means

| $k = 2$ | $k = 3$ | $k = 4$ |
|--|--|--|
| 1 [8.53381] Anthropology, Biology, Chemistry, Computer Science, Economics, Geography, Mathematics, Physics, Political Science, Psychology, Sociology 2 [6.12743] Art, Communication, Criminal Justice, English, History, Modern Languages, Music, Philosophy, Theatre $\Sigma = [12.0253]$ | 1 [2.83806] History, Music, Philosophy 2 [6.09107] Anthropology, Biology, Chemistry, Computer Science, Geography, Mathematics, Political Science 3 [7.12119] Art, Communication, Criminal Justice, Economics, English, Modern Languages, Physics, Psychology, Sociology, Theatre $\Sigma = [12.0253]$ | 1 [3.81771] Art, Communication, English, Modern Languages 2 [5.44416] Biology, Economics, Mathematics, Physics, Psychology, Sociology 3 [2.83806] History, Music, Philosophy 4 [5.64819] Anthropology, Chemistry, Computer Science, Criminal Justice, Geography, Political Science, Theatre $\Sigma = [12.0253]$ |

7.6.3. Đánh giá dựa trên dữ liệu gán nhãn

- *Phương pháp đánh giá dựa vào độ chính xác và tỷ lệ lỗi*

Phương pháp đánh giá ở mục 7.6.2 hoàn toàn dựa vào độ tương tự của các trang Web trong cùng một cụm. Tuy nhiên, khi phân thủ công các trang Web vào các cụm, người ta cần thêm một số tri thức khác nữa mà thông thường các tri thức này không có sẵn hay hiển thị rõ ràng trong nội dung của các trang Web. Do vậy, việc đánh giá thuật toán phân cụm chỉ dựa vào hàm điều kiện J như trên là không chính xác. Mục này sẽ tìm hiểu thêm một số phương pháp đánh giá các giải thuật phân cụm một cách chính xác hơn. Thông thường, dữ liệu gán nhãn thường được dùng để áp dụng cho các giải thuật học có giám sát, tuy nhiên, ngay cả giải thuật học không giám sát như các giải thuật phân cụm thì dữ liệu gán nhãn cũng hữu ích, cụ thể ta có thể

dùng để đánh giá chất lượng của giải thuật phân cụm bằng cách so sánh dữ liệu gán nhãn (dữ liệu phân cụm bằng tay) với kết quả của giải thuật phân cụm. Chú ý rằng, trong trường hợp này tuy đã có dữ liệu đã được gán nhãn (lớp/cụm), nhưng các nhãn của các trang Web không được dùng trong quá trình phân cụm mà chỉ dùng để đánh giá chất lượng của giải thuật phân lớp. Có một số độ đo độ đánh giá được dùng trong phương pháp này: *độ chính xác* (precision), *tỷ lệ lỗi* (error), *độ hồi tưởng* (recall) và *F-measure*. Giá sử dữ liệu phân lớp bằng tay gồm có 2 lớp (để phân biệt với cụm) A và B, và giải thuật phân cụm cũng phân thành 2 cụm. Đối với mỗi lớp, ví dụ lớp A, những trang Web thuộc vào lớp A được gọi là các *ví dụ dương* (positive), những trang Web không thuộc vào lớp A được gọi là các *ví dụ âm* (negative). Kết quả phân cụm của một giải thuật sẽ có một số khả năng sau:

- **Dung dương (true positive):** Trang Web là ví dụ dương và được giải thuật phân cụm dự đoán là ví dụ dương (phân cụm đúng), ký hiệu là TP.
- **Sai dương (false positive):** Trang Web là ví dụ dương, nhưng giải thuật phân cụm lại đoán là ví dụ âm (phân cụm sai), ký hiệu là FP.
- **Dung âm (true negative):** Trang Web là ví dụ âm và được giải thuật phân cụm đoán là ví dụ âm (phân cụm đúng), ký hiệu là TN.
- **Sai âm (false negative):** Trang Web là ví dụ âm và được giải thuật phân cụm đoán là ví dụ dương (phân cụm sai), ký hiệu là FN.

Để tính toán ra được các độ đo ở trên, dựa vào các khả năng liệt kê ở trên. Để dễ tính toán, có thể lập ma trận biểu diễn các trường hợp trên, ma trận này được gọi là *ma trận lắn lộn* (confusion matrix) như Bảng 7.7.

Bảng 7.7. Ma trận lắn lộn

| Lớp thực tế | Lớp được dự đoán bởi giải thuật phân cụm | |
|-------------|---|----|
| | Đương | Âm |
| Đương | TP | FN |
| Âm | FP | TN |

Với trường hợp chỉ có 2 lớp trên, từ ma trận lắn lộn này các công thức độ đo sẽ được tính toán cụ thể như sau:

Tỷ lệ lỗi tổng thể:

$$\text{Error} = \frac{FP + FN}{TP + FP + TN + FN} \times 100\% \quad (7.12)$$

Độ chính xác tổng thể:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (7.13)$$

Đối với từng lớp, ta có thể sử dụng thêm 2 độ đo đánh giá sau:
 Độ chính xác:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (7.14)$$

Độ hồi tưởng:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (7.15)$$

Ví dụ, Bảng 7.8 đưa ra kết quả phân cụm với thuật toán k-means với k là 2, thực thi trên tập dữ liệu được liệt kê ở Bảng 7.2, so sánh với tập dữ liệu đã được gán nhãn. Với kết quả phân cụm với thuộc tính A3, ta có các giá trị của các độ đo như sau:

$$\text{Error} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\% = \frac{0+3}{8+0+9+3} \times 100\% = 15\%$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\% = \frac{8+9}{8+0+9+3} \times 100\% = 75\%$$

Với chỉ riêng lớp A, ta có các giá trị của độ chính xác và độ hồi tưởng như sau:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% = \frac{8}{8+0} \times 100\% = 100\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% = \frac{8}{8+3} \times 100\% = 73\%$$

Tương tự, cũng có thể tính toán được độ chính xác (precision) của phân cụm với thuộc tính A6 cho lớp A là 0,6 và độ hồi tưởng là 0,82.

Bảng 7.8. Kết quả phân cụm với k-means chỉ sử dụng 1 thuộc tính

| Lớp thực tế | Thuộc tính A3 | | Thuộc tính A6 | |
|-------------|---|---|---|---|
| | Lớp được dự đoán bởi giải thuật phân cụm | | Lớp được dự đoán bởi giải thuật phân cụm | |
| | A | B | A | B |
| A | 8 | 3 | 9 | 2 |
| B | 0 | 9 | 6 | 3 |

Sо sánh kết quả độ chính xác và độ hồi tưởng của phân cụm với 2 thuộc tính khác nhau A3 và A6 như trên rất khó để có thể kết luận là kết quả nào tốt hơn, vì cái có độ chính xác cao hơn thì lại có độ hồi tưởng thấp hơn và ngược lại. Do vậy, một độ đo khác có tên F-measure đã kết hợp 2 loại độ đo

này lại dễ giúp đánh giá chính xác được kết quả nào tốt hơn. Công thức của độ đo này là:

$$F\text{-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7.16)$$

Như vậy, kết quả của giải thuật phân cụm với thuộc tính A3 có F-measure = 86%, và phân cụm với thuộc tính A6 có F-measure = 69%. Như vậy có thể kết luận là, kết quả của phân cụm với thuộc tính A3 tốt hơn phân cụm với thuộc tính A6. Cũng có thể mở rộng trường hợp có 2 lớp sang trường hợp có nhiều hơn 2 lớp/cụm. Gọi số lớp là m, số cụm là k, chú ý là m có thể khác k. Ma trận lẩn lộn tổng quát sẽ có dạng:

Bảng 7.9. Ma trận lẩn lộn để đánh giá thuật toán phân cụm bằng dữ liệu gán nhãn trong trường hợp tổng quát

| Lớp | Cụm | | | | |
|-----|----------|-----|----------|-----|----------|
| | i | ... | j | ... | k |
| 1 | n_{11} | ... | n_{1j} | ... | n_{1k} |
| ... | ... | ... | ... | ... | ... |
| i | n_{i1} | ... | n_{ij} | ... | n_{ik} |
| ... | ... | ... | ... | ... | ... |
| m | n_{m1} | ... | n_{mj} | ... | n_{mk} |

Và công thức dùng để tính toán các độ đo cho lớp i với cụm j là:

$$\text{Độ chính xác } P(i, j) = \frac{n_{ij}}{\sum_{i=1}^m n_{ij}} \times 100\% \quad (7.17)$$

$$\text{Độ hồi tưởng } R(i, j) = \frac{n_{ij}}{\sum_{j=1}^k n_{ij}} \times 100\% \quad (7.18)$$

$$\text{Độ đo F-measure } F(i, j) = \frac{2 \times P(i, j) \times R(i, j)}{P(i, j) + R(i, j)} \quad (7.19)$$

Để thu được kết quả F-measure trên toàn bộ các cụm, có thể dùng công thức:

$$F = \sum_{i=1}^m \frac{n_i}{n} \max_{j=1, \dots, k} F(i, j) \quad (7.20)$$

trong đó n_i là tổng số trang Web thuộc vào lớp i (hay tổng số hàng thứ i trong ma trận lẩn lộn) $n_i = \sum_{j=1}^k n_{ij}$, và n là tổng số trang Web có trong tập

dữ liệu $n = \sum_{i=1}^m \sum_{j=1}^k n_{ij}$. Tỷ lệ $\frac{n_i}{n}$ trong công thức trên cho biết được độ "quan trọng" của lớp thứ i.

Giả sử với kết quả phân cụm với thuộc tính A6 ở bảng 7.8, có thể tính toán các độ đo đánh giá như sau:

$$P(1, 1) = 9 \times 100\% / (9 + 6) = 60\%; R(1, 1) = 9 \times 100\% / (9 + 2) = 82\%;$$

$$F(1, 1) = 2 \times 0,6 \times 0,82 / (0,6 + 0,82) = 69\%;$$

$$P(1, 2) = 2 \times 100\% / (2 + 3) = 40\%; R(1, 2) = 2 \times 100\% / (9 + 2) = 18\%;$$

$$F(1, 2) = 2 \times 0,4 \times 0,18 / (0,4 + 0,18) = 25\%;$$

$$P(2, 1) = 6 \times 100\% / (6 + 3) = 67\%; R(2, 1) = 6 \times 100\% / (6 + 9) = 40\%;$$

$$F(2, 1) = 2 \times 0,67 \times 0,4 / (0,67 + 0,4) = 50\%;$$

$$P(2, 2) = 3 \times 100\% / (3 + 2) = 60\%; R(2, 2) = 3 \times 100\% / (3 + 6) = 33\%;$$

$$F(2, 2) = 2 \times 0,60 \times 0,33 / (0,60 + 0,33) = 43\%$$

và giá trị F-measure toàn cục $F = \frac{15}{20} \times 0,69 + \frac{5}{20} \times 0,50 = 64\%$.

• Phương pháp dựa vào entropy

Một phương pháp đánh giá này dựa vào lý thuyết xác suất bằng cách giả thiết nhãn lớp của các trang Web trong tập dữ liệu là các sự kiện ngẫu nhiên. Giả thiết này cho phép có thể đánh giá được phân bố xác suất trong mỗi cụm. Xác suất p_{ij} của lớp i ở trong cụm j có thể được ước lượng bằng tỷ lệ xuất hiện của các trang Web có nhãn i ở trong cụm j. Sử dụng ma trận lẩn lộn ta có thể tính được xác suất này là:

$$p_{ij} = \frac{n_{ij}}{\sum_{i=1}^m n_{ij}} \quad (7.21)$$

Nếu chú ý thi đây chính là độ chính xác $P(i, j)$ theo cách tính ở trên. Entropy là độ đo sự hỗn độn của thông tin, và entropy của cụm j được định nghĩa bằng:

$$H_j = - \sum_{i=1}^m p_{ij} \log p_{ij} \quad (7.22)$$

và entropy của toàn bộ các cụm là:

$$H = \sum_{j=1}^k \frac{n_j}{n} H_j \quad (7.23)$$

trong đó n_j là số lượng các trang Web nằm trong cụm j và n là tổng số các trang Web trong tập dữ liệu. Giải thuật phân cụm càng tốt thì entropy của nó

có kết quả càng nhỏ. Ví dụ, với kết quả phân cụm ở Bảng 7.8 sử dụng thuộc tính A6, ta có thể tính giá trị entropy như sau:

$$H = \frac{15}{20} \left(-\frac{9}{15} \log \frac{9}{15} - \frac{6}{15} \log \frac{6}{15} \right) + \frac{5}{20} \left(-\frac{2}{5} \log \frac{2}{5} - \frac{3}{5} \log \frac{3}{5} \right) = 0,292285253$$

7.7. Mô hình phân cụm kết quả tìm kiếm và gán nhãn cụm tiếng Việt

7.7.1. Vấn đề phân cụm kết quả tìm kiếm

Nhu cầu tìm kiếm, truy cập thông tin tiếng Việt ngày càng tăng. Theo đó, nhiều dịch vụ Web đã ra đời nhằm đáp ứng nhu cầu này, trong đó có *Xalo.vn* [Xalo08] của Công ty Cổ phần Truyền thông Tinh Văn, *Socbay* [Soc08] của Công ty Cổ phần Công nghệ Thông tin NAJSCORP, thư mục *Web* [Zing08] của Công ty Cổ phần Dịch vụ Phần mềm Trò chơi Vi Na – VinaGame, *Baamboo* [Baam08] của Công ty Cổ phần Truyền thông Việt Nam,... Mặc dù chất lượng của các máy tìm kiếm ngày càng được cải tiến, việc duyệt qua hàng trăm đến hàng nghìn các mẫu thông tin được gửi về từ máy tìm kiếm vẫn còn là một trở ngại của người dùng. Một nghiên cứu trên log của các máy tìm kiếm cho thấy "*hơn một nửa người dùng không muốn duyệt sang trang thứ hai trong kết quả máy tìm kiếm trả về, và hơn ba phần tư số người dùng không muốn duyệt sang trang thứ ba*" [JSB98]. Vì hầu hết các máy tìm kiếm trình diễn từ 10 đến 20 kết quả trên mỗi trang, phần lớn người dùng không muốn duyệt nhiều hơn 30 kết quả. Một giải pháp cho việc quản lý các kết quả tìm kiếm là thực hiện phân cụm. Giống như phân cụm Web, phân cụm kết quả tìm kiếm gộp các mẫu thông tin tìm kiếm với nhau dựa trên độ đo tương tự; vì thế các mẫu thông tin liên quan tới một chủ đề sẽ có thể được gộp vào một cụm. Việc này giúp người dùng xác định các thông tin cần thiết và nắm được tổng thể của các kết quả tìm kiếm một cách nhanh chóng và dễ dàng. Không giống với phân cụm tài liệu, phân cụm kết quả tìm kiếm cần được thực hiện đối với mỗi kết quả tìm kiếm và giới hạn trong các kết quả từ máy tìm kiếm [ZE99, Ngo03]. Các yêu cầu đối với phân cụm Web cũng đúng đối với phân cụm kết quả tìm kiếm.

Những yêu cầu về phân cụm, đặc biệt là yêu cầu về "sử dụng các mẫu thông tin" và "tốc độ" không dễ dàng được đáp ứng trong trường hợp phân cụm kết quả tìm kiếm do tính chất của các mẫu thông tin. Không giống với các tài liệu Web thông thường, các mẫu thông tin này thường có nhiều nội dung chủ đề không rõ ràng và thường rất ngắn, chỉ chứa từ một số từ tới một vài câu. Vì lý do này, các mẫu thông tin này thường không cung cấp đủ ngữ cảnh cho một độ đo phân cụm tốt.

Gần đây, có nhiều kết quả nghiên cứu nhằm giải quyết vấn đề đưa một số mẫu thông tin ngắn để thu được độ tương tự tốt hơn [PNH08]. Một giải pháp là sử dụng các máy tìm kiếm lấy về ngữ cảnh phong phú hơn cho dữ liệu [SH06, BMI07, YM07]. Với mỗi cặp mẫu thông tin ngắn này, sử dụng các thông kê trên kết quả tìm kiếm thêm của Google để do độ tương tự. Một nhược điểm của phương pháp này là phải thực hiện lặp lại việc truy vấn máy tìm kiếm, giải pháp này tương đối tốn thời gian và không phù hợp với các ứng dụng thời gian thực. Một giải pháp khác là tận dụng các kho dữ liệu trực tuyến như Wikipedia hay DMOZ như là một tài nguyên nền [BP01, Scho06, GM07]. Tuy nhiên, để có được kết quả tối, các tài nguyên này cần phải có định dạng tốt và phong phú. Tuy nhiên, những kiểu tài nguyên dạng này không đủ phong phú trong tiếng Việt.

Dựa trên ý tưởng sử dụng tài nguyên nền như trên, C.T. Nguyen và cộng sự [NPH09] đã đề xuất một mô hình cho phân lớp và gắn nhãn với chủ đề án được khai thác từ kho dữ liệu lớn. Kho dữ liệu này có thể giúp làm giảm tính thưa, ngắn của các mẫu thông tin cũng như cho kết quả phân cụm mang tính định hướng chủ đề tốt hơn. Ý tưởng chính là thu thập một kho dữ liệu đủ lớn (tập dữ liệu tổng thể) và sau đó tiến hành phân tích chủ đề cho tập dữ liệu này dựa trên một số các mô hình chủ đề như pLSA [Hon99], LDA [BNJ03]. Mô hình chủ đề sau khi được ước lượng có thể coi như một loại tri thức ngôn ngữ chứa các mối quan hệ giữa các từ. Dựa trên mô hình chủ đề, có thể tiến hành phân tích chủ đề cho các kết quả tìm kiếm để xác định các chủ đề án chứa trong các kết quả này. Các chủ đề sau đó được kết hợp với các mẫu thông tin gốc để tạo ra một biểu diễn mở rộng, giàu ngữ nghĩa hơn. Sử dụng các phương pháp đo độ tương tự phổ biến (như độ đo cosine), có thể áp dụng các phương pháp phân cụm như HAC, K-means để phân cụm các kết quả được làm giàu ngữ nghĩa này. Ưu điểm của lược đồ này gồm những điểm sau:

- Giảm độ thưa dữ liệu: Các mẫu thông tin tuy gần nhau về mặt ngữ nghĩa, nhưng khác nhau về cách dùng từ vựng sẽ không có được độ đo tương tự thích hợp nếu ta chỉ dựa trên từ vựng. Việc thêm các chủ đề án giúp cho các mẫu thông tin này gần nhau hơn so với các bản gốc.
- Giảm tính nhập nhằng: Một số mẫu thông tin có chia sẻ các từ vựng không quan trọng như các từ dừng (các từ này có thể không thể loại bỏ hết trong quá trình tiền xử lý). Bằng cách đưa các chủ đề án vào, độ đo tương tự giữa các cặp kết quả tìm kiếm chỉ chia sẻ các từ dừng mà không cùng chủ đề sẽ giảm đi so với tương quan chung với các cặp khác. Kết quả là phương pháp này vượt qua giới hạn của tương khớp nồng độ dựa trên từ vựng.

- Cho phép gắn các nhãn giàu ngữ nghĩa hơn: Các phương pháp gắn nhãn truyền thống dựa trên giả thuyết rằng, các từ cụm từ lập trong một cụm

có nhiều khả năng là nhãn cụm. Các phương pháp này không phải luôn cho kết quả tốt trong mọi trường hợp. Phương pháp kết hợp chủ đề để đo độ tương tự về mặt chủ đề giữa cụm và các từ vựng/cụm từ như là các đặc trưng quan trọng để xác định nhãn thích hợp.

- Tính linh động, dễ thực thi: Phương pháp này đơn giản và dễ thực thi. Tất cả những gì chúng ta cần là một tập hợp dữ liệu văn bản lớn (tập tổng thể) và dùng các chủ đề ăn được mô tả để tăng cường chất lượng phân cụm và gán nhãn. Vì tập dữ liệu tổng thể này không đòi hỏi phải có cấu trúc như Wordnet, Ontology,... đây là giải pháp kinh tế và hiệu quả với bài toán phân cụm và gán nhãn kết quả tìm kiếm trên Web trong tiếng Việt.

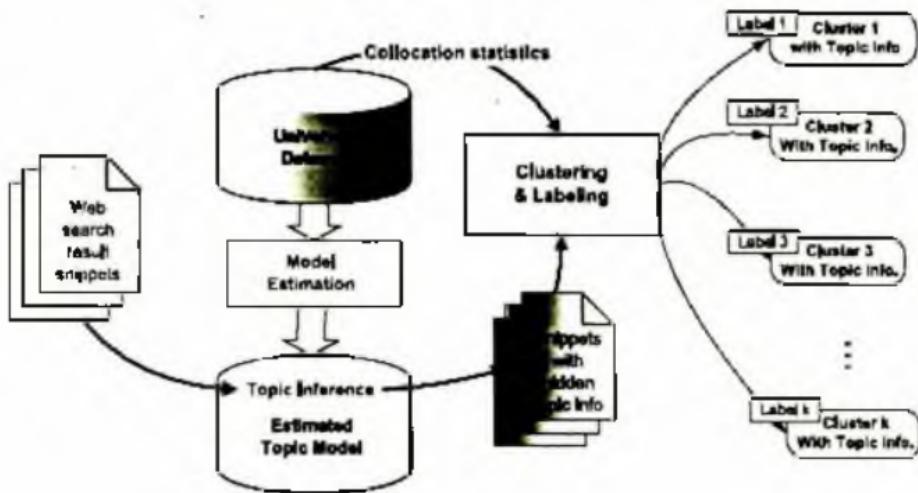
Thêm vào đó, giải pháp này mang tính tổng quát và không ràng buộc với bất kỳ phương pháp phân cụm nào. Trong công trình nghiên cứu này, nhóm tác giả tiến hành đánh giá cẩn thận cho vấn đề phân cụm kết quả trong tiếng Việt với tập dữ liệu tổng thể có dung lượng khoảng vài trăm megabytes các trang Web từ Wikipedia và VnExpress.

7.7.2. Mô hình phân cụm với chủ đề ăn

Lược đồ phân cụm và gán nhãn với chủ đề ăn được mô tả trong Hình 7.6 bao gồm 6 bước chính. Trong số 6 bước này, việc lựa chọn tập dữ liệu tổng thể (a) có thể coi là bước quan trọng nhất. Tập dữ liệu tổng thể cần đủ lớn và giàu thông tin để bao phủ tập từ vựng, khái niệm và chủ đề phong phú, thích hợp cho miền ứng dụng. Tuy nhiên, tập dữ liệu tổng thể này không nhất thiết phải có cấu trúc như DMOZ. Tập dữ liệu này cần phải được tiền xử lý để loại trừ dữ liệu nhiễu và các từ không phù hợp, từ đó bước (b) có thể có kết quả tốt. Bên cạnh phân tích chủ đề, sử dụng tập dữ liệu tổng thể để tìm các ngữ cố định (c). Các ngữ cố định này được dùng cho việc gán nhãn cụm trong bước (f). Một điểm đáng lưu ý là (a), (b) và (c) được thực hiện ngoại tuyến, không giám sát. Mô hình ước lượng tham số còn có thể được sử dụng cho nhiều bài toán khác nhau [PNH08].

Một cách tổng quát, bước phân tích chủ đề (b) có thể được tiến hành với một trong các mô hình chủ đề như pLSA [Hof99], LDA [BNJ03], DTC hay CTM. Cũng cần lưu ý rằng, các mô hình càng phức tạp thì việc cung cấp thông tin chủ đề càng phong phú, nhưng làm tăng độ phức tạp thời gian của hệ thống. Ở đây dùng mô hình LDA [BNJ03] vì LDA là mô hình sinh hoàn thiện hơn so với pLSA, nhưng không quá phức tạp. Với LDA, có thể vẫn thu được các quan hệ ngữ cảnh quan trọng trong giới hạn thời gian cho phép.

Trong trường hợp dựa trên phương pháp LDA, kết quả của bước (b) là một mô hình ước lượng bao gồm các chủ đề, phân phối xác suất của từ vựng trong chủ đề. Dựa trên mô hình này và một tập các kết quả tìm kiếm, có thể tiến hành lập luận chủ đề (d) cho các kết quả tìm kiếm. Chủ ý rằng, những mẩu thông tin này được coi như là tài liệu mới, và được phân tích chủ đề dựa trên mô hình đã được ước lượng trên tập dữ liệu tổng thể. Với mỗi một mẩu thông tin, đầu ra của bước (d) là phân phối chủ đề ẩn, trong đó các chủ đề có xác suất cao là các chủ đề chính tương ứng. Ví dụ, một mẩu thông tin được trả về như là kết quả của câu truy vấn "ma trận" có thể liên quan đến các chủ đề như "toán học" hay "phim ảnh". Việc kết hợp các thông tin chủ đề này với các dữ liệu gốc như thế nào cho việc phân cụm và gán nhãn phụ thuộc vào thuật toán phân cụm sử dụng. Tuy nhiên, phương pháp không giới hạn thuật toán phân cụm, có thể dùng với Kmeans hay HAC.....

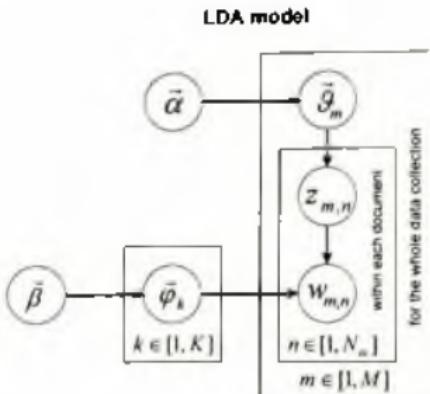


Hình 7.6. Mô hình phân cụm kết quả tìm kiếm Web với chủ đề ẩn [NPH09]

7.7.3. Phân tích chủ đề với tập dữ liệu tổng thể

a) Phân tích chủ đề ẩn với LDA

Latent Dirichlet Allocation (LDA) là một mô hình sinh xác suất cho tập dữ liệu rời rạc như text corpora. LDA dựa trên ý tưởng: mỗi tài liệu là sự trộn lẫn của nhiều topic. Về bản chất, LDA là một mô hình Bayesian 3 cấp (three – level hierarchical Bayes model: corpus level, document level, word level), trong đó mỗi phần của mô hình được coi như một mô hình trộn hữu hạn trên cơ sở tập các xác suất topic.



Hình 7.7. Mô hình sinh trong LDA [BNJ03]

Cho một corpus của M tài liệu biểu diễn bởi $D = \{d_1, d_2, \dots, d_M\}$, trong đó, mỗi tài liệu m trong corpus bao gồm N_m từ w , rút từ một tập từ vựng của các mục từ $\{t_1, \dots, t_V\}$, V là số lượng các mục từ t trong tập từ vựng. LDA cung cấp một mô hình sinh đầy đủ chỉ ra kết quả tốt hơn các phương pháp trước. Quá trình sinh ra văn bản như sau:

Các khối vuông trong Hình 7.7 biểu diễn các quá trình lặp.

Tham số đầu vào α và β (tham số mức).

- $\bar{\theta}_m$: phân phối của topic trong document thứ m (document-level parameter). $\bar{\theta}_m$ biểu diễn tham số cho $p(z|m)$, thành phần trộn topic cho tài liệu m . Một tỷ lệ cho mỗi tài liệu

$$\Theta = \left\{ \bar{\theta}_m \right\}_{m=1}^M \text{ (M} \times \text{K matrix)}$$

- $z_{m,n}$: chỉ số chủ đề (word n của văn bản m);
- $w_{m,n}$: word n của văn bản m chỉ bởi $z_{m,n}$ (word-level variable, observed word)

$$\Phi = \left\{ \bar{\phi}_k \right\}_{k=1}^K \text{ (K} \times \text{V matrix)}$$

- $\bar{\phi}_k$: phân phối của các từ được sinh từ chủ đề $z_{m,n}$. $\bar{\phi}_k$ biểu diễn tham số cho $p(t|z=k)$, thành phần trộn của topic k . Một tỷ lệ cho mỗi topic;
- M : số lượng các tài liệu;
- N_m : số lượng các từ trong tài liệu thứ m (hay còn gọi là độ dài của văn bản);
- K : số lượng các topic ẩn.

LDA sinh một tập các từ $w_{m,n}$ cho các văn bản d_m bằng cách:

- Với mỗi văn bản m , sinh ra phân phối topic θ_m cho văn bản theo $\text{Dir}(\alpha)$;
- Sinh ra chỉ số chủ đề $z_{m,n}$ dựa vào phân phối topic $\text{Mult}(\theta_m)$;
- Từ w được sinh ra dựa vào phân phối từ ϕ_z , (ở đây $z = z_{m,n}$).

Ở đây, Dir và Mult lần lượt là các phân phối Dirichlet, Multinomial. (Lấy mẫu theo phân phối Dirichlet, Multinomial). Ta cần phân biệt hai quá trình cơ bản:

+ Ước lượng mô hình: cho một tập dữ liệu, K – số lượng chủ đề, văn bản ước lượng mô hình là sinh ra các phân phối ϕ_z , với mỗi một chủ đề và một từ vựng.

+ Lập luận mô hình: dựa trên mô hình đã ước lượng, cho một tài liệu mới, ước lượng phân phối chủ đề cho tài liệu này.

Chi tiết về các bài toán này có thể được xem thêm trong [BNJ03] và [NPH09].

| Topic 3 | Topic 4 | Topic 7 | Topic 9 | Topic 10 | Topic 15 |
|----------------------------|---------------------------------------|----------------------|-----------------------|---|---|
| hàm (functions) | phần mềm (software) | chủ đề (football) | máy bay (aircraft) | tác giả (author) | quốc hội (congress) |
| thông tin (space) | chương trình (program) | player | sân bay | sách | tổng thống (president) |
| toán học (mathematics) | Windows ¹ (Windows) | H.I.V ² | hàng không | nghề | đảng chủ nhiệt ³ (democracy) |
| định nghĩa (definition) | Windows ¹ (Windows) | đội bóng | giáo thông | văn học ⁴ (literature) | bộ đồng |
| phân tử (elements) | phiên bản (version) | trận đấu | traffic | truyện ⁵ (stories) | chính quyền (government) |
| bài toán (problem) | Microsoft ⁶ (Microsoft) | tiền đạo | tai nạn | thơ /poems | nhân dân (people) |
| tý thuyết (theory) | hỗn hợp | hỗn hợp | tragedy | tiểu thuyết (novel) | cộng hòa (republic) |
| tính toán (calculation) | ứng dụng (applications) | hỗn hợp | quốc tế | nhất bản ⁷ (publish) | nhà nước (state) |
| sắc định (definite) | chi tiết | player | Boeing ⁸ | nhà thơ | biến pháp (constitution) |
| định lý (theorem) | giao diện | thủ môn | vận chuyển | Độc giả (reader) | lãnh đạo (leadership) |
| phương trình (equation) | interface | thủ môn | điện tử | văn chương ⁹ (literature) | bầu cử |
| ảnh xạ (mapping) | trình duyệt | trợ lý | phương tiện | nhà xuất | hội nghị (meeting) |
| đại số (algebra) | internet | trợ lý | transportation | báo | đảng cộng sản (communist party) |
| | server | đội tuyển | đường sắt | nhà xuất | |
| | | line-up | trainway | phát hành ¹⁰ (publisher) | |
| | | SLNA ¹¹ | nhà ga | phát hành ¹⁰ (publish) | |
| | | | station | | |

Hình 7.8. Top các từ vựng với xác suất cao nhất trên một số chủ đề

b) Ước lượng mô hình chủ đề ẩn LDA với tập dữ liệu tổng thể

Tập dữ liệu tổng thể bao gồm khoảng 480MB dữ liệu được lấy từ Wikipedia tiếng Việt và VnExpress, bao gồm các nội dung liên quan đến khoa học, lịch sử, chính trị, văn hóa, thể thao,... Tập dữ liệu này được làm sạch (loại bỏ các tài liệu quá ngắn, loại HTML, loại từ dừng,...) và tách câu, tách từ. Sau đó sử dụng GibbsLDA cho việc phân tích mô hình chủ đề. Ví dụ về một số chủ đề ẩn được sinh ra sau quá trình ước lượng mô hình như được cho dưới đây:

Một số quan sát thú vị có thể thu được như Hình 7.8. Chủ đề ẩn có thể gán nhãn ngữ nghĩa cho các từ viết tắt như HLV (huấn luyện viên) hay SLNA (Sông Lam Nghệ An); từ nước ngoài như Windows, internet, các từ gắn nghĩa như văn học hay văn chương....

7.7.4. Kết hợp thông tin chủ đề ẩn cho phân cụm và gán nhãn

a) Phân cụm với chủ đề ẩn

Với hai kết quả tìm kiếm các mẫu thông tin như dưới đây, sau khi tiến hành lập luận chủ đề ta có thể do độ tương tự giữa chúng dựa trên hai hướng nhìn: độ độ tương tự dựa trên từ vựng hoặc do độ tương tự dựa trên chủ đề.

Mẫu thông tin 1:

Ma trận (toán học) – Wikipedia tiếng Việt

Trong toán học, một **ma trận** là bảng chữ nhai chứa dữ liệu (thường là số thực hoặc số phức, nhưng có thể là bất kỳ dữ liệu gì) theo hàng và cột ...

Mô tả - Các loại ma trận đặc biệt - Các phép toán đại số trên ma trận

<http://wikipedia.org/> / [Ma_tran_\(toan_hoc\)](#) - [Đã lưu trong bộ nhớ cache](#) - [Tương tự](#) .

Mẫu thông tin 2:

Ma trận nghịch đảo (khá nghịch) « Maths 4 Physics(M4Ps) and more

3 Tháng Mười 2008 ... Cho A là một **ma trận** vuông cấp n trên K. Ta bảo A là **ma trận** khá nghịch, nếu tồn tại một **ma trận** B vuông cấp n trên K sao cho: A.B = B.A = I_n ...
<http://thunhan.wordpress.com/> / [matrix-inversion/](#) .

Phương pháp mà nhóm tác giả sử dụng để kết hợp hai độ đo này để thu được một độ đo tương tự cho phân cụm được thể hiện như sau:

$$\text{sim}_{d_i, d_j} (\text{topic - vectors}) = \frac{\prod_{k=1}^K t_{i,k} \times t_{j,k}}{\sqrt{\sum_{k=1}^K t_{i,k}^2} \sqrt{\sum_{k=1}^K t_{j,k}^2}}$$

$$\text{sim}_{d_i, d_j} (\text{term - vectors}) = \frac{\prod_{l=1}^{|V|} w_{i,l} \times w_{j,l}}{\sqrt{\sum_{l=1}^{|V|} w_{i,l}^2} \sqrt{\sum_{l=1}^{|V|} w_{j,l}^2}}$$

Ở đây, d_i và d_j là hai độ đo tương ứng, độ đo tương tự được tính toán dựa trên độ đo Cosine, t và w là các trọng số chủ đề và từ vựng. Kết hợp hai độ đo trên, ta thu được độ đo tương tự như sau:

$$\text{sim}(d_i, d_j) = \lambda \times \text{sim}(\text{topic - vectors}) + (1 - \lambda) \times \text{sim}(\text{term - vectors})$$

b) Gán nhãn cụm với chủ đề ẩn

Tư tưởng chính của gán nhãn cụm với chủ đề ẩn gồm các bước:

- Trích rút các cụm từ, từ vựng tiềm năng; tiến hành trích rút các cụm từ có nghĩa dựa trên xác định cụm từ và tần suất xuất hiện;
- Với mỗi ứng cử viên tiềm năng; tiến hành trích rút một số các đặc trưng như đặc trưng về từ, chủ đề,...
- Đo độ tương tự giữa từ/cụm từ với cụm tương ứng, sắp xếp các cụm từ và chọn cụm từ/từ đầu tiên làm nhãn cụm.

7.7.5. Kết quả phân cụm/gán nhãn dữ liệu tìm kiếm

Kết quả phân cụm và gán nhãn trong mô hình của [NPH09] cho kết quả tốt hơn so với Baseline và Vivisimo (Hình 7.9).

| Hồng Sơn (A personal name) | | |
|---|--|---|
| Clustering and Labeling with HT20 | The Baseline | Vivisimo |
| Bác sĩ Hồng Sơn (14) Doctor Pham Hong Son (14) Thái úy Dương Hồng Sơn (11) Guard Keeper Duong Hong Son Diễn viên (8) Actor/Actress Đạo diễn Võ Hồng Sơn (5) Director Vu Hong Son Ca sĩ (5) Singer Môn phái (5) Artemis Art Group Nghệ an, nghệ sỹ (4) Nghệ An, Temper Xã (4) Community | Hàm số Pham Hong Son (13) Doctor Pham Hong Son Thái úy Dương Hồng Sơn (11) Guard Keeper Duong Hong Son Diễn viên (8) Actor/actress Đạo diễn Vũ Hồng Sơn (5) Director Vu Hong Son Nguyễn Hồng Sơn (4) Nguyen Hong Son Xã (4) Community Việt Nam (4) Vietnam Ca sĩ (3) Singer | Nam, Việt (45) Nam, Viet Vietnamese (17) Vietnamese Phạm Hồng Sơn (28) Pham Hong Son Công (21) Cong Quang (13) Quang Đương Hồng Sơn (12) Duong Hong Son Thông tin (11) Information Bản tri (8) Dan tri |
| Không hàng (Cross) | | |
| Clustering and Labeling with HT20 | The Baseline | Vivisimo |
| Ngân hàng (3) Bank Khủng hoảng kinh thực (18) Economic Crisis Nền kinh tế (4) Economy Sản xuất và khung hoảng nhảm (4) (10) Human Powers, Human Resource Crisis Doanh nghiệp Việt Nam (10) Vietnam companies Xuất khẩu hàng (9) Export management Giáo dục Việt nam (7) Vietnam Education Nhà đất (4) Real estate Khủng hoảng chính trị (6) Political Crisis | Tin dung MS (22) United State Credit Tài sản (21) Wang Tài chính (17) Finance Việt Nam (15) Vietnam Chính trị (13) Politics Nhân sự (7) Human Resource Giáo dục (7) Education Xuất khẩu hàng (6) Export Management Thực phẩm thế giới (6) World Food | Chính, Không Kinh Tế (49) The phrase "American Crisis" but in the wrong order Việt Nam (43) Vietnam Vietnam (30) Vietnam Khủng hoảng kinh (21) Part of the phrase "Economic Crisis" Thông tin (19) Information Doanh (15) in "Doanh nghiệp" (Companies), Vietnamnet (12) Vietnamnet Thực, Không hoảng kinh (18) The phrase "Food Crisis" but in the wrong order |

Hình 7.9. Kết quả phân cụm và gán nhãn với phương pháp nền (HAC không dùng chủ đề), và phân cụm với chủ đề (mô hình HT20)

Câu hỏi và bài tập

1. Thu thập một tập dữ liệu các trang Web từ Internet về, sau đó viết chương trình để chuyển đổi các trang Web này về dạng các vector.
2. Cài đặt thuật toán phân cụm k-means với gán cứng, sau đó áp dụng phân cụm tập dữ liệu thu thập được từ bài 1.
3. Cài đặt thuật toán phân cụm k-means với gán mềm, sau đó áp dụng phân cụm tập dữ liệu thu thập được từ bài 1.
4. Cài đặt thuật toán phân cụm phân cấp tích tụ từ dưới lên với độ đo người láng giềng gần nhất, sau đó áp dụng phân cụm tập dữ liệu thu thập được từ bài 1.
5. Cài đặt thuật toán phân cụm phân cấp tích tụ từ dưới lên với độ đo người láng giềng xa nhất, sau đó áp dụng phân cụm tập dữ liệu thu thập được từ bài 1.
6. Cài đặt thuật toán phân cụm phân cấp tích tụ từ dưới lên với độ đo tương tự trung bình, sau đó áp dụng phân cụm tập dữ liệu thu thập được từ bài 1.
7. Cài đặt thuật toán phân cụm phân cấp từ trên xuống, sau đó áp dụng phân lớp tập tài liệu thu được từ bài 1.
8. Dùng phương pháp đánh giá dựa vào độ tương tự để đánh giá các thuật toán phân cụm từ bài 2 đến bài 7.
9. Phân loại bằng tay tập dữ liệu trong bài 1 ra một số lớp (ví dụ: 3 lớp). Dùng phương pháp đánh giá dựa trên dữ liệu gán nhãn để đánh giá các giải thuật phân cụm từ bài 2 đến bài 7.
10. Áp dụng thuật toán phân cụm trong bài 2 và 3 để phân cụm N kết quả đầu tiên trả về của máy tìm kiếm google đối với một câu truy vấn nào đó, chú ý là ta chỉ sử dụng các từ khóa hay đoạn văn bản Google trả về để dùng cho quá trình phân cụm, chứ không phải tải toàn bộ nội dung các trang Web nằm trong danh sách trả về.

Chương 8

PHÂN LỚP VĂN BẢN

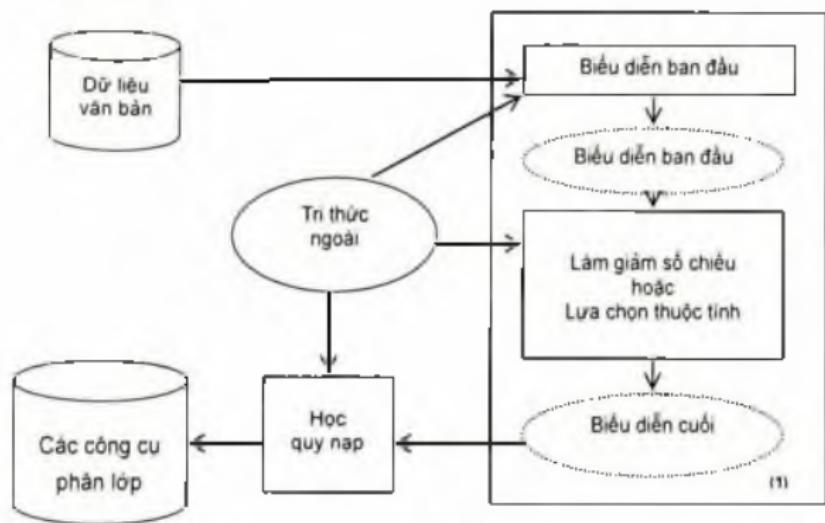
8.1. Giới thiệu

Phân lớp là một trong những mối quan tâm nhiều nhất của con người trong quá trình làm việc với một tập hợp đối tượng. Điều này giúp con người có thể tiến hành việc sắp xếp, tìm kiếm các đối tượng một cách thuận lợi. Khi biểu diễn đối tượng vào các hệ thống thông tin, tính chất lớp vốn có của đối tượng trong thực tế thường được biểu diễn tương ứng bằng một thuộc tính "lớp" riêng biệt. Chẳng hạn, trong hệ thống thông tin quản lý tài liệu của thư viện, thuộc tính về loại tài liệu có miền giá trị là tập tên chuyên ngành của tài liệu, gồm các giá trị như "Tin học", "Vật lý".... Trước đây các công việc gán các giá trị của thuộc tính lớp thường được làm một cách thủ công. Nhưng hiện nay, với sự bùng nổ của thông tin và các loại dữ liệu, việc đánh thuộc tính lớp một cách thủ công là rất khó khăn, có thể nói là không thể. Do vậy, các phương pháp phân lớp tự động, trong đó có phân lớp văn bản là rất cần thiết và là một trong những chủ đề chính trong khai phá dữ liệu.

Bài toán phân lớp văn bản còn được phân biệt một cách chi tiết hơn. Phân lớp nhị phân khi miền áp dụng chỉ có hai lớp ($|C| = 2$) và phân lớp đa lớp khi miền ứng dụng có nhiều hơn hai lớp ($|C| > 2$). Như vậy, phân lớp nhị phân chỉ là một trường hợp đặc biệt của bài toán phân lớp, song do xuất xứ cho nên phân lớp nhị phân có vị trí riêng cả về đặt bài toán lẫn về các giải pháp. Ngoài ra còn phân biệt các loại phân lớp đơn nhãn với đa nhãn. Trong phân lớp đơn nhãn, mỗi tài liệu được gán vào một và chỉ một lớp, trong khi đó, trong phân lớp đa nhãn một tài liệu có thể được gán nhiều hơn một lớp. Điều này có ý nghĩa thực tiễn lớn, vì một văn bản không chỉ liên quan tới một chủ đề duy nhất, ví dụ như, một trang Web nói về việc bùng phát bệnh cúm gia cầm tại một số tỉnh phía Bắc có thể thuộc về chủ đề dịch tễ, nhưng đồng thời cũng thuộc về lĩnh vực chăn nuôi. Trong những trường hợp như vậy, việc sắp xếp một tài liệu vào nhiều hơn một lớp là phù hợp với yêu cầu thực tế.

Phân lớp văn bản được các nhà nghiên cứu định nghĩa thông nhất như là việc gán tên các chủ đề (tên lớp nhãn lớp) đã được xác định cho trước vào các văn bản dựa trên nội dung của nó. Phân lớp văn bản là công việc được

sử dụng để hỗ trợ trong quá trình tìm kiếm thông tin (Information Retrieval), chiết lọc thông tin (Information Extraction), lọc văn bản hoặc tự động dẫn đường cho các văn bản tới những chủ đề xác định trước [DPH98, Lew92b, Yan99].



Hình 8.1. Lược đồ chung xây dựng bộ phân lớp văn bản

Hình 8.1 biểu diễn một lược đồ chung cho hệ thống phân lớp văn bản, trong đó bao gồm ba thành phần chính: thành phần đầu tiên là biểu diễn văn bản, tức là chuyển các dữ liệu văn bản thành một dạng có cấu trúc nào đó [Mla98]. Thành phần thứ hai là học quy nạp – sử dụng các kỹ thuật học máy để phân lớp văn bản vừa biểu diễn. Thành phần thứ ba là tri thức ngoài – bổ sung các kiến thức thêm vào do người dùng cung cấp để làm tăng độ chính xác trong biểu diễn văn bản hay trong quá trình học máy. Trong nhiều trường hợp, các phương pháp học hệ thống phân lớp có thể bò qua thành phần thứ ba này.

Thành phần thứ hai được coi là trung tâm của một hệ thống phân lớp văn bản. Trong thành phần này, có nhiều phương pháp học máy được áp dụng như mô hình học Bayes, cây quyết định, phương pháp k người láng giềng gần nhất, SVM,... [Seb02, HAN90, Joa98, CRM98, Mla98] là phù hợp.

Đại lượng đánh giá hiệu suất phân lớp:

Việc đánh giá độ phân lớp dựa trên việc áp dụng mô hình đối với các dữ liệu thuộc tập dữ liệu kiểm tra D_{test} , sử dụng mô hình cho từng trường hợp dữ liệu ở D_{test} mà kết quả ra là lớp c dự báo cho từng dữ liệu. Hai độ đo được dùng phổ biến để đánh giá chất lượng của thuật toán phân lớp là độ

hồi tưởng (recall) ρ và độ chính xác (precision) π . Ngoài ra, một số độ đo kết hợp được xây dựng từ các độ đo này cũng được sử dụng, trong đó điển hình nhất là độ đo f_1 (nhiều trường hợp bỏ qua chỉ số 1). Phần dưới đây trình bày các tính toán chi tiết giá trị của các độ đo hồi tưởng và chính xác trong bài toán phân lớp văn bản [Seb02].

Xét trường hợp lực lượng của tập C các lớp trong bài toán lớn hơn hai ($|C| > 2$) với lưu ý rằng, trường hợp tập C chỉ gồm có hai lớp là đơn giản. Đối với mỗi lớp c , cho thực hiện mô hình phân lớp vừa được xác định với các dữ liệu thuộc D_{test} nhận được các đại lượng TP_c , TF_c , FP_c , FN_c như bảng dưới đây:

| Lớp c | | Giá trị thực tế | |
|-------------------------|-------------------|-----------------|-------------------|
| | | Thuộc lớp c | Không thuộc lớp c |
| Giá trị qua bộ phân lớp | Thuộc lớp c | TP_c | TN_c |
| | Không thuộc lớp c | FP_c | FN_c |

Điển giải bằng lời cho từng giá trị trong bảng:

- TP_c (true positives): số lượng ví dụ dương (tài liệu thực sự thuộc lớp c) được thuật toán phân lớp gán cho giá trị đúng thuộc lớp c.
- TN_c (true negatives): số lượng ví dụ âm (tài liệu thực sự không thuộc c) nhưng lại được thuật toán phân lớp gán cho giá trị đúng thuộc lớp c.
- FP_c (false positives): số lượng ví dụ dương được thuật toán phân lớp gán cho giá trị sai là không thuộc lớp c.
- FN_c (false negatives): số lượng ví dụ âm được thuật toán phân lớp gán cho giá trị sai là không thuộc lớp c.

Khi đó, với mỗi lớp c , giá trị các độ đo ρ_c và π_c được tính như sau:

$$\rho_c = \frac{TP_c}{TP_c + FP_c} \quad \text{và} \quad \pi_c = \frac{TP_c}{TP_c + FN_c} \quad (8.1)$$

Với bài toán phân lớp nhị phân, các độ đo nói trên cho một lớp trong hai lớp là dù để đánh giá chất lượng bộ phân lớp, tuy nhiên, trong trường hợp bài toán phân lớp K lớp, các độ đo trung bình được sử dụng bao gồm trung bình mịn (microaveraging) và trung bình thô (macroaveraging).

Độ hồi tưởng trung bình thô (macroaveraging recall):

$$\rho^M = \frac{1}{K} \sum_{c=1}^K \rho_c \quad (8.2)$$

và độ chính xác trung bình thô (macroaveraging precision)

$$\pi^M = \frac{1}{K} \sum_{c=1}^K \pi_c \quad (8.2')$$

Dộ hồi tưởng trung bình mịn (microaveraging recall):

$$P^R = \frac{\sum_{c=1}^K TP_c}{\sum_{c=1}^K (TP_c + FP_c)} \quad (8.3)$$

và độ chính xác trung bình mịn (microaveraging precision)

$$\pi^R = \frac{\sum_{c=1}^K TP_c}{\sum_{c=1}^K (TP_c + TN_c)} \quad (8.3')$$

Các độ đo vi trung bình mịn được coi là các độ đo tốt hơn để đánh giá chất lượng thuật toán phân lớp đa lớp tài liệu.

8.2. Một số thuật toán phân lớp có giám sát

Lớp giải thuật phân lớp đầu tiên được áp dụng cho việc phân lớp văn bản là các giải thuật học máy có giám sát, trong đó việc học là phụ thuộc hoàn toàn vào dữ liệu đã được gán nhãn (trong bài toán cụ thể này, các tài liệu văn bản đã được gán vào các lớp cụ thể).

8.2.1. Phân lớp dựa trên hệ thống luật

Những hệ thống phân lớp đầu tiên là những hệ sử dụng luật phân lớp bằng cách sử dụng người dùng định nghĩa các luật này. Một trong những hệ nổi tiếng nhất là CONSTRUE được P.J. Hayes và cộng sự (một nhóm nghiên cứu tại Đại học Carnegie Mellon University – CMU) phát triển vào những năm 1980 để phân lớp các bài tin cho hãng tin Reuters [ADW94, Seb02]. Hệ thống này sử dụng luật dưới dạng chuẩn rời nhau (DNF – Disjunctive Normal Form). Một luật DNF ví dụ của hệ thống nói trên được trình bày trong Hình 8.2.

Hệ thống phân lớp dựa trên luật thường cho kết quả tương đối cao trong trường hợp người dùng có thể tạo đầy đủ các luật, bao gồm tất cả các trường hợp có thể xảy ra cho bộ phân lớp. Tuy nhiên, phương pháp này có điểm hạn chế là có thể có những luật mâu thuẫn nhau, hoặc có những trường hợp mà luật bù sót. Ngoài ra, khi dữ liệu thay đổi thì có thể phải cập nhật hoặc sửa đổi lại hệ thống luật phân lớp.

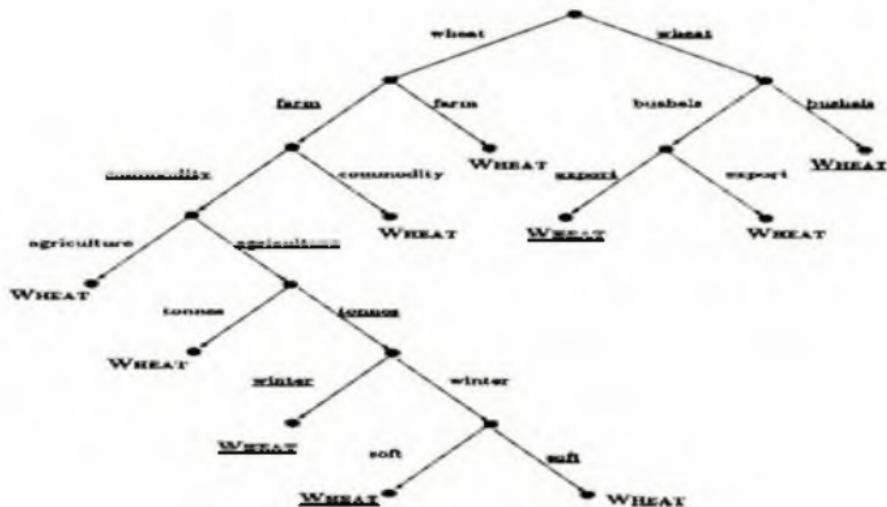
8.2.2. Thuật toán Bayes

Thuật toán phân lớp Bayes là một trong những thuật toán phân lớp điển hình nhất trong học máy và khai phá dữ liệu. Đây cũng là một trong những thuật toán được sử dụng rộng rãi nhất trong phân lớp văn bản. Trong học máy, Bayes thường được coi như thuật toán học máy chuẩn (baseline) để so sánh với các thuật toán khác [DC00, Seb02, LR94, Lew92, Mit97, Mla98].

wheat & farm → wheat
 wheat & commodity → wheat
 bushels & export → wheat
 wheat & agriculture → wheat
 wheat & tonnes → wheat
 wheat & winter & soft → wheat

| | | Test Cases | |
|-----------|-------|------------|--|
| | wheat | not wheat | |
| wheat | 73 | 8 | |
| not wheat | 14 | 3577 | |

a)



b)

Hình 8.2. (a) Luật phân lớp DNF cho lớp WHEAT [ADW94] và (b) cây quyết định tương đương (nhân cung là từ khoá, nhân là lớp) [Seb02]

Ý tưởng chính của thuật toán là tính xác suất hậu nghiệm của sự kiện c xuất hiện khi sự kiện x đã có trong không gian ngữ cảnh t thông qua tổng hợp các xác suất tiên nghiệm của sự kiện c xuất hiện khi sự kiện x đã có trong tất cả các điều kiện riêng T thuộc không gian t:

$$p(c|x, t) = \sum_{T \in t} p(c|x, T)p(T|x)$$

Trong trường hợp phân lớp văn bản, xét biểu diễn văn bản thông qua tập các từ khoá có trong văn bản đó. Gọi V là tập tất cả các từ vựng. Giá sữ có N lớp tài liệu là c_1, c_2, \dots, c_n . Mỗi lớp c_i có xác suất $p(c_i)$ và ngữđồng

CtgTsh.; Gọi $p(c|Doc)$ là xác suất để tài liệu Doc thuộc lớp c hay "xác suất để sự kiện c xuất hiện khi đã có sự kiện Doc".

Cho một lớp c và một tài liệu Doc, nếu xác suất $p(c|Doc)$ tính được lớn hơn hoặc bằng giá trị ngưỡng CtgTsh_c của lớp c thì kết luận tài liệu Doc thuộc vào lớp c.

Tài liệu Doc được biểu diễn như một vector có kích thước là số từ khoá trong tài liệu. Mỗi thành phần chứa một từ trong tài liệu và tần suất xuất hiện của từ đó trong tài liệu. Thuật toán được thực hiện trên tập từ vựng V, vector biểu diễn tài liệu Doc và các tài liệu có sẵn trong lớp, tính toán $p(c|Doc)$ và quyết định tài liệu Doc sẽ thuộc lớp nào.

Xác suất $p(c | Doc)$ được tính theo công thức sau:

$$p(c | Doc) = \frac{p(c) * \prod_{F_i \in V} (p(F_i | c))^{TF(F_i, Doc)}}{\sum_{i=1}^n p(c_i) * \prod_{F_i \in V} (p(F_i | c))^{TF(F_i, Doc)}} \quad (8.4)$$

với

$$P(F_j | c) = \frac{1 + TF(F_j, c)}{|V| + \sum_{i=1}^n TF(F_i | c)} \quad (8.5)$$

Trong đó: $|V|$: số lượng các từ khoá có trong từ vựng V; F_j : từ khoá thứ j trong từ vựng V; $TF(F_j | Doc)$: tần suất của từ F_j trong tài liệu Doc (bao gồm cả từ đồng nghĩa); $TF(F_j | C)$: tần suất của từ F_j trong lớp c (số lần F_j xuất hiện trong tất cả các tài liệu thuộc lớp c); $p(F_j | c)$: xác suất có điều kiện để từ F_j xuất hiện trong tài liệu của lớp c.

Công thức $TF(F_i | c)$ được tính theo trắc lượng xác suất Laplace. Sờ dĩ có số 1 trên tử số của công thức này để tránh trường hợp tần suất từ F_i trong lớp c bằng 0 khi F_i không xuất hiện trong lớp c.

Để giảm sự phức tạp và thời gian tính toán, để ý rằng, không phải tài liệu Doc đã cho đều chứa tất cả các từ trong tập từ vựng V. Do đó, $TF(F_i | Doc) = 0$ khi từ F_i thuộc V nhưng không thuộc tài liệu Doc, nên ta có $(p(F_i | c))^{TF(F_i, Doc)} = 1$. Vậy, công thức (8.4) được viết lại như sau:

$$p(c | Doc) = \frac{p(c) * \prod_{F_i \in Doc} (P(F_i | c))^{TF(F_i, Doc)}}{\sum_{i=1}^n p(c_i) * \prod_{F_i \in Doc} (P(F_i | c))^{TF(F_i, Doc)}} \quad (8.6)$$

Như vậy, trong quá trình phân lớp không dựa vào toàn bộ tập từ vựng mà chỉ dựa vào các từ khoá xuất hiện trong tài liệu Doc.

8.2.3. Thuật toán cây quyết định

Phương pháp học cây quyết định là một trong những phương pháp được sử dụng rộng rãi nhất cho việc học quy nạp từ một tập mẫu lớn. Đây là phương pháp học xấp xỉ các hàm mục tiêu có giá trị rời rạc. Một khác, cây quyết định còn có thể chuyển sang dạng biểu diễn tương đương dưới dạng cơ sở tri thức là các luật *Nếu – Khi* (*If – Then*).

Hình 8.3 là một ví dụ về cây quyết định. Mỗi một nút của cây biểu diễn một thuộc tính trong tập huấn luyện, mỗi một nhánh tới nút tương ứng với một trong những giá trị có thể cho thuộc tính này. Trong ví dụ này, các thuộc tính được biểu diễn dưới dạng nhị phân với các giá trị 0 và 1.

Dưới đây là mô tả thuật toán ID3 – thuật toán học cây quyết định đơn giản nhất [Mit97, Qui86].

a) ID3 (Examples, Target_attribute, Attributes)

Examples ở đây là tập các ví dụ huấn luyện; *Target_attribute* là những thuộc tính đầu ra cho cây quyết định dự đoán; *Attributes* là một danh sách các thuộc tính khác tham gia trong quá trình học của cây quyết định. Kết quả thủ tục trả về cây quyết định phân lớp đúng các mẫu ví dụ đưa ra.

- Tạo một nút gốc *Root* cho cây quyết định
- Nếu toàn bộ *Examples* đều là các ví dụ dương. Trả lại cây *Root* một nút đơn, với nhãn +.
- Nếu toàn bộ *Examples* đều là các ví dụ âm. Trả lại cây *Root* một nút đơn, với nhãn -.
- Nếu *Attributes* là rỗng thì trả lại cây *Root* một nút đơn với nhãn gán bằng giá trị phổ biến nhất của *Target_attribute* trong *Examples*.
- Ngược lại: Begin
 - * $A \leftarrow$ Thuộc tính từ tập *Attributes* mà phân lớp tốt nhất tập *Examples*.
 - * Thuộc tính quyết định cho *Root* $\leftarrow A$
- * For mỗi giá trị có thể *v*, của *A*.
 - Cộng thêm một nhánh cây con ở dưới *Root*, phù hợp với biểu thức kiểm tra $A = v$,
 - Đặt *Examples_v* là một tập con của tập các ví dụ có giá trị *v*, cho *A*
 - Nếu *Examples_v* rỗng

Thì dưới mỗi nhánh mới thêm một nút là với nhãn = giá trị phổ biến nhất của *Target_attribute* trong tập *Examples*.

Ngược lại thì dưới nhánh mới này thêm một cây con

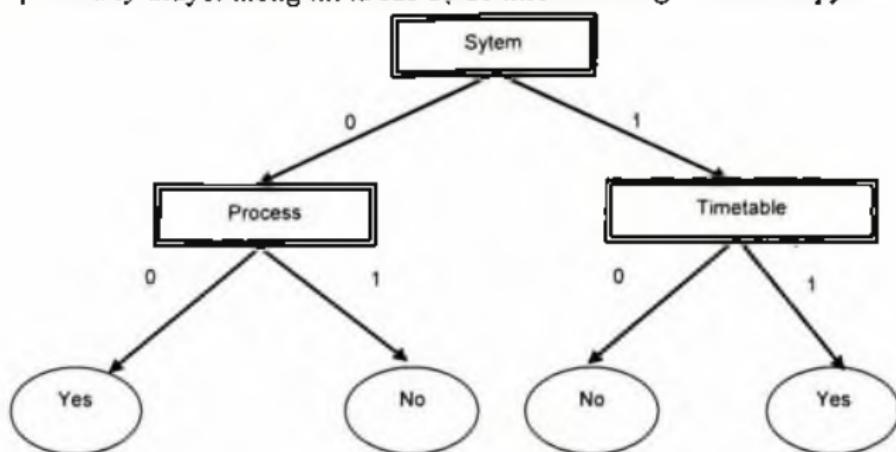
ID3 (Examples_v, Target_attribute, Attribute-{A})

- End
- Return *Root*

Thuộc tính tốt nhất là thuộc tính có độ đo thông tin (information gain) lớn nhất.

b) Chọn lựa thuộc tính tốt nhất

Vấn đề trung tâm của thuật toán ID3 là chọn lựa thuộc tính tốt nhất để đưa vào mỗi nút của cây. Để giải quyết vấn đề này, người ta sử dụng các kết quả của lý thuyết thông tin là các độ đo information gain và entropy.



Hình 8.3. Một ví dụ về cây quyết định

• Entropy

Entropy là đại lượng đo tính đồng nhất hay tính thuần nhất của các mẫu. Entropy là đại lượng hết sức quan trọng trong lý thuyết thông tin. Giả sử đưa ra một tập S có chứa các mẫu ví dụ dương (positive) và mẫu ví dụ âm (negative), như vậy ta có hai lớp phân biệt. Khi đó entropy của tập S được định nghĩa như sau:

$$\text{Entropy}(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$$

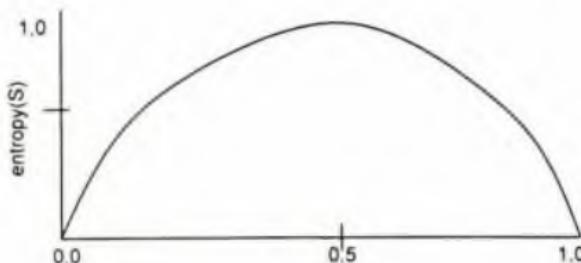
trong đó p_0 là phân bố của các ví dụ dương trong S và p_1 là phân bố của các ví dụ âm trong S, ta định nghĩa $0 \log 0 = 0$.

Để dễ hình dung, giả sử S là một tập hợp gồm 10 mẫu ví dụ (5 ví dụ âm và 5 ví dụ dương), ký hiệu là [5+, 5-]. Khi đó, đại lượng entropy S liên quan tới việc phân bố của hai lớp dương và âm trong tập S là:

$$\text{Entropy}([5+, 5-]) = -(5/10)\log_2(5/10) - (5/10)\log_2(5/10) = 1,0$$

Ta đã xét trường hợp đặc biệt đối với tập có số lượng phân bố mỗi lớp là như nhau. Đại lượng entropy trong trường hợp hai lớp nằm trong khoảng 0 và 1.

Các giá trị của entropy phụ thuộc vào phân bố của một lớp được mô tả trong hình 8.4.



Hình 8.4. Mối liên hệ giữa entropy và phân bố của p_i .

Chú ý rằng, entropy là 0 nếu tất cả các phần tử của S thuộc vào cùng một lớp. Chẳng hạn, nếu tất cả phần tử là dương ($p_0 = 1$), khi đó $p_0 = 0$ và $\text{Entropy}(S) = -1,0 \cdot \log_2(1) - 0 \cdot \log_2 0 = 0$. Entropy là 1 khi tập hợp chứa số mẫu dương bằng số mẫu âm.

Ở trên ta xét trường hợp phân lớp thành hai lớp, đối với trường hợp tổng quát thì đại lượng entropy được xác định như sau:

$$\text{Entropy}(S) = - \sum_{i=1}^k p_i \log_2 p_i \quad (8.7)$$

trong đó p_i là phân bố của S thuộc vào lớp i. Chú ý là đại lượng logarithm vẫn là cơ số 2 bởi entropy là đại lượng trong lý thuyết thông tin dựa vào việc mã hóa trên các bit. Lúc này, entropy có thể lớn hơn 1.

• Information Gain

Entropy là đại lượng đo độ không đồng nhất trong một tập các mẫu. Người ta đưa ra một độ đo xác định ảnh hưởng của một thuộc tính trong mẫu đó trong việc phân lớp, đại lượng đó là information gain, tạm dịch là độ lấy thông tin, phản tiếp theo vẫn giữ nguyên tên tiếng Anh như ban đầu.

Information Gain của một thuộc tính A trong tập hợp S, ký hiệu là $\text{Gain}(S, A)$ được xác định như sau:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (8.8)$$

trong đó $\text{Values}(A)$ là tập các giá trị có thể của thuộc tính A, còn S_v là tập con của S mà A có giá trị v (tức là $S_v = \{s \in S | A(s) = v\}$).

Biểu thức đầu $\text{Entropy}(S)$ là đại lượng entropy nguyên thuỷ của tập S. Biểu thức sau là giá trị kỳ vọng của entropy sau khi S được chia theo thuộc tính A. Giá trị kỳ vọng của entropy trong biểu thức thứ hai đơn giản chỉ là

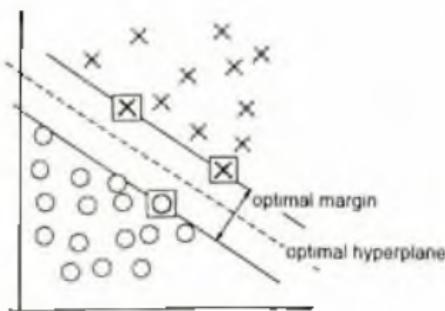
tổng các giá trị entropy của các S_v , nhân với tỷ số các ví dụ $\frac{|S_v|}{|S|}$ mà thuộc vào S_v . $\text{Gain}(S, A)$ do đó là độ giảm kỳ vọng (expected entropy) trong entropy khi biết giá trị các thuộc tính A. Nói cách khác, $\text{Gain}(S, A)$ là thông

tin cung cấp về giá trị hàm mục tiêu khi biết các giá trị của thuộc tính A. Giá trị của Gain(S, A) là số các bit được lưu khi mã hoá các giá trị mục tiêu của một thành phần của S, khi biết các giá trị của thuộc tính A.

8.2.4. Thuật toán SVM

Thuật toán máy vector hỗ trợ (Support Vector Machine – SVM) được áp dụng cho phân lớp nhị phân. Cơ sở của thuật toán là dựa trên phương pháp tiếp cận thống kê được Vapnik đề xuất [Vap98]. SVM rất hiệu quả để giải quyết các bài toán với dữ liệu có số chiều lớn, như các vector biểu diễn văn bản [CV95, DPH98, HAN90, KC05, VK06] và được coi là một trong 10 thuật toán khai phá dữ liệu điển hình nhất [WKQ08].

Ý tưởng cơ bản của SVM là cực tiểu hóa giá trị gọi là rủi ro cấu trúc thực nghiệm (empirical structure risk) [Vap98]. Bằng việc đề xuất khái niệm chiều VC (Vapnik và Chervonenkis), thực hiện một ánh xạ từ không gian dữ liệu đầu vào sang một không gian khác gọi là *không gian thuộc tính*. Qua ánh xạ này, bài toán phân lớp được chuyển thành bài toán đi tìm một siêu phẳng tốt nhất phân chia không gian thuộc tính đó thành hai nửa không gian, một chứa các ví dụ dương và nửa không gian còn lại chứa ví dụ âm. Bài toán tìm siêu phẳng tương đương với bài toán quy hoạch toàn phương (quadratic programming problem) và vì vậy sử dụng các thuật toán quy hoạch toàn phương để tìm ra siêu phẳng tốt nhất. Siêu phẳng tốt nhất là siêu phẳng mà khoảng cách từ nó tới tập ví dụ học là nhỏ nhất. Tập các vector hỗ trợ hình thành mặt phẳng tốt nhất được gọi là các vector hỗ trợ. Hình 8.5 mô tả một minh họa hình học cho thuật toán SVM bao gồm cả chi dẫn các vector hỗ trợ.



Hình 8.5. Minh họa hình học thuật toán máy vector hỗ trợ [CV95]

Điểm mang dấu cộng/hình tròn biểu diễn mẫu dương/mẫu âm. Các đường thẳng biểu diễn các siêu phẳng quyết định, trong đó đường rời nét (nằm giữa hai đường song song) là siêu phẳng tốt nhất vì khoảng cách của nó tới các tập huấn luyện là nhỏ nhất. Các hộp vuông nhỏ chứa các dấu cộng và hình tròn biểu diễn các vector hỗ trợ

Theo phương diện toán học, bài toán được phát biểu như sau:

Đầu vào: Cho tập dữ liệu học: $D = \{d_i = (x_i, y_i), i = 1, \dots, n\}$ với

$$y_i \in \{-1, 1\} \text{ xác định dữ liệu dương hay âm}$$

Đầu ra: Tìm siêu phẳng $f(d) = \alpha_{SVM} \cdot d + b$ phân chia tập dữ liệu học thành hai miền rời nhau với khoảng cách của siêu phẳng tới tập ví dụ học D là lớn nhất.

Sử dụng thuật toán SVM với một dữ liệu mới d , tính giá trị của $f(d)$ và phân d vào lớp dương khi $f(d) > 0$, d vào lớp âm khi $f(d) < 0$ (trường hợp $f(d) = 0$).

Trong [CV95], Corinna Cortes và Vladimir Vapnik trình bày khá chi tiết về thuật toán phân lớp SVM. Dưới đây giới thiệu sơ bộ về một số nội dung cơ bản về SVM.

a) Thuật toán siêu phẳng tối ưu

Một tập ví dụ mẫu $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) | y_i \in \{-1, 1\}\}$ được gọi là *tách* được *tuyến* nếu như tồn tại một vector w và một giá trị vô hướng b sao cho

$$w.x_i + b \geq 1 \text{ nếu } y_i = 1 \text{ và } w.x_i + b \leq -1 \text{ nếu } y_i = -1 \quad (8.9)$$

với mọi ví dụ mẫu thuộc D hay $\forall i = 1, 2, \dots, n$.

Đây là điều kiện cần và đủ để thuật toán SVM tìm ra siêu phẳng tối ưu tách hai miền dữ liệu dương và âm. Trong thực tiễn, vì w và b là hai tham số hệ thống cần phải tìm, cho nên trước hết áp dụng thuật toán để xác định hai tham số này và sau đó đánh giá hiệu lực bộ phân lớp để kiểm tra điều kiện (8.9) đối với các dữ liệu kiểm tra.

Các bất đẳng thức (8.9) được trình bày ở dạng thu gọn hơn là:

$$y_i(w.x_i + b) \geq 1, \forall i = 1, 2, \dots, n \quad (8.10)$$

Siêu phẳng tối ưu:

$$w^*.x + b^* = 0 \quad (8.11)$$

là siêu phẳng duy nhất tách tập dữ liệu học với lề tốt nhất theo nghĩa xác định hướng $w/|w|$ mà khoảng cách từ các hình chiếu của các vector học của hai lớp khác nhau là lớn nhất như Hình 8.5. Khoảng cách nói trên được ký hiệu là $\rho(w, b)$, được tính bằng

$$\rho(w, b) = \min_{\{x, y=1\}} \frac{x.w}{|w|} - \max_{\{x, y=-1\}} \frac{x.w}{|w|} \quad (8.12)$$

Như vậy, siêu phẳng tối ưu (w^*, b^*) là giá trị tham số làm cực đại khoảng cách (8.12). Từ (8.12) và (8.10) nhận được

$$\rho(w^*, b^*) = \frac{2}{|w^*|} = \frac{2}{\sqrt{w^*.w^*}} \quad (8.13)$$

Điều đó có nghĩa là siêu phẳng tối ưu là cái duy nhất làm cực tiểu $w.w$ dưới ràng buộc (8.10). Cực đại giá trị hàm mục tiêu (8.13) với ràng buộc (8.10) là một bài toán tối ưu toàn phương.

Vector x , thỏa mãn $y_i(w.x_i + b) = 1$ được gọi là vector hỗ trợ. Corinna Cortes và Vladimir Vapnik [CV95] chứng tỏ rằng vector w^* xác định siêu phẳng tối ưu có thể được biểu diễn dưới dạng tò hợp tuyến tính của các vector ví dụ học:

$$w^* = \sum_{i=1}^n y_i \alpha_i^* x_i \quad (8.14)$$

trong đó $\alpha_i^* \geq 0$. Do $\alpha > 0$ chỉ đối với vector hỗ trợ, công thức (8.14) cho một dạng biểu diễn có dạng của w^* . Như vậy, việc tìm ra các thành phần biểu diễn vector tham số w^* như sau:

$$\hat{w}^T = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*) \quad (8.15)$$

và đưa về giải bài toán quy hoạch toàn phương:

$$W(\hat{w}) = \hat{w}^T \cdot 1 - \frac{1}{2} \hat{w}^T \cdot D \cdot \hat{w} \quad (8.16)$$

với $\hat{w}^T = (\alpha_1, \alpha_2, \dots, \alpha_n)$, với các ràng buộc:

$$\hat{w} \geq 0 \quad (8.17)$$

$$\hat{w}^T Y = 0 \quad (8.18)$$

trong đó, $1^T = (1, 1, \dots, 1)$ là vector đơn vị n thành phần, $Y^T = (y_1, y_2, \dots, y_n)$ là vector n thành phần các nhãn, còn D là ma trận đối xứng với các phần tử:

$$D_{ij} = y_i y_j x_i \cdot x_j \quad \forall i, j = 1, 2, \dots, n \quad (8.19)$$

Khi dữ liệu học D tách được theo (8.9), tồn tại quan hệ sau đây giữa cực đại hamin (8.16), cặp (w^*, b^*) và cực đại lề từ (8.13):

$$W(w^*) = \frac{2}{\rho^2} \quad (8.20)$$

Nếu có \hat{w}^* nào đó và hằng W^* nào đó thoả mãn $W(\hat{w}^*) > W^*$, có thể khảng định được mọi siêu phẳng tách tập dữ liệu học sẽ có lề:

$$\rho < \sqrt{\frac{2}{W^*}}$$

Như vậy, lời giải bài toán toàn phương tách lề cực đại tập dữ liệu ví dụ học.

b) Phần mềm nguồn mở về SVM

Phần mềm nguồn mở thuật toán SVM đã được công bố trên nhiều trang Web, chẳng hạn như: <http://people.cs.uchicago.edu/~vikass/svmLin.html> [KC05, VK06], <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> và http://www-ai.cs.uni-dortmund.de/svm_light.

c) Tập dữ liệu học không tách tuyến tính

Khi áp dụng thuật toán SVM, song kết quả đánh giá cho chất lượng không như ý muốn, giả thiết hai lớp dữ liệu tách được tuyến tính bị vi phạm, cần phải điều chỉnh thuật toán SVM.

Trong một số trường hợp, sử dụng thuật toán SVM với siêu phẳng lề mềm (The soft margin hyperplane) [CV95, Joa99]. Nếu thuật toán SVM với siêu phẳng lề mềm cũng không hiệu quả, các phép biến đổi không gian vector được thực hiện bằng các phép biến đổi chiêu VC thích hợp khác. Một

trong những nội dung cốt lõi về các phép biến đổi chiêu VC là lựa chọn các hàm nhân phù hợp với miền dữ liệu của bài toán [Vap98].

8.2.5. Thuật toán k-người láng giềng gần nhất

Thuật toán hoạt động không dựa vào tập từ vựng. Tuy nhiên, nó vẫn sử dụng ngưỡng CtgTsh, và thực hiện theo các bước như đã đề cập ở trên. Đó là tiến hành ngẫu nhiên k tài liệu và tính xác suất $p(c|Doc)$ dựa trên sự giống nhau giữa tài liệu Doc và k tài liệu được chọn. Xác suất $p(c|Doc)$ được tính theo công thức sau:

$$p(c|DOC) = \frac{\sum_{j=1}^k Sm(Doc, D_j) * p(c|D_j)}{\sum_{i=1}^n \sum_{j=1}^k Sm(Doc, D_j) * p(c|D_j)} \quad (8.21)$$

Trong đó, n: số lớp; k: số tài liệu được chọn để so sánh; $p(c|D_j)$: có giá trị 0 hoặc 1, cho biết tài liệu D_j có thuộc lớp c_i không. Sở dĩ có giá trị này vì một tài liệu có thể thuộc hơn một lớp: $Sm(Doc, D_j)$ xác định mức độ giống nhau của tài liệu Doc với tài liệu được chọn D_j , được tính bằng cos của góc giữa hai vector biểu diễn tài liệu Doc và tài liệu được chọn D_j .

$$Sm(Doc, D_i) = \text{Cos}(Doc, D_i) = \frac{\sum X_i * Y_i}{\sqrt{\sum_i X_i^2 \sum_i Y_i^2}} \quad (8.22)$$

Cách biểu diễn tài liệu trong thuật toán này hoàn toàn tương tự như trong thuật toán phân lớp Bayes thứ nhất. Trong công thức (8.22), X_i là tần suất xuất hiện từ khoá thứ i trong tài liệu Doc, còn Y_i là tần suất xuất hiện của từ khoá thứ i trong tài liệu D_i .

8.3. Học bán giám sát và một số thuật toán phân lớp bán giám sát

Các thuật toán đã trình bày ở trên có đặc điểm là chỉ có thể học từ dữ liệu đã gán nhãn, việc tạo ra các dữ liệu gán nhãn thường là công việc buồn tẻ, nhưng lại tốn công sức. Ngược lại trong thực tế, các dữ liệu chưa gán nhãn thường tồn tại với số lượng lớn (chẳng hạn như nguồn dữ liệu các trang Web từ Internet) như mô tả tại Hình 8.8. Nếu tận dụng được các nguồn dữ liệu chưa gán nhãn thì sẽ làm giảm được công sức tạo dữ liệu cũng như làm tăng được chất lượng của các bộ phân lớp. Hiện tại đã có rất nhiều nghiên cứu và đề xuất các giải thuật có khả năng sử dụng dữ liệu gán nhãn, đồng thời tận dụng cả dữ liệu chưa gán nhãn để làm phong phú thêm

dữ liệu huân luyện, nhằm làm tăng chất lượng phân lớp. Các giải thuật có đặc điểm này được phân vào lớp giải thuật học bán giám sát.

Mục này sẽ trình bày một số nét khái quát về học bán giám sát và giới thiệu một số thuật toán bán giám sát.

8.3.1. Khái quát về học bán giám sát

Như đã giới thiệu tại Chương 3, Xiaojin Zhu

[Zhu08] không chỉ cung cấp một danh sách các công trình nghiên cứu liên quan tới các mô hình và giải pháp dựa trên đồ thị và học bán giám sát mà tác giả còn trình bày những nội dung cơ bản trong tiếp cận học bán giám sát. Những câu hỏi cần được làm rõ về học bán giám sát là:

- Học bán giám sát là gì?
- Tại sao lại sử dụng cách tiếp cận học bán giám sát?
- Nội dung cụ thể của học bán giám sát là gì hoặc học bán giám sát như thế nào?

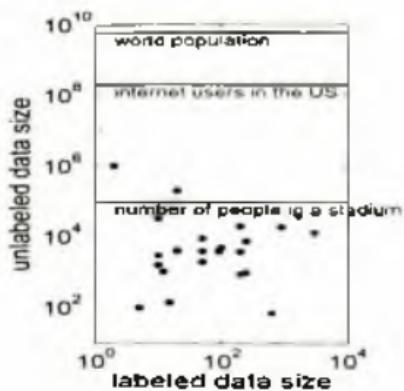
Câu hỏi cuối cùng được giải đáp thông qua việc khảo sát các thuật toán phân lớp bán giám sát được trình bày ở sau bao gồm thuật toán *học bán giám sát cực đại kỳ vọng địa phương*, thuật toán Self-training, hai loại thuật toán Co-training và thuật toán Tri-training. Các nội dung tiếp theo giải đáp cho hai câu hỏi đầu tiên.

• *Học bán giám sát là gì?*

Trong học giám sát, tập nhãn lớp đã cho $c = \{c_i\}$ và toàn bộ các ví dụ học đều đã có nhãn lớp. Trong học không giám sát, vì không có tập nhãn lớp c cho nên không có nhãn lớp đối với các ví dụ học. Các phân tích trên đây cho ra một gợi ý về khái niệm học bán giám sát là "*học bán giám sát là một kiểu học đặc biệt, trong đó sử dụng cả ví dụ có gán nhãn và ví dụ chưa có nhãn*". Học bán giám sát có thể được sử dụng cho cả học giám sát (bài toán phân lớp) và học không giám sát (bài toán phân cụm).

• *Tại sao lại là phương pháp học bán giám sát?*

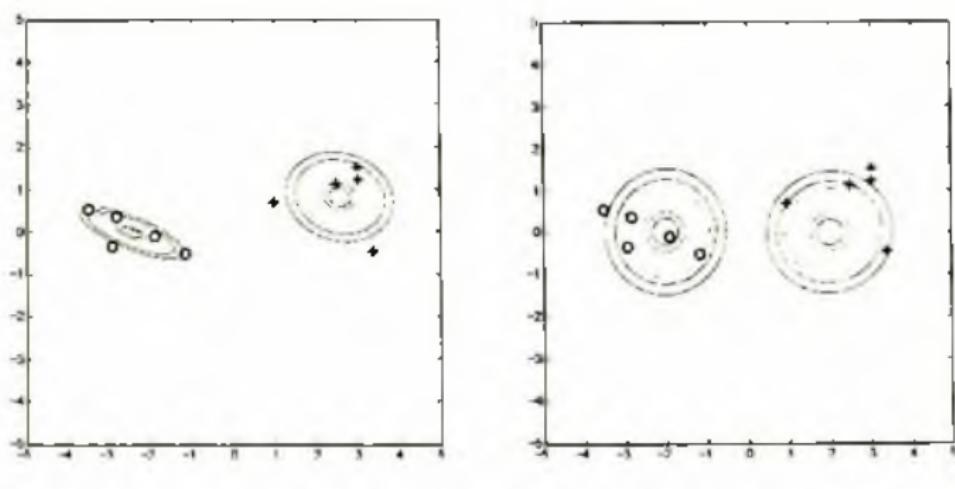
Một trong những yêu cầu cốt lõi của bài toán khai phá dữ liệu là dữ liệu học (dữ liệu đầu vào) của thuật toán khai phá dữ liệu phải đại diện cho toàn bộ dữ liệu của miền ứng dụng. Như đã được giới thiệu ở Chương 1, để đảm



Hình 8.6. Kích thước dữ liệu có nhãn và dữ liệu không có nhãn [Zhu08]

bảo tính đại diện cho toàn bộ miền dữ liệu thì phải nắm bắt được mô hình sinh tập dữ liệu trong miền dữ liệu. Trong nhiều trường hợp là không thể có mô hình sinh dữ liệu, vì vậy một cách suy nghĩ rất hợp tự nhiên là chọn một cách ngẫu nhiên và càng nhiều càng tốt các dữ liệu học từ miền ứng dụng. Trong học bán giám sát, ví dụ học trong tập ví dụ học phải được gán nhãn hoặc để làm ví dụ xây dựng mô hình học hoặc được dùng làm ví dụ kiểm tra. Công việc gán nhãn được tiến hành hầu như bằng thủ công (trong một số trường hợp có thể sử dụng cách thức bán tự động) và luôn đòi hỏi chúng ta phải dành nhiều thời gian, công sức và hiện nhiên là kinh phí để gán nhãn dữ liệu. Vì vậy, cách tiếp cận học bán giám sát được sử dụng để khai thác được dữ liệu chưa có nhãn. Như vậy, dữ liệu chưa có nhãn có thể bổ sung được thông tin cần thiết về miền ứng dụng, bù đắp được sự thiếu hụt của tập ví dụ có nhãn hiện đang có.

Hình 8.9 cung cấp một cái nhìn trực quan về việc sử dụng ví dụ chưa có nhãn đối với bài toán phân lớp nhị phân. Hình 8.9a cho thấy, nếu chỉ sử dụng các ví dụ có nhãn thì thuật toán phân lớp giám sát sẽ đưa ra mô hình lệch so với thực tiễn miền ứng dụng, trong đó bộ phân lớp mô tả lớp các dữ liệu 0 là quá lệch so với bản chất của nó. Trong khi đó, Hình 8.9b lại cho thấy, vì được bổ sung tri thức miền từ các ví dụ chưa có nhãn mang tới, mô hình học phân lớp phù hợp với miền dữ liệu hơn nhiều.



Hình 8.7. Sử dụng dữ liệu học trong mô hình học giám sát (a) và mô hình học bán giám sát (b) [Zhu08]

Về mặt lý luận, chúng ta xem xét tác dụng của ví dụ chưa có nhãn, chẳng hạn trong học máy Bayes thì xác suất hậu nghiệm được tính theo công thức sau (xem Chương 3):

$$p(e|D) = \frac{p(D|e)p(e)}{p(D)} \quad (8.23)$$

Các xác suất tiên nghiệm thuộc về phái của công thức (8.23) thường được ước lượng nhờ vào tập dữ liệu ví dụ mẫu. Chẳng hạn như, xác suất $p(w)$ về sự xuất hiện từ khoá w thường được xấp xỉ bằng tỷ số giữa tổng số lần xuất hiện của từ khoá w với tổng số lần xuất hiện của tất cả các từ khoá trong tập dữ liệu học có nhãn. Trong trường hợp số lượng ví dụ có nhãn rất nhỏ, giá trị tỷ số trên không thể đại diện tốt cho $p(w)$ được. Nếu giá trị xấp xỉ cho $p(w)$ cũng được tính bằng tỷ số nói trên những mở rộng trên tập ví dụ có nhãn và không nhãn thì tính đại diện của nó sẽ tốt hơn. Hình 8.9 cho một minh họa tốt về nội dung này.

• Phạm vi tác dụng của ví dụ không nhãn

Ví dụ không nhãn có tác dụng hỗ trợ để cung cấp một số tri thức miền toàn diện hơn, tuy nhiên, ví dụ không nhãn cũng có phạm vi tác dụng của nó. Ví dụ không nhãn cũng có thể làm sai lệch tri thức miền nếu việc lựa chọn chúng không đảm bảo tính đại diện cho miền ứng dụng. Chúng ta luôn nhấn mạnh về tính đại diện cho miền ứng dụng của dữ liệu học. Khi ví dụ có nhãn ít thì dù có được lựa chọn cẩn thận, song chúng vẫn có hạn chế về tính đại diện cho miền ứng dụng, và vì vậy cần khai thác dữ liệu không có nhãn. Việc lựa chọn dữ liệu không nhãn làm dữ liệu học cũng phải được thi hành theo định hướng về tính đại diện. Nếu việc lựa chọn không dựa theo mô hình sinh dữ liệu, không thực hiện một cách ngẫu nhiên thì sẽ tạo nên sự lệch lạc về tri thức miền. Theo Xiaojin Zhu [Zhu08], trong các nghiên cứu của Elworth (vào năm 1994) và Cozman và cộng sự (vào năm 2003), nội dung này đã được thảo luận.

Theo Xiaojin Zhu [Zhu08], tồn tại rất nhiều các thuật toán học bán giám sát thuộc vào khoảng 190 công trình khoa học được công bố về học bán giám sát. Dưới đây là một số thuật toán điển hình trong số đó.

8.3.2. Thuật toán bán giám sát cực đại EM địa phương

Thuật toán *học bán giám sát cực đại kỳ vọng* (EM) *địa phương* thuộc loại thuật toán trong mô hình sinh. Mô hình hoạt động dựa trên giả thiết Bayes $p(x, y) = p(y)*p(x|y)$. Với số lượng nhiều dữ liệu chưa nhãn cho $P(x|y)$ mô hình đồng nhất, tài liệu được phân thành các thành phần mà trong trường hợp lý tưởng, trong mô hình "đồng nhất", mọi đối tượng trong một thành phần cùng chung một nhãn, vì vậy, chỉ cần biết nhãn của một đối tượng nào đó trong thành phần là kết luận được nhãn toàn bộ các đối tượng trong thành phần đó.

Tính đồng nhất được phát biểu như sau: "Cho họ phân bố $\{p_\beta\}$ là đồng nhất nếu $\theta_1 \neq \theta_2$ thì $p_{\theta 1} \neq p_{\theta 2}$ ". Từ tính chất này dẫn tới tính khả tách của

phân bổ tới các thành phần. Khi quan tâm tới tính xác thực của mô hình, có thể thấy trong trường hợp khi giả thiết mô hình trọn là chính xác thì việc bổ sung các dữ liệu không nhãn sẽ làm tăng độ chính xác phân lớp.

Ký hiệu:

D: tập ví dụ đã có (có nhãn/chưa có nhãn)

D^K: tập ví dụ có nhãn trong D ($|D^K| < |D|$). D^K gồm D_{Train} và D_{Test}.

Thuật toán.

1. Cố định tập tài liệu không nhãn $D^U \subseteq D \setminus D^K$ dùng trong E-bước và M-bước
2. Dùng D_{Train} xây dựng mô hình ban đầu Θ_0
3. for $i = 0, 1, 2, \dots$ cho đến khi kết quả đảm bảo do
4. for mỗi tài liệu $d \in D^U$ do
5. E-bước: dùng phân lớp Bayes thứ nhất xác định xác suất hậu nghiệm $P(c|d, \Theta_i)$
6. end for
7. for mỗi lớp c và từ khoá t do
8. M-bước: xác định $\theta_{c,t}$ dùng công thức tính các xác suất tiên nghiệm để xây dựng mô hình $i + 1$
9. end for
10. end for

Một câu hỏi được đặt ra với thuật toán EM về điều kiện kết thúc thuật toán là "kết quả đảm bảo" được hiểu như thế nào? Sử dụng tập dữ liệu D_{Test} để kiểm tra các mô hình Θ_i ; hoặc chất lượng mô hình Θ_{i+1} không tốt hơn mô hình Θ_i , thì thuật toán dừng, hoặc đạt tới độ hiệu quả cho trước.

8.3.3. Thuật toán Self-training

Self-training là một phương pháp được sử dụng phổ biến trong học bài giám sát. Trong Self-training, một tập phân lớp ban đầu được huấn luyện cùng với số lượng nhỏ dữ liệu gán nhãn. Tập phân lớp sau đó sẽ được dùng để gán nhãn cho dữ liệu chưa gán nhãn. Điều hình là hầu hết các điểm chưa gán nhãn có tin cậy cao, cũng như cùng với các nhãn dự đoán trước của chúng, được chèn thêm vào tập huấn luyện. Sau đó tập phân lớp sẽ được huấn luyện lại và lặp lại các quy trình [Zhu05, Zhu08]. Chú ý rằng, tập phân lớp sử dụng các dự đoán của nó để dạy chính nó. Quy trình này được gọi là self-teaching hay là bootstrapping. Thuật toán EM địa phương được trình bày phía trên là dạng đặc biệt của self-training.

Self-training được áp dụng để xử lý các bài toán của một số ngôn ngữ tự nhiên. Ngoài ra, Self-training còn được áp dụng để phân tích và dịch máy. Trong thuật toán Self-training, sử dụng một thuật toán phân lớp giám sát h và gọi h là thuật toán "nền" của thuật toán Self-training.

Ký hiệu:

L: Tập các dữ liệu gán nhãn.

U: Tập các dữ liệu chưa gán nhãn

Thuật toán:

Loop (cho đến khi $U = \emptyset$)

 Huấn luyện bộ phân lớp giám sát h trên tập L

 Sử dụng h để phân lớp dữ liệu trong tập U

 Tìm tập con $U' \subseteq U$ có độ tin cậy cao nhất:

$$L + U' \Rightarrow L$$

$$U - U' \Rightarrow U$$

Trong nội dung thuật toán tổng quát trên còn một vấn đề cần xem xét chính là vấn đề tìm tập U' có "độ tin cậy cao nhất". Trong một số trường hợp có thể sử dụng thủ tục bootstrapping trong thuật toán Self-training.

8.3.4. Thuật toán Co-training dựa trên việc phân chia khung nhìn

Co-training dựa trên giả thiết rằng, các thuộc tính có thể được phân chia thành hai tập. Một tập có khả năng huấn luyện, còn một tập phân lớp tốt. Hai tập con này độc lập có điều kiện (conditionally independent) với lớp đã biết ban đầu. Đầu tiên hai tập phân lớp phân tách thành dữ liệu huấn luyện và dữ liệu gán nhãn trên hai tập thuộc tính được tách biệt. Sau đó, mỗi tập phân lớp lại phân lớp các dữ liệu chưa gán nhãn và "dạy" tập phân lớp khác cùng với một vài mẫu chưa gán nhãn (và các nhãn dự đoán) mà chúng cảm giác có độ tin cậy cao. Để có kết quả chính xác hơn, chỉ thêm các dữ liệu được gán nhãn tự động bởi các bộ phân lớp vào tập dữ liệu gán nhãn ban đầu nếu cả hai bộ phân lớp cùng có chung kết quả phân lớp với độ tin cậy cao. Cuối cùng, mỗi tập phân lớp sẽ được huấn luyện lại cùng với các dữ liệu huấn luyện chèn thêm được cho bởi tập phân lớp khác và bắt đầu tiến trình lặp. Giải thuật tổng quát có thể được mô tả như sau:

Đầu vào: Tập dữ liệu đã được gán nhãn (X_L, Y_L), trong đó Y_L là các nhãn; tập dữ liệu không gán nhãn X_U ; giải thuật phân lớp f;

Đầu ra: Tập dữ liệu gán nhãn (X_L, Y_L) đã được làm giàu và bộ phân lớp đã được huấn luyện từ tập dữ liệu đã được làm giàu (X_L, Y_L).

1. Tạo hai tập thuộc tính độc lập nhau trên tập dữ liệu huấn luyện X_L , ký hiệu là $X_L^{(1)}$ và $X_L^{(2)}$.

- Huấn luyện giải thuật phân lớp f trên tập thuộc tính $X_L^{(1)}$ của tập dữ liệu ($X_L^{(1)}, Y_L$) ta thu được bộ phân lớp f ⁽¹⁾,
- Huấn luyện giải thuật phân lớp f trên tập thuộc tính $X_L^{(2)}$ của tập dữ liệu ($X_L^{(2)}, Y_L$) ta thu được bộ phân lớp f ⁽²⁾.

2. Phân lớp X_U với f ⁽¹⁾ và f ⁽²⁾ tách biệt nhau.

- Chèn thêm k phần tử dữ liệu có độ tin cậy cao nhất do $f^{(1)}$ thực hiện trên dữ liệu X_U vào tập dữ liệu gán nhãn của bộ phân lớp $f^{(2)}$: $(X_L^{(2)}, Y_L) = (X_L^{(1)}, Y_L) \cup k\text{-most-confident}(x, f^{(1)}(x)), x \in X_U$.
- Chèn thêm k phần tử dữ liệu có độ tin cậy cao nhất do $f^{(2)}$ thực hiện trên dữ liệu X_U vào tập dữ liệu gán nhãn của bộ phân lớp $f^{(1)}$: $(X_L^{(1)}, Y_L) = (X_L^{(1)}, Y_L) \cup k\text{-most-confident}(x, f^{(2)}(x)), x \in X_U$. Chú ý là thực hiện việc thêm dữ liệu vào tập gán nhãn phải đảm bảo sao cho điều kiện sau được thoả mãn: $k\text{-most-confident}(x, f^{(2)}(x)) = k\text{-most-confident}(x, f^{(1)}(x))$. Điều này dựa trên ý tưởng khi hai bộ phân lớp (dựa trên hai khung nhìn độc lập) cùng có chung một kết quả phân lớp cho một phần tử dữ liệu thì khả năng phân tử dữ liệu đó được phân lớp đúng là cao, do đó có thể đưa thêm vào tập dữ liệu gán nhãn ban đầu.
- $X_U = X_U \setminus \{x \mid (x, f^{(1)}(x)) \in k\text{-most-confident}(x, f^{(1)}(x))\}$

3. Lặp lại từ bước 1.

Điều kiện dừng của thuật toán là hoặc tập dữ liệu chưa gán nhãn là rỗng, hoặc số vòng lặp đạt tới một ngưỡng được xác định trước.

Khi làm việc với bộ phân lớp Co-training cần lưu ý một số vấn đề sau đây. *Thứ nhất*, tập dữ liệu gán nhãn có ảnh hưởng lớn đến hiệu quả của thuật toán Co-training. Nếu tập này quá ít thì sẽ không hỗ trợ Co-training. Trong trường hợp quá nhiều, thì thực sự không thu được lợi ích từ Co-training. *Thứ hai*, cơ sở tăng hiệu quả Co-training là vấn đề thiết lập các tham số trong thuật toán như kích cỡ tập dữ liệu gán nhãn, kích cỡ tập dữ liệu chưa gán nhãn, số mẫu được thêm vào sau mỗi vòng lặp. Trong mọi trường hợp, việc chọn bộ phân lớp thành phần cho từng khung nhìn là rất quan trọng [ZL05, Zhu05, Zhu08].

8.3.5. Thuật toán Co-training dựa trên sự cộng tác của các giải thuật phân lớp khác nhau

Việc chia tập thuộc tính như trên thành 2 tập con không giao nhau và độc lập có điều kiện với nhau ở trên có hạn chế là tập con các thuộc tính thường không đại diện cho việc biểu diễn đầy đủ dữ liệu. Do đó, chất lượng của bộ phân lớp trên tập con thuộc tính $X_L^{(1)}$ sẽ có thể thấp hơn chất lượng của bộ phân lớp được thực hiện trên toàn bộ tập thuộc tính của X_L . Hay trong một số trường hợp, khi số lượng thuộc tính là rất ít, thi việc tạo ra 2 tập con có thể là không khả thi. Một trong các đề xuất giải pháp để khắc phục trường hợp trên là sử dụng các giải thuật phân lớp khác nhau, thay vì một giải thuật như trên. Việc chọn các giải thuật phân lớp được xây dựng trên các mô hình học máy khác nhau, sao cho việc phân lớp của các giải thuật phân lớp là độc lập nhau cho từng phần tử dữ liệu. Khi đảm bảo được điều này, chúng ta có thể dùng được toàn bộ tập thuộc tính để huấn luyện

cho các bộ phân lớp. Goldman và Zhou là tác giả của giải thuật Co-training theo tư tưởng này, chi tiết của giải thuật có thể xem tại [GZ00].

8.3.6. Thuật toán Tri-Training

Một đề xuất khác cho giải thuật Co-training để xoá bỏ hạn chế là phải dùng 2 giải thuật học khác nhau là của Zhi-Hua Zhou và cộng sự [ZLT05]. Đề xuất của Zhi-Hua Zhou và cộng sự cho phép ta không phải chia tập thuộc tính ra làm 2 khung nhìn khác nhau cũng như không phải sử dụng nhiều giải thuật học khác nhau. Trong giải thuật mới, thay vì sử dụng 2 bộ phân lớp huấn luyện lẫn nhau thì Zhi-Hua Zhou và cộng sự đề xuất sử dụng 3 bộ phân lớp, và cứ 2 bộ phân lớp lại chịu trách nhiệm "huấn luyện" bộ phân lớp còn lại. Giải thuật mới của Zhi-Hua Zhou và cộng sự được đặt tên là Tri-Training. Chi tiết của giải thuật này được minh họa như sau:

Đầu vào: Tập dữ liệu gán nhãn L; tập dữ liệu chưa gán nhãn U và một giải thuật phân lớp Learn;

Đầu ra: Bộ phân lớp kết hợp đã được huấn luyện;

```
for i = 1..3 do
    Si=BootstrapSample(L);
    hi=Learn(Si);
    ei' = 0.5;
    li' = 0;
endfor
repeat until không có hi nào thay đổi
    for i = 1..3 do
        Li = ∅, updatei=false;
        ei=MeasureError(hi & hk) (j, k≠i);
        if (ei< ei') then
            for every x ∈ U do
                if(hj(x) = hk(x)) (j, k≠i) then Li=Li ∪ {(x,hi(x))}
            endfor
            if (li'=0) then li'= ⌈ ei / (ei' - ei) + 1 ⌉;
            if (li' < |Li|) then
                if(ei|Li| < ei'|Li'|) then updatei=true;
                else if (li' > ei'|Li'|) then
                    Li=Subsample(Li, ⌈ ei'|Li'| - 1 ⌉);
                    updatei=true;
            endfor
    for i = 1..3 do
```

```

if(update,==true) then
    hi=Learn(L1 ∪ Li), ei'= ei; i=|Li|;
endfor
endrepeat
Output: h(x)= argmax  $\sum_{y \in \text{label}} \frac{1}{n_i(x)} e_i$ 

```

Vì không sử dụng các khung nhìn (tập con thuộc tính) khác nhau và cũng không sử dụng các giải thuật phân lớp khác nhau, do đó, để tạo cho các bộ phân lớp có khả năng dự đoán độc lập nhau, Tri-Training phải sử dụng phương pháp lấy mẫu (BootstrapSample) để tạo ra các tập huấn luyện ban đầu khác nhau cho 3 bộ phân lớp. Trong quá trình học, nếu 2 bộ phân lớp cùng có chung một kết quả dự đoán cho một phần tử dữ liệu chưa gán nhãn thì phần tử dữ liệu đó được đưa vào tập dữ liệu tiềm năng ký hiệu là L_i. Tập dữ liệu tiềm năng này sau này sẽ được dùng kết hợp với tập dữ liệu gán nhãn đầu vào L để huấn luyện bộ phân lớp h_i. Một điểm khá thú vị ở đây là, tập dữ liệu gán nhãn ban đầu L và tập dữ liệu chưa gán nhãn U sẽ không bị thay đổi trong quá trình học của giải thuật. Một điểm khác biệt nữa là số lượng dữ liệu mới được gán nhãn L, dùng để kết hợp với dữ liệu gán nhãn ban đầu L là được giới hạn. Thủ tục Subsample sẽ đảm bảo việc loại bỏ một số lượng dữ liệu trong tập L, để thu được số lượng dữ liệu phù hợp nhất định trước khi dùng để kết hợp với tập dữ liệu L dùng cho việc huấn luyện h_i. Kết quả cuối cùng của giải thuật là bộ 3 các bộ phân lớp đã được huấn luyện.

Khi dùng Tri-Training (đã được huấn luyện) để phân lớp các phần tử dữ liệu mới x, thì nhãn phân lớp cuối cùng của phần tử x sẽ là nhãn được dự đoán nhiều nhất trong 3 bộ phân lớp.

Giải thuật Tri-Training được đề xuất và chứng minh được tính hiệu quả của nó, tuy nhiên, trong một số bài toán cụ thể, khi mà số lượng các phần tử dữ liệu trong một lớp c nào đó có thể có rất ít. Khi thực hiện việc lấy mẫu BootstrapSample thì các phần tử dữ liệu thuộc lớp c đó có thể bị bỏ qua, dẫn đến khả năng lớp c sẽ không có dữ liệu để huấn luyện, dẫn đến kết quả phân lớp của từng bộ phân lớp h_i sẽ bị giảm đáng kể, và kết quả cuối cùng sẽ không cao. Một đề xuất khác được đưa ra trong khi gặp trường hợp này là thay vì chỉ sử dụng 1 giải thuật phân lớp, ta có thể sử dụng nhiều giải thuật phân lớp khác nhau, do đó, có thể sử dụng toàn bộ tập dữ liệu gán nhãn gốc L ban đầu để huấn luyện các bộ phân lớp [NMS08]. Đề xuất này đã được thử nghiệm trong bài toán phân lớp câu hỏi và đã đem lại kết quả khả quan. Một chú ý trong trường hợp này là, phải chọn các giải thuật phân lớp sao cho chất lượng phân lớp của chúng phải tương đương nhau.

Câu hỏi và bài tập

1. Tạo một tập dữ liệu các trang Web, sau đó chia làm dữ liệu huấn luyện và kiểm tra. Dùng giải thuật học có giám sát Bayes để huấn luyện trên tập dữ liệu huấn luyện, sau đó kiểm tra phân lớp cho tập dữ liệu kiểm tra.
2. Dùng giải thuật học có giám sát kNN để phân lớp cho tập dữ liệu kiểm tra. So sánh kết quả thu được so với thuật toán Bayes.
3. Dùng giải thuật học có giám sát SVM để huấn luyện trên tập dữ liệu huấn luyện, sau đó kiểm tra phân lớp cho tập dữ liệu kiểm tra. So sánh kết quả thu được với thuật toán Bayes và kNN.
4. Lấy thêm một tập dữ liệu mới dùng làm dữ liệu không gán nhãn. Cài đặt giải thuật Selftraining (dùng chung giải thuật phân lớp với bài tập 1) để xem chất lượng của nó có được cải thiện không.
5. Cài đặt giải thuật Co-Training (dùng chung giải thuật phân lớp với bài tập 1 – 3) để xem chất lượng của nó có được cải thiện không.

Chương 9

TRÍCH CHỌN THÔNG TIN TRÊN WEB

9.1. Giới thiệu

Như đã biết, sự phát triển không ngừng của Internet và các phương tiện lưu trữ đã tạo ra một lượng thông tin không lồ. Các dịch vụ tìm kiếm thông tin trên Web trả về kết quả là các trang Web phù hợp với nhu cầu tìm kiếm của người dùng. Trang Web kết quả có tính phù hợp với yêu cầu trong câu hỏi của người dùng. Cũng vậy, các bài toán phân cụm, phân lớp có đối tượng tác động là trang Web trong hệ thống tài liệu. Tuy nhiên, các thông tin cần khai thác còn tiềm ẩn trong một câu, một vùng văn bản, hay một phân vùng của trang Web. Đây thực sự là một nguồn tài nguyên rất có giá trị, chứa nhiều thông tin, dữ liệu tiềm ẩn có giá trị, và vì thế cần thiết được khai thác để sử dụng. Tuy nhiên, với khối lượng thông tin rất lớn, công việc nói trên là không dễ dàng được xử lý hoặc phân tích tự động bởi máy tính. Lĩnh vực trích chọn thông tin (Information Extraction) [CL96] có đối tượng nghiên cứu không chỉ là các trang Web nguyên vẹn, mà là những thông tin tiềm ẩn trong nội dung của các trang Web đó. Có thể hiểu rằng, trích chọn thông tin là quá trình chắt lọc các thông tin từ CSDL một cách tự động theo những tiêu chí nhất định.

Thông tin và dữ liệu trên Web được lưu trữ và trình bày bởi nhiều thể loại và định dạng khác nhau. Chúng ta có thể kể ra đây một số định dạng văn bản Web với các đặc thù, tính chất dữ liệu khác nhau và theo đó sẽ có các phương pháp trích chọn khác nhau.

- **Trích chọn thông tin từ văn bản Web phi cấu trúc** (unstructured Web documents): Hướng đến việc trích chọn các thông tin như thực thể (entities), mối quan hệ giữa các thực thể (relations),... từ các văn bản ngôn ngữ tự nhiên trên Web. Các phương pháp được sử dụng để trích chọn thông tin dạng này thường là các mô hình học máy thông kê mạnh.

- **Trích chọn thông tin từ văn bản Web bán cấu trúc** (semi-structured documents): Thông tin bán cấu trúc trên Web rất đa dạng phụ thuộc vào cách lưu trữ và trình bày của từng Web site cụ thể. Các Web site của các hàng thương mại điện tử (e-commerce) thường trình bày các sản phẩm hàng

hoa của mình rất đa dạng và sử dụng nhiều định dạng về bảng biều, màu sắc, font chữ, hình ảnh,... nhằm thu hút sự chú ý của người dùng Web. Các phương pháp trích chọn thông tin dạng bản cấu trúc thẻ này thường phải tận dụng được hết các đặc điểm của văn bản như, các từ khoá quan trọng, các định dạng font, màu sắc, thẻ và khuôn mẫu HTML....

• **Trích chọn thông tin chủ đề trên Web** (topic analysis/extraction on the Web): Có thể nói, thông tin trên Internet/Web hiện nay được xem là vô tận và không có bất kỳ ai có thể hiểu và có một cái nhìn bao quát về lượng thông tin cũng như xu hướng thông tin được xuất bản trực tuyến theo thời gian. Trích chọn thông tin trên Web theo chủ đề giúp chúng ta có một cái nhìn tổng thể, biết được những gì nổi bật trong quá khứ, đâu là xu hướng thông tin hiện thời, và đâu là những hướng sẽ nổi lên trong tương lai gần. Tổng hợp thông tin hướng chủ đề trên Web cũng giúp chúng ta sắp xếp lại thông tin và theo dõi các luồng thông tin tốt hơn.

• **Trích chọn thông tin từ các cộng đồng Web** (diễn đàn – forums, blogs, mạng tin nhắn nhanh – instant messenger networks và mạng xã hội trực tuyến – online social networks): Thông tin tiềm ẩn trong các cộng đồng Internet rất phong phú và đa dạng. Ví dụ, từ những nhận xét, đánh giá các mặt hàng, các sản phẩm mới của người tiêu dùng được đăng tải trên một diễn đàn hay blog nào đó có thể giúp chúng ta trích chọn những ý kiến để từ đó có thể biết được mức độ chấp nhận và thoả mãn của khách hàng. Chúng ta gọi đây là trích chọn ý kiến (opinion mining/extraction) [Liu08]. Những thông tin liên quan đến các cộng đồng người sử dụng trên diễn đàn, blogs, các mạng xã hội như (Facebook, MySpace,...) và thậm chí là các mạng tin nhắn nhanh (online instant messaging networks) đều chứa một hàm lượng tri thức cộng đồng cao. Trích chọn, tổng hợp và tìm ra được những thông tin hữu ích trên đó giúp chúng ta nắm bắt được cả những thông tin, tri thức cụ thể và những xu hướng chung của thế giới trực tuyến.

Còn khá nhiều những vấn đề thú vị khác về trích chọn, tổng hợp thông tin trên Web cũng như những phương pháp, kỹ thuật để giải quyết các bài toán đó.

9.1.1. Trích chọn thông tin từ văn bản Web phi cấu trúc

Trích chọn thông tin từ văn bản Web phi cấu trúc có thể được xem tương đương với việc trích chọn thông tin từ văn bản ngôn ngữ tự nhiên. Ở đây có thể bò qua các thông tin bản cấu trúc đi kèm trong văn bản Web như các thẻ HTML, các định dạng font chữ, màu sắc,... Các bài toán trích chọn thông tin từ văn bản phi cấu trúc cũng được định nghĩa rõ ràng từ Hội nghị MUC (Message Understanding Conference), từ MUC-1 (1987) đến MUC-7 (1998) [CM98]. Trong hội nghị MUC cuối cùng, ban tổ chức đã

định nghĩa đầy đủ năm bài toán trích chọn thông tin từ văn bản phi cấu trúc như sau:

- Bài toán trích chọn các thực thể (Named Entity Recognition – NER) trong văn bản như tên người (PERSON), tên địa điểm, nơi chôn (LOCATION), tên các tổ chức (ORGANIZATION), các đơn vị số (NUMBER), thời gian (DATE/TIME), tiền tệ (MONETARY),... Hình 9.1 minh họa bài toán trích chọn thông tin thực thể từ văn bản.

| | | | | | | | | | | | | | | |
|--|---|---|-----------|---------------------|---------------|---|------------|--|-----------|-------------------|---------|-------------|-------------|-----|
| William (Bill) H. Gates, born on October 28, 1955 and grew up in Seattle, Washington, is the co-founder, chairman, and chief software architect of Microsoft Corporation, the worldwide leader in software, services, and solutions that help people and businesses realize their full potential. Microsoft had revenues of US\$39.79 billion for the fiscal year ending June, 2005, and employs more than 61,000 people in 102 countries and regions. According to Forbes, Bill Gates is the richest person in the world with a net worth of about US\$50 billion, as of March, 2006. | | | | | | | | | | | | | | |
| *** | | | | | | | | | | | | | | |
| Since amassing his fortune, Gates has pursued a number of philanthropic endeavors, donating large amounts of money, about 51% of his total fortune, to various charitable organizations and scientific research programs through the Bill & Melinda Gates Foundation, founded in 2000. He and his wife, Melinda Gates, were collectively named by The Time as the 2005 Persons of the Year. | | | | | | | | | | | | | | |
| <table border="1"><tr><td>PERSON:</td><td>William (Bill) H. Gates; Bill Gates; Gates; Melinda Gates</td></tr><tr><td>LOCATION:</td><td>Seattle, Washington</td></tr><tr><td>ORGANIZATION:</td><td>Microsoft Corporation; Microsoft; Forbes; Bill & Melinda Gates Foundation; The Time</td></tr><tr><td>DATE/TIME:</td><td>October 28, 1955; June 2005; March, 2006; 2000</td></tr><tr><td>MONETARY:</td><td>US\$39.79; US\$50</td></tr><tr><td>NUMBER:</td><td>61,000; 102</td></tr><tr><td>PERCENTAGE:</td><td>51%</td></tr></table> | PERSON: | William (Bill) H. Gates; Bill Gates; Gates; Melinda Gates | LOCATION: | Seattle, Washington | ORGANIZATION: | Microsoft Corporation; Microsoft; Forbes; Bill & Melinda Gates Foundation; The Time | DATE/TIME: | October 28, 1955; June 2005; March, 2006; 2000 | MONETARY: | US\$39.79; US\$50 | NUMBER: | 61,000; 102 | PERCENTAGE: | 51% |
| PERSON: | William (Bill) H. Gates; Bill Gates; Gates; Melinda Gates | | | | | | | | | | | | | |
| LOCATION: | Seattle, Washington | | | | | | | | | | | | | |
| ORGANIZATION: | Microsoft Corporation; Microsoft; Forbes; Bill & Melinda Gates Foundation; The Time | | | | | | | | | | | | | |
| DATE/TIME: | October 28, 1955; June 2005; March, 2006; 2000 | | | | | | | | | | | | | |
| MONETARY: | US\$39.79; US\$50 | | | | | | | | | | | | | |
| NUMBER: | 61,000; 102 | | | | | | | | | | | | | |
| PERCENTAGE: | 51% | | | | | | | | | | | | | |

Hình 9.1. Ví dụ về trích chọn các thực thể trong văn bản phi cấu trúc

- Bài toán trích chọn đối tượng (Element Extraction), trong đó các đối tượng được trích chọn bao gồm cả các thuộc tính, đặc điểm (features/properties) thuộc về các đối tượng đó.

- Bài toán trích chọn quan hệ giữa các thực thể (Relation Extraction) hướng đến xác định các mối quan hệ giữa các thực thể đã xác định trong bài toán thứ nhất. Mỗi quan hệ có thể liên quan đến hai hoặc nhiều hơn các thực thể. Hình 9.2 nêu ví dụ về các quan hệ giữa "Bill Gates" với "Microsoft Corporation", "Bill Gates" với "Melinda Gates".... được trích chọn từ văn bản ở hình 9.1.

- Trích chọn các chuỗi đồng tham chiếu (Co-reference Resolution), trong đó các thực thể được nhắc tới trong văn bản hướng đến cùng một đối tượng trong thế giới thực sẽ được liên kết lại với nhau thành một chuỗi (co-reference chain). Hình 9.3 nêu ví dụ về 3 chuỗi đồng tham chiếu liên quan đến ba đối tượng trong thế giới thực là "Bill Gates", "Microsoft Corporation" và "Melinda Gates".

[Bill Gates] co-founder of [Microsoft Corporation]
[Bill Gates] chairman of [Microsoft Corporation]
[Bill Gates] chief software architect of [Microsoft Corporation]
[Melinda Gates] wife of [Bill Gates]
[Bill Gates and Melinda Gates] co-founders of [Bill & Melinda Gates Foundation]

Hình 9.2. Ví dụ về trích chọn quan hệ giữa các khái niệm

William (Bill) H. Gates, born on October 28, 1955 and grew up in Seattle, Washington, is the co-founder, chairman, and chief software architect of Microsoft Corporation, the worldwide leader in software, services, and solutions that help people and businesses realize their full potential. Microsoft had revenues of US \$39.79 billion for the fiscal year ending June, 2005, and employs more than 61,000 people in 102 countries and regions. According to Forbes, Bill Gates is the richest person in the world with a net worth of about US\$50 billion, as of March, 2006.

Since amassing his fortune, Gates has pursued a number of philanthropic endeavors, donating large amounts of money, about 51% of his total fortune, to various charitable organizations and scientific research programs through the Bill & Melinda Gates Foundation, founded in 2000. He and his wife, Melinda Gates, were collectively named by *The Time* as the 2005 Persons of the Year.

- Chain 1: William (Bill) H. Gates; the co-founder, chairman, chief software architect of Microsoft Corporation; Bill Gates; the richest person in the world; Gates; He; his
Chain 2: Microsoft Corporation; the worldwide leader in software, services, and solutions; that; Microsoft
Chain 3: his wife; Melinda Gates

Hình 9.3. Ví dụ về trích chọn các chuỗi đóng tham chiếu lời các đối tượng trong văn bản

– Trích chọn kịch bản (Scenario Extraction) hướng tới trích chọn thông tin liên quan đến các hoạt động hoặc sự kiện nêu trong các văn bản.

Có rất nhiều cách tiếp cận, phương pháp và kỹ thuật để giải quyết các bài toán trích chọn thông tin nêu trên. Nổi bật trong các hướng tiếp cận đó chính là hướng ứng dụng các mô hình máy trạng thái hữu hạn (finite state machines) hoặc còn có tên gọi khác là các mô hình học máy thống kê cho dữ liệu chuỗi (statistical sequence learning models). Nổi bật trong các phương pháp này là ba mô hình:

- Mô hình Markov ẩn (Hidden Markov Models – HMMs) [Rab89].
- Mô hình Markov cực đại hóa Entropy (Maximum Entropy Markov Models – MEMMs) [CFP00].
- Mô hình trường ngẫu nhiên (Conditional Random Fields – CRFs) [LCP01].

Đặc điểm chung của các mô hình này là coi các câu văn bản là các chuỗi dữ liệu đầu vào để từ đó đoán nhận, xác định đâu là thông tin cần trích chọn. Ví dụ, để trích chọn thông tin các thực thể trong câu văn bản "*According to Forbes, Bill Gates is the richest person in the world with a net worth of about US\$50 billion, as of March, 2006*". Thì các mô hình trên sẽ xem câu này là chuỗi dữ liệu đầu vào (input data sequence) và sẽ đoán nhận nhãn của các từ trong câu như sau:

| <u>Chuỗi câu dữ liệu đầu vào</u> | <u>Nhận câu đoán nhận</u> |
|----------------------------------|---------------------------|
| According | 0 |
| to | 0 |
| Forbes | ORGANIZATION |
| * | 0 |
| Bill | PERSON |
| Gates | PERSON |
| is | 0 |
| the | 0 |
| richest | 0 |
| person | 0 |
| in | 0 |
| the | 0 |
| world | 0 |
| with | 0 |
| a | 0 |
| net | 0 |
| worth | 0 |
| of | 0 |
| about | 0 |
| US\$50 | MONETARY |
| billion | 0 |
| * | 0 |
| in | 0 |
| of | 0 |
| March | DATE/TIME |
| * | DATE/TIME |
| 2006 | DATE/TIME |
| * | 0 |

Trong mục 9.2 sẽ trình bày cụ thể mô hình lý thuyết và các ứng dụng tương ứng của các mô hình này.

9.1.2. Trích chọn thông tin từ văn bản Web bán cấu trúc

Trích chọn thông tin, dữ liệu từ những trang Web bán cấu trúc là một vấn đề rất quan trọng trong trích chọn dữ liệu nói chung. Chúng ta có thể kể ra đây rất nhiều ví dụ, ứng dụng như trích chọn thông tin người dùng trên Web (personal data extraction) như tên tên, tuổi, địa chỉ email, số điện

thoại,... trích chọn các thông tin quảng cáo về nghề nghiệp (job advertising), trích chọn các thông tin về các đối tượng như các sản phẩm (product descriptions) trên các Web site thương mại điện tử, thông tin về các nhà hàng, khách sạn,... Tất cả những thông tin trích chọn được lưu vào CSDL nhằm phục vụ các nhu cầu tìm kiếm của người dùng Web.

Đặc điểm chính là thông tin, dữ liệu dạng này tồn tại ở dạng bản cấu trúc, có nghĩa là ngoài những từ khoá (ngôn ngữ tự nhiên) thì còn những căn cứ (evidence) khác như bảng biểu, danh sách, kích thước font chữ, màu sắc, định dạng, các thẻ HTML,... giúp quá trình trích chọn dễ dàng hơn.

| | | | |
|---|----------------|--|---|
| Barto, Andrew G. | (413) 545-2109 | barto@cs.umass.edu | CS276 |
| Professor. | | |   |
| Computational neuroscience, reinforcement learning, adaptive motor control, artificial neural networks, adaptive and learning control, motor development. | | | |
| Berger, Emery D. | (413) 577-4211 | emery@cs.umass.edu | CS344 |
| Assistant Professor. | | |   |
| Brock, Oliver | (413) 577-0334 | oli@cs.umass.edu | CS246 |
| Assistant Professor. | | |   |
| Clarke, Lori A. | (413) 545-1328 | clarke@cs.umass.edu | CS304 |
| Professor. | | |   |
| Software verification, testing, and analysis; software architecture and design. | | | |
| Cohen, Paul R. | (413) 545-3638 | cohen@cs.umass.edu | CS278 |
| Professor. | | |   |
| Planning, simulation, natural language, agent-based systems, intelligent data analysis, intelligent user interfaces. | | | |

Hình 9.4. Ví dụ về văn bản Web bán cấu trúc có bảng biểu, định dạng font chữ,...



More Great Deals

| | | |
|--|---|--|
|  Acer 20" Widescreen LCD Monitor 1600 x 1050 Resolution, 2500:1 Contrast Ratio Click Here |  Navigon 5100 3.5-inch GPS Text-to-Speech, Real-time traffic, Received Click Here |  Western Digital 1TB SATA Hard Drive Minimize Your Carbon Footprint Click Here |
| \$129.99  | \$69.99  | \$79.99  |

Hình 9.5. Ví dụ về trang Web trình bày các sản phẩm tại CompUSA.com

Hình 9.4 là một ví dụ về danh sách các giáo sư được liệt kê dưới dạng bản cấu trúc mà chúng ta có thể dễ dàng nhận ra các trường thông tin như họ tên, học hàm/học vị, số điện thoại, địa chỉ email, địa chỉ phòng làm việc,... Hình 9.5 và 9.6 là ví dụ về các mẫu sản phẩm trình bày trên hai Web site thương mại điện tử nổi tiếng là CompUSA.com và Amazon.com. Các sản

phẩm cùng với đặc điểm được mô tả đều được trình bày dưới dạng bản cấu trúc, giúp ta có nhiều thông tin hơn khi trích chọn. Trong mục 9.3 sẽ trình bày cụ thể hơn các cách tiếp cận, các phương pháp trích chọn thông tin từ các trang Web bán cấu trúc như vừa nêu.

The Nine Rules of Marketing and PR: How to Use News Releases, Blogs, Podcasting, Viral Marketing and Online Media to Reach Buyers Directly by David M. Scott (Paperback - Nov 3, 2008)
Buy new \$16.96 \$11.33 49 used & new from \$9.64
Get it by Friday, Sep 4 if you order in the next 2 hours and choose one-day shipping
Eligible for FREE Super Saver Shipping
Audiobook (179)
Other Editions: Kindle Edition, Hardcover, Audio Download, Audio CD

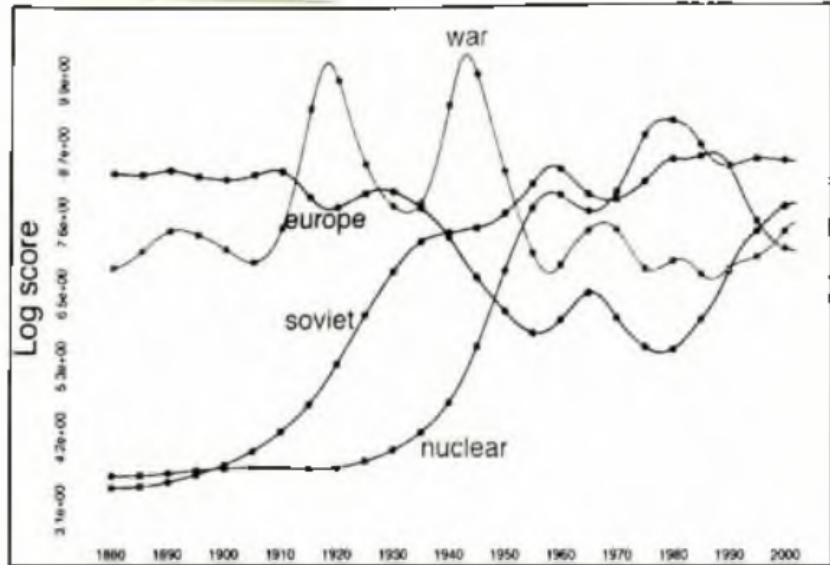
Predictably Irrational, Revised and Expanded Edition: The Hidden Forces That Shape Our Decisions by Dan Ariely (Hardcover - May 19, 2009)
Buy new \$22.99 \$10.47 38 used & new from \$16.91
Get it by Friday, Sep 4 if you order in the next 5 hours and choose one-day shipping
Eligible for FREE Super Saver Shipping
Audiobook (122)
Other Editions: Kindle Edition

Hình 9.6. Thông tin bán cấu trúc trên Amazon.com

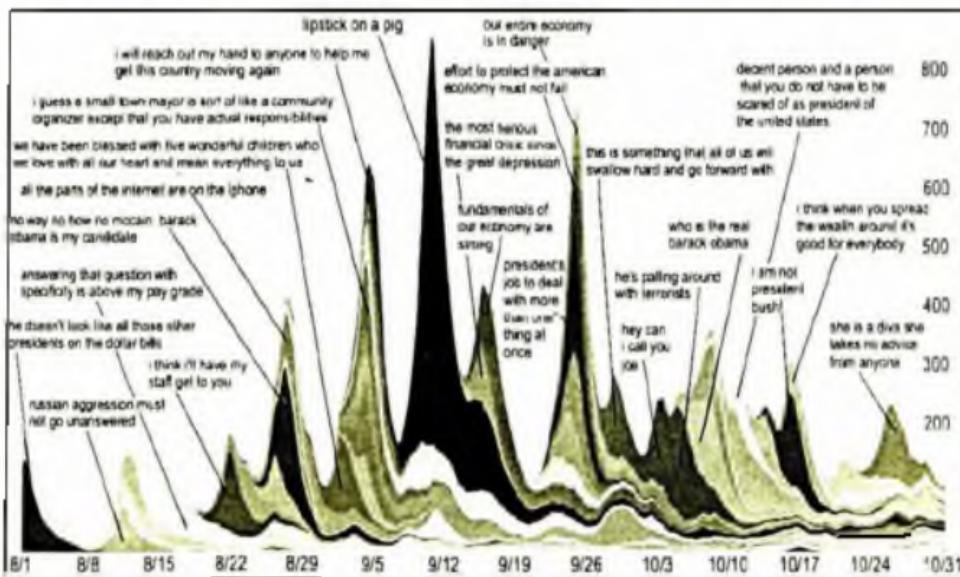
9.1.3. Trích chọn thông tin chủ đề trên Web

Phân tích, trích chọn thông tin các chủ đề trên Web là một trong những hướng nghiên cứu nhận được nhiều quan tâm hiện nay. Vấn đề trở nên quan trọng hơn khi Web là một kho thông tin văn bản khổng lồ và mức độ phức tạp tăng lên không ngừng cả về số lượng trang Web lẫn các hình thức đăng tải nội dung (tin tức, blogs, mạng xã hội,...). Những công cụ phân tích các luồng và các chủ đề thông tin trở nên cần thiết khi giúp người dùng có một cái nhìn phổ quát và định hướng thông tin tốt hơn. Những người làm việc với các tập dữ liệu văn bản lớn như các nhà làm luật, các nhà báo, những nhà thống kê và những người làm trong thư viện... có thêm công cụ để duyệt qua các mảng thông tin dễ dàng hơn theo các chủ đề.

Hình 9.7 hiển thị một số chủ đề thông tin được Blei và Lafferty [BL08] phân tích từ tập các bài báo của Tạp chí Science kể từ năm 1880 tới năm 2002. Chúng ta thấy, các chủ đề theo thời gian sẽ thay đổi, hoặc nổi bật lên, hoặc ít được đề cập hơn tuỳ thuộc vào thời sự, diễn biến khoa học, văn hoá, chính trị của thời gian đó. Ví dụ, ở chủ đề "chiến tranh" (war), ta thấy hai khoảng thời gian mà chủ đề này nổi bật rơi vào giai đoạn của chiến tranh thế giới lần thứ I và thứ II. Hình 9.8 là ví dụ về 50 luồng thông tin nổi bật trong vòng 3 tháng (từ 1/8/2008 đến 31/10/2008) được phân tích từ 90 triệu trang Web lấy từ 1.6 triệu nguồn thông tin khác nhau trên Web [LBK09].



Hình 9.7. Một số chủ đề nổi bật được phân tích từ Tạp chí Science từ 1880 đến 2002 [BL08]



Hình 9.8. 50 luồng thông tin nổi bật nhất trong khoảng 1/8/2008 đến 31/10/2008 được thống kê/phân tích từ 90 triệu Web pages từ 1.6 triệu nguồn thông tin khác nhau [LBK09]

Ở mục 9.4 sẽ phân tích kỹ hơn vấn đề này cùng một vài phương pháp được sử dụng rộng rãi trong phân tích chủ đề và dõi theo chủ đề trên Web.

9.1.4. Trích chọn thông tin từ các cộng đồng Web

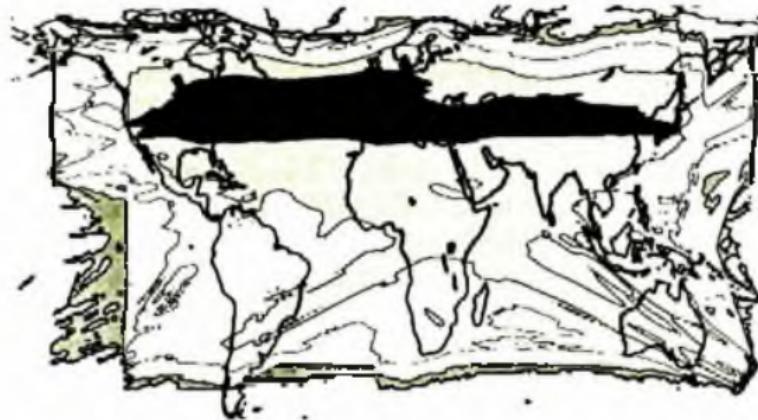
Nghiên cứu các tính chất và trích chọn những thông tin quan trọng từ các cộng đồng trực tuyến như từ các diễn đàn (forums), blogs, mạng tin nhắn nhanh (instant messenger networks) và mạng xã hội trực tuyến (online social networks) là một trong những hướng thu hút được sự chú ý của cộng đồng khai phá Web hiện nay. Thông tin tiềm ẩn từ các cộng đồng này rất đa dạng, có sự phối hợp và góp sức của hàng ngàn, thậm chí hàng triệu thành viên, và do đó nếu nắm bắt được những thông tin này, có thể hiểu được xu hướng, thị hiếu, quan điểm của người dùng Web và theo đó sẽ có những điều chỉnh, cải tiến kịp thời để đáp ứng nhu cầu của người dùng Web. Ví dụ, từ những nhận xét, đánh giá các mặt hàng, các sản phẩm mới của người tiêu dùng được đăng tải trên một diễn đàn hay blog nào đó có thể giúp chúng ta trích chọn những ý kiến, để từ đó có thể biết được mức độ chấp nhận và thỏa mãn của khách hàng. Ta gọi đây là trích chọn ý kiến (opinion mining/extraction) [HL06, Liu08]. Những thông tin liên quan đến các cộng đồng người sử dụng trên diễn đàn, blogs, các mạng xã hội như (Facebook, MySpace,...) và thậm chí là các mạng tin nhắn nhanh (online instant messaging networks) đều chứa một hàm lượng tri thức cộng đồng cao. Trích chọn, tổng hợp và tìm ra được những thông tin hữu ích trên đó, giúp nắm bắt được cả những thông tin, tri thức cụ thể và những xu hướng chung của thế giới trực tuyến.



Hình 9.9. Số lượng người sử dụng mạng đối thoại trực tuyến của Microsoft (MSN Messenger) được thống kê từ 30 tỷ hội thoại giữa 240 triệu người dùng [LH08]

Hình 9.9 trực quan hóa phân bố số lượng người sử dụng dịch vụ hội thoại trực tuyến của Microsoft. Hình 9.10 minh họa lưu lượng thông tin hội thoại giữa người dùng ở các vùng miền, lãnh thổ khác nhau trên toàn thế

giới [LH08]. Phân tích và trích chọn được những thông tin này giúp những người thiết kế và phát triển dịch vụ hội thoại trực tuyến có một cái nhìn toàn cục, tổng quan; giúp tối ưu hoá về đường truyền, tốc độ, chất lượng dịch vụ,...



Hình 9.10. Biểu đồ (heatmap) đối thoại trực tuyến sử dụng mạng MSN giữa các vùng, lãnh thổ trên thế giới [LH08]

9.2. Các phương pháp trích chọn thông tin từ văn bản Web phi cấu trúc

Thông tin bán cấu trúc trên Web rất đa dạng, phong phú và do đó có rất nhiều các phương pháp trích chọn khác nhau. Mỗi phương pháp có một điểm mạnh riêng. Ví dụ, phương pháp xây dựng các luật quy nạp (induction rules) thì thường có độ chính xác rất cao, nhưng có một nhược điểm là cần sự can thiệp của con người, vì các phương pháp này đòi hỏi các ví dụ mẫu cho các trang Web, hoặc Web site có cấu trúc khác nhau. Các phương pháp tự động thì cố gắng xây dựng các thuật toán tự nhận dạng cấu trúc HTML, các định dạng font chữ, màu sắc, quy luật chung để tự động trích chọn các thông tin cần thiết.

9.2.1. Mô hình Markov ẩn (Hidden Markov Models – HMMs)

a) Mô hình lý thuyết HMMs

HMMs [Rab89] là mô hình xác suất khá hiệu quả để xây dựng công cụ mô hình hóa dữ liệu chuỗi (sequence data), đã được ứng dụng rất thành công đối với các bài toán liên quan đến xử lý, nhận dạng tiếng nói, xử lý ngôn ngữ tự nhiên như tìm cụm từ, phân đoạn, nhận dạng, tìm chủ đề của đoạn, hay phân đoạn và đoán nhận gene trong tin – sinh học. Mô hình sử dụng khái niệm các trạng thái ẩn (hidden states) và khái niệm quan sát

(observations – các đối tượng dữ liệu được sinh ra bởi các trạng thái ẩn). Trong trường hợp trích chọn thông tin văn bản, mỗi từ hoặc một đoạn trong câu được xem là một quan sát (data observation) X_i , còn các trạng thái ẩn Y_i , chính là các nhãn cần gắn cho từ hay quan sát X_i . Nhãn cần gắn cho từ trong trường hợp cụ thể có thể là từ loại (danh từ, tính từ, động từ...), có thể là định danh người (PERSON), địa danh (LOCATION), giá cả (PRICE)... Tập nhãn phụ thuộc vào từng bài toán trích chọn thông tin cụ thể.

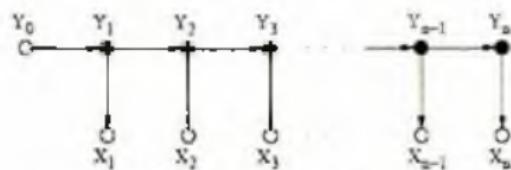
HMM là một mô hình sinh (*generative model*), mô tả quá trình sinh ra các dữ liệu quan sát bằng cách xác định xác suất đồng thời (*joint probability*) của chuỗi quan sát và chuỗi trạng thái. Chuỗi quan sát được sinh ra theo quá trình bắt đầu từ trạng thái đầu tiên, sinh ra một quan sát tương ứng với trạng thái đó, chuyển tới trạng thái tiếp theo, sinh ra một quan sát tương ứng với trạng thái đó, chuyển tới trạng thái tiếp theo,... Hình 9.11 mô tả cấu trúc và cách thức hoạt động của HMMs với X_i là quan sát, còn Y_i , là trạng thái ẩn tương ứng. Theo sơ đồ ở Hình 9.11, trạng thái tại thời điểm n chỉ phụ thuộc vào trạng thái tại thời điểm $n - 1$. Tương tự, quan sát sinh ra tại thời điểm n chỉ phụ thuộc vào trạng thái tại thời điểm n .

Giả thiết rằng, chuỗi trạng thái $\{Y_n\}_{n=0}^{\infty}$ là chuỗi Markov trong không gian trạng thái hữu hạn, gồm J trạng thái $S = \{1, 2, \dots, J\}$. Hình 9.12 minh họa chuỗi Markov ẩn với $J = 5$. Chuỗi quan sát $\{X_n\}_{n=0}^{\infty}$ có giá trị trong không gian quan sát hữu hạn $O = \{o_1, o_2, \dots, o_K\}$.

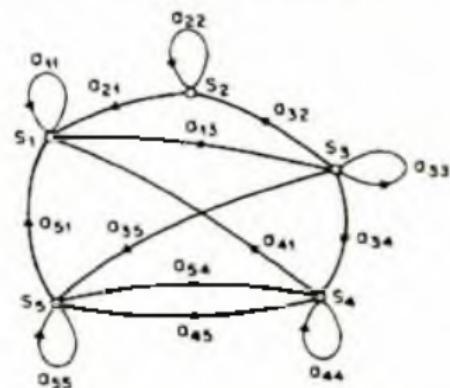
Với giả thiết trên, mô hình HMM được xác định theo các tham số là ma trận chuyển, ma trận sinh và phân bố xác suất xuất phát.

- Xác suất chuyển từ trạng thái i sang trạng thái j là:

$$a_{ij} = p(Y_n = j | Y_{n-1} = i); \quad n \geq 1; \quad i, j \in S$$



Hình 9.11. Cấu trúc của mô hình Markov ẩn



Hình 9.12. Sơ đồ chuyển trạng thái trong HMMs

Ma trận chuyển (transition matrix) $A = (a_{ij})_{i=1,j=1}^{I,J}$, trong đó xác suất chuyển trạng thái thỏa mãn điều kiện: $a_{ij} \geq 0, \sum_{j=1}^J a_{ij} = 1$.

- Tại thời điểm $n = 0$, trạng thái X_0 được xác định bởi phân bố xác suất ban đầu

$$\pi(0) = (\pi_1(0), \pi_2(0), \dots, \pi_I(0)), \text{ với } \pi_j(0) = p(X_0 = j)$$

- Hai quá trình $\{X_n\}_{n=0}^{\infty}$ và $\{Y_n\}_{n=0}^{\infty}$ quan hệ với nhau qua phân phối xác suất $b_j(k) = p(X_k = o_k | Y_k = j)$ là xác suất để trạng thái j sinh ra quan sát o_k . Ma trận sinh (emission matrix) $B = \{b_{jk}\}_{j=1,k=1}^{I,K}$, trong đó các b_{jk} thỏa mãn điều kiện: $b_{jk} \geq 0$ và $\sum_{k=1}^K b_{jk} = 1$.

Khi đó, mô hình HMM hoàn toàn được xác định với $\lambda = (A, B, \pi(0))$. Dưới đây xem xét hai bài toán chính của mô hình HMM là bài toán huấn luyện (dạy) và bài toán giải mã.

Vấn đề huấn luyện được phát biểu như sau: "Làm thế nào điều chỉnh các tham số của mô hình $\lambda = (A, B, \pi(0))$ để làm cực đại $p(\lambda)$?".

Bài toán giải mã được phát biểu như sau: "Cho chuỗi quan sát $O = \{o_1, o_2, \dots, o_K\}$ và mô hình $\lambda = (A, B, \pi(0))$. Tìm chuỗi trạng thái phù hợp nhất với chuỗi quan sát đưa ra, tức là tìm chuỗi j_0^*, \dots, j_n^* sao cho xác suất:

$$p(Y_0 = j_0, \dots, Y_n = j_n, X_0 = o_0, \dots, X_n = o_n; \lambda)$$
 đạt giá trị lớn nhất.

• Vấn đề huấn luyện:

Đặt: $V^*(\lambda) = \max p(Y_0 = j_0, \dots, Y_n = j_n, X_0 = o_0, \dots, X_n = o_n; \lambda)$

Ta phải ước lượng các tham số của λ để làm cực đại $V^*(\lambda)$.

Các bước để Training:

Bước 1. Chọn một mô hình HMM ban đầu.

Bước 2. Sử dụng thuật toán Viterbi để tìm các chuỗi $o^{(1)}, \dots, o^{(n)}$ với mô hình HMM hiện tại.

Bước 3. Tính các số liệu thống kê sau:

$n_i(k) =$ số lần trạng thái i xuất hiện trong chuỗi $o^{(k)}$,

$n_{ij}(k) =$ số lần chuyển từ trạng thái i sang trạng thái j trong chuỗi $o^{(k)}$,

$m_{ji}(k) =$ số lần quan sát o_j được sinh ra từ trạng thái j của chuỗi $o^{(k)}$,

Sau đó tính:

$$e_{rj} = \sum_{k=1}^I n_{rj}(k); e_r = \sum_{k=1}^I n_r(k); d_{ri} = \sum_{k=1}^I m_{ri}(k)$$

Chuẩn hóa các tham số để đánh giá lại mô hình.

Bước 4. Lặp lại bước 2 và 3 đến khi tham số đánh giá hội tụ.

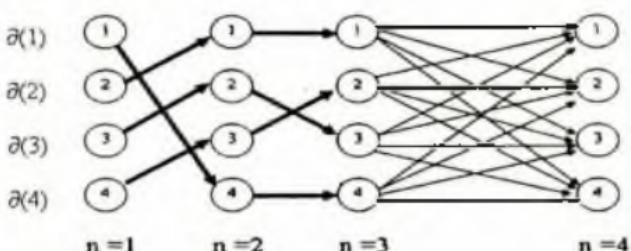
• *Vấn đề giải mã:*

Chương 3 đã trình bày nội dung cơ bản về thuật toán Viterbi và đã giới thiệu một số nét sơ bộ về việc áp dụng thuật toán trong một số bài toán trích chọn thông tin. Mục này sẽ giới thiệu thêm một số nét chi tiết hơn việc áp dụng thuật toán.

Với hướng tiếp cận truyền thống, để tìm chuỗi phù hợp nhất với chuỗi quan sát bằng cách liệt kê các chuỗi trạng thái sẽ có độ phức tạp là hàm mũ với độ dài của chuỗi. Vì vậy, sử dụng thuật toán Viterbi với độ phức tạp là $O(TN^2)$ với T là độ dài của chuỗi quan sát và N là số trạng thái trong mô hình HMM.

Để tìm chuỗi trạng thái $S = \{s_1, s_2, \dots, s_k\}$ phù hợp nhất với chuỗi quan sát $O = \{o_1, o_2, \dots, o_k\}$ ta phải xác định được xác suất lớn nhất tại thời điểm n của chuỗi trạng thái được sinh ra bởi chuỗi quan sát với trạng thái hiện tại là j :

$$\hat{\sigma}_n(j) = \max_{i_0, \dots, i_{n-1}} p(X_0 = o_0, \dots, X_n = o_n, Y_0 = j_0, \dots, Y_n = j)$$



Hình 9.13. Cách hoạt động của thuật toán Viterbi

Hình 9.13 mô tả sơ đồ chuyển trạng thái theo thuật toán. Theo các kết quả nghiên cứu, ta nhận được:

$$\hat{\sigma}_n(j) = [\max_{i_{n-1}} \hat{\sigma}_{n-1}(i).a_{ij}].b_j(o_n)$$

Để thu được chuỗi trạng thái, với mỗi n và j cần phải xác định các tham số để làm cực đại xác suất trên. Xác định mảng $\psi_n(j)$ với

$$\psi_n(j) = \arg \max_{i_{n-1}} \hat{\sigma}_{n-1}(i).a_{ij}$$

Quá trình tìm chuỗi trạng thái bao gồm các bước sau:

Bước 1. Tại $n = 0$, với mỗi $j \in S$ tính $\hat{\sigma}_0(j) = \pi_j(0).b_j(o_0)$; $\psi_0(j) = 0$.

Bước 2. Bước lặp của thuật toán Viterbi: Tính

$$\hat{\sigma}_{n-1}(j) = [\max_{i_{n-1}} \hat{\sigma}_{n-1}(i).a_{ij}].b_j(o_n)$$

$$\psi_n(j) = \arg \max_{i_{n-1}} \hat{\sigma}_{n-1}(i).a_{ij}$$

Bước 3. Tính:

$$p^* = \max_{i=1, \dots, n} \partial_N(i)$$

$$j^*(n) = \arg \max_{i=1, \dots, n} \partial_N(i)$$

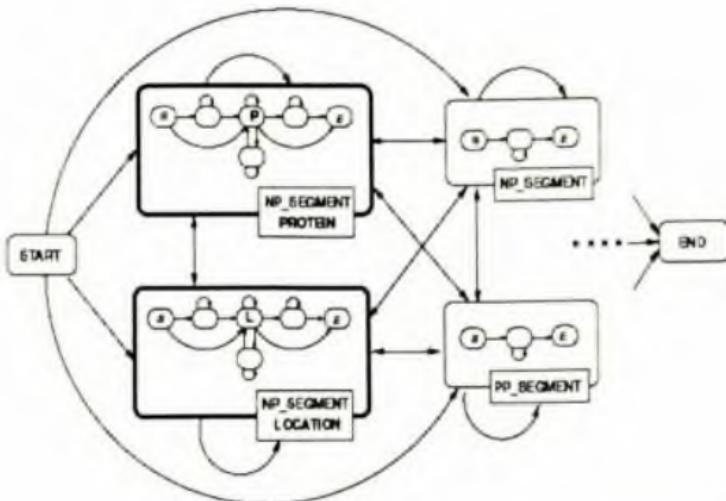
Chuỗi trạng thái cuối cùng thu được là:

$$j^*(n) = \psi_{n+1}(j^*(n+1)), n = N-1, N-2, \dots, 0$$

HMM là một công cụ mạnh cho việc mô hình hóa các chuỗi dữ liệu và đã được ứng dụng thành công trong nhiều bài toán liên quan đến văn bản. Tuy nhiên, HMM có hai hạn chế. Hạn chế thứ nhất, tập quan sát thường là các từ, nên khó tích hợp các đặc trưng (features) phụ thuộc lẫn nhau (dependent features) hoặc liên quan với nhau (overlapping features) như vị trí của từ trong câu, chữ cái đầu tiên viết hoa hay không, cả từ có viết hoa hay không, vị trí của từ trong văn bản, từ có bắt đầu bằng số hay không, hoặc có đứng trước hay đứng sau những từ đặc biệt nào không,... Một khác, trong các bài toán, tập quan sát thường rất lớn, khó có thể liệt kê hết được, điều này làm giảm sự chính xác khi thực hiện, đồng thời làm tăng độ phức tạp của bài toán vì ma trận sinh lớn (do tập quan sát lớn). Hạn chế thứ hai, mặc dù đã được sử dụng rộng rãi, nhưng HMM và các mô hình sinh khác không phù hợp cho những bài toán gán nhãn cho dữ liệu chuỗi. Theo cách tiếp cận truyền thống, khi xác định các tham số để cực đại hóa xác suất của chuỗi quan sát đưa ra, mô hình phải xác định xác suất đồng thời (joint probability) $p(x, y)$ qua chuỗi quan sát và chuỗi dữ liệu được gán nhãn. Điều này hữu ích nếu như mô hình huấn luyện được sử dụng để sinh ra dữ liệu. Tuy nhiên, phân phối xác suất của tập được gán nhãn là phân phối có điều kiện $p(x|y)$. Nói cách khác, sử dụng mô hình sinh để giải quyết vấn đề phân phối có điều kiện là không phù hợp. Xác định phân phối đồng thời khi giải mã qua các nhãn và chuỗi quan sát có nghĩa là, tất cả các chuỗi quan sát có thể phải được liệt kê. Điều này sẽ gặp khó nếu các phần tử của chuỗi quan sát có sự phụ thuộc lớn. Vì vậy, hạn chế của mô hình sinh là thừa nhận rằng, các quan sát độc lập với nhau. Trong mô hình HMM, quan sát tại thời điểm t chỉ phụ thuộc vào trạng thái t , mỗi quan sát được xử lý như một đơn vị riêng biệt, không phụ thuộc vào các quan sát khác trong chuỗi. Tuy nhiên thực tế, hầu hết các chuỗi dữ liệu đều không được biểu diễn chính xác như tập hợp các đối tượng riêng biệt.

b) Ứng dụng HMMs trong trích chọn thông tin

Trước khi có sự xuất hiện của các mô hình MEMMs và CRFs, HMMs được xem là công cụ chính trong các ứng dụng về trích chọn thông tin. Cụ thể, Seymore và các đồng nghiệp [SCR99] đã huấn luyện cấu trúc cho mô hình HMMs nhằm trích chọn các thông tin từ đầu đề các bài báo nghiên cứu (Hình 9.14). Thuật toán này có thể ứng dụng cho các bài toán khác.



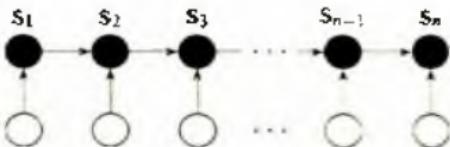
Hình 9.14. Ứng dụng mô hình HMMs phân cấp (Hierarchical HMMs) để trích chọn thông tin đa mức [SKR03]

Để tăng thêm khả năng trích chọn thông tin, Skounakis và đồng nghiệp [SCR03] đã giới thiệu mô hình HMM phân cấp, trong đó các mô hình HMMs được lồng nhau để trích chọn thông tin ở các cấp khác nhau (Hình 9.15). Mô hình này được ứng dụng vào trích chọn các thực thể từ các văn bản sinh y học và cho độ chính xác khá quan.

HMMs còn có nhiều ứng dụng khác trong trích chọn thông tin từ văn bản và từ dữ liệu Web. Nhiều công bố khoa học đã trình bày về ứng dụng đa dạng của HMMs trong trích chọn thông tin, chẳng hạn như [Lee97, Fre98, Fre99, Fre00, FC99, RC01].

9.2.2. Mô hình Maximum Entropy Markov Models (MEMMs)

MEMMs [MFP00] cho phép giải quyết hai hạn chế của mô hình HMM. Giống như HMM, MEMMs cũng là mô hình hữu hạn trạng thái theo xác suất. HMM được mô tả bởi tập S . Ở và hai xác suất có điều kiện $p(s'|s)$, $p(o|s)$. Trong mô hình MEMMs, hai phân phối xác suất là phân phối xác suất $p(s'|s)$ chuyển trạng thái s sang trạng thái s' và xác suất sinh $p(o|s)$ là xác suất trạng thái s sinh ra quan sát o trong mô hình HMM được thay thế bởi duy nhất một ham $p(s'|o) = p(s'|s, o)$ là xác suất chuyển từ trạng thái s sang trạng thái s' với quan sát hiện tại o .



Hình 9.15 Cấu trúc của mô hình Markov cục đại hóa Entropy (MEMMs)

Khác với mô hình HMM, quan sát hiện tại chỉ phụ thuộc vào trạng thái hiện tại, trong MEMMs quan sát hiện tại không chỉ phụ thuộc vào trạng thái hiện tại mà còn phụ thuộc vào cả trạng thái trước đó (Hình 9.15).

MEMMs dùng thuật toán GIS để ước lượng các tham số của mô hình. GIS ước lượng giá trị các tham số của λ_a giải quyết vấn đề entropy cực đại cho mỗi hàm chuyển. Với mỗi cặp (o, s) thuật toán đòi hỏi rằng, tổng giá trị của đặc trưng là hằng số C. Nếu điều này chưa đúng, tức là tổng giá trị các đặc trưng không bằng C, cộng thêm một giá trị đặc trưng f_x với $x = n + 1$, như $f_x(o, s) = C - \sum_{a=1}^n X_a f_a(o, s)$ và C được chọn đủ lớn sao cho $f_x(o, s) \geq 0$ với mọi o và s.

- **Ước lượng các tham số của mô hình MEMMs:**

Dầu vào: Chuỗi quan sát $o_1 o_2 \dots o_m$ tương ứng với chuỗi được gán nhãn $s_1 s_2 \dots s_m$, từ đó có tập quan sát trạng thái tương ứng.

Nội dung.

- Xác định chuỗi trạng thái ứng với chuỗi quan sát dựa vào các tập quan sát – trạng thái ở trên.

- Tìm xác suất chuyển trạng thái $p(s'|s, o)$.

- Dùng thuật toán GIS để tìm entropy cực đại cho mỗi hàm trạng thái.

Kết quả nhận được mô hình khi cho một chuỗi quan sát chưa được gán nhãn và tìm chuỗi trạng thái tương ứng với chuỗi quan sát.

Thuật toán GIS để huấn luyện các hàm chuyển p_s cho trạng thái s' gồm các bước thực hiện sau:

Bước 1. Tính trung bình của mỗi đặc trưng

$$F_a = \frac{1}{m_s} \sum_{k=1}^{m_s} f_a(o_{t_k}, s_{t_k})$$

Bước 2. Chọn $\lambda_a^{(0)} = 1$

Bước 3. Tại bước lặp j, sử dụng giá trị hiện tại $\lambda_a^{(j)}$ trong $p_s^{(j)}(s|o)$ để tính giá trị kỳ vọng của mỗi đặc trưng

$$E_a^{(j)} = \frac{1}{m_s} \sum_{k=1}^{m_s} p_s^{(j)}(s|o_{t_k}) f_a(o_{t_k}, s)$$

Bước 4. Tính: $\lambda_a^{(j+1)} = \lambda_a^{(j)} + \frac{1}{C} \log \left(\frac{F_a}{E_a^{(j)}} \right)$

Bước 5. Quay lại bước 3 cho đến khi hội tụ.

• **Vấn đề giải mã của MEMMs:**

Mặc dù có sự khác nhau giữa hai mô hình HMM và MEMMs, nhưng chúng vẫn đồng nhất nhau về cách giải quyết vấn đề giải mã, đó là sử dụng thuật toán Viterbi.

Bước lặp trong HMM được tính bởi công thức:

$$\hat{\delta}_{n+1}(j) = \left[\max_{i=1, \dots, J} \hat{\delta}_n(i) \cdot a_{ij} \right] \cdot b_j(o_{n+1})$$

Bước lặp trong MEMMs được tính bởi công thức sau:

$$\hat{\delta}_{n+1}(j) = \max_{i=1, \dots, J} \hat{c}_n(i) \cdot p_i(j | o_{n+1})$$

Trong MEMM, mỗi hàm $p_s(s'|o) = p(s'|s, o)$ được mô tả như một mô hình mǔ:

$$p_s(s|o) = \frac{1}{Z(o, s)} \exp \left(\sum \lambda_a f_a(o, s) \right)$$

với: $Z(o, s)$ là thừa số;

λ_a là tham số được huấn luyện;

$f_a(o, s)$ là hàm đặc trưng với hai đối số là quan sát hiện tại và trạng thái tiếp theo.

Mỗi đặc trưng a là một cặp $a = \langle b, s \rangle$, trong đó b là đặc trưng nhị phân của quan sát và s là trạng thái đích. Chẳng hạn, một đặc trưng:

$$b(x) = \begin{cases} 1 & \text{nếu quan sát là '$';} \\ 0 & \text{trường hợp còn lại.} \end{cases}$$

Hàm đặc trưng nhận giá trị 1 (true), 0 (false) với:

$$f_{\langle b, s \rangle}(o_t, s_t) = \begin{cases} 1 & \text{nếu } b(o_t) \text{ là đúng và } s = s_t; \\ 0 & \text{trường hợp còn lại.} \end{cases}$$

Thông thường, mỗi trạng thái trước s' và đặc trưng a , hàm chuyên trạng thái có tính chất sau:

$$\frac{1}{m} \sum_{k=1}^m f_a(o_{t_k}, s_{t_k}) = \frac{1}{m} \sum_{s' \in S} \sum_{s \in S} p_{s'}(s | o_{t_k}) f_a(o_{t_k}, s)$$

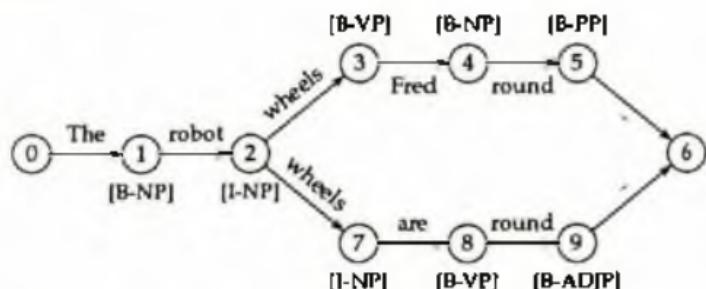
với t_1, t_2, \dots, t_m là bước đồi hỏi hàm chuyên $p(s'|s, o)$.

Cách sử dụng hàm chuyên trạng thái quan sát thay cho hàm chuyên trạng thái và hàm quan sát riêng rẽ của mô hình HMM cho phép mô tả được các đặc trưng không độc lập nhau của các quan sát.

Mặc dù mô hình MEMMs đã khắc phục được hai hạn chế của HMM, tuy nhiên, MEMMs lại gặp vấn đề "label bias". Tại mỗi thời điểm, tông xác suất chuyên trạng thái s_{n+1} sang trạng thái s_n với quan sát o_n bằng 1.

$$\sum p(s_n | s_{n+1}, o_n) = 1$$

Xét ví dụ với tập dữ liệu học có hai văn bản "the robot wheels Fred round" và "the robot wheels are round" [CM03]. Sơ đồ chuyển trạng thái như Hình 9.116.



Hình 9.16. Ví dụ về vấn đề "label bias" trong bước chuyển xác suất của mô hình MEMMs

Cần tìm chuỗi trạng thái ứng với câu: "the robot wheels are round".

Xác suất để chuỗi trạng thái là s với điều kiện chuỗi quan sát là x được tính bởi công thức sau:

$$p(s|x) = \prod_{i=1}^n p(s_i | s_{i-1}, x_i)$$

Vì vậy có:

$$p(1, 2 | \text{the robot}) = p(1 | 0, \text{the})p(2 | 1, \text{robot})$$

Với quan sát là "The" và "robot" chỉ có một cách chuyển từ trạng thái 0 chuyển sang trạng thái 1 và từ trạng thái 1 chuyển sang trạng thái 2. Nhưng từ trạng thái 2 với quan sát "wheels" có hai cách chuyển trạng thái: chuyển sang trạng thái 3 hoặc chuyển sang trạng thái 7. Vì vậy, để xác định được chuỗi trạng thái ta phải tính xác suất theo cả hai con đường đó, so sánh và chọn chuỗi trạng thái có xác suất lớn hơn.

Vì tại mỗi thời điểm, tổng xác suất chuyển trạng thái s_{n-1} sang trạng thái s_n với quan sát x_n bằng 1:

$$\sum p(s | 7, \text{Fred}) = 1$$

nên $p(4 | 3, \text{Fred}) = p(8 | 7, \text{Fred}) = 1$

Nhưng từ thực tế không có bước chuyển từ trạng thái 7 sang trạng thái 8 với quan sát "Fred":

$$p(8 | 7, \text{Fred}) = 0$$

xuất hiện tình không chính xác giữa bước chuyển trạng thái này.

Với ví dụ trên, giả sử ngoài bước chuyển trạng thái 2, xác suất chuyển trạng thái bằng nhau, thì chuỗi trạng thái thu được chỉ phụ thuộc vào xác suất chuyển trạng thái từ trạng thái 2 sang trạng thái 3, hay trạng thái 4. Trong ví dụ này, $p(3|2, \text{wheels}) = p(4|2, \text{wheels})$ nên cuối cùng xác suất của hai chuỗi trạng thái trên là bằng nhau bất chấp chuỗi quan sát đưa vào.

Đây chỉ là một ví dụ đơn giản, trong đó $p(3|2, \text{wheels}) = p(4|2, \text{wheels})$

Nếu bước chuyển từ trạng thái 2 sang trạng thái 3 xảy ra nhiều hơn sang trạng thái 4, thì cuối cùng sẽ thu được chuỗi trạng thái ứng với chuỗi "the robot wheels Fred round" mặc dù chuỗi quan sát đưa vào là "the robot wheels are round".

Trong các bài toán thực tế, với tập dữ liệu huấn luyện lớn, khả năng phân nhánh của các trạng thái cao, thì vấn đề label bias sẽ ảnh hưởng nhiều tới sự chính xác của mô hình. Đây chính là hạn chế lớn nhất của MEMMs. Trong khi đó, HMM không gặp phải vấn đề này vì HMM tách riêng xác suất chuyển trạng thái và xác suất sinh quan sát.

9.2.3. Mô hình trường ngẫu nhiên (Conditional Random Fields – CRFs)

Chương 4 đã đề cập tới một số nội dung về mô hình trường ngẫu nhiên có điều kiện và ứng dụng vào bài toán tách từ. Trong mục này, mô hình được giới thiệu với những nội dung bổ sung và phù hợp với bài toán trích chọn thông tin.

a) Mô hình lý thuyết CRFs

CRF được Lafferty và các đồng nghiệp giới thiệu lần đầu vào năm 2001 [LCP01]. Giống như MEMMs, CRFs là mô hình dựa trên xác suất điều kiện, chúng có thể tích hợp được các thuộc tính đa dạng của chuỗi dữ liệu quan sát nhằm hỗ trợ cho quá trình phân lớp. Tuy vậy, khác với MEMMs, CRFs là các mô hình đồ thị vô hướng. Điều này cho phép CRFs có thể định nghĩa phân phối xác suất của toàn bộ chuỗi trạng thái với điều kiện biết chuỗi quan sát cho trước, thay vì phân phối trên mỗi trạng thái với điều kiện biết trạng thái trước đó và quan sát hiện tại như trong các mô hình MEMMs. Chính vì cách mô hình hóa như vậy, CRFs có thể giải quyết được vấn đề 'label bias' mà MEMMs gặp phải.

Trong các bài toán thực tế với tập dữ liệu huấn luyện lớn, khả năng phân nhánh của các trạng thái cao thì vấn đề label bias sẽ ảnh hưởng nhiều tới sự chính xác của mô hình. Đây chính là hạn chế lớn nhất của MEMMs. Trong khi đó, HMM không gặp phải vấn đề này vì HMM tách riêng xác suất chuyển trạng thái và xác suất sinh quan sát.

- Một số ký hiệu:

- Chữ viết hoa X, Y, Z,... ký hiệu các biến ngẫu nhiên.

- Chữ thường đậm x, y, f, g,... ký hiệu các vector như vector biểu diễn chuỗi các dữ liệu quan sát, vector biểu diễn chuỗi các nhãn,...

- Chữ in đậm có chỉ số là ký hiệu của một thành phần trong một vector, ví dụ x_i chỉ một thành phần tại vị trí i trong vector x.

- Chữ viết thường không đậm như x, y,... là ký hiệu các giá trị đơn như một dữ liệu quan sát hay một trạng thái.

- S: Tập hữu hạn các trạng thái của một mô hình CRFs.

- Khái niệm CRFs

Ký hiệu X là biến ngẫu nhiên nhận giá trị là chuỗi dữ liệu cần phải gán nhãn và Y là biến ngẫu nhiên nhận giá trị là chuỗi nhãn tương ứng. Mỗi thành phần Y_i của Y là một biến ngẫu nhiên nhận giá trị trong tập hữu hạn các trạng thái S. Trong bài toán nhận diện loại thực thể, X có thể nhận giá trị là các câu trong ngôn ngữ tự nhiên, Y là một chuỗi ngẫu nhiên các tên thực thể tương ứng với các câu này và mỗi một thành phần Y_i của Y có miền giá trị là tập tất cả các nhãn tên thực thể (tên người, tên địa danh,...).

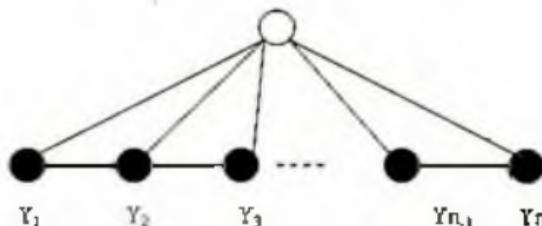
Cho một đồ thị vô hướng không có chu trình G = (V, E), ở đây V là tập các đỉnh của đồ thị và E là tập các cạnh vô hướng nối các đỉnh đồ thị. Các đỉnh V biểu diễn các thành phần của biến ngẫu nhiên Y sao cho tồn tại ánh xạ một – một giữa một đỉnh và một thành phần của Y_v của Y. Ta nói (X, Y) là một trường ngẫu nhiên điều kiện (CRFs) khi với điều kiện X, các biến ngẫu nhiên Y tuân theo tính chất Markov đối với đồ thị G:

$$p(Y_v | X, Y_{\omega}, \omega \neq v) = p(Y_v | X, Y_{\omega}, \omega \in N(v)) \quad (9.1)$$

Ở đây, $N(v)$ là tập tất cả các đỉnh kề với v.

Như vậy, một CRF là một trường ngẫu nhiên phụ thuộc toàn cục vào X. Trong các bài toán xử lý dữ liệu dạng chuỗi, G đơn giản chỉ là dạng chuỗi $G = (V = \{1, 2, \dots, m\}, E = \{(i, i+1)\})$. Ký hiệu $X = (X_1, X_2, \dots, X_n)$, $Y = (Y_1, Y_2, \dots, Y_n)$. Mô hình đồ thị cho CRF có dạng như hình 9.17.

Gọi C là tập hợp tất cả các đồ thị con đầy đủ của đồ thị G – đồ thị biểu diễn cấu trúc của một CRFs. Áp dụng kết quả của Hammerley – Clifford cho các trường ngẫu nhiên Markov, thừa số hoá được phân phối $p(y|x)$ thành tích của các hàm tiềm năng:



Hình 9.17. Cấu trúc của CRFs

$$p(y | x) = \prod_{A \in C} \psi_A(A | x) \quad (9.2)$$

Vì trong các bài toán xử lý dữ liệu dạng chuỗi, đồ thị biểu diễn cấu trúc của một CRFs có dạng đường thẳng như trong Hình 9.17, nên tập C phải là hợp của E và V , trong đó E là tập các cạnh của đồ thị, G và V là tập các đỉnh của G , hay nói cách khác, đồ thị con A hoặc là gồm một đỉnh, hoặc gồm một cạnh của G . Bằng cách áp dụng nguyên lý cực đại hóa entropy, Lafferty xác định hàm tiềm năng của một CRFs có dạng một hàm mũ:

$$\psi_A(A | x) = \sum_k \gamma_k t_k(A | x) \quad (9.3)$$

Ở đây t_k là một thuộc tính của chuỗi dữ liệu quan sát và γ_k là thừa số Lagrange liên kết với t_k , nói cách khác, γ_k là trọng số chỉ mức biểu đạt thông tin của thuộc tính t_k .

Có hai loại thuộc tính là thuộc tính cạnh (ký hiệu là f) và thuộc tính đỉnh (ký hiệu là g) tùy thuộc vào A là đồ thị gồm một đỉnh hay một cạnh của G . Thay các hàm tiềm năng vào công thức (9.2) và thêm vào một thừa số chuẩn hóa $Z(x)$ (để đảm bảo tổng xác suất của tất cả các chuỗi nhẫn tương ứng với một chuỗi dữ liệu quan sát bằng 1) ta được:

$$p(y | x) = \frac{1}{Z(x)} \exp \left(\sum_i \sum_k \lambda_k f_k(y_{i-1}, y_i, x) + \sum_i \sum_k \mu_k g_k(y_i, x) \right) \quad (9.4)$$

Ở đây, x, y là chuỗi dữ liệu quan sát và chuỗi trạng thái tương ứng; f_k là thuộc tính của toàn bộ chuỗi quan sát và các trạng thái tại vị trí $i - 1, i$ trong chuỗi trạng thái; g_k là thuộc tính của toàn bộ chuỗi quan sát và trạng thái tại vị trí i trong chuỗi trạng thái. Thừa số chuẩn hóa $Z(x)$ được cho như sau:

$$Z(x) = \sum_y \exp \left(\sum_i \sum_k \lambda_k f_k(y_{i-1}, y_i, x) + \sum_i \sum_k \mu_k g_k(y_i, x) \right) \quad (9.5)$$

$\theta(\lambda_1, \lambda_2, \dots, \mu_1, \mu_2, \dots)$ là các vector các tham số của mô hình. θ sẽ được ước lượng giá trị nhờ các phương pháp ước lượng tham số cho mô hình.

• *Ước lượng tham số mô hình CRFs*

Kỹ thuật được sử dụng để đánh giá tham số cho một mô hình CRFs là làm cực đại hóa độ đo likelihood của tập huấn luyện.

Giả sử dữ liệu huấn luyện gồm một tập N cặp, mỗi cặp gồm một chuỗi quan sát và một chuỗi trạng thái tương ứng $D = \{(x^{(i)}, y^{(i)})\} \forall i = 1 \dots N\}$. Độ đo likelihood giữa tập huấn luyện và mô hình điều kiện tương ứng $p(y|x, \theta)$ là:

$$L(\theta) = \prod_{i,y} p(y | x, \theta)^{\hat{p}_{i,y}} \quad (9.6)$$

Ở đây $\theta(\lambda_1, \lambda_2, \dots, \mu_1, \mu_2, \dots)$ là các tham số của mô hình và $\bar{p}(x, y)$ là phân phối thực nghiệm đồng thời của x, y trong tập huấn luyện.

- **Nguyên lý cực đại likelihood:**

Các tham số tối đa của mô hình là các tham số làm cực đại hàm likelihood.

$$\theta_{ML} = \arg \max_{\theta} L(\theta) \quad (9.7)$$

θ_{ML} đảm bảo những dữ liệu mà chúng ta quan sát được trong tập huấn luyện sẽ nhận được xác suất cao trong mô hình. Nói cách khác, các tham số làm cực đại hàm likelihood sẽ làm phân phối trong mô hình gần nhất với phân phối thực nghiệm trong tập huấn luyện. Vì việc tính θ dựa theo công thức (9.7) rất khó khăn, nên thay vì tính toán trực tiếp, cần xác định θ làm cực đại logarit của hàm likelihood:

$$l(\theta) = \sum_{x,y} \bar{p}(x,y) \log(p(y|x, \theta)) \quad (9.8)$$

Vì hàm logarit là hàm đơn điệu, nên việc làm này không làm thay đổi giá trị của θ được chọn.

Thay $p(y|x, \theta)$ của mô hình CRF vào công thức (9.8), ta có:

$$l(\theta) = \sum_{x,y} \bar{p}(x,y) \left[\sum_{i=1}^{n+1} \lambda^* f_i + \sum_{i=1}^n \mu^* g_i \right] - \sum_x \bar{p}(x) * \log Z \quad (9.9)$$

ở đây, $\lambda(\lambda_1, \lambda_2, \dots, \lambda_{n+1})$ và $\mu(\mu_1, \mu_2, \dots, \mu_n)$ là các vector tham số của mô hình. f là vector các thuộc tính ($f_1(y_{i-1}, y_i, x), f_2(y_{i-1}, y_i, x), \dots$), g là vector các thuộc tính ($g_1(y_i, x), g_2(y_i, x), \dots$).

Hàm log likelihood cho mô hình CRFs là một hàm lôm và trơn trong toàn bộ không gian của tham số. Bàn chất hàm lôm của log-likelihood cho phép ta có thể tìm được giá trị cực đại toàn cục θ bằng cách thiết lập các thành phần của vector gradient của log-likelihood bằng không. Mỗi thành phần trong vector gradient của hàm log-likelihood là đạo hàm của hàm log-likelihood theo một tham số của mô hình. Đạo hàm hàm log-likelihood theo tham số λ_k , nhận được:

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \lambda_k} &= \sum_{x,y} \bar{p}(x,y) \sum_{i=1}^n f_k(y_{i-1}, y_i, x) \\ &\quad - \sum_x \bar{p}(x) p(y|x, \theta) \sum_{i=1}^n f_k(y_{i-1}, y_i, x) \\ &= E_{\bar{p}(x,y)} [f_k] - E_{p(y|x,\theta)} [g_k] \end{aligned} \quad (9.10)$$

Việc thiết lập phương trình trên bằng 0 tương đương với việc đưa ra một ràng buộc cho mô hình: giá trị trung bình của f_k theo phân phối $p(x)p(y|x,\theta)$ bằng giá trị trung bình của f_k theo phân phối thực nghiệm $\bar{p}(x,y)$.

Về phương diện toán học, bài toán ước lượng tham số cho một mô hình CRFs chính là bài toán tìm cực trị của hàm log-likelihood. Trong các phương pháp tìm cực trị của hàm đa biến bằng cách sử dụng các thông tin về vector gradient, phương pháp L-BFGS được đánh giá là hội tụ nhanh và hiệu quả hơn so với các phương pháp khác. Ưu điểm của phương pháp này là tránh được việc tính toán trực tiếp ma trận Hessian của hàm log-likelihood trong quá trình tìm cực trị.

- *Thuật toán gán nhãn cho dữ liệu dạng chuỗi:*

Tại mỗi vị trí i trong chuỗi quan sát, ta xác định ma trận $|S|^*|S|$ như sau:

$$M_i(x) = [M_i(y', y, x)] \quad (9.11)$$

$$M_i(y', y, x) = \exp \left(\sum_k \lambda_k f_k(y', y, x) + \sum_k \mu_k g_k(y, x) \right) \quad (9.12)$$

Chuỗi trạng thái y^* mô tả tốt nhất cho chuỗi dữ liệu quan sát x là nghiệm của phương trình:

$$y^* = \operatorname{argmax}(p(y|x)) \quad (9.13)$$

- *Thuật toán Viterbi tìm chuỗi y^* :*

Gọi $\partial_i(y)$ là xác suất của "chuỗi trạng thái độ dài i kết thúc bởi trạng thái y và có xác suất lớn nhất", biết chuỗi quan sát là x . Với mọi trạng thái y' trong tập trạng thái:

$$\partial_i(y) = \max(\partial_{i-1}(y') * M_i(y', y, x)) \quad (9.14)$$

$$\text{Đặt } \text{Pre}_i(y) = \operatorname{argmax}(\partial_{i-1}(y') * M_i(y', y, x)) \quad (9.15)$$

Giả sử chuỗi dữ liệu quan sát x có độ dài n , sử dụng kỹ thuật quay lui để tìm chuỗi trạng thái y^* tương ứng như sau:

Bước 1: Với mọi y thuộc tập trạng thái tìm

$$y^*(n) = \operatorname{argmax}(\partial_n(y))$$

$$i \leftarrow n$$

Bước 2: Chứng nào $i > 0$

$$i \leftarrow i - 1$$

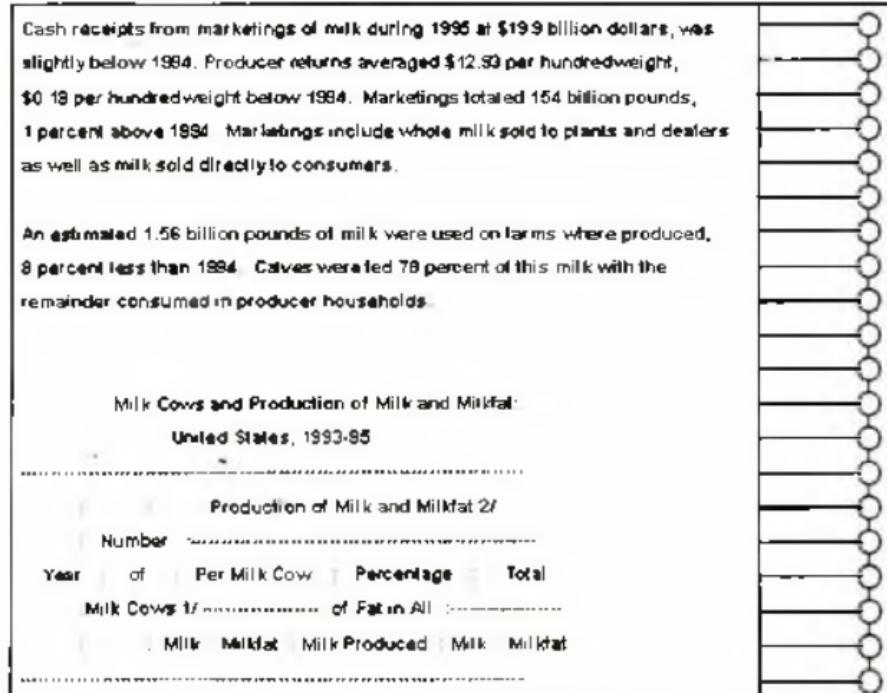
$$y \leftarrow \text{Pre}_i(y)$$

$$y^*(i) = y$$

b) Ứng dụng CRFs trong trích chọn thông tin

CRFs được ứng dụng thành công rất nhiều trong các lĩnh vực như tin – sinh học, xử lý ngôn ngữ tự nhiên và khai phá Text/Web. Ở đây điểm qua một số ứng dụng thành công của CRFs trong trích chọn thông tin.

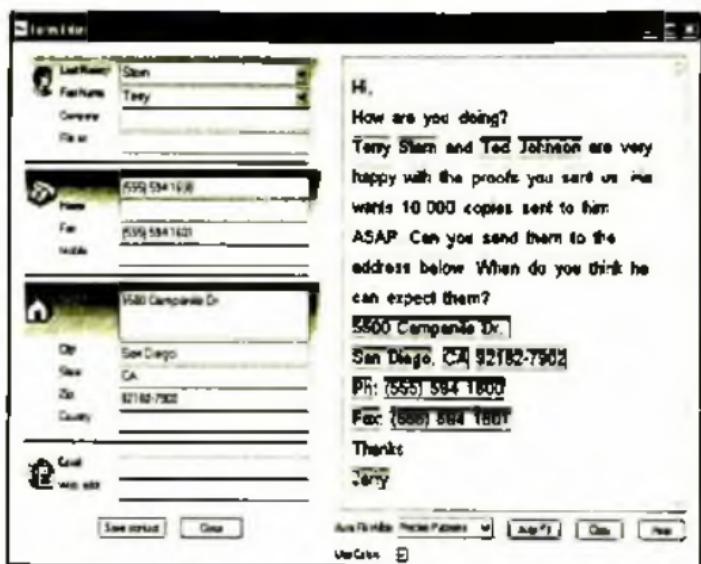
- Ứng dụng CRFs trong trích chọn thông tin bảng biểu từ văn bản [PCW03]. Trong ứng dụng này, CRFs được sử dụng để đoán nhận các dòng trong văn bản xem dòng nào thuộc các đoạn văn thông thường, dòng nào chứa thông tin về bảng biểu. Thông tin để đoán nhận dựa rất nhiều vào đặc điểm của các dòng văn bản và các ký tự đặc biệt (ký tự tạo bảng). Các dòng nằm kề nhau cũng có tính phụ thuộc lẫn nhau và rất thích hợp để mô hình hoá bảng CRFs (Hình 9.18). Kết quả cho thấy CRFs cho độ chính xác cao hơn đáng kể so với các phương pháp khác.



Hình 9.18. CRFs được sử dụng để trích chọn thông tin bảng biểu trong văn bản [PCW03]

- Ứng dụng CRFs trong trích chọn thông tin từ văn bản nhằm hỗ trợ quá trình điền form (form filling) bán tự động [KCV04] (Hình 9.19). Trong ứng dụng này CRFs được thay đổi một chút với các ràng buộc (constraints) được tích hợp vào thuật toán giải mã Viterbi cho phép thông tin trích chọn từ các văn bản khi điền vào form có tính phụ thuộc và ràng buộc lẫn nhau. Ví dụ, nếu một trường đã là họ tên thì các trường khác không thể là họ tên, hoặc nếu một trường đã là số điện thoại văn phòng thì các trường khác ít có

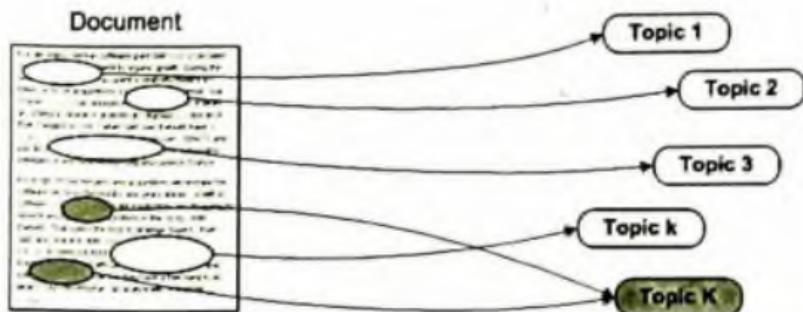
kết quả là số điện thoại văn phòng nữa, vì thường chỉ có một số điện thoại văn phòng. Bằng cách đó, người ta nâng độ chính xác của điện form bán tự động lên đáng kể. Bài toán này quan trọng khi các công ty nhận được những đơn đặt hàng hoặc hợp đồng qua email bằng văn xuôi và hệ thống sẽ trích chọn thông tin từ các email này một cách bán tự động để điền vào cơ sở dữ liệu. Ứng dụng này sẽ giảm đáng kể công sức của nhân viên nhập dữ liệu ở những công ty, tổ chức có lượng giao dịch lớn.



Hình 9.19. CRFs được sử dụng để trích chọn thông tin nhằm hỗ trợ cho việc điền form bán tự động [KCV04]

| | HMM | CRF | SVM |
|---------------|-------|--------------|-------|
| Overall acc. | 93.1% | 93.3% | 92.9% |
| Instance acc. | 41.3% | 73.3% | - |
| | acc. | F1 | acc. |
| Title | 98.2 | 82.2 | 99.7 |
| Author | 98.7 | 81.0 | 99.8 |
| Affiliation | 98.3 | 85.1 | 99.7 |
| Address | 99.1 | 84.8 | 99.7 |
| Note | 97.8 | 81.4 | 98.8 |
| Email | 99.9 | 92.5 | 99.9 |
| Date | 99.8 | 80.6 | 99.9 |
| Abstract | 97.1 | 98.0 | 99.6 |
| Phone | 99.6 | 53.8 | 99.9 |
| Keyword | 98.7 | 45.6 | 99.7 |
| Web | 95.9 | 65.6 | 99.9 |
| Degree | 95.4 | 65.8 | 99.8 |
| Pubmed | 77.5 | 64.2 | 93.5 |
| Average F1 | 75.6 | 93.9 | 89.7 |

Hình 9.20. Trích chọn thông tin từ các bài báo nghiên cứu [PC04] và so sánh kết quả với hai phương pháp HMMs và Support Vector Machines (SVMs)



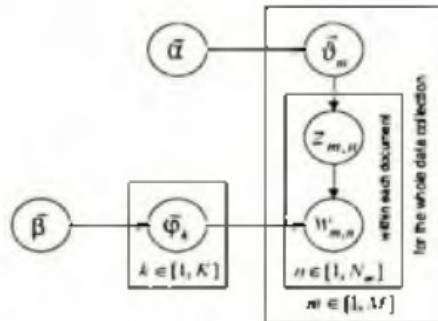
Hidden Topic Analysis/Discovery

Topics {1, 2, ..., K} are unknown (i.e., hidden and need to be discovered)

Hình 9.21. Mỗi văn bản có thể chứa nhiều chủ đề (topics), trong đó LDA được thiết kế dựa trên giả thiết sinh văn bản dựa vào các phân phối Dirichlet được minh họa trong hình 9.22 và hình 9.23.

Ngoài ra CRFs còn có nhiều ứng dụng khác như trong bài toán nhận dạng thực thể, trích chọn thông tin từ đầu mục của các bài báo nghiên cứu. Ngoài ra CRFs cũng được chỉnh sửa thay đổi để phù hợp với các bài toán trích chọn dữ liệu trên Web. Sau đây là một số ứng dụng:

- Ứng dụng CRFs cho bài toán nhận biết thực thể (named entity recognition) [CL03, LC03, SC05].
- Ứng dụng CRFs trong trích chọn thông tin từ các bài báo nghiên cứu (Hình 9.20) [PC04].
- Ứng dụng CRFs để trích chọn các nguồn ý kiến đánh giá [CCR05].
- Ứng dụng CRFs hai chiều để trích chọn thông tin trên Web [ZNW05].



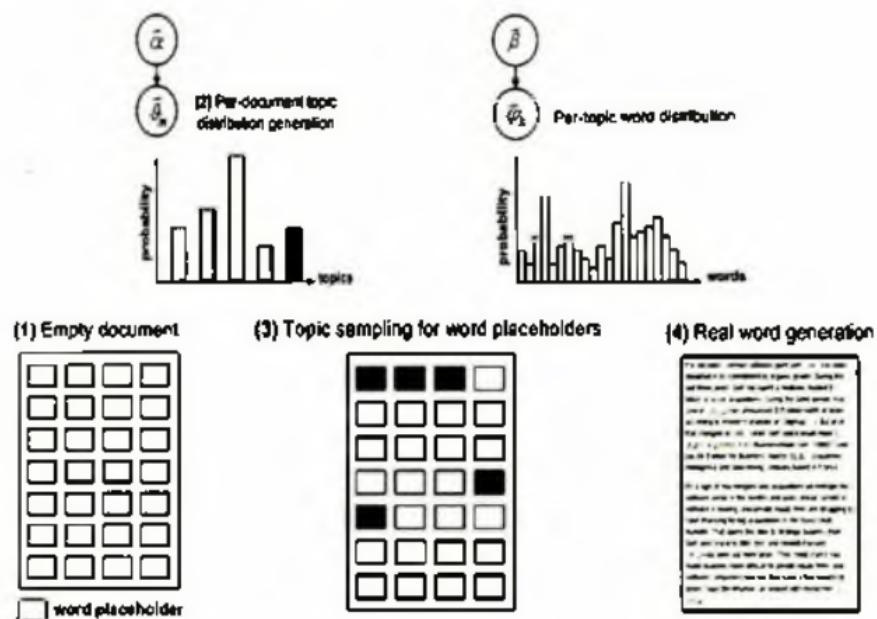
Hình 9.22. Cấu trúc của LDA

9.3. Các phương pháp trích chọn thông tin chủ đề trên Web

9.3.1. Mô hình lý thuyết

Phân tích chủ đề cho văn bản nói chung và cho dữ liệu Web nói riêng có vai trò quan trọng trong việc "hiểu" và định hướng thông tin trên Web.

Khi ta hiểu một trang Web có chứa những chủ đề hay thông tin gì, thì dễ dàng hơn cho việc xếp loại, sắp xếp và tóm tắt nội dung của trang Web đó. Trong phân lớp văn bản, mỗi văn bản thường được xếp vào một lớp cụ thể nào đó. Trong phân tích chủ đề, giả sử mỗi văn bản để cập đến nhiều hơn một chủ đề (K chủ đề) như minh họa ở Hình 9.23.



Hình 9.23. Sinh văn bản theo phương pháp thống kê của mô hình LDA [BNR03]

Có rất nhiều phương pháp phân tích thông tin chủ đề từ văn bản, điển hình là mô hình LDA (latent Dirichlet allocation) [BNR03]. LDA là một mô hình sinh (generative model) và thực hiện phân tích chủ đề từ các tập dữ liệu văn bản hoàn toàn phi giám sát (fully unsupervised). Về mặt phân tích chủ đề, LDA hướng tới mục tiêu như LSA (latent semantic analysis) [DFL90]. Về mặt trực quan, LSA tìm những cấu trúc chủ đề (topics) và khái niệm (concepts) trong tập văn bản dựa trên thông tin về đồng xuất hiện (co-occurrence) của các từ khoá trong văn bản, cho phép mô hình hoá các khái niệm đồng nghĩa (synonymy) và đa nghĩa (polysemy). Về mặt mô hình hoá, LDA hoạt động tương đối giống với pLSA (probabilistic LSA) [Hon99]. Tuy vậy, LDA ưu việt hơn pLSA ở một vài điểm như tính đầy đủ và tính khai quát cao hơn [BNR03, GS04, Hei05].

Quá trình sinh văn bản được giải thích như sau.

Trong LDA, mỗi văn bản $w_m = \{w_{m,n}\}_{n=1}^{N_m}$ được sinh bởi một phân phối chủ đề v_m từ một phân phối Dirichlet ($Dir(\alpha)$). Phân phối chủ đề này sẽ

quyết định việc gắn chủ đề cho từng vị trí từ trong văn bản. Việc gắn chủ đề $z_{m,n}$ cho mỗi từ trong văn bản được sinh ngẫu nhiên từ một phân phối multinomial $\text{Mult}(\vec{\nu}_m)$. Sau khi mỗi vị trí từ được gắn chỉ số chủ đề, việc sinh các từ sẽ được lấy ngẫu nhiên từ phân phối multinomial $\text{Mult}(\vec{\varphi}_{z_{m,n}})$.

```

for all topics  $k \in [1, K]$  do
    sample mixture components  $\vec{\varphi}_k \sim \text{Dir}(\vec{\beta})$ 
end for
for all documents  $m \in [1, M]$  do
    sample mixture proportion  $\vec{\theta}_m \sim \text{Dir}(\vec{\alpha})$ 
    sample document length  $N_m \sim \text{Poiss}(\xi)$ 
    for all words  $n \in [1, N_m]$  do
        sample topic index  $z_{m,n} \sim \text{Mult}(\vec{\theta}_m)$ 
        sample term for word  $w_{m,n} \sim \text{Mult}(\vec{\varphi}_{z_{m,n}})$ 
    end for
end for


---



- $M$ : the total number of documents
- $K$ : the number of (hidden/latent) topics
- $V$ : vocabulary size
- $\vec{\alpha}, \vec{\beta}$ : Dirichlet parameters
- $\vec{\theta}_m$ : topic distribution for document  $m$
- $\Theta = \{\vec{\theta}_m\}_{m=1}^M$ : a  $M \times K$  matrix
- $\vec{\varphi}_k$ : word distribution for topic  $k$
- $\Phi = \{\vec{\varphi}_k\}_{k=1}^K$ : a  $K \times V$  matrix
- $N_m$ : the length of document  $m$
- $z_{m,n}$ : topic index of  $n$ th word in document  $m$
- $w_{m,n}$ : a particular word for word placeholder  $[m, n]$

```

Hình 9.24. Thuật toán sinh văn bản của LDA

Tùy cấu trúc LDA ở hình 9.23, ta có công thức xác suất đồng thời của tất cả các biến ngẫu nhiên (bao gồm cả biến ẩn) như sau:

$$p(\vec{w}_m, \vec{z}_m, \vec{v}_m, \Phi | \vec{\alpha}, \vec{\beta}) = p(\Phi | \vec{\beta}) \prod_{n=1}^{N_m} p(w_{m,n} | \vec{\varphi}_{z_{m,n}}) p(z_{m,n} | \vec{v}_m) p(\vec{v}_m | \vec{\alpha})$$

Và hàm likelihood của một văn bản được lấy tích phân từ công thức trên và có dạng như sau:

$$p(\vec{w}_m | \vec{\alpha}, \vec{\beta}) = \int \int p(\vec{v}_m | \vec{\alpha}) p(\Phi | \vec{\beta}) \prod_{n=1}^{N_m} p(w_{m,n} | \vec{v}_m, \Phi) d\Phi d\vec{v}_m$$

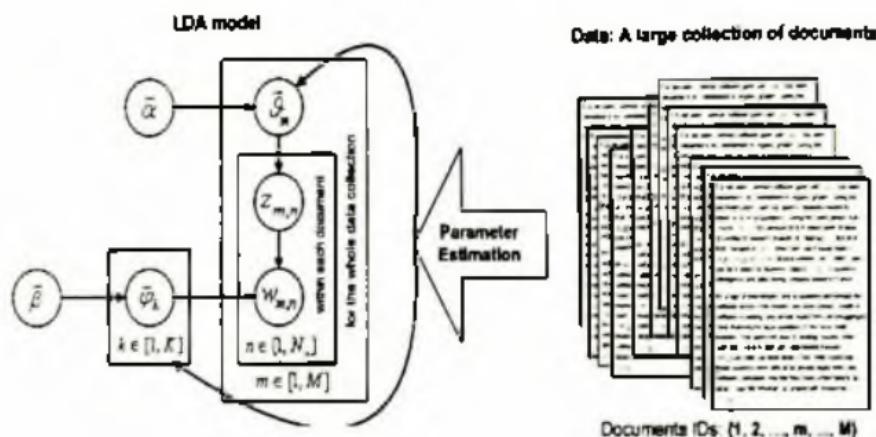
Và cuối cùng hàm likelihood của toàn bộ tập dữ liệu văn bản có dạng như sau:

$$p(w | \vec{\alpha}, \vec{\beta}) = \prod_{m=1}^M p(\vec{w}_m | \vec{\alpha}, \vec{\beta})$$

Ước lượng giá trị tham số cho mô hình LDA (Hình 9.25):

Ước lượng tham số cho mô hình LDA bằng phương pháp cực đại hóa hàm likelihood trực tiếp và một cách chính xác có độ phức tạp thời gian rất

cao, không khả thi trong thực tế. Người ta thường sử dụng các phương pháp xấp xỉ như Variational Methods [BNR03] và Gibbs Sampling [GS04]. Gibbs Sampling được xem là một thuật toán nhanh, đơn giản và hiệu quả để huấn luyện LDA.



Hình 9.25. Ước lượng tham số cho mô hình LDA từ tập dữ liệu văn bản

Các nội dung chi tiết hơn về LDA được trình bày trong nhiều công trình khoa học, chẳng hạn các công trình của Blei D. và cộng sự [BNR03], Heinrich G. [Hei05], ...

Ngoài mô hình LDA, tồn tại một số mô hình chủ đề khác xuất phát từ LDA để phân tích chủ đề với các yêu cầu khác nhau:

- Mô hình chủ đề động (Dynamic Topic Models), chẳng hạn trong [BL06, WBH08].
- Mô hình chủ đề liên kết (Correlated Topic Models), chẳng hạn trong [BL07, BL08, BL09].

9.3.2. Ứng dụng phân tích chủ đề trên Web

Có rất nhiều ứng dụng về phân tích chủ đề trên Web.

– Phân tích thông tin chủ đề trên Web để hỗ trợ phân lớp văn bản. Để hỗ trợ phân lớp cho các văn bản ngắn trên Web, Hieu X.P. và cộng sự đã phân tích hơn 70.000 trang Wikipedia bằng tiếng Anh [PNH08]. Kết quả phân tích được minh họa một phần ở Hình 9.27. Việc sử dụng các chủ đề phân tích từ tập dữ liệu này giúp nâng cao đáng kể kết quả phân lớp văn bản.

– Phân tích thông tin chủ đề của các trang Web để "hiểu" sơ bộ nội dung trang Web nhằm hướng đến đặt các thông tin quảng cáo phù hợp trên trang Web đó (contextual advertising). Thu D.L. và cộng sự [LNH08] tiến hành thử nghiệm phân tích hơn 40.000 trang Web tiếng Việt lấy từ các

nguồn VnExpress, VietnamNet, TuoiTre Online. Một phần kết quả phân tích được minh họa ở hình 9.28. Các tác giả sử dụng kết quả phân tích này trợ giúp tìm thông tin quảng cáo phù hợp cho các trang Web và độ chính xác tăng đáng kể.

| Topic 3 | Topic 15 | Topic 44 | Topic 48 | Topic 56 |
|------------------------|------------------------|------------------------|-------------------------|-----------------------|
| bác sĩ (doctor) | thời trang (fashion) | thiết bị (equipment) | chứng khoán (stock) | bánh (cake) |
| bệnh viện (hospital) | người mẫu (model) | sản phẩm (product) | công ty (company) | mc donald (McDonald) |
| thuốc (medicine) | mặc (wear) | máy (machine) | đầu tư (investment) | thịt (meat) |
| bệnh (disease) | trang phục (clothes) | màn hình (screen) | ngân hàng (bank) | pizza (pizza) |
| phẫu thuật (surgery) | thiết kế (design) | công nghệ (technology) | cổ phần (stock) | bánh mì (bread) |
| dùi (tool) (realmen) | đẹp (beautiful) | điện thoại (telephone) | thị trường (market) | bánh mì (bread) |
| bệnh nhân (patient) | váy (dress) | hãng (company) | giao dịch (transaction) | bánh ngọt (pie) |
| y tế (medical) | sưu tập (collection) | sử dụng (use) | đóng (VND) | cửa hàng (shop) |
| ung thư (cancer) | mang (wear) | thị trường (market) | mua (buy) | súc tích (hot dog) |
| tình trạng (condition) | phong cách (style) | usd (USD) | phát hành (publish) | kem (ice-cream) |
| cơ thể (body) | quần áo (costume) | pin (battery) | niềm vui (posh) | khai trương (open) |
| sức khỏe (health) | nổi tiếng (famous) | cho phép (allow) | bán (sell) | nguội (cold) |
| đau (hurt) | quần (pants) | samsung (Samsung) | tài chính (finance) | hamburger (hamburger) |
| gây (cause) | thành diễn (perform) | di động (mobile) | đấu giá (auction) | thịt (meat) |
| khám (examine) | thích (like) | sony (Sony) | trung tâm (center) | nhà hàng (restaurant) |
| kết quả (result) | quý phón rú (charming) | nhạc (music) | thông tin (information) | đồ ăn (food) |
| điều kiện (illness) | sang trọng (luxurious) | máy tính (computer) | doanh nghiệp (business) | sandwich (sandwich) |
| rặng (serious) | vẻ đẹp (beauty) | hỗ trợ (support) | cổ đông (shareholder) | khẩu vị (taste) |
| cho biết (inform) | gái (girl) | điện tử (electronic) | nhà đầu tư (investor) | lò bánh (bakery) |
| máu (blood) | gương mặt (figure) | tính năng (feature) | nhà nước (government) | bảo đảm (ensure) |
| kết nghiệm (test) | tiêu (super) | khả năng (connect) | tổ chức (organization) | nướng (grill) |
| chữa (cure) | áo dài (aoda) | thiết kế (design) | triệu (million) | bí quyết (secret) |
| chứng (trouble) | giày (shoes) | chức năng (function) | quỹ (budget) | ngon (delicious) |

Hình 9.26. Một số chủ đề (topics) được phân tích từ hơn 40 000 trang Web tiếng Việt (trong đó có VnExpress). Có thể xem đầy đủ lập các chủ đề tại <http://gibbslda.sourceforge.net/vnexpress-200topics.txt> [LNH08]

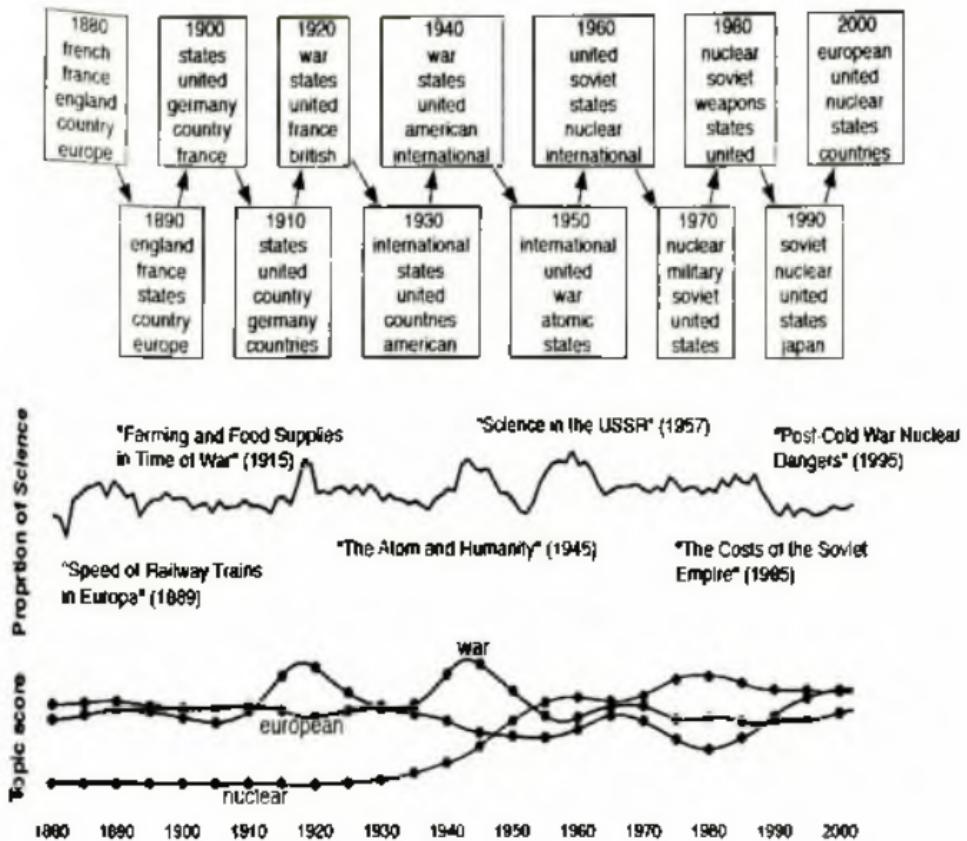
– Phân tích thông tin chủ đề văn bản (Web) để xâu chuỗi nội dung thông tin, trợ giúp tìm kiếm và duyệt các văn bản liên quan trong các tập văn bản (Web). Hình 9.29 là một ví dụ về việc sử dụng mô hình phân tích chủ đề để hỗ trợ duyệt các tập văn bản. Mô hình được sử dụng ở đây là Correlated Topic Models [BL07], cho phép không những phân tích các chủ đề mà còn khai thác mối quan hệ giữa các chủ đề với nhau. Hình 9.29 cho thấy, khi ta chọn một chủ đề thì hệ thống sẽ hiển thị các chủ đề liên quan, bên trái sẽ hiển thị các từ khóa liên quan và bên phải sẽ hiển thị các văn bản liên quan. Tương tự khi ta lựa chọn một văn bản, thì các văn bản và chủ đề liên quan cũng được hiển thị. Bằng cách đó, có thể di chuyển từ chủ đề này sang chủ đề kia, từ văn bản này sang văn bản kia theo một mối liên hệ hữu cơ được phát hiện bởi mô hình phân tích.

T0: medical health medicine care practice patient training treatment patients hospital surgery clinical physicians physician hospitals doctors therapy physical nursing doctor
T1: memory intel processor instruction processors cpu performance instructions architecture hardware data address com cache computer processing operating program
T4: signal radio frequency signals digital transmission channel antenna frequencies receiver communication transmitter analog modulation transmitted mhz data channels
T10: theory quantum universe physics mechanics particles matter particle relativity einstein model space physical light classical field theories principle energy fundamental
T18: economic trade economy world government growth countries country industry foreign production sector gdp development domestic billion industrial market policy nature
T19: film film movie production movies director cinema studio hollywood released pictures picture studios directed motion release shot sound scene actors ...
T20: party election vote elections parties voting votes candidate candidates majority political voters seats electoral democratic elected opposition coalition government ballot
T22: tax income taxes pay paid rate revenue taxation government benefit plan sales benefits rates value plans money cost property federal ...
T27: philosophy philosophers world philosophical knowledge mind reality aristotle existence nature plato ideas experience philosopher view consciousness kant physical idea
T28: space function functions vector linear theory geometry matrix equations mathematics equation field theorem algebra mathematical spaces differential product continuous
T33: insurance debt risk rate credit bonds pay loss loan cash policy payment bond money paid rates loans cost payments financial ...
T34: university college degree students universities school research academic student degrees campus colleges education graduate professor master institute institutions
T38: law act rights laws court constitution federal united legal government supreme legislation amendment civil constitutional congress public process justice power ...
T45: network networks protocol server data internet client ip nodes node connection servers protocols address packet layer connections service routing link ...
T58: government house parliament minister prime president power executive elected office council constitution assembly appointed powers head cabinet parliamentary
T57: cell cells protein proteins membrane molecules amino enzymes enzyme structure binding acids processes bacteria acid cellular receptor antibodies receptors atp ...
T80: radio television tv stations broadcast channel news network station cable broadcasting bbc satellite programming channels service media networks broadcasts program
T62: music jazz dance folk blues songs musicians style musical styles traditional american song rhythm country pop performers artists played dances ...
T64: gold currency dollar coins silver value money coin issued exchange euro inflation monetary rate pound currencies paper standard dollars mini ...
T73: internet online users site com content sites community web website user virtual information websites people software media personal forums yahoo ...
T81: art artists painting paintings artist style arts movement artistic sculpture museum painted aesthetic abstract visual painters figures architecture beauty gallery ...
T84: race sports spon racing olympic events world event competition races games team golf course olympics track international championship teams formula ...
T93: military army service officers forces force officer rank training command war armed united personnel units air soldiers ranks corps navy ...
T98: bc ancient egyptian egypt civilization period culture bronze bce age king city maya archaeological stone cities egyptians temple millennium discovered ...
T101: magic harry potter magical house witch hook witchcraft wizard witches magician books people spell wizards hogwarts rowling black paranormal phoenix ...
T103: card cards stores store chain department items retail customer customers shopping credit chains service retailers cash item shop merchant target ...
T104: software windows file microsoft operating version user files os applications linux source system map versions application users released code release ...
T107: market price stock value exchange trading markets prices sell options buy spread index stocks risk selling trade features shares contracts ...
T137: bank money banks account credit financial banking central accounts reserve balance funds federal savings services deposit loans transactions deposits commercial ...
T152: economics economic value market theory price demand production capital economy cost economists costs prices marginal utility money output labor inflation ...
T191: distribution probability test random sample variables statistical variable data error analysis function value mean tests inverse statistics values hypothesis correlation ...

Hình 9.27. Một số chủ đề mẫu được phân tích từ hơn 70.000 trang Wikipedia tiếng Anh.

Có thể xem danh sách toàn bộ các chủ đề tại

<http://gibbsida.sourceforge.net/wikipedia-topics.txt> [PNH08]



Hình 9.28. Sự thay đổi của chủ đề "chiến tranh" (war) theo thời gian được trích chọn từ tạp chí Science từ năm 1880 đến 2002 [BL09]

– Phân tích thông tin chủ đề nhằm hiểu và nắm bắt được các xu hướng thông tin trên Internet. Hình 9.27 và hình 9.28 minh họa khả năng sử dụng các mô hình phân tích chủ đề để phát hiện các chủ đề thời sự của từng giai đoạn thời gian và các chủ đề biến đổi theo thời gian như thế nào? Ví dụ, Hình 9.28 minh họa chủ đề về chiến tranh (war) thay đổi như thế nào trong giai đoạn từ 1880 tới 2002? Ở mỗi giai đoạn, từ khoá nào, khái niệm nào được bàn tới nhiều nhất và chủ đề được bàn đến nhất vào những giai đoạn nào, ít được bàn luận ở giai đoạn nào? Có hàng trăm, thậm chí hàng ngàn chủ đề như thế được bàn luận trên Web/Internet và những phân tích như thế này giúp chúng ta xâu chuỗi, định hướng và có một cái nhìn tổng thể về thông tin được đăng tải trên Internet.

| ice | volcanic year fly deposit rock | "Revolutionary Phase of the Mid-Pleistocene Arctic Interval" (1996) |
|-------------|--------------------------------------|--|
| climate | age rate rate card mortality | "Northern Hemisphere Ice-Sheet Influence on Global Climate Change" (1995) |
| cosmo | atmosphere air atmosphere atmosphere | |
| ice | carbon dioxide energy water gas | "Impact Events 500,000 to 140,000 Years Ago: Evidence from Subpolar North Atlantic Sediment" (1998) |
| temperature | glacier forest population population | "Climate Records Covering the Last Deglaciation" (1995) |
| year | ice climate ocean sea temperature | "A 0.5-Million-Year Record of Millennial-Scale Climate Variability in the North Atlantic" (1999) |
| surface | | |
| water | | "Structure of Atmospheric Variability Induced by Sea Surface Temperature and Implications for Global Warming" (1994) |
| changes | | "Inter-decadal Climate Fluctuations That Depend on Exchange Between the Tropics and Extratropics" (1993) |
| global | | |
| change | | "Decadal Trends in the North Atlantic: Oscillation, Regional Temperature and Precipitation" (1993) |
| data | | |
| atmospheric | | "Di-Nine-Southern Oscillation Displacements of the Western Equatorial Pacific Warm Pool" (1990) |
| time | | |
| weather | | "Mechanism of the Zonal Displacements of the Pacific Warm Pool by El Niño in C1850" (1994) |
| north | | |
| record | | "14.5 °C Rapid Temperature Variation in Central Greenland 70,000 Years Ago" (1999) |
| fly | | "Past and Present Subarctic Summer Glaciation" (1996) |
| pacific | | "Climate Change During the Last Deglaciation in Antarctica" (1995) |

Demo of Information Navigation in Journal Science

<http://www.cs.cmu.edu/~lemur/science/>

This was done with Correlated Topic Model (CTM) – a more advanced variant of LDA.

Hình 9.29. Phân tích chủ đề hỗ trợ duyệt và tìm kiếm trong các tập tài liệu [BL07]

Câu hỏi và bài tập

1. Dùng một phiên bản cài đặt mô hình HMM để thực nghiệm nhận dạng các thực thể tên trong tập dữ liệu MUC.
2. Dùng một phiên bản cài đặt mô hình MEMM để thực nghiệm nhận dạng các thực thể tên trong tập dữ liệu MUC.
3. Dùng một phiên bản cài đặt mô hình CRF để thực nghiệm nhận dạng các thực thể tên trong tập dữ liệu MUC. So sánh chất lượng của 3 mô hình này với nhau.

Chương 10

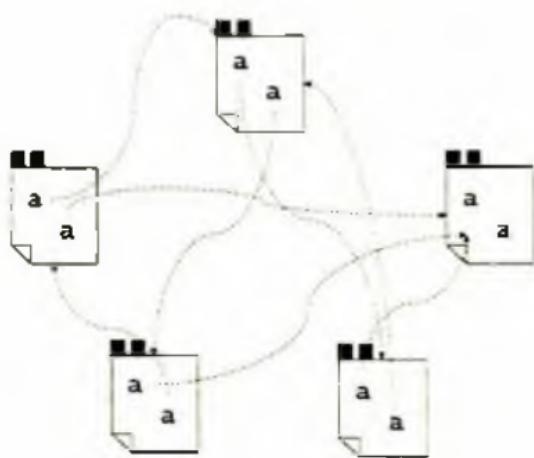
WEB NGỮ NGHĨA

10.1. Giới thiệu Web ngữ nghĩa

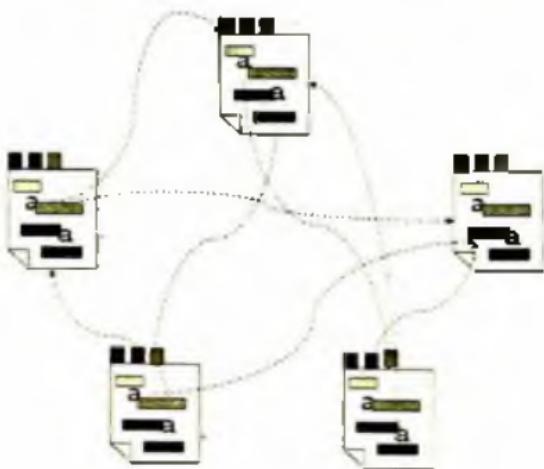
Với sự ra đời của Internet và sự phổ biến của Web, số lượng các trang Web ngày càng nhiều đến mức nếu thiếu sự ra đời của các máy tìm kiếm thì người dùng sẽ chẳng khai thác được nội dung của các Web site trên thế giới. Với sự hỗ trợ của các máy tìm kiếm (ví dụ như <http://www.google.com> hay <http://www.yahoo.com>), người dùng có thể tìm ra các nội dung mình cần ở một địa chỉ mà mình chưa bao giờ biết đến bằng cách gõ các từ khoá trong nội dung mình cần tìm hiểu, sau đó một loạt các trang Web liên quan đến nội dung đó sẽ được trả về gần như ngay lập tức. Lúc này người sử dụng chỉ làm công việc đơn giản là duyệt qua các trang Web trả về để tìm ra cái mình cần. Tuy nhiên, hiện tại nội dung các trang Web được viết cho đối tượng là con người, nên các máy tìm kiếm đều "hiểu" nội dung các trang Web ở dưới dạng khá đơn giản, đó là dưới dạng một tập các từ khoá, khi người dùng gõ vào một câu truy vấn dưới dạng một danh sách các từ khoá, hệ thống máy tìm kiếm sẽ tìm tất cả các trang Web chứa các từ khoá đó, sau đó sắp xếp (ranking) theo độ liên quan và trả về cho người dùng. Có những trường hợp kết quả trả về có thể lên đến hàng triệu trang Web, nên người dùng khó mà có thời gian để duyệt toàn bộ nội dung của các trang Web này. Thông thường, người dùng chỉ duyệt khoảng 10 trang kết quả trả về, do đó nếu trang người dùng cần tìm không nằm trong khoảng 10 trang đầu tiên sẽ không được người dùng tìm được.

Từ nhược điểm trên của các máy tìm kiếm, Web ngữ nghĩa (semantic Web) được ra đời như là sự mở rộng của Web, trong đó bên cạnh các thông tin (nội dung) dành cho người dùng, các trang Web còn được bổ sung thêm các thông tin dành cho máy (giúp máy có thể hiểu được "nội dung" của trang Web và xử lý được các thông tin này). Các thông tin dành cho máy này được gọi là siêu dữ liệu (meta-data), tức là dữ liệu dùng để mô tả dữ liệu.

WWW hiện tại là một mạng các tài liệu được diễn tả bằng ngôn ngữ tự nhiên (minh họa như Hình 10.1) dưới dạng các chuỗi ký tự, nhằm phục vụ đối tượng là con người đọc và hiểu chứ không phải dành cho máy tính. Với Web ngữ nghĩa, bằng việc thêm các siêu dữ liệu và cách tổ chức, biểu diễn thích hợp, các máy tính có thể không những hiểu được "nội dung" của các trang Web mà còn có thể suy diễn ra các tri thức mới, nhằm đáp ứng những yêu cầu của người dùng. Các siêu dữ liệu có thể đơn thuần là các thông tin mô tả các *thực thể tên* (như tên người, tên địa danh...); *sự kiện* (ví dụ như Olympic thế giới năm 2008); *các con số* (ví dụ như con số biểu thị giá vàng của một ngày cụ thể nào đó, hay tổng thu nhập của một doanh nghiệp trong một năm nào đó); *ngày tháng* (ngày tháng có thể được diễn tả bằng các cách khác nhau trong tài liệu như 20/02/2009, hay "ngày 20 tháng 2 năm 2009"); *tiền tệ* (ví dụ 30\$, 40VNĐ); *các tài liệu* (cuốn sách, hay bài báo). Các siêu dữ liệu phức tạp hơn có thể là các *khái niệm*, *quan hệ*, *ràng buộc*,... Tập hợp tất cả các siêu dữ liệu tạo ra một cơ sở tri thức (CSTT) để dựa vào đó có thể suy diễn ra các tri thức mới. Hình ảnh của Web ngữ nghĩa có thể được minh họa trong hình 10.2, trong đó các hộp được tô màu là các siêu dữ liệu, hay các thông tin đã được thêm thông tin siêu dữ liệu. Tổ chức World Wide Web Consortium (W3C) (<http://www.w3c.org>) là người đi tiên phong trong việc xây dựng ra các



Hình 10.1. Kiến trúc của WWW hiện tại



Hình 10.2. Hình ảnh của Web ngữ nghĩa

chuẩn, ngôn ngữ dùng để tổ chức, lưu trữ, thao tác, suy diễn với các cơ sở tri thức phục vụ Web ngữ nghĩa⁽¹⁾. Và cộng đồng phát triển Web ngữ nghĩa ngày càng thu hút được nhiều người tham gia⁽²⁾.

Để so sánh sự khác nhau giữa Web hiện tại và Web ngữ nghĩa, ta có thể so sánh một máy tìm kiếm thông thường và một máy tìm kiếm ngữ nghĩa. Với máy tìm kiếm thông thường, câu truy vấn là một dãy các từ khoá, chẳng hạn như "công ty, viễn thông", máy tìm kiếm sẽ trả về một danh sách các trang Web có chứa các từ khoá trên và sắp xếp (ranking) kết quả theo "mức độ liên quan". Với một máy tìm kiếm ngữ nghĩa, ta có thể đưa một câu truy vấn có "ý nghĩa", chẳng hạn như tìm một "công ty thuộc ngành viễn thông", máy tìm kiếm sẽ trả về danh sách các tài liệu chứa tên các công ty thuộc ngành viễn thông, hay danh sách các công ty thoả mãn điều kiện trên. Điểm khác biệt ở đây là máy tìm kiếm ngữ nghĩa phải "hiểu" được tên các công ty, và "phân loại" được nó thuộc ngành nào, từ đó có thể lọc ra được các công ty thuộc ngành "viễn thông" làm kết quả trả về cho người dùng.

10.2. Kiến trúc của Web ngữ nghĩa

Kiến trúc chung của Web ngữ nghĩa được minh họa trong hình 10.3. Ở tầng thấp nhất là các tài nguyên (chẳng hạn như một tài liệu, một đối tượng, một thực thể,...) được mô tả bằng các *định danh tài nguyên thông nhất*: Uniform Resource Identifier (URI). Một URI là một chuỗi được chuẩn hoá để xác định một tài nguyên duy nhất. Tập con của URI là *định vị tài nguyên thông nhất*: Uniform Resource Locator (URL), nó chứa phương thức truy cập và vị trí của một tài liệu (ở trên mạng), ví dụ, một URL là địa chỉ trang Web về Ontology: http://en.wikipedia.org/wiki/Web_Ontology_Language, ở đây phương thức truy cập là giao thức HTTP. Một tập con khác của URI là *tên tài nguyên thống nhất*: Uniform Resource Name (URN), nó cho phép xác định một tài nguyên mà không cần phải chỉ rõ địa chỉ và phương thức truy cập đến nó. Ví dụ về một URN là số ISBN của một cuốn sách: urn:isbn:0-123-45678-9. Việc sử dụng URI là quan trọng, vì nó cho phép ta xây dựng một hệ thống phân tán, trong đó các tài nguyên có thể nằm ở nhiều nơi khác nhau trên mạng. Một biến thể khác của URI là *định danh tài nguyên được quốc tế hóa*: Internationalized Resource Identifier (IRI), nó cho phép sử dụng các ký tự Unicode⁽³⁾ trong định danh.

Để mã hóa các thông tin, dữ liệu ta sử dụng chuẩn mã hóa Unicode, đây là chuẩn thông nhất dùng để mã hóa các tập ký tự quốc tế. Nó cho phép tất cả các ngôn ngữ tự nhiên của các nước có thể được mã hóa thống nhất, tránh

⁽¹⁾ <http://www.w3.org/2001/sw/>

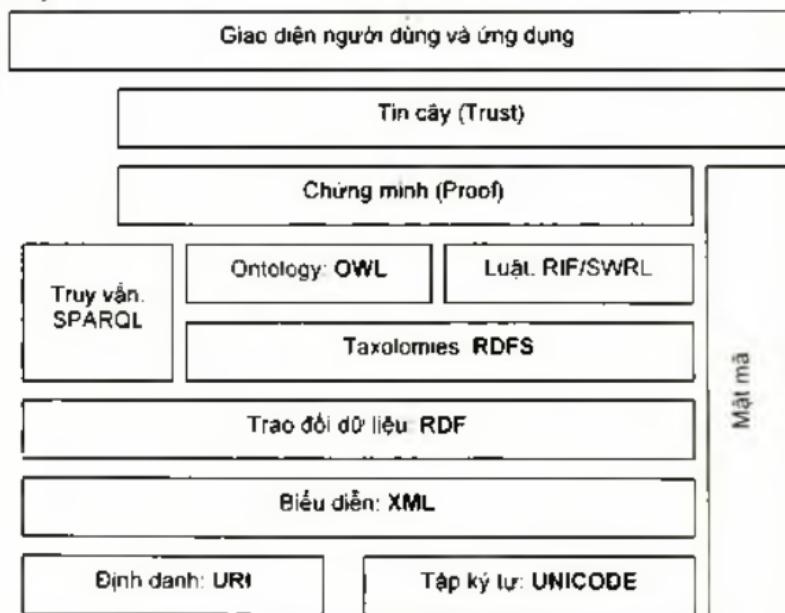
⁽²⁾ <http://www.SemanticWeb.org>

⁽³⁾ <http://unicode.org>

hiện tượng mỗi quốc gia lại sử dụng một chuẩn mã hoá riêng, gây khó khăn cho việc trao đổi dữ liệu.

Tảng tiếp theo là *ngôn ngữ đánh dấu mở rộng*: Extensible Markup Language⁽¹⁾ (XML), dùng để biểu diễn dữ liệu dưới dạng máy tính có thể hiểu và tự động xử lý được. Để giúp máy tính có thể biết được cấu trúc của file XML, cũng như thẩm định được tính chính xác của dữ liệu XML, ngôn ngữ định nghĩa kiểu của tài liệu: Document Type Definition (DTD) hay ngôn ngữ định nghĩa lược đồ XML: XML Schema Definition Language⁽²⁾ (XSDL). XML là một ngôn ngữ đánh dấu phổ dụng dùng để biểu diễn các tài liệu có cấu trúc hay bản cấu trúc ngày càng được sử dụng rộng rãi.

Ngôn ngữ biểu diễn dữ liệu cốt lõi của Web ngữ nghĩa là ngôn ngữ *Cơ cấu mô tả tài nguyên*: Resource Description Framework⁽³⁾ (RDF). Đây là ngôn ngữ dùng để biểu diễn thông tin về các tài nguyên dưới dạng đồ thị. Mục đích chủ yếu của RDF là biểu diễn các siêu dữ liệu của các tài nguyên trên WWW như *tiêu đề, tên tác giả, ngày tạo, ngày sửa đổi* của một trang Web (tất nhiên ngoài ra RDF có thể được sử dụng để biểu diễn bất kỳ loại dữ liệu nào).



Hình 10.3. Kiến trúc của Web ngữ nghĩa

(1) <http://www.w3.org/TR/REC-xml>

(2) <http://www.w3.org/TR/xmlschema-0>

(3) <http://www.w3.org/TR/REC-rdf-syntax>

Để xác định ra cấu trúc và ngữ nghĩa của RDF, ngôn ngữ lược đồ RDF: RDF Schema⁽¹⁾ (RDFS) đã được đề xuất, nó cho phép ta định nghĩa ra các lớp đối tượng, các thuộc tính của các đối tượng, mối quan hệ giữa các đối tượng/ thuộc tính, cũng như các ràng buộc của các thuộc tính.

Đối với mỗi một miền ứng dụng cụ thể, sẽ có một tập hợp các khái niệm (concept) và các mối quan hệ, ràng buộc giữa chúng tạo thành một Ontology. Một khái niệm có thể là một lớp (class), một thuộc tính (property) của một lớp hay tập từ vựng (vocabulary) sử dụng trong miền ứng dụng đó. Ontology có thể được dùng để định nghĩa ra một miền ứng dụng cũng như các cơ chế suy diễn trong miền ứng dụng đó. Để biểu diễn Ontology có thể dùng RDFS, tuy nhiên RDFS có nhược điểm là không có khả năng suy diễn. Để xoá bỏ hạn chế này, rất nhiều ngôn ngữ đã được đề xuất, chẳng hạn như ngôn ngữ suy diễn Ontology: Ontology Inference Layer⁽²⁾ (OIL), DARPA Agent Markup Language (DAML–OIL) hay ngôn ngữ Ontology cho Web: Web Ontology Language⁽³⁾ (OWL).

OWL có thể dùng để định nghĩa ngữ nghĩa và ngữ nghĩa có thể dùng để suy diễn trong một Ontology và cơ sở tri thức tương ứng. Để cung cấp các luật suy diễn dựa trên các cấu trúc trong OWL, một số ngôn ngữ luật (rule language) đang được phát triển và chuẩn hoá, hai trong số các ngôn ngữ này đang nổi lên là ngôn ngữ: Rule Interchange Format⁽⁴⁾ (RIF) và Semantic Web Rule Language⁽⁵⁾ (SWRL).

Để truy vấn (query) dữ liệu RDF trong một cơ sở tri thức, ngôn ngữ query RDF: Simple Protocol and RDF Query Language⁽⁶⁾ (SPARQL) đã được đề xuất. SPARQL là ngôn ngữ giống như ngôn ngữ truy vấn có cấu trúc: Structured Query Language (SQL) dùng để thao tác với CSDL quan hệ. Vì RDFS và OWL đều được xây dựng dựa trên RDF, nên SPARQL có thể dùng để truy vấn ontology và cơ sở tri thức một cách đồng thời. Hơn nữa, SPARQL không chỉ đơn thuần là một ngôn ngữ, mà nó còn là một giao thức để truy xuất dữ liệu RDF.

Tất cả các thao tác suy diễn đều được thực hiện ở dưới tầng chứng minh (Proof) và kết quả sẽ được sử dụng để tầng Proof chứng minh kết quả suy luận (deduction). Chứng minh hình thức cùng với dữ liệu đầu vào đáng tin cậy sẽ cho kết quả đáng tin cậy để chuyên cho tầng cao nhất là lớp ứng dụng. Để đảm bảo dữ liệu đầu vào đáng tin cậy, có thể sử dụng mật mã

⁽¹⁾ <http://www.w3.org/TR/rdf-schema/>

⁽²⁾ <http://www.ontoknowledge.org/oil/>

⁽³⁾ <http://www.w3.org/TR/owl-features/>

⁽⁴⁾ <http://www.w3.org/TR/rif-bld/>

⁽⁵⁾ <http://www.w3.org/Submission/SWRL/>

⁽⁶⁾ <http://www.w3.org/TR/rdf-sparql-query/>

(cryptography), chẳng hạn như sử dụng *chữ ký điện tử* (digital signature) để thẩm định xuất xứ của nguồn dữ liệu.

Ngoài các ngôn ngữ được liệt kê trong hình 10.3, hiện tại còn có rất nhiều ngôn ngữ tương đương đã được đề xuất. Thậm chí còn có nhiều kiến trúc về Web ngữ nghĩa khác như một số tác giả đề xuất trong [Heff01, JDF02].

10.3. Các ngôn ngữ nền tảng cho Web ngữ nghĩa

10.3.1. Ngôn ngữ trao đổi dữ liệu XML

Những viền gạch đề xây dựng lên Web ngữ nghĩa là một loạt các ngôn ngữ. XML (Extensible Markup Language) là ngôn ngữ ở mức thấp nhất trong hệ thống các ngôn ngữ này. Khác với ngôn ngữ HTML (HyperText Markup Language) – ngôn ngữ dùng để biểu diễn cách hiển thị nội dung các văn bản có cấu trúc hữu hạn trên các trình duyệt Web – XML là một ngôn ngữ dùng để định nghĩa và miêu tả cấu trúc dữ liệu bất kỳ. XML là một tập các luật để định nghĩa các thẻ, để chia tài liệu thành các phần và xác định rõ các phần khác nhau của tài liệu. Nó là một ngôn ngữ siêu đánh dấu (meta-markup language), nó có cơ chế định nghĩa cú pháp được sử dụng trong các tài liệu có cấu trúc, có ngữ nghĩa và được áp dụng cho một lĩnh vực cụ thể.

Điều đầu tiên cần hiểu về XML là nó không chỉ là một ngôn ngữ đánh dấu như ngôn ngữ đánh dấu siêu văn bản như HTML. Vì ngôn ngữ HTML có một tập các thẻ được định nghĩa cố định. Nếu trong ngôn ngữ HTML không có thẻ mà ta muốn, thì đây là điều ta không thể nào thêm được thẻ minh yêu cầu. HTML là một ngôn ngữ đánh dấu vừa bao hàm ngữ nghĩa, cấu trúc và định dạng hiển thị. Ví dụ, thẻ **** là thẻ định dạng sẽ báo cho trình duyệt hiển thị chữ đậm; **** là một thẻ ngữ nghĩa có nghĩa là nội dung nằm trong phần này là quan trọng; **<TD>** là một thẻ cấu trúc, nó chỉ ra nội dung là một ô trong một bảng. Trong thực tế, một thẻ có thể có 3 ý nghĩa, ví dụ: thẻ **<H1>** có ý nghĩa báo cho trình duyệt cần hiển thị kích thước 20 point font chữ Helvetica và in đậm, là tiêu đề của trang.

Để biểu diễn một bài hát, trong HTML ta có thể dùng các thẻ định nghĩa tiêu đề **<dt>**, thẻ định nghĩa dữ liệu **<dd>**, thẻ danh sách ****, nhưng các thẻ này chẳng có ý nghĩa gì với bài hát cả. Nội dung tài liệu HTML có dạng như sau:

```
<dt>Hot Cop  
<dd>by Jacques Morali, Henri Belolo, and Victor Willis  
<ul>  
  <li>Producer:Jacques Morali  
  <li>Publisher:PolyGram Records
```

```

<li>Length:6:20
<li>Written:1978
<li>Artist:Village People
</ul>

```

Tuy nhiên, XML lại là một ngôn ngữ đánh dấu, trong đó nó cho phép định nghĩa ra các thẻ ta cần và có gắn một ý nghĩa cụ thể. Những thẻ này phải được tổ chức theo một nguyên tắc cụ thể nào đó, nhưng ý nghĩa của nó lại rất mềm dẻo. Dữ liệu ở trên có thể được biểu diễn bằng XML như sau:

```

<?xml version="1.0" standalone="yes"?>
<SONG>
  <TITLE>Hot Cop</TITLE>
  <COMPOSER>Jacques Morali</COMPOSER>
  <COMPOSER>Henri Belolo</COMPOSER>
  <COMPOSER>Victor Willis</COMPOSER>
  <PRODUCER>Jacques Morali</PRODUCER>
  <PUBLISHER>PolyGram Records</PUBLISHER>
  <LENGTH>6:20</LENGTH>
  <YEAR>1978</YEAR>
  <ARTIST>Village People</ARTIST>
</SONG>

```

Thay vì sử dụng các thẻ chung như `<dt>` và ``, tài liệu này sử dụng các thẻ rất có ý nghĩa như `<SONG>`, `<TITLE>`, `<COMPOSER>` và `<YEAR>`. Điều này có một số ưu điểm: nó làm cho con người dễ hiểu và dễ đọc tài liệu và có thể hiểu được mục đích của tác giả định làm điều gì.

XML cũng làm cho các thực thể không phải là con người như robot cũng có thể tự động tìm ra tất cả các bài hát trong tài liệu. Nếu đó là một tài liệu HTML, thì robot không thể nào biết được ý nghĩa của từng thẻ và nó không thể nào xác định được đâu là tiêu đề của bài hát, đâu là tác giả của bài hát.

Tên các phần tử trong XML có thể được lựa chọn sao cho nó có ý nghĩa thực tiễn, chẳng hạn nó có thể là tên của một CSDL. Các thẻ đánh dấu có 3 kiểu ý nghĩa: *cấu trúc* (structure), *ngữ nghĩa* (semantics) và *kiểu dáng* (style). Tính cấu trúc chia tài liệu thành một cây, ngữ nghĩa liên kết từng phần tử đến thế giới thực bên ngoài tài liệu, kiểu dáng xác định cách hiển thị nội dung của các phần tử. Cấu trúc liên quan đến dạng của tài liệu, các tài liệu XML thường có chung một cấu trúc, nó sẽ có một gốc. Tên của các thẻ không ảnh hưởng đến cấu trúc của tài liệu. Ngữ nghĩa tồn tại bên ngoài tài liệu, nó là ý tưởng của tác giả hoặc chương trình máy tính tạo ra hoặc đọc các file XML. Máy tính thì không thể hiểu được ý nghĩa của bất kỳ cái gì cả, do đó cho dù ta dùng thẻ nào đi chăng nữa, thì với máy tính cũng giống nhau. Do vậy, sẽ rất hữu ích nếu chọn các thẻ có ý nghĩa gần với thông tin nó chưa. Tuy ngôn ngữ XML có mục đích chính là dùng để biểu diễn dữ liệu, song nó được bổ sung một ngôn ngữ Cascading Style Sheets (CSS) và

Extensible Style Language (XSL), cho phép các trình duyệt đưa vào đó để biết cách hiển thị nội dung của file XML theo một cách thức nào đó được thể hiện trong nội dung của file CSS (hay XSL) tương ứng.

• Định nghĩa kiểu dữ liệu của cấu trúc file XML:

Các thẻ có thể có trong một file, cũng như cấu trúc của file XML được mô tả cụ thể trong phần **định nghĩa kiểu tài liệu**: Document Type Definition (DTD). Ta sẽ tìm hiểu một phần DTD, hiện tại ta hiểu DTD như là một bảng từ vựng và cú pháp cho một loại tài liệu nào đó. Phân định nghĩa kiểu tài liệu cung cấp một danh sách các phần tử, các thuộc tính, các ký hiệu, các thực thể trong tài liệu và các quan hệ giữa chúng. DTD xác định một tập các luật cho cấu trúc của tài liệu. Ví dụ, một DTD có thể ràng buộc mỗi phần tử BOOK chỉ có thẻ có một phần tử con là ISBN, một phần tử con là TITLE và có một hay nhiều phần tử con AUTHOR. Mọi phần tử BOOK có thẻ có hoặc không một phần tử con SUBTITLE. Khai báo kiểu dữ liệu, cấu trúc của file XML còn có ý nghĩa là để thâm định tính chính xác của file XML (chẳng hạn, nó có được viết đúng cấu trúc, nội dung của một phần tử có đúng định dạng?).

Mỗi thẻ sử dụng trong một tài liệu XML phải được khai báo trong DTD. Một khai báo phần tử sẽ xác định tên và một số nội dung có thẻ có của phần tử này. Danh sách nội dung của một phần tử còn được gọi là đặc tả nội dung. Đặc tả nội dung này sử dụng ngữ pháp đơn giản để chỉ ra cái gì được phép và không được phép có mặt trong tài liệu. Ngoài ra, DTD cho phép ta xác định quan hệ giữa các phần tử trong tài liệu, chẳng hạn nó quy định phần tử GIVEN_NAME phải xuất hiện dang trước phần tử SURNAME, và SURNAME phải xuất hiện trước POSITION,...

• Khai báo phần tử gốc (root) của tài liệu:

Trong khai báo DTD, đầu tiên là khai báo cho phần tử gốc, ví dụ để phần tử SEASON là phần tử gốc của tài liệu XML ta sử dụng cú pháp:

```
<!DOCTYPE SEASON [ ... ]>
```

Ở bên trong dấu ba chấm (...) là khai báo các phần tử con của nút gốc SEASON. Để khai báo các phần tử con trong tài liệu ta sử dụng cú pháp ELEMENT như sau:

```
<ELEMENT YEAR (#PCDATA)>
```

Trong đó #PCDATA thể hiện rằng, kiểu dữ liệu của phần tử YEAR là chuỗi văn bản (text). Cụ thể hơn, khai báo này bắt buộc nội dung của phần tử YEAR chỉ có thẻ là chuỗi văn bản, chứ không thể có một phần tử con (element). Do đó, các phần tử sau là hợp lệ:

```
<YEAR>1998</YEAR>.  
<YEAR>1998 C.E.</YEAR>  
<YEAR>
```

The year of our lord one thousand,nine
hundred,&ninety-eight

```
</YEAR>
```

Nhưng phần tử sau lại không hợp lệ vì nó chưa phân tử con:

```
<YEAR>
  <MONTH>January</MONTH>
</YEAR>
```

Khoảng trắng và dấu tab trong khai báo DTD không có ý nghĩa và ngay cả thứ tự khai báo các phần tử cũng không có ý nghĩa.

- **Danh sách các phần tử con:**

Khi muốn điều khiển các phần tử trong phần nội dung của một phần tử, chẳng hạn, muốn với mỗi phần tử LEAGUE phải có phần tử LEAGUE_NAME, sử dụng cú pháp như sau:

```
<!ELEMENT LEAGUE (LEAGUE_NAME) >
<!ELEMENT LEAGUE_NAME (#PCDATA) >
```

Mỗi phần tử nên được khai báo trong phần `<!ELEMENT>` và chỉ khai báo một lần, thứ tự khai báo của các phần tử này không làm thay đổi nội dung khai báo. Khai báo DTD có thể được chèn vào đầu nội dung của một file XML, hay cũng có thể đứng độc lập ở một file và được tham chiếu trong file XML. Ví dụ về trường hợp chèn nội dung DTD vào đầu file như sau:

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON [
<!ELEMENT YEAR (#PCDATA) >
<!ELEMENT LEAGUE (LEAGUE_NAME) >
<!ELEMENT LEAGUE_NAME (#PCDATA) >
]>
<SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
    <LEAGUE_NAME>American League</LEAGUE_NAME>
  </LEAGUE>
  <LEAGUE>
    <LEAGUE_NAME>National League</LEAGUE_NAME>
  </LEAGUE>
</SEASON>
```

- **Thứ tự của các phần tử trong tài liệu XML:**

Ta thêm các quy định chặt hơn cho phần tử SEASON: một phần tử SEASON chỉ chứa một phần tử YEAR, sau đó là 2 phần tử LEAGUE. Để làm điều này sử dụng cú pháp như sau:

```
<!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE) >
```

Một danh sách có thứ tự các phần tử con được phân cách bởi dấu phẩy là khai báo các phần tử con của một phần tử. Theo cú pháp trên thì một phần tử SEASON hợp lệ phải chứa một phần tử YEAR sau đó là 2 phần tử LEAGUE và không chứa gì thêm cả. Kết hợp với tài liệu ở trên sẽ có khai báo sau:

```
<!DOCTYPE SEASON [
<!ELEMENT YEAR (#PCDATA) >
<!ELEMENT LEAGUE (LEAGUE_NAME) >
```

```
<!ELEMENT LEAGUE_NAME (#PCDATA)>
<!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE) >
]>
```

• **Khai báo có một hay nhiều phần tử con:**

Gia sư muốn khai báo mỗi phần tử DIVISION đều có DIVISION_NAME và một hoặc nhiều phần tử con TEAM. Để làm được điều này, thêm dấu cộng (+) vào sau phần tử TEAM như khai báo sau:

```
<!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
<!ELEMENT DIVISION_NAME (#PCDATA)>
<!ELEMENT TEAM (#PCDATA)>
```

• **Khai báo không có hoặc nhiều phần tử con:**

Khi muốn khai báo một phần tử có thể không có hoặc có rất nhiều phần tử con nào đó, thì thêm dấu hoa thị (*) vào cuối tên phần tử đó. Chẳng hạn, muốn khai báo một đội bóng cháy có tên thành phố, tên đội bóng và có thể không có cầu thủ nào hoặc có nhiều cầu thủ ta khai báo như sau:

```
<!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
<!ELEMENT TEAM_CITY (#PCDATA)>
<!ELEMENT TEAM_NAME (#PCDATA)>
```

• **Khai báo không có hoặc một phần tử con nào đó:**

Khi muốn khai báo một phần tử không có hoặc một phần tử con nào đó, thì thêm dấu chấm hỏi vào cuối tên của phần tử con đó. Ví dụ, muốn khai báo cho phần tử PLAYER phải có một phần tử SURNAME, một phần tử GIVEN_NAME, một phần tử POSITION và có thể có 0 hoặc 1 phần tử GAMES, khai báo như sau:

```
<!ELEMENT PLAYER (GIVEN_NAME, SURNAME, POSITION, GAMES?)>
```

• **Lựa chọn:**

Nếu muốn cho phép người dùng có thể lựa chọn các phần tử xuất hiện ở một số vị trí tùy chọn. Ví dụ, khi muốn mô tả các mặt hàng mà khách hàng mua, khi thanh toán thì khách hàng có thể sử dụng tiền mặt hoặc thẻ tín dụng, nhưng không thẻ có cả hai. Để khai báo ta sử dụng dấu thẳng đứng (!) giữa các phần tử. Chẳng hạn, áp dụng cho trường hợp trên:

```
<!ELEMENT PAYMENT (CASH | CREDIT_CARD)>
```

Kiểu khai báo này được gọi là lựa chọn, có thể liệt kê nhiều phần tử con giữa các dấu thẳng đứng khi ta chỉ muốn một phần tử được lựa chọn. Ví dụ khai:

```
<!ELEMENT PAYMENT (CASH | CREDIT_CARD | CHECK)>
```

• **Khai báo các phần tử con với cặp ngoặc:**

Các phân tử có thể nhóm lại với nhau bằng cách sử dụng cặp ngoặc, mỗi cặp ngoặc chứa một số phân tử tạo thành một phân tử duy nhất. Những phân tử trong ngoặc này lại có thể nằm trong cặp ngoặc khác, hơn nữa, có thể kết hợp với các khai báo khác như dấu cộng (+), dấu phẩy (,) hay dấu chấm than (!). Việc kết hợp này sẽ tạo ra các cấu trúc phức tạp.

Một số phần tử có thể xuất hiện theo một thứ tự ngẫu nhiên nào đó. Chẳng hạn, các bài báo (ARTICLE) thường có một tiêu đề (TITLE) theo sau là các đoạn văn bản (P) hoặc ảnh (PHOTO), biểu đồ (GRAPH)... và cuối cùng thì có thể có câu kết. Dưới đây là phần khai báo cho ví dụ trên:

```
<!ELEMENT ARTICLE (TITLE, (P | PHOTO | GRAPH  
| SIDEBAR | PULLQUOTE | SUBHEAD) *, BYLINE?)>
```

- **Nội dung trộn:**

Ta có thể khai báo một phần tử có thể chứa cả dữ liệu text lẫn phân tử con, trường hợp này gọi là nội dung trộn. Ví dụ:

```
<!ELEMENT TEAM (#PCDATA | TEAM_CITY | TEAM_NAME | PLAYER)*>
```

- **Các phần tử rỗng:**

Trong một số trường hợp, có thể định nghĩa một phần tử rỗng (phân tử không có nội dung). Chẳng hạn trong HTML ta có các thẻ rỗng , <HR> và
. Để định nghĩa các thẻ rỗng sử dụng từ khoá EMPTY. Ví dụ:

```
<!ELEMENT BR EMPTY>
```

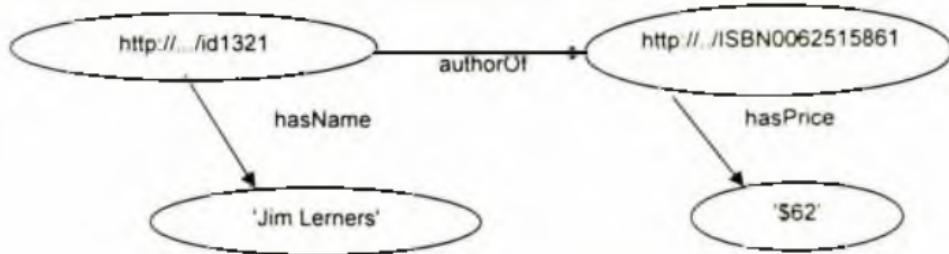
Ngoài ngôn ngữ DTD, để định nghĩa cấu trúc của một file XML, có thể dùng ngôn ngữ định nghĩa lược đồ XML: XML Schema Definition Language (XSDL). Vì ra đời sau DTD, XSDL cung cấp khả năng khai báo đa dạng và chặt chẽ hơn nhiều so với DTD. Cuốn giáo trình này sẽ không đi sâu vào chi tiết ngôn ngữ này, thông tin đầy đủ về ngôn ngữ này có thể xem tại tài liệu ở địa chỉ <http://www.w3.org/TR/xmlschema11-1/>, hay tham khảo ở [Har99].

10.3.2. Ngôn ngữ mô tả tài nguyên RDF và RDFS

Ngôn ngữ cơ cấu mô tả tài nguyên: Resource Description Framework⁽¹⁾ (RDF) cũng là một đề xuất của tổ chức W3C để chuẩn hóa cách định nghĩa và sử dụng siêu dữ liệu của các tài nguyên trên Web: *tài nguyên* (resource) ở đây có thể là bất kỳ đối tượng nào được xác định bằng một URI; *mô tả* (description) ở đây là *khai báo* các thuộc tính của tài nguyên được xác định bằng một URI nào đó; và *cơ cấu* (framework) là một *mô hình* hay *ngữ pháp* dùng trong các câu khai báo. Với một miền ứng dụng cụ thể, cũng có thể coi *mô hình* là một tập hợp các *khai báo* (hay mô tả). Cú pháp của một *khai báo* của ngôn ngữ RDF là bộ ba: *vị ngữ* (predicate), *chủ ngữ* (subject) và *đối tượng* (Object). *Chủ ngữ* ở đây là một tài nguyên, *vị ngữ* cũng là một tài nguyên, còn *đối tượng* có thể là một tài nguyên hay một giá trị có kiểu đơn giản nào đó. Cũng có thể coi một khai báo là một bộ ba: *đối tượng_O* (tương đương với chủ ngữ) – *thuộc_tính_A* (tương đương với vị ngữ) – *giá_trị_V* (tương đương với đối tượng) và thường được viết dưới dạng $A(O, V)$ để mô tả đối tượng O có thuộc tính A , có giá trị V . Ta cũng có thể biểu diễn mô tả này là quan hệ giữa đối tượng O với giá trị V là một

⁽¹⁾ <http://www.w3.org/RDF>

cạnh có hướng được gán nhãn *A* như được minh họa trên Hình 10.4. Về tổng quát mô hình trong RDF được dựa trên mô hình đồ thị như Hình 10.4.



Hình 10.4. Một ví dụ về một đoạn mô tả tài nguyên

Quan hệ trên có thể biểu diễn bằng cú pháp A(O, V) như sau:

```
hasName ('http://www.w3.org/employee/id1321',
          'Jim Lerners')
authorOf ('http://www.w3.org/employee/id1321',
          'http://www.books.org/ISBN0062515861')
hasPrice ('http://www.books.org/ISBN0062515861', "$62")
```

Để máy tính có thể xử lý được quan hệ này, RDF được biểu diễn bằng cách sử dụng XML, do đó các quan hệ A(O, V) trên được biểu diễn như sau:

```
<rdf:Description rdf:about=
    "http://www.w3.org/employee/id1321">
<hasName rdf:resource="Jim Lerners"/>
</rdf:Description>
<rdf:Description rdf:about=
    "http://www.books.org/ISBN0062515861">
<hasPrice rdf:resource="$62"/>
</rdf:Description>
<rdf:Description rdf:about =
    "http://www.w3.org/employee/id1321">
<authorOf rdf:resource=
    "http://www.books.org/ISBN0062515861"/>
</rdf:Description>
```

Trong trường hợp muốn liệt kê một nhóm các tài nguyên cùng loại (chẳng hạn như, trong trường hợp một người là tác giả của nhiều cuốn sách), RDF cũng cung cấp cơ chế khai báo để hỗ trợ biểu diễn dữ liệu ngắn gọn hơn là *container*. Container là một tài nguyên nó chứa các tài nguyên giá trị khác. Các tài nguyên được chứa trong container được gọi là thành viên. RDF định nghĩa 3 loại container khác nhau đó là: rdf:Bag (chứa một nhóm các tài nguyên, cho phép các thành viên có thể trùng nhau), rdf:Seq (chứa một danh sách có thứ tự các tài nguyên, cho phép các thành viên có thể trùng nhau) và rdf:Alt (chứa danh sách các tài nguyên/giá trị có thể dùng

thay thế được cho nhau, chẳng hạn một người có nhiều tên gọi). Giả sử người trong hình 10.4 là tác giả của 3 cuốn sách, thì có thể khai báo như sau thay vì phải khai báo 3 thuộc tính authorOf:

```
<rdf:Description
  rdf:about="http://www.w3.org/employee/id1321">
  <authorOf>
    <rdf:Bag>
      <rdf:li
        rdf:resource="http://www.books.org/ISBN0062515861"/>
      <rdf:li
        rdf:resource="http://www.books.org/ISBN0062515862"/>
      <rdf:li
        rdf:resource="http://www.books.org/ISBN0062515863"/>
    </rdf:Bag>
  </authorOf>
</rdf:Description>
```

Tuy nhiên, RDF chỉ cho phép định nghĩa các quan hệ, chứ không nói rõ chỉ có thể có các loại quan hệ nào, hay các kiểu đối tượng có thể có trong miền hiện tại. Để bổ sung chức năng này, *luật đồ RDF*: RDF Schema – (RDFS) được đề xuất. RDFS cho phép định nghĩa *tập từ vựng* (vocabulary) cho một miền ứng dụng nào đó, *các lớp* (class), *lớp con* (sub-class), *thuộc tính* (property), *thuộc tính con* (sub-property), *miền xác định* và *miền giá trị* của một thuộc tính,... Mọi thứ trong RDF đều có thể coi là tài nguyên, các lớp về bản chất cũng là tài nguyên, nhưng nó còn là một tập hợp các tài nguyên cùng loại. RDFS định nghĩa ý nghĩa của các khái niệm này. Các khai báo của RDFS cũng được biểu diễn giống như các khai báo trong RDF, do đó chúng cũng có thể biểu diễn thông qua XML. Một số lớp được định nghĩa sẵn dùng trong RDFS được liệt kê trong Bảng 10.1 và một số thuộc tính được định nghĩa sẵn liệt kê trong Bảng 10.2.

Quay lại ví dụ trong Hình 10.4, có thể dùng RDFS để khai báo các lớp, các thuộc tính cùng các ràng buộc cho nó. Quan sát ta thấy, ở đây có các lớp đối tượng: Person (người), một lớp Book (sách), lớp con của lớp Book là HardCover (sách in đóng bìa cứng để phân biệt với sách điện tử); thuộc tính là hasName có miền xác định là một người nào đó, miền xác định là một chuỗi ký tự; hasPrice có miền xác định là một đối tượng có kiểu Book, còn miền giá trị có kiểu chuỗi; thuộc tính authorOf có miền giá trị là Person, còn miền xác định là Book.

```
<rdfs:Class rdf:about="Person"/>
<rdfs:Class rdf:about="Book"/>
<rdfs:Class rdf:about="HardCover">
  <rdfs:subClassOf rdf:resource="#Book"/>
</rdfs:Class>
```

Bảng 10.1. Các lớp trong RDFS

| Phản từ | Mô tả | Lớp con của rdfs:subClassOf | Kiểu rdf:type |
|--------------------------------------|-------------------------------|--------------------------------|---------------|
| rdfs:Resource | Tất cả các tài nguyên | rdfs:Resource | rdfs:Class |
| rdfs:Class | Tất cả các lớp | rdfs:Resource | rdfs:Class |
| rdfs:Literal | Giá trị đơn giản | rdfs:Resource | rdfs:Class |
| rdfs:Datatype | Kiểu dữ liệu | rdfs:Class | rdfs:Class |
| rdf:XMLLiteral | Giá trị đơn giản XML | rdfs:Literal | rdfs:Datatype |
| rdf:Property | Các thuộc tính | rdfs:Resource | rdfs:Class |
| rdf:Statement | Các khai báo | rdfs:Resource | rdfs:Class |
| rdf:List | Các danh sách | rdfs:Resource | rdfs:Class |
| rdfs:Container | Container | rdfs:Resource | rdfs:Class |
| rdf:Bag | Container không có thứ tự | rdfs:Container | rdfs:Class |
| rdf:Seq | Container có thứ tự | rdfs:Container | rdfs:Class |
| rdf:Alt | Container thay thế | rdfs:Container | rdfs:Class |
| rdfs:Container MembershipProperty | Thuộc tính quan hệ thành viên | rdf:Property | rdfs:Class |

Bảng 10.2. Một số thuộc linh dung trong RDFS

| Phản từ | Mô tả | Miền xác định rdfs:domain | Miền giá trị rdfs:range |
|--------------------|---|------------------------------|----------------------------|
| rdfs:range | Hạn chế các đối tượng | rdf:Property | rdfs:Class |
| rdfs:domain | Hạn chế các đối tượng | rdf:Property | rdfs:Class |
| rdf:type | Thể hiện của | rdfs:Resource | rdfs:Class |
| rdfs:subClassOf | Lớp con của | rdfs:Class | rdfs:Class |
| rdfs:subPropertyOf | Thuộc tính con của | rdf:Property | rdf:Property |
| rdfs:label | Nhân có nghĩa (dành cho người dùng dễ hiểu) | rdfs:Resource | rdfs:Literal |
| rdfs:comment | Lời chú giải (dành cho người dùng dễ hiểu) | rdfs:Resource | rdfs:Literal |
| rdfs:member | Thành viên của container | rdfs:Resource | rdfs:Resource |
| rdf:first | Phản từ đầu tiên | rdf:List | rdfs:Resource |
| rdf:rest | Phản còn lại của danh sách | rdf:List | rdf:List |
| rdf:_1, rdf:_2, | Thành viên của container | rdfs:Container | rdfs:Resource |
| rdfs:seeAlso | Thông tin xem thêm | rdfs:Resource | rdfs:Resource |
| rdfs.isDefinedBy | Định nghĩa bởi | rdfs:Resource | rdfs:Resource |
| rdf:value | Các giá trị có cấu trúc | rdfs:Resource | rdfs:Resource |
| rdf:object | Đối tượng của khai báo | rdf:Statement | rdfs:Resource |
| rdf:predicate | Predicate khai báo | rdf:Statement | rdfs:Resource |
| rdf:subject | Chủ thể của khai báo | rdf:Statement | rdfs:Resource |

Tiếp đến ta có thể khai báo thuộc tính:

```
<rdfs:Property rdf:about="hasName">
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdfs:Property>
<rdfs:Property rdf:about="hasPrice">
    <rdfs:domain rdf:resource="#Book"/>
    <rdfs:range
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdfs:Property>
<rdfs:Property rdf:about="authorOf">
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#Book"/>
</rdfs:Property>
```

10.3.3. Ngôn ngữ OIL và DAML-OIL

Ngôn ngữ RDF có nhiều hạn chế vì nó không cung cấp khả năng suy diễn. Để bổ sung khả năng suy diễn, ngôn ngữ Ontology Inference Layer (OIL) được đề xuất. Ngôn ngữ OIL được phát triển dựa trên các khái niệm dùng trong *logic mô tả*: Description Logic (DL) và các hệ thống dựa trên frame (frame còn gọi là lớp). Ngôn ngữ DAML-OIL⁽¹⁾ là phiên bản tiếp theo của OIL. DAML-OIL giống ngôn ngữ OIL ở nhiều mặt, nhưng nó được tích hợp chặt chẽ hơn với RDFS, do đó có ưu điểm là có thể tận dụng được những hạ tầng sẵn có của RDFS. Ngoài ra, DAML-OIL còn bổ sung một số khả năng để loại bỏ được một số hạn chế của RDFS.

Ví dụ, để mô tả một số thông tin liên quan đến một cửa hàng bán đồ thể thao có tên "Super Sports Inc.", để mô tả lớp sản phẩm "Product" và số sản phẩm "productNumber" là một thuộc tính của lớp Product, với RDFS ta có thể khai báo các lớp, các thuộc tính trực tiếp:

```
<rdfs:Class rdf:ID="Product">
    <rdfs:label>Product</rdfs:label>
    <rdfs:comment>An item sold by Super Sports
Inc.</rdfs:comment>
</rdfs:Class>
<rdfs:Property rdf:ID="productNumber">
    <rdfs:label>Product Number</rdfs:label>
    <rdfs:domain rdf:resource="#Product"/>
    <rdfs:range
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdfs:Property>
```

⁽¹⁾ <http://www.daml.org/2000/10/damlont.html>

Tiếp đến ta có thể khai báo một thẻ hiện của lớp trên như sau:

```
<Product rdf:ID="WaterBottle">
  <rdfs:label>Water Bottle</rdfs:label>
  <productNumber>38267</productNumber>
</Product>
```

Người dùng sẽ kỳ vọng rằng, giá trị của thuộc tính productNumber là một con số nào đó. Khai báo RDFS đã định nghĩa giá trị của thuộc tính productNumber là có kiểu literal. Tuy nhiên, một chuỗi bất kỳ (không phải là các con số) cũng có thể có kiểu là literal. Do đó, ngôn ngữ RDFS là không đủ chặt chẽ. DAML+OIL cho phép khai báo giá trị của thuộc tính được giới hạn vào các kiểu dữ liệu được định nghĩa trong *ngôn ngữ định nghĩa lược đồ XML*. XSDL hay thậm chí là kiểu dữ liệu do người dùng định nghĩa. Với trường hợp này, có thể khai báo bằng ngôn ngữ DAML (sử dụng thuộc tính của RDF là DatatypeProperty) như sau:

```
<daml:DatatypeProperty rdf:ID="productNumber">
  <rdfs:label>Product Number</rdfs:label>
  <rdfs:domain rdf:resource="#Product"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#nonNegative
Integer"/>
</daml:DatatypeProperty>
```

Chú ý là, trong khai báo trên sử dụng tiền tố "daml" để biểu diễn không gian tên của DAML+OIL được xác định tại địa chỉ <http://www.w3.org/2001/10/daml+oil#>. Một ví dụ khác là DAML-OIL cho phép định nghĩa một thuộc tính có giá trị duy nhất (giống như thuộc tính khoá của một bảng trong CSDL quan hệ). Chẳng hạn, khi muốn số hiệu sản phẩm productNumber phải có giá trị duy nhất, khai báo như sau:

```
<daml:DatatypeProperty rdf:ID="productNumber">
  <rdfs:label>Product Number</rdfs:label>
  <rdfs:domain rdf:resource="#Product"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#nonNegative
Integer"/>
  <rdf:type rdf:resource=
"http://www.w3.org/2001/10/daml+oil#UniqueProperty"/>
</daml:DatatypeProperty>
```

10.3.4. Ngôn ngữ OWL

Ngôn ngữ Ontology cho Web: Ontology Web Language¹¹ (OWL) mở rộng ngôn ngữ RDF và RDFS. Cũng có thể coi OWL là một phiên bản nâng cấp của ngôn ngữ DAML-OIL, dựa trên nghiên cứu (research-based). Tô

¹¹ <http://www.w3.org/2004/owl>

chức W3C đã thành lập nhóm "Web Ontology Working Group" vào ngày 1/11/2001 để nghiên cứu ngôn ngữ này, và tài liệu chính thức của OWL đã được phát hành vào ngày 10/02/2004. Mục đích chính của OWL là mang lại khả năng suy diễn của logic mô tả cho Web ngữ nghĩa.

Ví dụ về một số luật suy diễn: a) *quan hệ thành viên*: Giả sử X là một thể hiện của lớp C, còn C lại là một lớp con của D, thì có thể suy ra X cũng là một thể hiện của D; b) *quan hệ tương đương của các lớp*: nếu lớp A tương đương với lớp B, còn lớp B có quan hệ tương đương với lớp C, có thể suy ra lớp A là tương đương với lớp C; c) *thẩm định tính nhất quán*: X là một thể hiện của lớp A và B, nhưng lớp A và B là không giao nhau, thì có thể suy ra Ontology là không nhất quán; d) *điều kiện phân lớp*: nếu có một cặp *thuộc tính – giá trị* nào đó là điều kiện dù để xác định quan hệ thành viên của một lớp A, khi đó nếu có một cá thể X thoả mãn điều kiện trên, thì có thể suy ra X là một thể hiện của lớp A. Cơ chế suy diễn này rất quan trọng, vì có thể kiểm tra tính nhất quán của Ontology và cơ sở tri thức; tìm ra được các quan hệ không mong muốn giữa các lớp; hay tự động phân các thể hiện vào các lớp. Do đó, tạo cho ta khả năng phát triển một Ontology lớn có sự tham gia của nhiều người; hay tích hợp và chia sẻ các nguồn Ontology với nhau mà vẫn đảm bảo được tính đúng đắn, toàn vẹn và nhất quán. Để các máy tính có thể tự động suy diễn được với Ontology sẵn có, OWL ánh xạ các phát biểu sang logic mô tả như đã nói ở trên, sau đó có thể sử dụng các bộ *suy diễn* (reasoner) như FaCT++⁽¹⁾ (tiền thân là FaCT) và RACER⁽²⁾.

Tuy nhiên, không phải mọi thứ trong RDF đều có thể diễn đạt bằng ngôn ngữ DL. Chẳng hạn như lớp của các lớp không thể diễn đạt bằng ngôn ngữ DL, và một số biểu thức của bộ ba A(O, V) không có ý nghĩa gì trong DL. Đây là nguyên nhân tại sao OWL chỉ được coi là sự mở rộng cú pháp của RDF/RDFS. Để giải quyết một phần nhược điểm này và cung cấp khả năng phân tầng trong OWL, 3 ngôn ngữ con của OWL đã được đề xuất là: OWL Lite, OWL DL và OWL Full. OWL Lite là thành phần đơn giản nhất có thể sử dụng để diễn tả một phân hoạch và các ràng buộc đơn giản. OWL DL hỗ trợ khả năng diễn đạt nhiều nhất, nó hỗ trợ tính tương thích với ngôn ngữ DL. OWL Full sử dụng tất cả các quy tắc của OWL cũng như cho phép kết hợp các quy tắc này theo một cách bất kỳ và hoàn toàn tương thích với RDF và RDFS. Tuy ngôn ngữ OWL ra đời sau ngôn ngữ DAML-OIL và có

⁽¹⁾ <http://owl.man.ac.uk/factplusplus/>

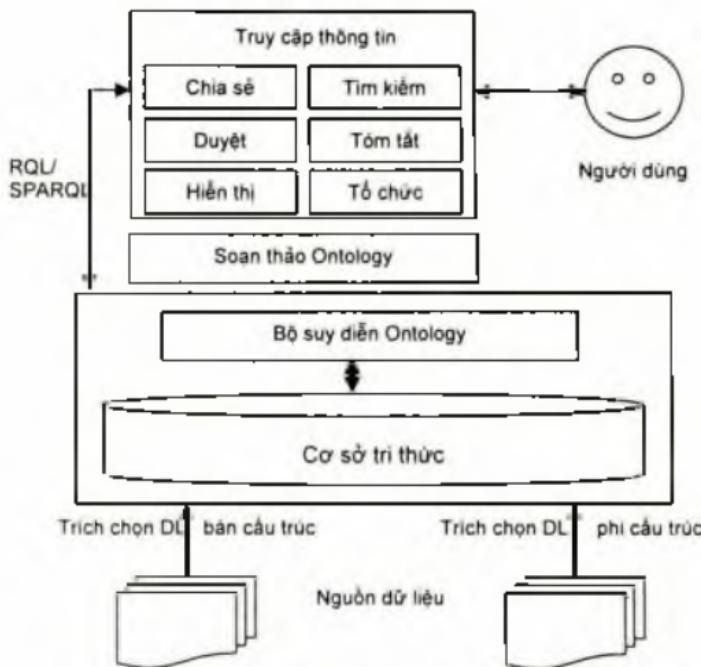
⁽²⁾ <http://www.sts.tu-harburg.de/~r.f.moeller/racer/>

nhiều ưu điểm, song trên thực tế tùy vào từng ứng dụng mà ngôn ngữ OWL hay DAML-OIL được chọn để sử dụng.

10.4. Tiệm cận tới Web ngữ nghĩa

10.4.1. Kiến trúc của một hệ thống ứng dụng Web ngữ nghĩa

Web ngữ nghĩa được thiết kế và được coi là thể hệ tiếp theo của WWW, cách thứ nhất để tạo ra hệ thống ứng dụng Web ngữ nghĩa là tạo siêu dữ liệu (bằng tay) cho các tài liệu trong ứng dụng này, hay chuyên hệ thống WWW hiện tại sang các ứng dụng Web ngữ nghĩa bằng cách thêm các siêu dữ liệu cho chúng. Nhưng việc tạo ra một ứng dụng Web ngữ nghĩa cũng cần nhiều thời gian để tạo siêu dữ liệu và đường như là không thể thực hiện được đối với các hệ thống rất lớn, chẳng hạn như WWW hiện tại.



Hình 10.5. Kiến trúc của một ứng dụng Web ngữ nghĩa

Một cách tiếp cận khác là tận dụng các nguồn tài nguyên sẵn có là WWW hiện tại, bằng cách sử dụng các phương pháp khai phá dữ liệu (như trích chọn thông tin) để có thể tự động tạo ra các siêu dữ liệu cho các ứng dụng. Hình 10.5 mô tả kiến trúc của một hệ thống ứng dụng Web ngữ nghĩa bằng cách khai thác nguồn tài nguyên WWW sẵn có [JDF02]. Đối với mỗi miền ứng dụng, có thể xây dựng một Ontology tương ứng. Sau đó bằng cách

ứng dụng các giải thuật trích chọn thông tin từ các tài liệu sẵn có để trích ra các lớp, các đối tượng, hay các thực thể trong nội dung các trang Web tương ứng với Ontology để tạo ra một cơ sở tri thức gồm các dữ liệu đã được gắn nhãn. Vì các phương pháp trích chọn thông tin có nhược điểm là phụ thuộc vào miền ứng dụng và dữ liệu, nên có thể phải sử dụng các giải thuật trích chọn khác nhau cho dữ liệu bán cấu trúc và không có cấu trúc. Dựa trên cơ sở tri thức vừa thu được, có thể suy diễn để tạo ra các tri thức mới, đáp ứng các truy vấn tri thức không sẵn có thông qua ngôn ngữ truy vấn: RDF Query Language (RQL) hay SPARQL. Trên cùng là lớp ứng dụng phục vụ người dùng để khai thác các dữ liệu đã trích chọn được như: hiển thị, duyệt, tìm kiếm, chia sẻ,... Trong phần tiếp theo chúng ta sẽ tìm hiểu một phương pháp tiếp cận theo hướng thứ hai này.

10.4.2. Nền tảng quản lý thông tin và tri thức KIM

Nền tảng quản lý thông tin và tri thức: The Knowledge and Information Management Platform⁽¹⁾ (KIM) [PKM03, KPM04], được phát triển bởi phòng thí nghiệm Công nghệ tri thức và ngôn ngữ thuộc Sirma (Knowledge and Language Engineering Lab of Sirma), là cách tiếp cận hướng tới khả năng tự động hóa trong việc xây dựng siêu dữ liệu cho các tài liệu dựa trên phương pháp *trích chọn thông tin*: Information Extraction (IE). Quá trình xây dựng siêu dữ liệu này được gọi là quá trình *chú giải ngữ nghĩa*⁽²⁾ (Semantic Annotation).

Với giả thiết rằng, các *thực thể tên*⁽³⁾ (Named Entity) được đề cập trong một tài liệu đóng một vai trò quan trọng về ngữ nghĩa của tài liệu đó. Do đó, quá trình xây dựng siêu dữ liệu trong KIM là quá trình gắn nhãn cho các thực thể tên trong tài liệu. Kiến trúc của KIM được minh họa trong Hình 10.6 có thể coi là một thể hiện của kiến trúc được đề cập trong Hình 10.5. KIM được chia làm 2 phần: máy chủ (KIM server) và các máy trạm dành cho các ứng dụng, chúng giao tiếp với KIM server thông qua các giao diện lập trình ứng dụng (Application Programming Interface – API).

– Phần quan trọng nhất trong kiến trúc hệ thống KIM là module trích chọn thông tin "Custom IE" (hay module chú giải ngữ nghĩa, sẽ được đề cập chi tiết ở phần sau), nhiệm vụ của nó là tự động phát hiện ra các thực thể, chú giải ngữ nghĩa cho nó và đưa vào trong cơ sở tri thức. Vì đây là một nền tảng mở nên KIM cho phép có thể tùy biến lại module này theo từng miền ứng dụng cụ thể. Module này được xây dựng dựa trên *kiến trúc công nghệ*

⁽¹⁾ <http://www.ontotext.com/KIM>

⁽²⁾ Đây là khái niệm được đề xuất bởi nhóm nghiên cứu và phát triển KIM.

⁽³⁾ Thực thể tên là các thực thể trong tài liệu như người, công ty, tổ chức, địa điểm và các đối tượng khác được gắn một tên riêng cụ thể.

– Cơ sở tri thức của KIM: nơi chứa Ontology và các siêu dữ liệu phục vụ cho quá trình suy diễn được lưu trữ sử dụng phần mềm Sesame⁽²⁾.

– Để hỗ trợ tính năng tìm kiếm trong KIM, máy tìm kiếm Lucene⁽³⁾ đã được tích hợp nhằm hỗ trợ tìm kiếm theo từ khóa.

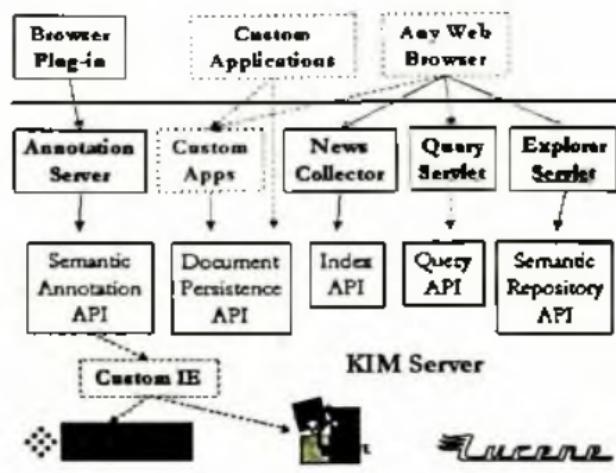
– Lớp trên cùng là các giao diện lập trình ứng dụng dành cho các máy trạm, nơi chứa các chương trình ứng dụng, khai thác sử dụng.

– Ngoài ra, KIM còn cung cấp thêm một số tính năng như lưu trữ tài liệu (Document Persistence), đánh chỉ mục tài liệu (Indexing) để hỗ trợ quá trình tìm kiếm (Search) hay truy vấn (Query) sau này.

Chú giải ngữ nghĩa:
Vì KIM tập trung vào

việc xây dựng ngữ nghĩa cho các tài liệu dựa vào các thực thể tên trong đó, nên module này có nhiệm vụ sinh ra các siêu dữ liệu (thông tin ngữ nghĩa) để mô tả các thực thể tên, các thuộc tính, cũng như các quan hệ giữa chúng trong các tài liệu đầu vào. Một ví dụ về quá trình chú giải ngữ nghĩa được minh họa trong Hình 10.7, với văn bản đầu vào ở phía trên, module này sẽ phát hiện ra các thực thể tên trong đó, phân loại, xác định các thuộc tính và quan hệ liên quan của chúng, sau đó ánh xạ vào trong cơ sở tri thức bên dưới. Chẳng hạn, "Bulgaria" được xác định và phân loại là một thực thể có kiểu quốc gia (country).

Quá trình chú giải ngữ nghĩa trong KIM cũng có thể coi là bài toán nhận dạng thực thể tên⁽⁴⁾ (Named Entity Recognition) truyền thống, ở đó các thực thể tên trong tài liệu sẽ được phát hiện và phân loại. Điểm khác biệt



Hình 10.6. Kiến trúc của KIM

⁽¹⁾ <http://gate.ac.uk>

⁽²⁾ <http://sourceforge.net/projects/sesame>

⁽³⁾ <http://lucene.apache.org>

⁽⁴⁾ Nhận dạng thực thể tên là ứng dụng phân tích văn bản đầu vào để tìm ra các thực thể tên xuất hiện trong đó

của nó so với các hệ nhận dạng thực thể tên là & chỗ; các hệ nhận dạng thực thể tên truyền thống chỉ nhận dạng một số lượng nhỏ các lớp thực thể chung (không phụ thuộc vào miền ứng dụng như *tên*, *chỗ*, *người*, *ngày tháng*, *địa điểm*, *con số*, *phản trả*, và *giá trị tiền tệ*).

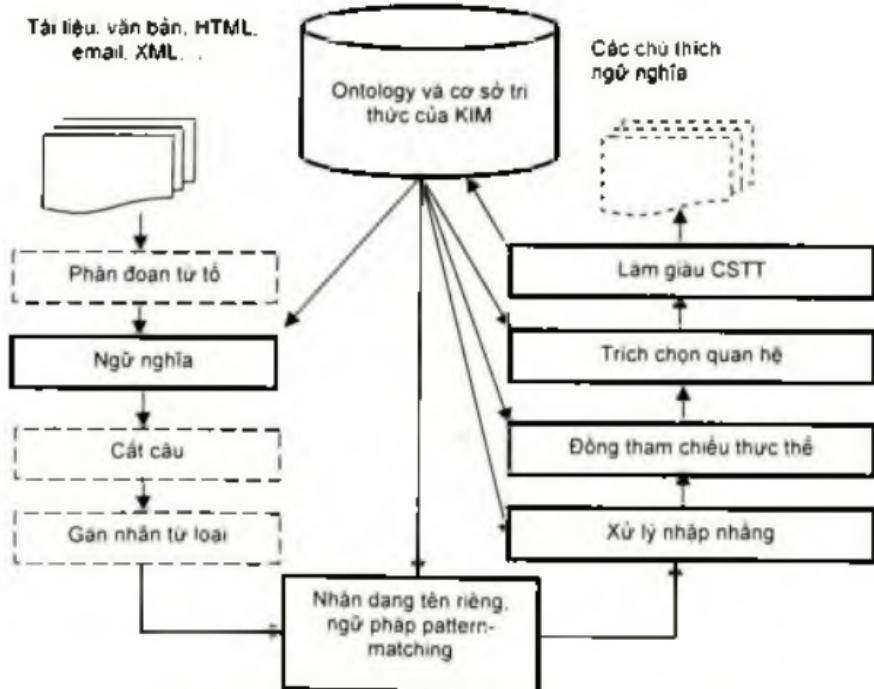
Trong thực tế, người dùng có thể yêu cầu (truy vấn) các loại thực thể tên mịn hơn (chẳng hạn *công ty viễn thông*), do đó KIM hỗ trợ gán nhãn một hệ thống phân cấp gồm nhiều loại thực thể mịn hơn. Ngoài ra KIM còn phát hiện thêm các thuộc tính cũng như các quan hệ của các thực thể tên này. Các bước trong quá trình chủ giải nghĩa ngữ nghĩa của KIM được dựa trên kiến trúc GATE và được minh họa trong hình 10.8.

... announced profits in Q3 planning to build a \$120M plant in Bulgaria, ... and more and more and more and more text ...

| KIM Ontology & KB | | |
|-------------------|-----------|-------------|
| Company | Location | |
| type | City | Country |
| XYZ | HQ London | type type |
| estab1on | partOF | UK Bulgaria |
| "03/11/1978" | | * |

Hình 10.7. Cơ sở tri thức của KIM

Tài liệu, văn bản, HTML, email, XML, ...



Hình 10.8. Các bước trong quá trình trích chọn thông tin

Điểm khác biệt so với GATE là trong KIM có sự tham gia Ontology và cơ sở tri thức (ở các bước có mũi tên đi ra từ cơ sở tri thức), do đó bên cạnh các thực thể tên thì các thông tin ngữ nghĩa của chúng cũng được phát hiện. Module này có thể xử lý được nhiều loại đầu vào như: văn bản, file HTML, XML, email, các trường văn bản trong cơ sở dữ liệu...

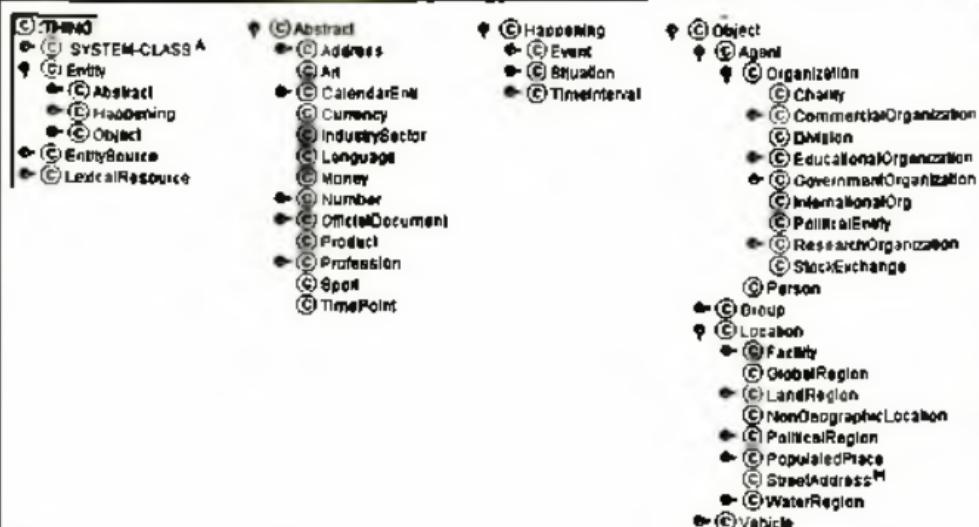
– *Ontology và cơ sở tri thức của KIM*: Đây là thành phần quan trọng thứ 2 của KIM, nó hỗ trợ quá trình chủ giải ngữ nghĩa cũng như quá trình khai thác dữ liệu ngữ nghĩa của các ứng dụng sau này, đồng thời tạo ra sự khác biệt giữa module trích chọn thông tin của KIM với các hệ nhận dạng thực thể tên thông thường. Cơ sở tri thức của KIM là một phép chiêu của thế giới Web tương ứng với một miền ứng dụng cụ thể. Hiện tại miền ứng dụng của KIM là các tài liệu bản tin quốc tế (International news), do đó nó phản ánh phần lớn các thực thể tên quan trọng và nổi tiếng trên thế giới. Ontology của KIM được tạo bởi khoảng 250 khái niệm (lớp thực thể tên) được chia thành 3 nhóm lớn: *Entity*, *EntitySource* và *LexicalResource* có ý nghĩa như sau:

– Nhóm quan trọng nhất là *Entity* lại được chia nhỏ thành các nhóm con là *Happening* (các sự kiện và tình huống), *Object* (là các thực thể có thực) và *Abstract* (các khái niệm còn lại không phải là *Object* hay *Happening*). Các nhóm con này lại tiếp tục được chia nhỏ thành phân lớp mịn hơn (như tổ chức chính phủ, tổ chức thương mại,...). Ngoài các khái niệm, Ontology của KIM còn có 100 thuộc tính (chẳng hạn như thuộc tính *Vĩ độ* (*Latitude*) của thực thể tên *địa điểm*, thuộc tính *hasPosition* của thực thể tên người) và quan hệ giữa các thực thể (chẳng hạn như, quan hệ *địa điểm* là *subRegionOf* của *địa điểm*, quan hệ *tổ chức locatedIn địa điểm*).

– Nhánh *LexicalResource* được tạo ra để chứa các dữ liệu hỗ trợ cho quá trình chủ giải ngữ nghĩa, chẳng hạn như tên tổ của các công ty (như AG, Ltd.), tên người (firstname),.... Một lớp con quan trọng của nhánh này là lớp *Bí danh* (*Alias*) – là một tên gọi khác của một thực thể tên. Quan hệ *hasAlias* được dùng để liên kết một thực thể tên tới các bí danh của nó, và tên chính thức của một thực thể được tham chiếu bởi thuộc tính *hasMainAlias*.

– Các thê hiện của lớp *EntitySource* được dùng để phân biệt thông tin tin cậy (được trích chọn từ nguồn thông tin đáng tin cậy) và các thông tin được trích chọn tự động trong cơ sở tri thức. Các thực thể trong cơ sở tri thức sẽ được chỉ định nguồn thông tin bởi thuộc tính *generatedBy*.

Hình 10.9 minh họa một phần của Ontology trong KIM, và Bang 10.3 thống kê thông tin về các thực thể tên ban đầu của KIM ở phiên bản 0305 được trích chọn từ một số nguồn thông tin tin cậy. KIM sử dụng ngôn ngữ RDF, RDFS để biểu diễn cơ sở tri thức và OWL để biểu diễn Ontology và được quản lý bởi phần mềm Sesame, nó hỗ trợ các thao tác quản lý, truy vấn, suy diễn trong cơ sở tri thức.



Hình 10.9. Một phần Ontology của KIM

Bảng 10.3. Thống kê các thực thể tên ban đầu của KIM

| Loại thực thể tên | Tổng số | Loại thực thể tên | Tổng số |
|-------------------|---------|----------------------|---------|
| Thực thể tên | 77,561 | Bí danh thực thể tên | 110,308 |
| Địa điểm | 49,348 | Thành phố | 4,720 |
| Công ty | 7,906 | Công ty công cộng | 5,150 |
| Người | 5,500 | Tổ chức | 8,365 |

– *Một số ứng dụng của KIM:* Với cơ sở tri thức khá lớn và module trích chọn thông tin có thể tự động xử lý được một khối lượng dữ liệu lớn, mô hình chủ giải nghĩa ngữ nghĩa trong KIM hy vọng sẽ hỗ trợ nhiều ứng dụng Web ngữ nghĩa mới dựa trên các thông tin ngữ nghĩa: như cơ chế đánh chỉ mục và tìm kiếm tài liệu/thực thể tên mới, phân loại tài liệu dựa vào ngữ nghĩa, hay sinh ra các dữ liệu ngữ nghĩa mới,... Nhóm nghiên cứu và phát triển KIM đã đề xuất 2 ứng dụng để so sánh sự khác biệt của Web ngữ nghĩa với Web thông thường, cũng như chứng minh khả năng của KIM. Ứng dụng thứ nhất của KIM là dùng để đánh dấu (highlighting) các thực thể tên trong trình duyệt khi xem một trang Web nào đó. Hình 10.10 là hình ảnh của một trang Web được bổ sung tính năng đánh dấu các thực thể tên. Tính năng này được viết dưới dạng một plugin của trình duyệt Web Internet Explorer.



Hình 10.10. Ứng dụng của KIM trong đánh dấu thực thể tên trên trình duyệt

Ở phía bên trái của trình duyệt, ta thấy liệt kê các loại thực thể tên được gán các màu khác nhau và nó cho phép người dùng có thể chọn. Ở phía bên phải của trình duyệt là nội dung của trang Web đang được duyệt, trong đó các loại thực thể tên được chọn đã được xác định và đánh dấu bằng các màu tương ứng. Khi nhấn chuột vào một thực thể tên nào đó, các thông tin chi tiết của nó sẽ được hiển thị ra.

- Ứng dụng thứ hai của KIM là máy tìm kiếm ngữ nghĩa, nó cho phép tìm kiếm các thực thể theo một số mẫu (pattern) ràng buộc như minh họa trên Hình 10.11. Chẳng hạn, khi tìm kiếm thực thể tên có kiểu là *công ty* (*Company*), thì có thể có ràng buộc: vị trí của công ty ở một *địa điểm* (*Location*) và công ty thuộc một ngành nào đó. Với hệ thống này có thể tìm kiếm các tài liệu chứa các thực thể tên, hay có thể là một danh sách các thực thể tên thỏa mãn câu truy vấn. Với ứng dụng này, có thể thấy sự khác biệt so với một máy tìm kiếm thông thường.

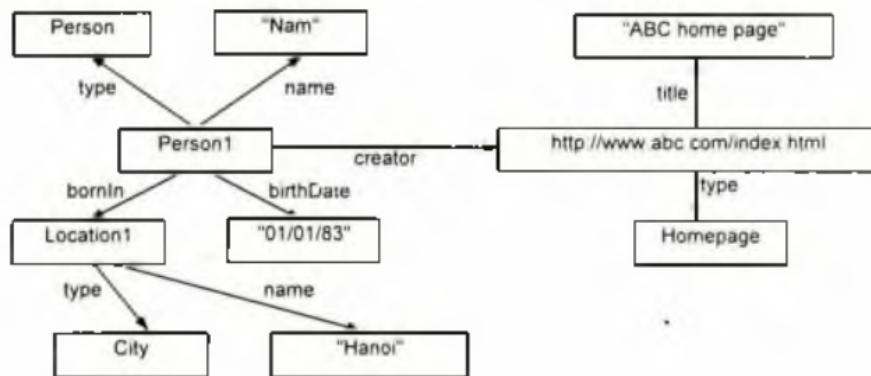
- Ngoài hai ứng dụng trên, KIM đã cung cấp các giao diện lập trình ứng dụng cho phép phát triển các ứng dụng tùy biến theo miền ứng dụng cụ thể nào đó.

Dựa vào KIM, một nhóm phát triển một hệ thống tương tự cho ngôn ngữ tiếng Việt mang tên Vietnam Knowledge and Information Management¹¹ (VN-KIM) [CDP05]. Cũng dựa trên cách tiếp cận khai thác

nguồn tài nguyên sẵn có để xây dựng Web ngữ nghĩa, Jennifer Chu-Carroll và cộng sự [CPC06] đã đề xuất một ứng dụng Web ngữ nghĩa bằng cách chuyển đổi dữ liệu văn bản sang định dạng XML. trong quá trình chuyên đổi thì cũng thêm các chủ giải ngữ nghĩa. Tiếp theo đó, một số toán tử thao tác với XML cũng được bổ sung thêm để hỗ trợ khả năng suy diễn trên dữ liệu XML [CPC06].

Câu hỏi và bài tập

1. Viết một tài liệu XML mô tả một danh sách điểm các môn học của một số lớp học.
2. Dùng ngôn ngữ DTD để đặc tả cấu trúc của file XML chứa nội dung của các danh sách điểm trong bài 1.



3. Dùng ngôn ngữ RDF để mô tả các đối tượng và thuộc tính của chúng được mô tả trong đồ thị trên.
4. Dùng ngôn ngữ RDF để mô tả các đối tượng cùng các thuộc tính của chúng có thể có trong bài tập 1.
5. Dùng ngôn ngữ RDFS để mô tả các lớp (và các thuộc tính) của các đối tượng xuất hiện trong bài 3.
6. Cài đặt môi trường phát triển GATE và thử nghiệm một số chức năng.
7. Cài đặt ứng dụng KIM và viết một số chương trình thử nghiệm sử dụng các chức năng sẵn có của nó thông qua các giao diện lập trình ứng dụng.

TÀI LIỆU THAM KHẢO

- [ACG00] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, Shram Raghavan (2000) Searching the Web Technical Report Computer Science Department Stanford University 2000
- [ACM03] L. Arlota, V. Crescenzi, G. Mecca, and P. Menaldo (2003) Automatic annotation of data extraction from large Web sites, *Proceedings of the International Workshop on the Web and Databases*, 7-12, San Diego USA, 2003
- [ADW94] Apte, C., Damerau, F. J., and Weiss, S. M. (1994). Automated learning of decision rules for text categorization. *ACM Trans on Inform Syst.* 12 (3) 233-251
- [AFD03] Andreu, C., Freitas, N., Doucel, A., and Jordán, M. (2003). An introduction to MCMC for machine learning. *Machine learning* 50, 5-43
- [AFM03] El-Sayed, Attam, M. Fukela, K. Morita, Jun-ichi Aoe (2003). Documents similarity measurement using field association terms. *Information Processing and Management* 39 (2003), 809-824
- [AG00] Eugene Agichtein, Luis Gravano (2000) Snowball: extracting relations from large plain-text collections. *ACM DL* 2000, 85-94
- [AGY01] Chauru C. Aggarwal, Falirna Al-Garawi, Philip S. Yu (2001). Intelligent Crawling on the World Wide Web with Arbitrary Predicates. *The 10 th International World Wide Web Conference*, Hong Kong, May 2001
- [AKM03] Alani, H., Kim, S., Millard, D. E., Weal, M. J., Lewis, P. H., Hall, W. and Shadbolt, N. R. (2003) Automatic Extraction of Knowledge from Web Documents. *2nd International Semantic Web Conference - Workshop on Human Language Technology for the Semantic Web and Web Services*. October 20-23, Sanibel Island, Florida USA
- [Alb05] Massimiliano Albanese (2005). Extracting and summarizing information from large data repositories, *Università Degli studi di Napoli Federico II Novembre 2005*
- [Ale99] Aleksander Ohm (1999) Discernibility and Rough Sets in Medicine Tools and Applications PhD Thesis, Norwegian University of Science and Technology, Trondheim Norway 1999
- [ALN05] K Avrachenkov N Litvak O Nemirovsky N Osipova (2005) Monte Carlo methods in PageRank computation When one iteration is sufficient. *INRIA Sophia Antipolis* 2005
- [AM03] Arvind Arasu, Hector Garcia-Molina (2003). Extracting Structured Data from Web Pages *SIGMOD Conference 2003* 337-348
- [AN06] Sophia Ananiadou, John McNaught (Editors. 2006) Text Mining for Biology and Biomedicine ARTECH HOUSE INC. 2006
- [Ana08] Sophia Ananiadou (2008) Selected Bibliography for Text Mining for Biomedicine. National Centre for Text Mining, University of Manchester.
http://www.nactem.ac.uk/nle_course/ISBM_BIBLIOGRAPHY_SA.pdf
- [ASG06] Markus Ackermann, Carlos Soares, Bettina Guidemann (2006) ECML/PKDD 2006 Workshop on Practical Data Mining: Applications, Experiences and Challenges, Berlin, September 22nd, 2006
- [BB98] Bagga, A. and Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. *Proceedings of the 17th international conference on Computational linguistics*. Association for Computational Linguistics, Morristown, NJ, USA, 79-85
- [BBM02] Holger Billehrt, Daniel Borrajo, Victor Mauro (2002) Context Vector Model for Information Retrieval. *Journal of American Society for Information Science and Technology (JASIS)* 53 (3) 2002, 236-249
- [BC98] L. D. Baker and A. K. McCallum (1998) Distributional clustering of words for text classification *Proc of SIGIR-98* 96-103, 1998
- [BCS04] Paolo Baldi, Bruno Codenotti, Massimo Santini, Sebastiano Vigna (2004). UbiCrawle: a scalable fully distributed web crawler. *Software-Practice & Experience* 711-725, 34 (8), July 2004
- [BCS05] Andras A. Benczur, Karoly Csalogany, Tamas Sarlos and Mate Uhe (2005). SpamRank - Fully Automatic Link Spam. *First International Workshop on Adversarial Information Retrieval on the Web*. WWW2005 2005
- [Ber00] Bellina Berendt (2000) Web usage mining: site semantics and the support of navigation. *Humboldt University Berlin Institute of Pedagogy and Informatics*, Berlin Germany, 2000
- [BFS03] Pierre Baldi, Paolo Frasconi, Padhraic Smyth (2003) Modeling the Internet and the Web Probabilistic Methods and Algorithms. Wiley 2003

- [BKV09] Timothy Baldwin, Valia Kordonis, Aline Villavicencio (2009) Prepositions in Applications: A Survey and Introduction to the Special Issue. *Computational Linguistics* 35(2) 119-149. 2009.
- [BL06] Blei, D and Lafferty, J. Dynamic topic models. *Proceedings of the 23rd International Conference on Machine Learning (ICML-2006)*
- [BL07] Blei, D and Lafferty, J. (2007) A correlated topic model of Science. *Annals of Applied Statistics* 1(1) 17-35 (2007).
- [BL08] Blei, D and Lafferty, J. (2008) Modeling Science.
<http://www.cs.princeton.edu/~blei/modeling-science.pdf>
- [BL09] Blei, D and Lafferty, J. (2009) Topic models. In *Text Mining: Theory and Applications* (A. Srivastava and M. Sahami, editors).
<http://www.cs.princeton.edu/~blei/papers/BleiLafferty2009.pdf>
- [BLP01] Buttler, D., Liu, L., and Pu, C. (2001) A fully automated extraction system for the world wide web. *Proceedings of IEEE ICDCS 2001*
- [BMD06] Helmut Berger, Dieter Merkl, Michael Dittenbach (2006). Exploiting Partial Decision Trees for Feature Subset Selection in e-Mail Categorization. *SAC'06*, April 23-27, 2006, Dijon, France
- [BMI07] Bollegala, D., Matsuo, Y., and Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. *Proceedings of WWW* 757-766. 2007
- [BMS97] Biket, D., Miller, S., Schwartz, R., and Weischedel, R. Nymble (1997). A high-performance learning name-finder. *Proceedings of ANLP 1997*. 194-201
- [BNJ03] Blei, D., Ng, A., and Jordan, M. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 993-1022 (2003)
- [BP03] Banerjee, S. and Pedersen, T. (2003). The design, implementation and use of the ngram statistics. *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*: 370-381, 2003.
- [BP98] Sergey Brin and Lawrence Page (1998). The Anatomy of a Large-Scale Hypertextual Web search Engine. Technical report, Stanford University, 1998
- [BPP96] Berger, A., Pietra, A. D., and Pietra, J. D. (1996) Maximum entropy approach to natural language processing. *Computational Linguistics* 22(1) 39-71. 1996.
- [BRG07] Banerjee, S., Ramanathan, K., and Gupta, A. (2007) Clustering short texts using wikipedia. *Proceedings of ACM SIGIR*, 787-788, 2007
- [Bri98] Sergey Brin (1998). Extracting Patterns and Relations from the World Wide Web. *WebDB Workshop at EDBT '98*, 1998. <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/ltheo-21/www/handouts/SergeiBrinBootstrapping1999.pdf>
- [Bro95] Eric W. Brown (1995). Execution Performance Issues in Full-Text Information Retrieval. *PhD Thesis*, University of Massachusetts at Amherst, 1995
- [BRR04] Allan Borodovsky, Gareth O. Roberts, Jeffrey S. Rosenthal, Panayiotis Tsaparas (2004) *Link Analysis Ranking Algorithms, Theory, and Experiments*. *ACM Trans. Inter. Tech.* 5 (1) 231-297, February 2005
- [Cal04] Janko Calic (2004) Highly Efficient Low-Level Feature Extraction For Video Representation and Retrieval. *PhD Thesis*, University of London, 2004.
- [CBD99] Soumen Chakrabarti, Martin van den Berg, Byron Dom (1999). Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. *Computer Networks*, 31(11-16) 1623-1640 (1999)
- [CCR05] Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. Identifying sources of opinions with conditional random fields and extraction patterns. *Proceedings of HLT/EMNLP*. 2005.
- [CCZ06] Olivier Chapelle, Mingmin Chi, Alexander Zien (2006) A Continuation Method for Semi-Supervised SVMs. *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.
- [CD01] Chen, H. and Dumais, S. (2001). Bringing order to the web: Automatically categorizing search results. *Proceedings of CHI'01, Human Factors in Computing Systems* 145-152
- [CDP05] Cao, T. H. & Do, H. T. & Pham, B. T. N. & Huynh, T. N. & Vu, D. Q. (2005). Conceptual graphs for knowledge querying in VN-KIM. *Contributions to the 13th International Conference on Conceptual Structures* 27-40. Kassel, Germany, Kassel University Press, 2005
- [CFP00] McCallum, A., Freitag, D., and Pereira, F. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of ICML-2000*
- [CH03] Cai, L. and Hofmann, T. (2003). Text categorization by boosting automatically extracted concepts. *Proceedings of ACM SIGIR*
- [Cha03] Soumen Chakrabarti (2003). *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003
- [Chi06] Mingmin Chi (2006). Advanced Semi-Supervised Techniques for the Classification of Remote Sensing Data. *PhD Thesis*, DIT, University of Trento, Mar. 2006

- [CJG06] Rich Caruana, Thorsten Joachims, Johannes Gehrke, Benyah Shaparenko (2006) Patterns and Key Players in Document Collections, KDD Challenge 2005
http://velblod.videolectures.net/ijc/solomon/2006/caruana_rich/solomon_caruana_wslmw_01.ppt
- [CKP92] Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tokey, J. W. (1992) Scatter/gather: A cluster-based approach to browsing large document collections. *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*: 318-329.
- [CKP93] Cutting, D. R., D. R. Karger, and J. O. Pedersen (1993) Constant interaction-limescatter/gather browsing of very large document collections. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '93)*, New York, NY, USA, 126-134. ACM Press, 1993.
- [CL01] Chang, C. H. and Lui, S. L. (2001) Information extraction based on pattern discovery. *Proceedings of the World Wide Web Conference (WWW)*, 2001.
- [CL03] McCallum, A. and Li, W. (2003) Early results for named entity recognition with conditional random fields: feature induction and Web-enhanced lexicons. *Proceedings of the 7th Conference on Natural Language Learning (CoNLL-2003)*.
- [CL96] Cowie, J. R. and Lehnert, W. G. (1996) Information extraction. *Communication of the ACM*, 39(1): 80-91, 1996.
- [CM03] William W. Cohen and Andrew McCallum (2003) Information Extraction from the World Wide Web. *KDD 2003*.
- [CM98] Chinchor, N. and Marsh, E. Information extraction task definition (version 5.1). *Proceedings of the 7th Message Understanding Conference (MUC-1998)*.
- [CMB02] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan (2002) GATE: an Architecture for Development of Robust HLT Applications. *Recent Advances in Language Processing*, 2002.
- [Coo00] Robert Walker Cooley (2000). *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. PhD Thesis, University of Minnesota, 2000.
- [CPC06] Jennifer Chu-Carroll, John Prager, Krzysztof Czuba, David Ferrucci, and Pablo Duboue (2006). Semantic Search via XML Fragments: A High-Precision Approach to IR. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*: 445-452, 2006.
- [CPS02] Soumen Chakrabarti, Kunal Punera, Mallela Subramanyam (2002) Accelerated focused crawling through online relevance feedback. *WWW 2002*: 148-159.
- [CRM98] A. McCallum, R. Rosenfeld, T.M. Mitchell, and A.Y. Ng. (1998) Improving text classification by shrinkage in a hierarchy of classes. *Proc. of ICML-98 Conference on Machine Learning*: 359-367, 1998.
- [CS07] Chang, Y.K. & Spink, A. (2007) Multimedia Chinese Web Search Engines: A Survey. *Proceedings of the International Conference on Information Technology*: 481-486.
<http://eprints.qut.edu.au/14311>
- [CSB03] Guihong Cao, Dawei Song, Peter Brzuska (2003) Suffix Tree Clustering on Post-retrieval Documents. *The University of Queensland, QLD 4072, Australia*, 28 July, 2003.
- [CV95] Corinna Cortes, Vladimir Vapnik (1995) Support-Vector Networks. *Machine Learning*, 20(3): 273-297.
- [CYC07] Tao Cheng, Xifeng Yan, Kevin Chen-Chuan Chang (2007) EntityRank: Searching Entities Directly and Holistically. *VLDB 2007*: 387-398.
- [CYC07a] Tao Cheng, Xifeng Yan, Kevin Chen-Chuan Chang (2007) Supporting entity search: a large-scale prototype search engine. *SIGMOD Conference 2007*: 1144-1146.
- [DC00] S. Dumais and H. Chen (2000) Hierarchical classification of web content. *Proc. Of SIGIR*, 00: 256-263, 2000.
- [DCL00] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gorri (2000) Focused crawling using context graphs. *Proc. 26th International Conference on Very Large Databases (VLDB 2000)*: 527-534, Cairo, Egypt, 2000.
- [DFD90] S. Deerwester, G.W. Furnas, S. Dumais, and T.K. Landauer (1990) Indexing by latent semantic indexing. *Journal of the American Society for Information Science and Technology (JASIST)*, 41(6): 391-407, 1990.
- [DFL90] Deerwester, S., Furnas, G., and Landauer, T. (1990) Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6): 319-407, 1990.
- [DFL90] Deerwester, S., Furnas, G., and Landauer, T. (1990) Indexing by latent semantic analysis. *Journal of the American Society for Info. Science* 41: 391-407.

- [DGM04] Michelangelo Diligenti, Marco Gori, and Marco Maggini (2004). A Unified Probabilistic Framework for Web Page Scoring Systems. *IEEE Transactions on Knowledge and Data Engineering*, 16 (1), January 2004. 4-16
- [DGN09] Gianluca Demartini, Julien Gaugaz, Wolfgang Nejdl (2009). A Vector Space Model for Ranking Entities and Its Application to Expert Search. ECIR 2009. 189-201
- [DH99] Dean and M Henzinger (1999). Finding Related Pages in the World Wide Web. *Proceedings of WWW8*. 1999
- [DPH98] S Dumais, J Platt, D Heckerman and M Sahami (1998). Inductive learning algorithms and representations for text categorization. *Proceeding of the 1998 ACM 7th International Conference on Information and Knowledge Management*. 148-156. 1998
- [ECJ99] Embley, D W., Campbell, D M., Jiang, Y S., Liddle, S W., Lonsdale, D W., Ng, Y K., and Smith, R. D. (1999). Conceptual-model-based data extraction from multiple-record web pages. *Data and Knowledge Engineering* 31(3) 227-251. 1999
- [Eib07] Sven Meyer zu Eiben (2007). On Information Need and Categorizing Search. PhD Thesis University of Paderborn, Germany
- [EKS02] Martin Ester, Hans-Peter Kriegel, Matthias Schubert (2002). Web Site Mining. A new way to spot Competitors, Customers and Suppliers in the World Wide Web. *Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. July 23-26, 2002. Alberta, Canada. 249-258
- [EM03] L Egghe, C Michel (2003). Construction of weak and strong similarity measures for ordered sets of documents using fuzzy set techniques. *Information Processing and Management* 39 (2003). 771-807
- [EPE05] Secrets of success. From *The Economist* print edition. Sep 8th 2005
http://www.usnews.com/usnews/edu/college/rankings/brief/natudoc/tier1/t1natudoc_brief.php
- [ER61] P Erdos and A Renyi (1961). On the evolution of random graphs. *Théorie de l'Information* 343-347. 1961. http://www.renyi.hu/~p_erdos/1961-15.pdf
- [FC99] Freitag, D. and McCallum, A. (1999). Information extraction using HMMs and shrinkage. *Proceedings AAAI-1999 Workshop on Machine Learning for Information Extraction*. 1999
- [FF02] Brian Fox, Christopher J. Fox (2002). Efficient stemmer generation. *Information Processing and Management* 38 (2002). 547-558
- [FG05] Paolo Ferragina, Antonio Gulli (2005). A Personalized Search Engine Based on WebSnippet Hierarchical Clustering. *WWW2005*, May 10-14, 2005, Chiba, Japan
- [FHK91] N Fuhri, S Hartmann, G Knorz, G Lustig, M Schwanter, and K Tzeras (1991). Airix - a rule-based multistage indexing system for large subject fields. *Proc. of RIAO-91*. 606-623. 1991
- [FK99] Freitag, D and Kushmerick, D. (1999). Boosted wrapper induction. *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-1999)*
- [FL03] Federico Michele Facca, Pier Luca Lanzi (2003). Recent Developments in Web Usage Mining Research. *DaWaK 2003*. 140-150
- [FPS96] Fayyad, Piatetsky-Shapiro, Smyth (1996). From Data Mining to Knowledge Discovery. An Overview. In Fayyad, Piatetsky-Shapiro, Smyth, Uthurusamy, *Advances in Knowledge Discovery and Data Mining*. AAAI Press/ The MIT Press, Menlo Park, CA. 1996. 1-34
- [Fre00] Freitag, D. (2000). Machine learning for information extraction in informal domains. *Machine Learning* 39(2/3). 99-101 (2000)
- [Fre98] Freitag, D. (1998). Information extraction from HTML - application of a general machine learning approach. *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-1998)*
- [Fre99] Freitag, D. (1999). Machine learning for information extraction in informal domains. Ph.D. Thesis. Carnegie Mellon University (1999)
- [Gar05] Ken McGarry (2005). A Survey of Interestingness Measures for Knowledge Discovery. *The Knowledge Engineering Review*. Vol. 20(1). 39-61. 2005. Cambridge University Press
- [GG05] Z Gyongyi and H Garcia-Molina (2005). Web Spam Taxonomy. *Proc. of the Fourteenth International World Wide Web Conference*, Chiba, Japan, 2005
- [GG05a] Zoltan Gyongyi and Hector Garcia-Molina (2005). Link spam alliances. *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, Trondheim, Norway. 2005
- [GGP04] Z Gyongyi, H Garcia-Molina, and J Pendersen (2004). Combating Web Spam with TrustRank. *Proceedings of the 30th International VLDB Conference*. 2004
- [Gir07] R Girju (2007). Improving the Interpretation of Noun Phrases with Cross-linguistic Information. In the proceedings of the Association for Computational Linguistics Conference (ACL 2007). Prague, June 2007.

- [Gir08] Roxana Giju (2008) Semantic Relation Extraction and its Applications European Summer School in Logic, Language and Information (ESSLLI 2008) (Invited Tutorial). Hamburg, Germany. August 2008
- [Gir09] R. Giju (2009) The Syntax and Semantics of Prepositions in the Task of Automatic Interpretation of Nominal Phrases and Compounds a Cross-linguistic Study. In Computational Linguistics - Special Issue on Prepositions in Applications. 35 (2): 185-229. 2009
- [GM07] Garilovich, E. and Markovitch, S. (2007) Computing semantic relatedness using wikipedia-based explicit semantic analysis, Proceedings of IJCAI
- [GPM06] Geraci, F., Pellegrini, M., Maggini, M. and Sebastiani, F. (2006) Cluster generation and cluster labelling for web snippets A fast and accurate hierarchical solution. LNCS String Processing and Information Retrieval 25-36
- [Gra08] Robert M. Gray (2008) Entropy and Information Theory. Springer Verlag. 2008
- [GS04] Griffiths, T. and Steyvers, M. (2004) Finding scientific topics. The National Academy of Science. 1(101) 5226-5235. 2004
- [GS05] A. Gulli, A. Signorini (2005). Building an Open Source MetaSearch Engine. WWW 2005, May 10-14 2005 Chiba, Japan
- [GZ00] Sally Goldman và Yan Zhou (2000) Enhance Supervised Learning with Unlabeled Data. Proceedings of ICML. 327-334. 2000
- [HA09] P. S. Hiramath, Siddu P. Algur (2009) Extraction of Data from Web Pages A Vision Based Approach. International Journal of Computer and Information Science and Engineering. 3 (1): 50-59. 2009
- [HAN90] P.J. Hayes, P.M. Andersen, I.B. Nirenburg, and L.M. Schmandt (1990) TCS: a shell for content-based text categorization. Proc. of 8th IEEE Conference on Artificial Intelligence Applications. 320-326. 1990
- [Har99] Elliott Rusty Harold (1999) XML Bible. IDG Books Worldwide, Inc. 1999
- [Hav02] Taher H. Haveliwala Y (2002) Topic-Sensitive PageRank. Stanford University, Computer Science Department, Stanford, CA 94305, 2002.
- [Hav03] Taher H. Haveliwala (2003). Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. Stanford University, Computer Science Department, Stanford, CA 94305, 2003
- [Hav99] Taher H. Haveliwala (1999) Efficient Computation of PageRank. Technical report, Stanford University.
- [HCZ08] Hu, J., Fang, L., Cao, Y., Zeng, H.-J., Li, H., and Cheng, Q. Y. Z (2008) Enhancing text clustering by leveraging wikipedia semantics. SIGIR '08 Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. 179-186. ACM, New York, NY, USA.
- [HD98] Hsu, C. N. and Dung, M. T. (1998). Generating finite-state transducers for semi-structured data extraction from the Web. Information Systems. 23(8): 521-538, 1998
- [Hef01] Heflin, J. (2001). Towards the Semantic Web. Knowledge Representation in a Dynamic, Distributed Environment, PhD Thesis, University of Maryland, Computer Sciences Dept. 2001.
- [Hei05] Heinrich, G. (2005) Parameter estimation for text analysis. Technical Report. 2005
- [HGI00] T H Haveliwala, A Gionis, and P Indyk (2000) Scalable Techniques for Clustering the Web. Informal Proceedings of the International Workshop on the Web and Databases. WebDB. 2000.
- [HGK02] Taher H. Haveliwala, Aristides Gionis, Dan Klein, Piotr Indyk (2002) Evaluating Strategies for Similarity Search on the Web. WWW2002. USA
- [Hie06] Hieu X P (2006) A Study of Discriminative and Exponential Models for Labeling and Segmenting Text and Web Data. PhD Thesis, Japan Advanced Institute of Science and Technology. 2006
- [HJM98] Michael Hersovici, Michal Jacovi, Yoav S. Maarek, Dan Peleg, Menachem Shlaiman, Sigalit Ur (1998) The shark-search algorithm: An application tailored Web site mapping. Proceedings of the seventh international conference on World Wide Web. 317-326. <http://www.cs.cmu.edu/~dpelleg/bin/360.html>
- [HK0106] J. Han and M. Kamber (2006) Data Mining: Concepts and Techniques. Morgan Kaufmann. 2006 (tai bản năm 2006)
- [HL06] Hu M and Liu, B (2006) Opinion extraction and summarization on the Web. Proceedings of the 21st National Conference on Artificial Intelligence (AAAI). 2006
- [HM99] Allan Heydon, Marc Najork (1999) Mercator: A Scalable Extensible Web Crawler. World Wide Web 2(4): 219-229. 1999

- [HMS02] Monika Henzinger, Rajeev Motwani, and Craig Silverstein (2002) Challenges in web search engines. *SIGIR Forum*, 36(2):1573-1579, 2002
- [Hof99] Hoffmann, T. (1999). Probabilistic latent semantic analysis. *Proceedings of Uncertainty in Artificial Intelligence*. 1999
- [Hof99a] Hofmann, T. (1999) Probabilistic LSA, *UAI 1999* 289-298
- [Hop07] John E Hopcroft (2007) Future Directions in Computer Science.
<http://www.cs.cornell.edu/jeh/China%202007.ppt>
- [HP03] Enrique Herrera-Viedma, Eduardo Peis (2003) Evaluating the informative quality of documents in SGML format from judgements by means of fuzzy linguistic techniques based on computing with words. *Int. Process. Manage.* 39(2):233-249
- [HRG00] J Hirai, S Raghavan, H Garcia-Molina, and A Paepcke (2000). WebBase: A Repository of Web Pages. *Proceedings of WWW9*, 2000
- [HSS03] Andreas Holho, Steffen Staab, Gerd Stumme (2003) Text Clustering Based on Background Knowledge. *Technical Report No. 425*, April 2003, University of Karlsruhe. D-76128 Karlsruhe, Germany
- [HWL05] Xian-Sheng Hua, Zengzhi Wang, Shipeng Li (2005) LazyCut - Content-Aware Template-Based Video Authoring. *MM'05*. November 6-11, 2005. Singapore
- [HZ03] Jingyu Hou and Yanchun Zhang (2003) Effectively Finding Relevant Web Pages from Linkage Information, *IEEE Transactions on Knowledge and Data Engineering*, 15 (4) July/August 2003
- [Inm02] W H Inmon (2002) Building the Data Warehouse (Third Edition). Wiley Computer Publishing, 2002
- [IV06] Renata Ivancsy, Istvan Vajk (2006). Frequent Pattern Mining in Web Log Data. *Acta Polytechnica Hungarica*, 3(1):77-90, 2006
- [JDF02] John Davies, Dieter Fensel, Frank van Harmelen (2002). Towards the Semantic Web Ontology-driven Knowledge Management, John Wiley & Sons, Ltd. 2002
- [J002] Jay Sethuraman and Leyla Ozsen (2002) Optimal Crawling Strategies for Web Search Engine Columbia University. May 8, 2002
- [Joe02] Thorsten Joachims (2002) Optimizing Search Engines using Clickthrough Data. Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. July 23-26, 2002. Alberta, Canada. 133-142
- [Joa98] T Joachims (1998) Text categorization with support vector machines. Learning with many relevant features. *Proceedings 10th European Conference on Machine Learning(ECML)* 137-142, 1998
- [Joa99] Joachims (1999). Transductive Inference for Text Classification using Support Vector Machines. *International Conference on Machine Learning (ICML)*, 1999.
- [JSB98] Jansen, B. J., Spink, A., Bateman, J., and Saracevic, T. 1998 Real life information retrieval a study of user queries on the web. *SIGIR Forum* 32, 1 5-17
- [KB00] Raymond Kosala, and Hendrik Blockeel (2000) Web Mining Research Survey. *ACM SIGKDD*, 2(1) 1-15. July, 2000.
- [KC05] S Sathiya Keerthi, Dennis DeCoste (2005) A Modified Finite Newton Method for Fast Solution of Large Scale Linear SVMs. *Journal of Machine Learning Research* 6 341-361
- [Kee77] E M Keen (1977) On the generation and searching of entries in printed subject indexes. *Journal of Documentation*, 33(1):15-45, 1977
- [KH00] George Karypis and Eui-Hong (Sam) Han (2000). Concept Indexing A Fast Dimensionality Reduction Algorithm with Applications to Document Retrieval & Categorization, Technical Report, University of Minnesota, Department of Computer Science / Army HPC Research Center
- [KHG03] Sepandar Kamvar, Taher Haveliwala, and Gene Golub (2003) Adaptive Methods for the Computation of PageRank Technical report, Stanford University
- [KHM03] S D Kamvar, T H Haveliwala, C D Manning, and G H Golub (2003) Extrapolation methods for accelerating PageRank computations. *Proceedings of the Twelfth International World Wide Web Conference*, 2003.
- [KHM03b] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, Gene H. Golub (2003) Exploiting the Block Structure of the Web for Computing PageRank. Technical report, Stanford University.
- [Kis95] Kiselev, M. V. (1995). PolyAnalyst 2.0 Combination of Statistical Data. Preprocessing and Symbolic KDD Technique. *Proceedings of ECML-95 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, Heraklion, Greece. 187-192
- [KJ97] R Kohavi and G John (1997) Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2) 273-324 1997

- [Kle99] Jon Kleinberg (1999) Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5) 604-632. November 1999.
- [KN03] Dhillion, J. Kogan, and C. Nicholas (2003). Feature selection and document clustering. M W Berry, editor, *A comprehensive survey of text mining*, 191-200. Springer-Verlag, 2003.
- [KP04] Kotsiantis, S. and Pintelas, P. E (2004) Recent advances in clustering. A brief survey. *WSEAS Transactions on Information Science and Applications* 1 1 73-81.
- [KPM04] Atanas Kiryakov, Borislav Popov, Dimitar Manov, Damyan Ognyanoff, Rosen Marinov, Ivan Terziev (2004) Automatic Semantic Annotation with KIM. *The 3rd International Semantic Web Conference (ISWC)*, 2004.
- [KS04] Tamer Kahveci and Ambuj K. Singh (2004) Optimizing Similarity Search for Arbitrary Length Time Series Queries. *IEEE Transaction on Knowledge and Data Engineering*, 16 (4), April 2004. 418-433.
- [KSS00] R. Kohavi, M. Spiliopoulou, J. Srivastava (editors, 2000) *Web Mining for E-Commerce: Challenges and Opportunities*. KDD-2000 workshop proceedings. WEBKDD'2000 August 2000, Boston, MA.
- [Kus00] Kushmerick, N. (2000). Wrapper induction, efficiency and effectiveness. *Artificial Intelligence* 118(1-2) 15-68. 2000.
- [KV01] Boris Kovalerchuk and Evgenii Vilyaev (2001). *Data Mining in Finance Advances in Relational and Hybrid Methods*. Kluwer Academic Publishers. Boston, Dordrecht - London. 2001.
- [KCV04] Kristjansson, T., Culotta, A., Viola, P., and McCallum (2004) A Interactive information extraction with constrained conditional random fields. *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-2004)*.
- [LAT06] Lan N. Bui, Anh Q. Tran, Thuy Q. Ha (2006) User authentic Rating based on Email Networks. In *First International Conference on Mobile Computing, Communications and Applications*. 16-17 August, 2006. ICMCCA 2006. Seoul, Korea. 144-148.
- [LBK09] Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. *Proceedings of ACM SIGKDD 2009*.
- [LC03] Li, W. and McCallum (2003). A Rapid development of Hindi named entity recognition using conditional random fields and feature induction. *ACM Transactions on Asian Language Information Processing (ACM TALIP)*. 2003
- [LC06] N C Liu, and Y. Cheng (2006). Academic Ranking of World Universities - Methodologies and Problems. *Institute of Higher Education, Shanghai Jiao Tong University* (<http://ed.sjtu.edu.cn/frank/file/ARWU-M&P.pdf>)
- [LCH02] Shian - Hua Lin, Meng Chang Chen, Jan-Ming Ho (2002). ACIRD: Intelligent Internet Document Organization and Retrieval. *IEEE transaction on knowledge and data engineering*, 14(3), May/June 2002
- [LCP01] Lafferty, J., McCallum, A., and Pereira, F. (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proceedings of ICML-2001*
- [Lee06] Wookey Lee (2006) Hierarchical Web Structure Mining. *Proc of Data Engineering Workshop (DEWS 2006)*. Japan. Feb 28-Mar 2. 2006.
- [Lew92] David Donald Lewis (1992) Representation and Learning in Information Retrieval. *PhD Thesis*. Univ Massachusetts at Amherst, 1992
- [Lew92b] D.D. Lewis (1992). An evaluation of phrasal and clustered representation on a text categorization task. *Proc of 19th ACM SIGIR*. 289-297. 1992
- [LGZ03] Bing Liu, Robert L Grossman, Yanhong Zhai (2003). Mining data records in Web pages. *KDD 2003* 601-606
- [LH08] Leskovec, J. and Horvitz, E (2008) Planetary-scale views on a large instant-messaging network. *Proceedings of International World Wide Web Conference (WWW-2008)*
- [Li07] Jiye Li (2007) Rough Set Based Rule Evaluations and Their Applications. *PhD Thesis*. University of Waterloo, Ontario, Canada. 2007
- [Liu08] Liu B. (2008) Opinion mining and summarization, *Tutorial given at the World Wide Web Conference (WWW)*. 2008
- [Liu09] Ying Liu (2009) On Document Representation and Term Weights in Text Classification. *MY09* 1-22
- [LM98] H. Liu and H. Motoda (1998) *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic. 1998
- [LNH08] Le, Dieu Thu Nguyen, Cam Tu, Ha Quang Thuy, Phan Xuan Hieu and Honguchi, Susumu (2008) Matching and ranking with hidden topics towards online contextual advertising. *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence*. 888-891. 2008

- [LNS02] Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran Soares da Silva (2002). DEByE - Data Extraction By Example. *Data Knowl. Eng.* 40(2) 121-154
- [LR94] D. Lewis and M. Ringuelet (1994) A comparison of two learning algorithms for text categorization. *3rd Annual Symposium on Document Analysis and Information Retrieval* 81-93. 1994
- [LS03] Xiaohui Long and Torsten Suel (2003) Optimized Query Execution in Large Search Engines with Global Page Ordering. *Proceeding of the 29th VLDB Conference*. Berlin Germany, 2003
- [Luh58] H.P. Luhn (1958) The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2) 159-165. 1958
- [MCL05] Gilad Mishne, David Carmel, Ronny Lempel (2005) Blocking Blog Spam with Language Model Disagreement, *AIRWeb '05 - First International Workshop on Adversarial Information Retrieval on the Web, at the 14th International World Wide Web Conference (WWW2005)* 2005
- [MD05] Panagiotis T. Metaxas, Joseph DeStefano (2005). Web Spam, Propaganda and Trust. *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*
- [MDL01] Bamshad Mobasher, Honghua Dai, Tao Luo, Miki Nakagawa (2001). Effective personalization based on association rule discovery from web usage data. *WIDM 2001* 9-15
- [Meh03] Mehmed Kavardzic (2003) Data Mining: Concepts, Models, Methods, and Algorithms. John Wiley & Sons. 2003
- [Mit97] T.M. Mitchell (1997) Machine Learning. McGraw Hill. 1997
- [Mla98] Dunja Mladenic' (1998). Machine Learning on Non-homogeneous, Distributed Text Data. *PhD Thesis*. University of Ljubljana, Slovenia
- [Mla98a] D. Mladenic (1998). Feature subset selection in text learning. *Proc of European Conference on Machine Learning(ECML)* 95-100. 1998
- [MMK01] Muslea, I., Minton, S., and Knoblock, C. A. (2001) Hierarchical wrapper induction for semi-structured information sources. *Autonomous Agents and Multi-Agent* 4(1-2):93-114. 2001
- [MMM02] Giansalvatore Mecca, Alberto O. Mendelzon, and Paolo Menaldo (2002). Efficient Queries over Web Views. *IEEE Transactions on Knowledge and Data Engineering*, 14 (6). November/December 2002
- [MNR00] Andrew McCallum, Kamal Nigam, Jason Rennie, Kristie Seymore (2000) Building Domain-Specific Search Engine with Machine Learning Techniques. *AAAI Spring Symposium on Intelligent Agents in Cyberspace*, Stanford University, USA, 1999.
- [Mou96] I. Moulinier (1996) A framework for comparing text categorization approaches. *AAAI Symposium on Machine Learning and Information Access*. Stanford University. 1996
- [MRS09] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schulze (2009) An Introduction to Information Retrieval. Online edition (c)2009 Cambridge UP (Draft of April 1. 2009). <http://nlp.stanford.edu/IR-book/pdf/rbookprint.pdf>
- [MS99] C.D. Manning and H. Schütze (1999) Foundations of Statistical Natural Language Processing. The MIT Press. 1999
- [MS99a] Aleix Martinez and Joan R. Serra (1999) Semantic Access to a Database of Images: An approach to object-related image retrieval. *Proceedings of IEEE Multimedia Computing and Systems (ICMCS'99)*, 1999
- [MS99b] B. Masand B. and M. Spiliopoulou (editors, 1999). Web Usage Analysis and User Profiling. *WebKDD-99 Workshop Proceedings*. August. 1999
- [MSZ07] Mei, Q., Shen, X., and Zhai, C. (2007) Automatic labeling of multinomial topic models. *Proceeding of KDD'07* 490-499
- [Mue95] Andreas Mueller (1995) Fast Sequential and Parallel Algorithms for Association Rule Mining. A Comparison Report CS-TR-3515. Dept. of Computer Science, Univ. of Maryland. College Park, MD. August 1995.
- [Mus02] Ion Alexandru Muslea (2002) Active Learning with multiple views. *PhD Thesis*. University of Southern California. 2002
- [MY09] Min Song and Yi-fang Brook Wu (2009) Handbook of Research on Text and Web Mining Technologies. *Information Sciences Reference*. 2009.
- [Ngo03] Ngo, Chi Lang (2003) A tolerance rough set approach to clustering web search results. *M.S Thesis*. Warsaw University
- [NMS08] Nguyen Tri Thanh, Nguyen Le Minh và Akira Shimazu (2008) Using Semi-supervised Learning for Question Classification. *Journal of Natural Language Processing* 3(1) 112-130. 2008.
- [NNP06] Nguyen, C.-T., Nguyen, T.-K., Phan, X. H., Nguyen, L. M., and Ha, Q. T. (2006) Vietnamese word segmentation with crfs and svms: An investigation. *Proceedings of the*

- [NNT05] Đỗ Thị Diệu Ngọc, Nguyễn Hoài Nam, Nguyễn Thu Trang, Nguyễn Yen Ngọc (2005) Giải pháp tinh chỉnh trang Modified Adaptive PageRank trong máy tìm kiếm. Chuyên san "Các công trình nghiên cứu về CNTT và Truyền thông". Tạp chí Bưu chính Viễn thông. 14:65-71. 4-2005.
- [NPH09] Cam-Tu Nguyen, Xuan-Hieu Phan, Susumu Honguchi, Thu-Trang Nguyen, Quang-Thuy Ha (2009) Web Search Clustering and Labeling with Hidden Topics. *ACM Transactions on Asian Language Information Processing*. 8(3). 12 (August 2009). 40 pages
- [OKI04] Satoshi Oyama, Takashi Kokubo and Toru Ishida (2004) Domain-Specific Web Search with Keyword Spices. *IEEE Transactions on Knowledge and Data Engineering*. 16 (1). January 2004
- [Osi03] Osinski, S. (2003) An algorithm for clustering web search result. MS Thesis Poznan University of Technology, Poland.
- [Payn08] Nigel Payne (2008) A Longitudinal Study of Academic Web Links Identifying and Explaining Change. PhD Thesis, University of Wolverhampton. 2008
- [PBM98] Page, L., Brin, S., Motwani, R. and Winograd, T. (1998) The PageRank citation ranking bringing order to the Web. Technical report Stanford University.
- [PC04] Peng, F. and McCallum, A. (1998) Accurate information extraction from research papers using conditional random fields. *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAAACL)*. 2004
- [PCW03] Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003) Table extraction using conditional random fields. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR 2003)*
- [PHH04] Phan, X. H., Honguchi, S., and Ho, T. B. Product extraction from the Web based on entropy estimation. *The 2004 IEEE/WIC/ACM International Joint Conference on Web Intelligence (WI-2004) and Intelligent Agent Technology (IAT-2004)*.
- [Pia06] Gregory Piatetsky-Shapiro (2006) Data Mining Course (Power Point Version) <http://www.kdnuggets.com/index.html>.
- [PKM03] Borislav Popov, Alanas Kiryakov, Dimitar Manov, Angel Kirilov, Damyan Ognyanoff, Miroslav Goranov (2003) Towards Semantic Web Information Extraction. *Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003)*. 2003
- [PNH08] Phan, X. H., Nguyen, L. M., and Honguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *Proceedings of WWW* 91-100
- [Por80] Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program*. 14(3). 130-137. 1980.
- [PRU02] Pandurangan, G., Raghavan, P. and Upfal, E. (2002) Using PageRank to characterize Web structure. Proc 8th Ann. Int. Computing and Combinatorics Conf (COCOON). Lecture Notes in Computer Science. 2387. 330-390. Springer. 2002
- [PS07] Simone Paolo Ponzetto, Michael Strube (2007) Knowledge Derived From Wikipedia For Computing Semantic Relatedness. *Journal of Artificial Intelligence Research* 30 (2007) 181-212
- [PSM04] Gautam Pant, Parminni Srinivasan, and Filipo Menczer (2004) Crawling the Web. *Web Dynamics 2004*. 153-178
- [PTL93] F Pereira, N Tishby and L Lee (1993) Distributional clustering of english words. Proc of 30th Annual Meeting of the Association for Computational Linguistics (ACL). 183-190. 1993
- [PU00] Popescul, A. and Ungar, L. (2000) Automatic labeling of document clusters <http://cileseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.141>
- [PVT08] Jovan Pečović, Anne-Marie Vercoustre, James A. Thom (2008) Exploiting Locality of Wikipedia Links in Entity Ranking. *ECIR 2008*. 258-269
- [Qui86] J.R. Quinlan (1986) Induction of decision trees. *Machine Learning*. 1(1). 81-155. 1986
- [Qui93] J.R. Quinlan (1993) C4.5 Program for Machine Learning. Morgan Kaufmann. 1993
- [Rab89] Rabiner, L. R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*. 77(2). 257-286. 1989
- [Rad09] Dragomir R. Radov (2009) Bibliography Webgraph Papers. January 23. 2009 <http://langra.si.umich.edu/~radov/webgraph/webgraph.ps>
- [RC01] Ray, S. and Craven, M. (2001) Representing sentence structure in hidden Markov models for information extraction. *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)*

- [RGS04] Davi de Castro Reis, Paulo B. Golgher, Aligran S. da Silva, Alberto H. F. Laender (2004). Automatic Web News Extraction Using Tree Edit Distance. *Proceedings of the Thirteenth International World Wide Web Conference*: 502-601. ACM Press, New York, NY, May 2004. ISBN 1581139126
- [RK07] Tarmo Robal, Ahti Kalja (2007). Applying User Profile Ontology for Mining Web Site Adaptation Recommendations, *ADBIS Research Communications* 2007
- [Ros03] Sheldon Ross (2003). Introduction to probability models. 8th Edition. Academic Press, January 2003
- [Ros04] Charles Joseph Rosenberg (2004). Semi-Supervised Training of Models for Appearance-Based Statistical Object Detection Methods. *PhD Thesis*. CMU-CS-04-150, May 2004.
- [SA01] Yuri Slyntko and Sergei Ananyan (2001). *WebAnalyst Server™ - universal platform for intelligent e-business*. Megaputer, 2001
- [Sal71] G Salton (1971). The SMART retrieval system. *Experiments on Automatic Document Processing*. Prentice Hall, 1971
- [SC05] Sarawagi, S. and Cohen (2005). W. Semi-Markov conditional random fields for information extraction. *Proceedings of Advances in Neural Information Processing Systems (NIPS 17)*, 2005
- [Sch09] Johannes C. Scholles (2009) Text Mining. The next step in Search Technology. Technical Report. ZyLAB North America LLC, 2009
- [Scho06] Schonhofen, P. (2006). Identifying document topics using the wikipedia category network. *WI '06. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, 456-462.
- [SCR03] Skounakis, M., Craven, M., and Ray, S. (2003). Hierarchical hidden Markov models for information extraction. *Proceedings of IJCAI 2003*.
- [SCR99] Seymore, K., McCallum, A., and Rosenthal, R. (1999). Learning hidden Markov model structure for information extraction. *Proceedings of AAAI Workshop on Machine Learning for Information Extraction* 1999
- [SDK02] J Srivastava, Prasanna Desikan, Vipin Kumar (2002). Web Mining Accomplishments & Future Directions. <http://www.csse.umbc.edu/~kolan1/Mining/papers/srivastava.pdf>
- [Seb02] Fabrizio Sebastiani (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1-47, 2002.
- [SEO03] SEO research labs (2003). Special report "How to prosper with the new Google". 2003
- [Ser01] Sergei Ananyan (2001). Text Mining Application and Technology. *Megaputer Intelligence Inc.*, 2001 (<http://www.megaputer.com>)
- [SH06] Sahami, M. and Heilman, T. (2006). A web-based kernel function for measuring the similarity of short text snippets. *WWW 2006*: 377-386.
- [SH99] Craig Silverstein, Monika Henzinger (1999). Analysis of a Very Large Web Search Engine Query Log. *ACM SIGIR Forum*, 33(1):6-12
- [SHM98] Craig Silverstein, Monika Henzinger, Hannes Marais, and Michel Moniz (1998). Analysis of a very large AltaVista Query Log. *Digital SRT Technical Note #1998-014*, 0-1998
- [SJ04] Spink, A. & Jansen, B.J. (2004). A study of Web search trends. *Webology*, 1(2), 4. <http://www.webology.nl/2004/v1n2/a4.html>
- [SK06] V. Sindhwani, S. S Keerthi (2006). Large Scale Semi-supervised Linear SVMs. *SIGIR 2006*
- [SK07] V. Sindhwani, S. S. Keerthi (2007). Newton Methods for Fast Solution of Semi-supervised Linear SVMs. *Large Scale Kernel Machines*, MIT Press, 2007
- [SKC02] Junhyeok Shim, Dongsook Kim, Jeongwon Cha, Gary Geunbae Lee, Jungyun Seo (2002). Integrated multi-strategic Web document pre-processing for sentence and word boundary detection. *Inf Process Manage*, 38(4):509-527
- [SKK00] M. Steinbach, G. Karypis, and V. Kumar (2000). A comparison of document clustering techniques. *TextMining Workshop*, KDD, 2000
- [Slx02] Sen Slattery (2002). Hypertext Classification. *PhD Thesis* (CMU-CS-02-142). School of Computer Science Carnegie Mellon University, 2002
- [Son05] Son Doan (2005). Study of Text Representation and Feature Selection for Text categorization. *PhD Thesis*. Japan Advanced Institute of Science and Technology, 2005
- [Son07] Nguyen Hung Son (2007). *Proceeding of the ECML/PKDD 2007 discovery Challenge*. September 17, 2007, Warsaw, Poland
- [Sra07] Rastislav Šramek (2007). The on-line Viterbi algorithm. *Master's Thesis*. Faculty of Mathematics, Physics and Informatics Comenius University, Bratislava
- [STH06] Son Doan, Quang Thuy Ha, and Susumu Horiguchi (2006). A General Fuzzy-based Framework for Text Representation and its Application to Text Categorization. *Lecture Notes on Artificial Intelligence (LNAI)*, 4423: 611-620, 2006

- [SWY75] G. Salton, A. Wong, and C.S. Yang (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11) 613-620, 1975.
- [TC06] Treeratpituk, P. and Callan, J. (2006). Automatically labeling hierarchical clusters. *Proceedings of the 2006 International Conference on Digital government research* 167-276. San Diego, California, USA.
- [Tha08] Nguyen Tri Thanh (2008). Study on Acquisition and Use of Linguistic Semantic Information for Searching. PhD Thesis, Japan Advanced Institute of Science and Technology 2008.
- [TM69] Jeffrey Travers, Stanley Milgram (1969). An Experimental Study of the Small World Problem. *Sociometry*, 32(4): 425-443, Dec., 1969.
- [TNT06] Thuy Q. Ha, Nam H. Nguyen, and Trang T. Nguyen (2006). Improve Performance of PageRank Computation with Connected-Component PageRank. *International Journal of Natural Sciences and Technology*, 1(1): 53-60, 2006 (selected from ICMOCCA2006 15-17 August, Seoul Korea).
- [Tur91] Howard Robert Turtle (1991). Inference Networks for Document Retrieval. PhD Thesis, University of Massachusetts, 1991.
- [UGP96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy (1996). Advances in Knowledge Discovery and Data Mining. AAAI Press / The MIT Press, 1996.
- [Vap98] Vladimir N. Vapnik (1998). Statistical Learning Theory. Wiley, 1998.
- [Vig05] Sebastiano Vigna (2005). TruRank: Taking PageRank to the Limit. *Proc. of the Fourteenth International World Wide Web Conference (Poster Session)*. Chiba, Japan, 2005. ACM Press.
- [VK06] Vikas Sindhwani and S. Sathiya Keerthi (2006). Large Scale Semi-supervised Linear SVMs. *SIGIR 2006* 477-484.
- [VTP08] Anne-Marie Vercoustre, James A. Thom, Jovan Pechevski (2008). Entity ranking in Wikipedia. *SAC 2008*: 1101-1106.
- [WAD99] S.M. Weiss, C. Apte, F.J. Damerau, D.E. Johnson, F.J. Oles, T. Goetz, and T. Hampp (1999). Maximizing text mining performance. *IEEE Intelligent systems*, 14(4) 63-69, 1999.
- [WBH08] Wang, C., Blei, D., and Heckerman, D. (2008). Continuous time dynamic topic models. *Proceedings of Uncertainty in Artificial Intelligence, (UAI-2008)*.
- [WCW07] Wang, X., McCallum, A., and Wei, X. (2007). Topical n-grams: Phrase and topic discovery with an application to information retrieval. *Proceedings of Seventh IEEE International Conference on Data Mining*: 697-702.
- [WD05] Baoning Wu and Brian D. Davison (2005). Identifying Link Farm Spam Pages. *Proceedings of the 14th International World Wide Web Conference, Industrial Track*. May 2005.
- [WHW09] Junfeng Wang, Xiaofei He, Can Wang, Jian Pei, Jiajun Bu, Chun Chen, Ziyu Guan, Lu Gang (2009). News Article Extraction with Template-Independent Wrapper. *WWW 2009* 1085-1086 April 20-24, 2009, Madrid, Spain.
- [Wit06] Witte, R. (2006). Prelude Overview: Introduction to Text Mining Tutorial. *EDBT, 2006* <http://www.edbt2006.de/edbt-share/IntroductionToTextMining.pdf>
- [WKQ08] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg (2008). Top 10 algorithms in data mining. *Knowl Inf Syst* (2008) 14 1-37.
- [WZT05] Jason T.L. Wang, Mohammed J. Zaki, Hannu T.T. Toivonen and Dennis Shasha (eds 2005). *Data Mining in Bioinformatics*. Springer, 2005.
- [Yan99] Y. Yang (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval Journal* 1:69-90, 1999.
- [YHC02] Hwano Yu, Jiawei Han, Kevin Chen-Chuan (2002). PEBL: Positive Example Based Learning for Web Page Classification Using SVM. *Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. July 23-26, 2002. Alberta Canada, 239-248.
- [YM07] Yih, W. and Meek, C. (2007). Improving similarity measures for short segments of text. *Proceedings of AAAI07* 1489-1494.
- [YN99] Ricardo Baeza-Yates, Berthier Ribeiro-Neto (1999). *Modern Information Retrieval*. Addison-Wesley, 1999.
- [Yoo06] Ilhoi Yoo (2006). Semantic Text Mining and its Application in Biomedical Domain. PhD Thesis, Drexel University, 2006.
- [YSG02] Y. Yang, S. Slattery, and R. Ghani (2002). A study of approaches to hypertext categorization. *Journal of Intelligent Information System* 18(2/3): 69-90, 2002.
- [YW06] Qiang Yang and Xindong Wu (2006). 10 Challenging Problems in Data Mining Research. *International Journal of Information Technology and Decision Making* 5(4): 597-614, 2006.

- [YZ03] Qiang Yang and Haining Henry Zhang (2003) Web-Log Mining for Predictive Web Caching. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), July/August 2003.
- [ZE99] Zamir, O. and Etzioni, O. (1999) Grouper: a dynamic clustering interface to Web search results. *Computer Networks (Amsterdam, Netherlands)*, 31 (11-16) 1361-1374.
- [ZHC04] Zeng, H. J., He, Q. C., Chen, Z., Ma, W. Y., and Ma, J. (2004) Learning to cluster web search results. *SIGIR 2004* 210-217.
- [ZHF05] Baoyao Zhou, Siu Cheung Hui, Alvis C. M. Fong (2005) Web Usage Mining for Semantic Web Personalization. *Workshop on Personalization on the Semantic Web*. 66-72. Edinburgh, UK, 2005. <http://www.win.tue.nl/persweb/Camera-ready/9-Zhou-full.pdf>
- [Zhu05] Zhu, X. (2005) Semi-supervised learning with graphs. *PhD Thesis*, Carnegie Mellon University. CMU-LTI-05-192
- [Zhu08] Xiaojin Zhu (2008) Semi-Supervised Learning Literature Survey. http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_Survey.pdf
- [Zia94] Wojciech P. Ziarko (Ed.), 1994). Rough Sets, Fuzzy Sets and Knowledge Discovery. *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery (RSKD'93)*, Banff, Alberta, Canada, 12-15 October 1993 Springer-Verlag
- [Zip49] G.K. Zipf (1949) Human Behavior and the Principle of Least-Effort, Addison-Wesley, 1949
- [ZL05] Zhou, Z.-H. & Li, M. (2005) Semi-supervised regression with co-training. *International Joint Conference on Artificial Intelligence (IJCAI)*
- [ZLT05] Zhi-Hua Zhou và Ming Li. Tri-Training (2005) Exploiting Unlabeled Data Using Three Classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11) : 1529-1541, 2005.
- [ZNW05] Zhu, J., Nie, Z., Wen, J. R., Zhang, B., and Ma, W. Y. (2005). 2D conditional random fields for Web information extraction. *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005.
- [Aspseek] ASP Open search engine <http://www.aspseek.org>
- [Baa08] Baamboo (2008). Vietnamese search engine <http://mp3.baamboo.com>
- [InfoWor] <http://www.InfoWorld.com>
- [Kdn] <http://www.kdnuggets.com>
- [Nutch] <http://lucene.apache.org/nutch>.
- [OpenDir] Open Directory Project (ODP). <http://www.dmoz.org/> [Dmoz]
<http://www.dmoz.com>
- [Soc08] Socbay (2008). Vietnamese search engine, <http://www.socbay.com>
- [Viterbi] http://www.cim.mcgill.ca/~latorre/Viterbi/va_elg.htm
- [Vivi08] Vivisimo (2008). Clustering engine. <http://vivisimo.com>
- [Vnni08] Vnnic (2008). Vietnam internet center. <http://www.thongkeinternet.vn>
- [Xalo08] Xalo (2008). Vietnamese search engine. <http://xalo.vn>.
- [Yahoo] <http://www.yahoo.com>
- [Zing08] Zing (2008). Vietnamese website directory. <http://directory.zing.vn>.

Chịu trách nhiệm xuất bản :

Chủ tịch HDQT kiêm Tổng Giám đốc NGÔ TRẦN ÁI
Phó Tổng Giám đốc kiêm Tổng biên tập NGUYỄN QUÝ THAO

Tổ chức bản thảo và chịu trách nhiệm nội dung :

Chủ tịch HDQT kiêm Giám đốc Công ty CP Sách DH – DN
TRẦN NHẬT TÂN

Biên tập nội dung và sửa bản in :

ĐỖ HỮU PHÚ

Thiết kế mỹ thuật và trình bày bìa :

BÍCH LA

Thiết kế sách và chế bản :

ĐỖ PHÚ

GIÁO TRÌNH KHAI PHÁ DỮ LIỆU WEB

Mã số: 7B753Y9 – DAI

In 1.000 bản (QĐ : 66), khổ 16 x 24 cm. In tại Nhà in Đại học Quốc Gia Hà Nội.

Địa chỉ : 16 Hàng Chuối, Hà Nội.

Số ĐKKH xuất bản : 375 – 2009/CXB/8 – 726/GD.

In xong và nộp lưu chiểu tháng 10 năm 2011