ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÙI NGUYÊN KHỞI

NGHIÊN CỨU MỘT SỐ PHƯƠNG PHÁP PHÂN LỚP CẢI TIẾN, ỨNG DỤNG VÀO HỆ TRUY TÌM VĂN BẢN

Chuyên ngành: KHOA HỌC MÁY TÍNH
Mã số: 60 48 01

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

HƯỚNG DẪN KHOA HỌC:
TS. VŨ THANH NGUYÊN

TP Hồ Chí Minh - 2009

MỤC LỤC

Trang

DANH MỤC BẢNG

Trang

DANH MỤC HÌNH VẼ

Trang

MỞ ĐẦU

Ngày nay, việc tìm kiếm thông tin nói chung cũng như thông tin văn bản nói riêng có vai trò rất quan trọng trong mọi lĩnh vực hoạt động của con người, nó trở đã thành một nhu cầu thiết yếu không thể thiếu. Với sự xuất hiện của internet thì khối lượng thông tin văn bản trên mạng ngày càng tăng, hình thành một kho văn bản khổng lồ, làm cho việc tìm kiếm những thông tin văn bản cần thiết, hữu ích thì ngày càng trở nên khó khăn hơn.

Xuất phát từ thực tế đó, đã có một số nghiên cứu xây dựng các hệ truy tìm văn bản theo các mô hình khác nhau, trong đó hệ truy tìm văn bản theo mô hình không gian vector được đánh giá là có nhiều ưu điểm nhất. Tuy nhiên, đối với một hệ truy tìm văn bản theo mô hình không gian vector cơ bản, việc xử lý truy tìm phải thực hiện trên toàn bộ tập văn bản. Điều này làm mất rất nhiều thời gian xử lý, tốc độ truy tìm sẽ chậm, đồng thời phải tiêu tốn nhiều không gian lưu trữ, tài nguyên tính toán, nếu tập văn bản lớn (hoặc số lượng từ đặc trưng lớn).

Bài toán đặt ra là làm thế nào để xây dựng một hệ thống tự động phân lớp và phục vụ truy tìm thông tin văn bản theo mô hình không gian vector VSM có cải tiến so với hệ thống truy tìm theo mô hình không gian vector VSM cơ bản, để việc truy tìm được nhanh chóng và hiệu quả hơn.

Hướng tiếp cận giải quyết như sau: Việc cải tiến hệ thống truy tìm văn bản theo mô hình không gian vector VSM được thực hiện bằng cách kết hợp sử dụng các kết quả phân lớp văn bản trên kho văn bản trước khi thực hiện các kỹ thuật xử lý truy tìm. Kết quả của việc cải tiến này là phân hệ truy tìm văn bản sẽ cải thiện đáng kể tốc độ, hiệu quả truy tìm vì không phải thực hiện xử lý truy tìm trên toàn bộ kho văn bản mà chỉ thực hiện truy tìm trên một hoặc vài nhóm văn bản có liên quan với câu truy vấn.

Hiện tại, đã có một số nghiên cứu về kỹ thuật phân lớp văn bản cũng như về kỹ thuật truy tìm thông tin văn bản. Luận văn này nhằm mục đích tìm hiểu các kỹ

thuật trên và áp dụng vào việc xây dựng thử nghiệm một hệ thống tự động phân lớp và phục vụ truy tìm thông tin văn bản thực tế.

Đối với các kỹ thuật phân lớp văn bản, luận văn tìm hiểu cụ thể kỹ thuật phân lớp văn bản Support Vector Machines (SVM) do kết quả phân lớp rất tốt của phương pháp này theo các đề tài đã nghiên cứu trước đây. Ý tưởng chính của SVM là tìm một siêu phẳng "tốt nhất" trong không gian n-chiều để phân chia các điểm dữ liệu (văn bản) sao cho các điểm dữ liệu thuộc 2 lớp khác nhau nằm ở 2 phía của siêu phẳng. Luận văn cũng nghiên cứu các thuật toán phân lớp văn bản cải tiến dựa trên kỹ thuật SVM là thuật toán Fuzzy SVM cho phép loại bỏ các dữ liệu nhiễu trong quá trình huấn luyện và cải thiện độ chính xác của quá trình phân lớp, nghiên cứu và cài đặt áp dụng thuật toán SVM Nearest Neighbor với việc kết hợp ý tưởng của thuật toán K-Nearest Neighbor và thuật toán SVM để cải thiện hiệu quả phân lớp. Đồng thời luận văn còn nghiên cứu và cài đặt áp dụng các chiến lược phân lớp văn bản đa lớp OAR (One - against - Rest), OAO (One - against - One) và kỹ thuật cải tiến việc phân lớp đa lớp này là phân lớp đa lớp mờ Fuzzy OAO (Fuzzy One - against - One).

Đối với các kỹ thuật phục vụ truy tìm văn bản, luận văn tìm hiểu sử dụng mô hình truy tìm văn bản theo mô hình không gian vector VSM (Vector Space Model). Nguyên lý hoạt động cốt lõi của hệ truy tìm văn bản VSM là tự động hóa quy trình tìm kiếm các văn bản có liên quan bằng cách tính độ đo tương tự giữa câu truy vấn và các văn bản đó.

Từ kết quả nghiên cứu trên, các kỹ thuật phân lớp và phục vụ truy tìm văn bản sẽ được cài đặt áp dụng để xây dựng thử nghiệm một hệ thống tự động phân lớp và phục vụ truy tìm thông tin văn bản thực tế theo mô hình không gian vector VSM có cải tiến so với hệ thống truy tìm theo mô hình VSM cơ bản.

Nội dung luận văn gồm 6 chương:

- Chương 1: Tổng quan về bài toán phân lớp văn bản.
- Chương 2: Các phương pháp phân lớp văn bản truyền thống.
- Chương 3: Các thuật toán cải tiến dựa trên phương pháp phân lớp văn bản Support Vector Machines.
- Chương 4: Tổng quan về bài toán truy tìm văn bản.
- Chương 5: Xây dựng thử nghiệm hệ phân lớp và truy tìm văn bản.
- Chương 6: Kết luận.

CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN PHÂN LỚP VĂN BẢN

1.1 Giới thiệu bài toán

Bài toán *Phân lớp văn bản (Text Categorization, Text Classification)* được mô tả như sau*:* cho một số lớp văn bản đã được xác định trước, nhiệm vụ của phân lớp văn bản là gán các văn bản vào một (hay một số) lớp văn bản thích hợp dựa vào nội dung của văn bản.

Trong những thập kỷ 80 hầu hết các cách tiếp cận (ít nhất là trong việc thiết đặt thao tác) để phân lớp văn bản tự động gồm các kỹ thuật điều khiển bằng tay bởi chuyên gia tri thức (Knowledge Engineering).

Theo thời gian, cách tiếp cận để giải quyết bài toán phân lớp đã có sự thay đổi. Đầu thập kỷ 90, cách tiếp cận máy học (Machine Learning) để phân lớp văn bản được coi là nổi tiếng và trở thành thống trị, ít nhất là trong cộng đồng người nghiên cứu.

1.1.1 Phân lớp văn bản dựa trên cách tiếp cận hệ chuyên gia

Theo cách tiếp cận này, việc phân lớp văn bản tự động được điều khiển bằng tay bởi các chuyên gia tri thức và hệ chuyên gia có khả năng đưa ra quyết định phân lớp. Hệ chuyên gia bao gồm một tập các luật logic định nghĩa bằng tay, cho mỗi loại, có dạng:

If *(DNF formula)* **then** *(category).*

Công thức DNF (“Disjunctive Normal Form”) là hợp của các mệnh đề liên kết, tài liệu được phân lớp vào *category* nếu nó thỏa mãn công thức, nghĩa là, nếu nó thỏa mãn ít nhất một mệnh đề trong công thức.

Đây là ví dụ về một luật logic định nghĩa bằng tay:

If *((“lúa mì” & “nông trại”)* **or** *(“lúa mì” & “hàng hóa”)* **or** *(“thúng để đong lúa mì” & “hàng xuất khẩu”)* **or** *(“lúa mì” & “hàng tấn”)* **or** *(“lúa mì” & “mùa đông” & ¬ “sự ôn hòa”))*

then *“lúa mì”*

else ¬ *“lúa mì”*

Điều trở ngại của cách tiếp cận này là hạn chế trong quá trình thu nhận tri thức từ tài liệu của các hệ thống chuyên gia. Nghĩa là, các luật phải được định nghĩa bằng tay bởi kỹ sư tri thức với sự giúp đỡ của chuyên gia về lĩnh vực được nêu trong tài liệu. Nếu tập hợp của các loại được cập nhật, thì hai nhà chuyên gia phải can thiệp lại, và nếu phân lớp được chuyển hoàn toàn sang một phạm vi khác, một chuyên gia về lĩnh vực này cần thiết phải can thiệp vào và công việc phải được bắt đầu lại từ tập tài liệu hỗn tạp ban đầu.

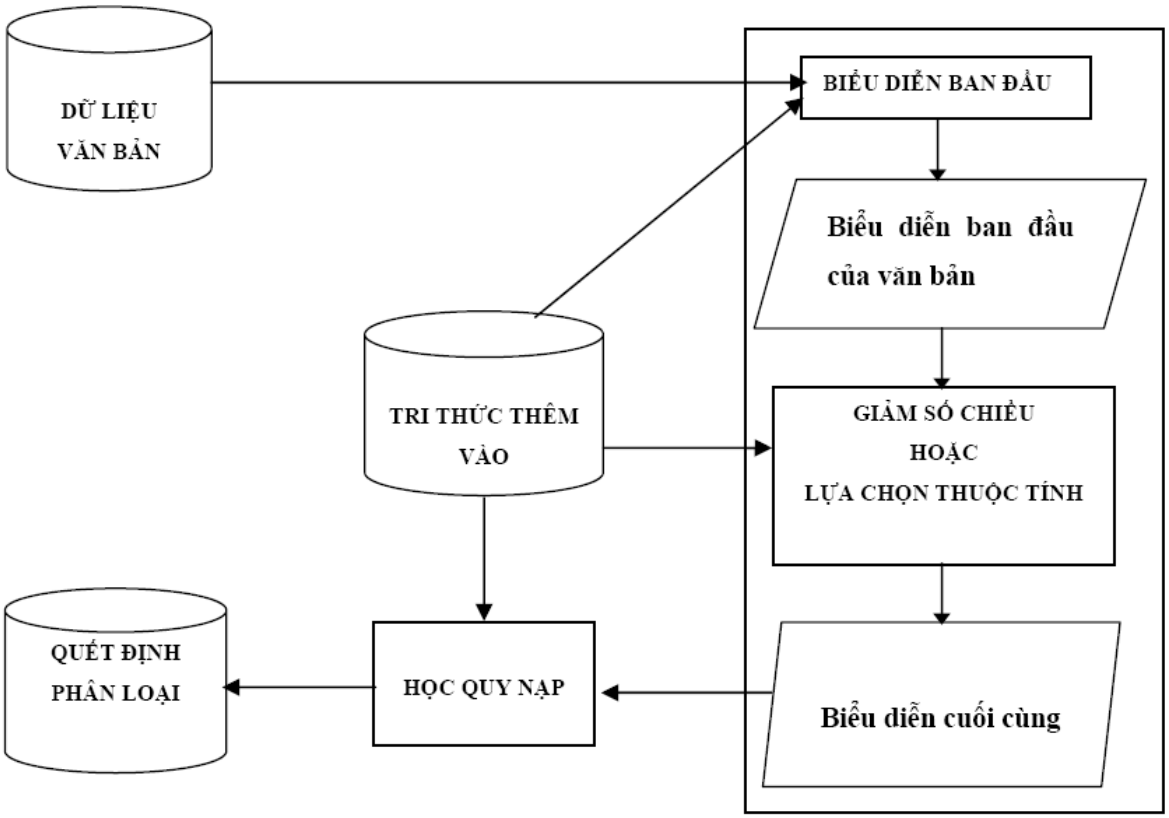
1.1.2 Phân lớp văn bản dựa trên cách tiếp cận máy học [3]

Theo cách tiếp cận này, *một quá trình xử lý quy nạp chung (cũng được gọi là quá trình học) xây dựng tự động một phân lớp cho một loại* c_i *bằng quan sát các đặc trưng của tập hợp các tài liệu đã được phân bằng tay vào* c_i *hay* $\overline{c_i}$ *bởi chuyên gia về lĩnh vực này; từ đó, quá trình qui nạp thu lượm các đặc trưng để phân lớp một tài liệu mới (không nhìn thấy) vào* c_i. Trong kỹ thuật máy học, bài toán phân lớp là hoạt động học có giám sát, quá trình học được "giám sát" bởi tri thức của các phân lớp và của các mẫu huấn luyện thuộc chúng.

Với phương pháp máy học, sự cố gắng về phương diện công việc của kỹ sư theo hướng không phải xây dựng một phân lớp mà xây dựng một phân lớp tự động (học) từ một tập hợp các tài liệu đã được phân lớp bằng tay. Trong các tiếp cận máy học, các tài liệu đã được phân lớp trở thành nguồn. Trường hợp thuận lợi nhất, chúng đã có sẵn, khi đó quá trình phân lớp bắt đầu bằng việc học từ tập dữ liệu này, sau đó sẽ thực hiện phân lớp tự động với các tài liệu khác. Trường hợp ít thuận lợi, không có sẵn tài liệu đã phân lớp bằng tay; khi đó quá trình phân lớp bắt đầu một hành động phân lớp và chọn một phương pháp tự động ngay lập tức. Do đó, cách tiếp cận máy học là thuận lợi hơn cách tiếp cận kỹ sư tri thức.

Các phân lớp xây dựng theo nghĩa của kỹ thuật máy học ngày nay gây được ấn tượng về mức độ hiệu quả, khiến cho phân lớp tự động trở thành một lựa chọn tốt để thay thế phân lớp bằng tay (không chỉ về phương diện kinh tế).

Chúng ta có thể hình dung các công việc của bài toán phân lớp văn bản dựa trên cách tiếp cận máy học như sau:



Hình 1.1: Bài toán phân lớp văn bản dựa trên cách tiếp cận máy học

Bài toán phân lớp văn bản dựa trên kỹ thuật máy học gồm các bước sau:

Bước 1: Chuẩn bị tập dữ liệu huấn luyện (Training Set) và tập dữ liệu kiểm tra (Test Set).

Bước 2: Tách từ trong văn bản.

Bước 3: Biểu diễn văn bản.

Bước 4: Phương pháp học để phân lớp văn bản.

Bước 5: Đánh giá hiệu quả của phương pháp học.

Bước 1: Chuẩn bị tập dữ liệu huấn luyện và tập dữ liệu kiểm tra.

Cách tiếp cận máy học dựa trên một tập dữ liệu có sẵn từ đầu $\Omega = \{d_1, \ldots, d_{|\Omega|}\} \subset D$, trong đó D tập tất cả các tài liệu đã được phân lớp trước, d_j là văn bản thứ j, Tập các lớp $C=\{c_1, \ldots, c_{|C|}\}$, c_i là kí hiệu của lớp thứ i. Hàm

$\breve{\Phi}$ $DxC \rightarrow$ $rue, False$ với mọi $\langle d_j, c_i \rangle \in$ 2 , Một tài liệu d_j thuộc lớp c_i nếu $\breve{\Phi}$ $_j, c_i$ = rue, d_j không thuộc lớp c_i nếu $\breve{\Phi}$ $_j, c_i$ = $alse$.

Với mỗi cách phân lớp được đưa ra, người ta mong muốn đánh giá được hiệu quả phân lớp của chúng. Bởi vậy, trước khi xây dựng phân lớp người ta chia tập dữ liệu ban đầu thành 2 tập hợp:

- Tập huấn luyện $Tr=\{d_1, ..., d_{|TV|}\}$. Phân lớp Φ cho các phân lớp $C=\{c_1, ..., c_{|C|}\}$ được xây dựng quy nạp dựa trên sự quan sát các đặc trưng của các tài liệu trong *Tr*.

- Tập kiểm tra $Te=\{d_{|TV+1|}, ...d_{|\Omega|}\}$, được sử dụng để kiểm tra hiệu quả của phân lớp. Mỗi $d_j \in T_e$ được đưa vào hệ thống phân lớp để xác định giá trị, và so sánh giá trị này với quyết định Φ $,c$ của chuyên gia. Hiệu quả của phân lớp dựa trên sự phù hợp giữa Φ $,c$ và $\breve{\Phi}$ $,c$. *Số tài liệu trong tập luyện và tập kiểm tra thường được chọn theo tỷ lệ tương ứng là 70% và 30%.* Trong đó, $Tr \cap Te =$, nếu điều kiện này bị vi phạm thì kết quả đánh giá hiệu quả của mô hình mất đi yếu tố khách quan, khoa học.

Bước 2: Tách từ trong văn bản

Hầu hết các phương pháp phân lớp văn bản dựa trên kỹ thuật máy học hiện nay đều dựa vào tần xuất xuất hiện (số lần xuất hiện) của từ hoặc cụm từ trong văn bản, hoặc dựa vào tần xuất xuất hiện của từ trong văn bản và tần xuất văn bản (số các văn bản trong tập dữ liệu huấn luyện có chứa từ đó). Độ chính xác của kết quả tách từ có ảnh hưởng rất lớn đến kết quả của phân lớp, không thể có một kết quả phân lớp tốt nếu như không tách được đúng các từ trong văn bản. Bởi vậy, một vấn đề quan trọng đối với phân lớp văn bản là phải tách được chính xác các từ trong văn bản. Các văn bản được viết bằng các ngôn ngữ khác nhau thì có đặc trưng riêng của ngôn ngữ đó, và không có một phương pháp chung nào để tách các từ trong các văn bản được viết bằng các ngôn ngữ khác nhau.

Bước 3 : Biểu diễn văn bản.

Các văn bản ở dạng thô cần được chuyển sang một dạng biểu diễn nào đó để xử lý. Quá trình này được gọi là quá trình biểu diễn văn bản, dạng biểu diễn của văn bản phải có cấu trúc và dễ dàng xử lý.

Việc biểu diễn lại văn bản được coi là một khâu quan trọng trong quá trình xử lý văn bản. Mỗi tài liệu được mô tả như một chuỗi các ký tự, cần phải được biến đổi thành những mô tả phù hợp với nhiệm vụ và thuật toán xử lý văn bản. Có rất nhiều phương pháp biểu diễn văn bản, mỗi phương pháp thích hợp với từng bài toán cụ thể. Trong luận văn này chúng ta sẽ tìm hiểu sâu về phương pháp biểu diễn văn bản theo mô hình không gian vector.

Bước 4: Phương pháp học để phân lớp văn bản

Phương pháp học để phân lớp văn bản thường được sử dụng trong quá trình xây dựng quy nạp của các phân lớp.

Cho đến nay, đã có nhiều đề xuất xây dựng bài toán phân lớp văn bản tự động như Neive Bayes, DBSCAN, K-trung bình, K-láng giềng gần nhất, cây quyết định, mạng nơron, Support Vector Machines, … Các phương pháp phân lớp này, đạt được những thành công đáng kể đối với các văn bản tiếng Anh, Pháp, Nhật, Trung Quốc, và đã được ứng dụng trong thực tế như trong các hệ tìm tin của Yahoo, Altavista, Google, … Trong đó, Support Vector Machines và các thuật toán cải tiến của nó được đánh giá cho độ chính xác phân lớp văn bản cao hơn nhiều phương pháp phân lớp khác.

1.2 Phương pháp tách từ

Để máy tính có thể tự động phân lớp văn bản, các văn bản được trình bày dưới dạng chuỗi các ký tự cần phải được biến đổi thành một biểu diễn thuận lợi cho thuật toán huấn luyện và bài toán phân lớp, nghĩa là văn bản được chuyển từ dạng không có cấu trúc (hoặc bán cấu trúc) sang dạng có cấu trúc. Có rất nhiều cách biểu diễn văn bản, nhưng dù theo cách này hay cách khác thì việc biểu diễn văn bản đều dựa vào sự xuất hiện của từ trong văn bản.

1.2.1 Các đặc điểm của văn bản tiếng Việt

Tiếng Việt là ngôn ngữ đơn âm tiết, và thuộc nhóm ngôn ngữ Đông Nam Á. Nó có đặc điểm riêng về ký hiệu, ngữ pháp và ngữ nghĩa, khác với các ngôn ngữ Ấn-Âu. Đây không chỉ là khó khăn đối với việc học các ngôn ngữ Châu Âu, mà còn là khó khăn trong việc ứng dụng các kỹ thuật phát triển để xử lý ngôn ngữ tự nhiên. Mặt khác, dù là ngôn ngữ đơn âm tiết nhưng không giống như các ngôn ngữ đơn âm tiết khác như Trung Quốc, Thái, tiếng Việt được viết bằng các ký tự Latin mở rộng. Vì vậy, cách thực hiện của các ngôn ngữ này cũng không thể ứng dụng cho tiếng Việt, và hiện tại một trong số các việc còn chưa được giải quyết trong xử lý ngôn ngữ tự nhiên của tiếng Việt là bài toán xác định các biên giới của từ (word boundaries) trong văn bản tiếng Việt.

Tiếng

Ngôn ngữ Việt Nam có một đơn vị đặc biệt gọi là *tiếng*. Mỗi *tiếng* trong tiếng Việt được viết thành một chữ, ngược lại mỗi chữ đọc thành một tiếng, mỗi chữ nằm giữa hai dấu phân cách trong câu. Tiếng được dùng để tạo thành từ, tiếng có thể có nghĩa rõ ràng hoặc không có nghĩa rõ ràng.

Ví dụ:

- Từ "*lạnh lẽo*" (có nghĩa): tiếng "*lạnh*" (có nghĩa), tiếng "*lẽo*" (nghĩa không rõ).
- Từ "*bồ kết*" (có nghĩa): tiếng "*bồ*" và tiếng "*kết*" (đều có nghĩa).

Tiếng gồm có ba bộ phận kết hợp lại: *âm đầu*, *vần* và *thanh*. Ví dụ, tiếng "*đà*" có âm đầu "*đ*" vần "*a*" và thanh "*huyền*". Hai bộ phận *vần* và *thanh*, tiếng nào cũng phải có. *Âm đầu* thì có tiếng có, có tiếng không. Ví dụ, tiếng "*ở*", chỉ có vần "*o*" và thanh "*hỏi*", không có âm đầu. Mỗi bộ phận của tiếng do một âm hay kết hợp một số âm tạo thành. Bộ phận *âm đầu* do một âm tạo thành. *Âm đầu là phụ âm*. Bộ phận vần có thể do một hoặc 2, 3 âm tạo thành, nhưng bao giờ cũng phải có một âm chính. Âm chính là nguyên âm. Âm cuối của vần cũng có thể là phụ âm. Ví dụ, tiếng "*nam*" có âm đầu là *n*, âm cuối của vần là phụ âm *m*, nguyên âm làm âm chính là *a*.

Tiếng Việt dùng chữ cái để ghi âm. Mỗi âm được ghi bằng 1 hoặc nhiều chữ cái ghép lại. Trật tự bảng chữ cái trong tiếng Việt: a, ă, â, b, c, d, đ, e, ê, g, h, I, k, l, m, n, o, ô, ơ, p, q, r, s, t, u, ư, v, x, y.

Từ

Tồn tại nhiều định nghĩa khác nhau về từ trong tiếng Việt, nhưng tất cả các nghiên cứu ngôn ngữ đều đồng ý từ trong tiếng Việt có những đặc điểm sau (Đinh Điền, 2001):

- Từ phải đầy đủ về phương diện hình thức, ngữ nghĩa và độc lập về mặt ngữ pháp.
- Từ được xây dựng từ *tiếng*.
- Chúng có thể gồm các từ đơn (1-tiếng), hoặc các từ phức (n-tiếng, n<5).

Xét về mặt cấu tạo từ có thể chia thành các loại sau:

- *Từ đơn*: do 1 tiếng tạo thành.
- *Từ ghép*: do 2, 3 hoặc 4 tiếng tạo thành.

Xét về mặt ngữ loại từ trong tiếng Việt được chia thành một số loại cơ bản sau: danh từ, đại từ, động từ, tính từ, phụ từ, trợ từ, thán từ.

1.2.2 Phương pháp tách từ bằng cách xây dựng các Ôtômát [2]

Công việc đầu tiên và có ảnh hưởng lớn đến chất lượng của quá trình phân lớp là kết quả của việc tách từ trong văn bản. Cho đến nay, đã có một số phương pháp tách từ tiếng Việt được đánh giá là hiệu quả. Phần dưới đây sẽ trình bày *phương pháp tách từ bằng cách xây dựng các Ôtômát để đoán nhận các từ*.

Bài toán

Nhập vào một câu tiếng Việt bất kỳ, hãy tách câu đó thành những đơn vị từ vựng (từ), hoặc chỉ ra những âm tiết nào không có trong từ điển (phát hiện đơn vị từ vựng mới).

Giải quyết

Với phương pháp này, chúng ta cần tập dữ liệu gồm từ điển âm tiết (khoảng 6700 âm tiết) và từ điển từ vựng tiếng Việt (khoảng 30.000 từ).

Bài toán gồm các bước giải quyết như sau:

Bước 1: Xây dựng ôtômát âm tiết đoán nhận tất cả các âm tiết tiếng Việt

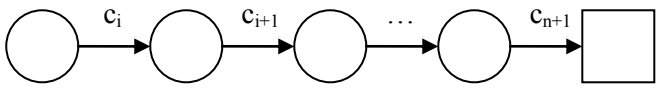
Bước 2: Xây dựng ôtômát từ vựng đoán nhận tất cả các từ vựng tiếng Việt.

Bước 3: Dựa trên các ôtômát nêu trên, xây dựng đồ thị tương ứng với câu cần phân tích và sử dụng thuật toán tìm kiếm trên đồ thị để liệt kê các cách phân tích có thể.

Ý tưởng của phương pháp này là: xây dựng dần dần dựa trên ôtômát đã có ở bước trước và âm tiết (hoặc từ vựng) mới học được từ tệp dữ liệu ở bước hiện tại.

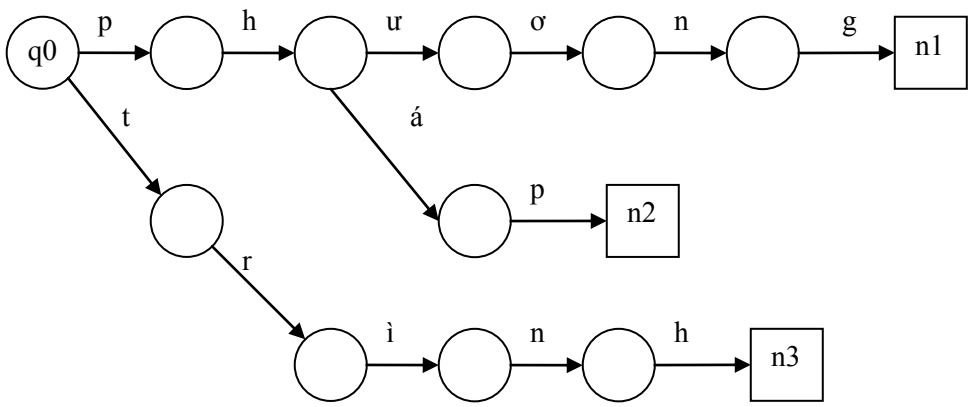
Bước 1: Xây dựng ôtômát âm tiết đoán nhận tất cả các âm tiết tiếng Việt

Bảng chữ cái của ôtômát âm tiết là bảng chữ cái tiếng Việt, mỗi cung chuyển được ghi trên đó một ký tự, ban đầu ôtômát âm tiết chỉ gồm một trạng thái khởi đầu được đánh số hiệu 0. Giả sử tại bước nào đó ta đọc được âm tiết **a** có độ dài n (tính bằng số ký tự) từ tệp dữ liệu. Xuất phát từ trạng thái khởi đầu $p=q_0$ ta lấy ra từng ký tự c_i của **a** và tìm xem từ p có cung chuyển đến trạng thái q nào đó mà trên đó ghi ký tự c_i hay không. Nếu có trạng thái q như thế, ta chuyển p thành q và lặp lại bước trên với ký tự c_{i+1} tiếp theo. Nếu không có q nào như thế, ta ra khỏi vòng lặp và xây dựng mới các trạng thái và cung chuyển tương ứng trên đó ghi các ký tự c_i, c_{i+1}, ..., c_{n-1} theo sơ đồ sau (ô vuông chỉ rằng đó là trạng thái kết thúc).



Hình 1.2: Sơ đồ chuyển trạng thái giữa các ký tự

Ví dụ, với ba âm tiết *phương, pháp, trình* ta sẽ có ôtômát âm tiết như sau:



Hình 1.3: Phương pháp xây dựng ôtômát âm tiết

Bước 2: Xây dựng ôtômát từ vựng đoán nhận tất cả các từ vựng tiếng Việt.

Ôtômát từ vựng được xây dựng tương tự, với điểm khác nhau như sau: thay vì ghi trên mỗi cung chuyển một ký tự, ta ghi một số. Số này là số hiệu của trạng thái (kết) của ôtômát âm tiết tại đó đoán nhận mỗi âm tiết của từ. Với cách tổ chức này, ta làm giảm được kích thước của ôtômát từ vựng mà không làm mất thông tin của nó, bởi vì mỗi âm tiết được xác định bằng một trạng thái kết duy nhất trong ôtômát âm tiết. Ví dụ, với hai từ *phương pháp* và *phương trình*, giả sử khi đưa lần lượt các âm tiết *phương, pháp, trình* qua ôtômát âm tiết, ta đến được các trạng thái kết ghi số n_1, n_2, n_3 thì trên các cung chuyển tương ứng ta ghi các số n_1, n_2, n_3

Bước 3: Dựa trên các ôtômát nêu trên, xây dựng đồ thị tương ứng với câu cần phân tích và sử dụng thuật toán tìm kiếm trên đồ thị để liệt kê các cách phân tích có thể.

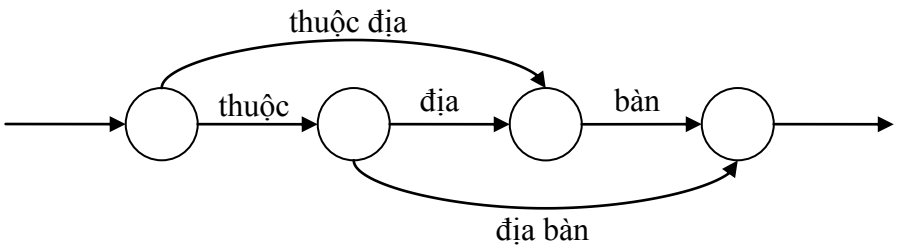
Sau khi đã xây dựng xong hai ôtômát, ta ghi chúng vào hai tệp định kiểu để dùng trong bước phân tách từ vựng. Đến lúc này, hai từ điển ban đầu không còn cần thiết nữa, mọi dữ liệu của ta nằm trong hai tệp ghi hai ôtômát này. Nếu mỗi ký tự (char) được ghi vào tệp với kích thước 2 byte (mã Unicode), mỗi số nguyên (int) có kích thước 4 byte thì tệp lưu ôtômát âm tiết có kích thước 146KB, tệp ôtômát từ vựng có kích thước 1MB.

Ý tưởng của thuật toán phân tách từ vựng là *quy việc phân tách câu về việc tìm đường đi trên một đồ thị có hướng, không có trọng số.*

Giả sử câu ban đầu là một dãy gồm $n+1$ âm tiết $s_0, s_1, ..., s_n$. Ta xây dựng một đồ thị có $n+2$ đỉnh $v_0, v_1, ..., v_n, v_{n+1}$, sắp thứ tự trên một đường thẳng từ trái sang phải; trong đó, từ đỉnh v_i đến đỉnh v_j có cung ($i<j$) nếu các âm tiết $s_i, s_{i+1}, ..., s_{j-1}$ theo thứ tự lập thành một từ. Khi đó mỗi cách phân tách câu khác nhau tương ứng với một đường đi trên đồ thị từ đỉnh đầu v_0 đến đỉnh cuối v_{n+1}. Trong các cách phân tách câu đó, cách phân tích câu đúng đắn nhất ứng với đường đi qua ít cung nhất trên đồ thị.

Trong trường hợp câu có sự nhập nhằng thì đồ thị sẽ có nhiều hơn một đường đi ngắn nhất từ đỉnh đầu đến đỉnh cuối, ta liệt kê toàn bộ các đường đi ngắn nhất trên đồ thị, từ đó đưa ra tất cả các phương án tách câu có thể và để người dùng quyết định sẽ chọn phương án nào, tùy thuộc vào ngữ nghĩa hoặc văn cảnh.

Ví dụ, xét một câu có cụm "*thuộc địa bàn*", ta có đồ thị sau:



Hình 1.4: Một tình huống nhập nhằng

Cụm từ này có sự nhập nhằng giữa *thuộc địa* và *địa bàn* và ta sẽ có hai kết quả phân tách là *"thuộc địa/bàn"* và *"thuộc/địa bàn"*. Ta có thể chỉ ra rất nhiều những cụm nhập nhằng trong tiếng Việt, chẳng hạn như "*bằng chứng cớ*", …

Trường hợp trong câu có âm tiết không nằm trong từ điển thì rõ ràng ôtômát âm tiết không đoán nhận được âm tiết này. Kết quả là đồ thị ta xây dựng từ câu đó là *không liên thông*. Dựa vào tính chất này, ta thấy rằng nếu đồ thị không liên thông thì dễ dàng phát hiện ra rằng đơn vị âm tiết không đoán nhận được không nằm trong từ điển âm tiết, tức nó bị viết sai chính tả hoặc là một đơn vị âm tiết (từ vựng) mới.

Đánh giá kết quả

Với cách tiếp cận như trên, bài toán phân tách từ vựng trong câu tiếng Việt về cơ bản đã được giải quyết, đặc biệt là vấn đề tách các tổ hợp từ tương đương với một đơn vị từ vựng, thường là các cụm từ cố định, ngữ cố định hoặc các thành ngữ trong tiếng Việt. Nếu chúng ta chỉ sử dụng một danh sách từ vựng thông thường và thực hiện các thao tác tìm kiếm trên danh sách này thì không thể đảm bảo thời gian tách từ vựng đối với câu có chiều dài lớn.

Với những câu nhập vào có sự nhập nhằng từ vựng, có nhiều hơn một cách phân tách thì chương trình liệt kê toàn bộ các phương án tách từ có thể và giành quyền lựac họn kết quả cho người sử dụng. Trong tất cả các phương án phân tách đó bao giờ cũng tồn tại một phương án đúng.

Dưới đây là một số câu nhập vào và kết quả tách từ tương ứng:

1. Nó | là | một | bản | tuyên ngôn | đặc sắc | của | chủ nghĩa nhân đạo | , một | tiếng | chuông | cảnh tỉnh | trước | hiểm họa | lớn lao | của | hành tinh | trước | sự | điên rồ | của | những | kẻ | cuồng tín

2. Trong khi | các | thành phần | tư bản chủ nghĩa | có | những | bước | phát triển | mạnh | hơn | thời kì | trước | thì | thế lực | của | giai cấp | địa chủ | vẫn | không hề | suy giảm.

Như vậy, còn một số vấn đề khó khăn cần phải tiếp tục nghiên cứu giải quyết:

- Vấn đề giải quyết nhập nhằng phân tách. Cần phải chọn một phương án đúng giữa nhiều phương án. Các hướng tiếp cận khả thi cho vấn đề này có thể là:

  + Dùng các quy tắc ngữ pháp do chuyên gia ngôn ngữ xây dựng. Tiến hành phân tích cú pháp của câu với những phương án tách từ vựng có thể, từ đó loại ra những phương án sai cú pháp.

  + Dùng phương pháp xác suất - thống kê. Phải thống kê trong kho văn bản tương đối lớn của tiếng Việt để tìm ra xác suất của các bộ đôi hay bộ ba

từ loại hoặc từ vựng đi cạnh nhau. Từ đó lựa chọn phương án phân tách có xác suất sai ít nhất.

- Vấn đề giải quyết tên riêng, tên viết tắt và tên có nguồn gốc nước ngoài có mặt trong câu. Hiện tại chương trình phân tách chưa nhận ra được các cụm từ dạng "Nguyễn Văn A" hoặc "ĐT. 8.20.20.20", "1.000$", "0,05%"...

1.3 Phương pháp biểu diễn văn bản

Trước tiên, ta có một số định nghĩa như sau:

- *Từ (Thuật ngữ)*: là một chuỗi các kí tự xuất hiện trong văn bản, mà không phải là dấu câu, con số, từ dừng.

- *Từ đặc trưng:* Sau khi dùng các phương pháp trích chọn thuật ngữ để biểu diễn văn bản, ta thu được một tập các thuật ngữ T' từ tập thuật ngữ ban đầu T ($T' << T$), thì mỗi thuật ngữ trong T' được gọi là từ đặc trưng (dùng để biểu diễn văn bản), hay thuật ngữ đặc trưng.

- *Từ dừng:* từ dừng là từ không mang lại ý nghĩa nội dung cho văn bản, vì nó xuất hiện trong hầu hết các văn bản.

Lưu ý : Các chuỗi tách được trong văn bản có thể là từ theo đúng định nghĩa trong tiếng Việt, nhưng cũng có khi là các ký hiệu viết tắt, các từ phiên âm tên nước ngoài, … Ví dụ: cty (công ty), btc (ban tổ chức), lđbđvn (liên đoàn bóng đá Việt Nam), … Mà các kí hiệu này nhiều khi lại có giá trị thông tin cao để biểu diễn văn bản.

1.3.1 Các kỹ thuật trích chọn đặc trưng của văn bản

1.3.1.1 Loại bỏ các từ dừng

Trong ngôn ngữ tự nhiên, có rất nhiều từ dùng để biểu diễn cấu trúc câu, nhưng hầu như không mang ý nghĩa về mặt thể hiện nội dung của văn bản, ví dụ như các loại từ: các từ quan hệ, kết từ, ... Các loại từ này xuất hiện thường xuyên trong văn bản nhưng không hề mang bất cứ một thông tin nào về nội dung của văn bản, những từ này gọi là từ dừng (stop word). Việc loại bỏ các từ này, đồng nghĩa với việc giảm số chiều của văn bản, tăng độ chính xác và tốc độ xử lý văn bản.

Ví dụ: Một số từ dừng trong tiếng Việt:

Bảng 1.1: Một số từ dừng trong văn bản tiếng Việt

có thể	khi mà	Là	rõ ràng
sau khi	Bởi	Không thể	với
trước khi	vì	thay vì	quả thật
trước hết	nhưng	vì vậy	với lại
tóm lại	nếu	cho nên	tất cả
tất cả	thì	nếu không	hầu hết
phần lớn	do	loại trừ	...
hầu như	vì thế	ngoài ra	
khi đó	cho nên	một số	

Xuất phát từ định nghĩa, *từ dừng là từ không mang lại ý nghĩa nội dung cho văn bản, vì nó xuất hiện trong hầu hết các văn bản.* Chúng ta có thể loại bỏ từ dừng trong văn bản bằng cách đặt ngưỡng để phát hiện từ dừng, ví dụ nếu chúng ta thấy một từ nào đó xuất hiện trong hơn một nửa số văn bản thì có thể coi đó là từ dừng. Tùy thuộc vào từng bài toán cụ thể mà ta đưa ra một ngưỡng phát hiện từ dừng thích hợp.

1.3.1.2 Giảm số chiều

Giảm số chiều thực chất là giảm số thuật ngữ trong tập hợp T, nghĩa là, giảm kích thước của không gian vector từ $|T|$ thành $|T'|<<|T|$.

Giảm số chiều có khuynh hướng làm giảm hiện tượng *overfiting*. Có hai hướng khác nhau trong việc giảm số chiều, phụ thuộc vào nhiệm vụ giảm số chiều là bộ phận hay tổng thể:

- Giảm số chiều bộ phận: Cho một loại c_i, một tập các thuật ngữ , $|T'|<<|T|$, được chọn chỉ để thực hiện phân lớp cho loại c_i.
- Giảm số chiều tổng thể: Một tập các thuật ngữ T', với $|T'|<<|T|$, được chọn để thực hiện phân lớp cho tất cả các loại $C=\{c_1,\ldots,c_{|C|}\}$.

Có rất nhiều phương pháp giảm số chiều. Trong đó phương pháp giảm số chiều dựa trên lý thuyết thông tin được xem là rất tốt cho việc giảm số chiều văn bản. Phương pháp này sử dụng các hàm lựa chọn thuật ngữ dựa trên lý thuyết thông tin

Các hàm lựa chọn thuật ngữ dựa trên lý thuyết thông tin

Kí hiệu: $P(\bar{t}_k, c_i)$ là xác suất cho một tài liệu ngẫu nhiên *x*, thuật ngữ t_k không xảy ra trong *x* và *x* thuộc loại c_i, xác suất này được đánh giá bởi đếm số lần xảy ra trong tập huấn luyện. $P(t_k, c_i)$ là xác xuất chọn ngẫu nhiên một tài liệu thì tài liệu đó có chứa từ t_k, và thuộc lớp c_i. $P(c_i)$ là xác xuất chọn ngẫu nhiên một tài liệu, thì tài liệu này thuộc vào lớp c_i. $P(t_k)$ là xác xuất chọn ngẫu nhiên một tài liệu trong tập dữ liệu thì tài liệu đó có chứa từ t_k. Tất cả các hàm được xác định là cục bộ cho phân lớp xác định c_i.

Trong trường hợp cần đánh giá giá trị của thuật ngữ t_k trong tổng thể tất cả các phân lớp độc lập, thì hàm $f(t_k)$ được tính như sau:

$$f(t_k) = f_{sum}(t_k) = \sum_{i=1}^{|C|} f(t_k, c_i)$$

hoặc $$f(t_k) = f_{wsum}(t_k) = \sum_{i=1}^{|C|} P(c_i) f(t_k, c_i)$$

hoặc $$f(t_k) = f_{\max}(t_k) = \max_{i=}^{|C|} f(t_k, c_i)$$

Trong đó, $f(t_k, c_i)$ là giá trị của từ t_k trong lớp c_i.

Theo phương pháp này người ta dựa vào các hàm tính toán giá trị thông tin của từ t_k đối với phân lớp c_i để quyết định xem có nên lựa chọn từ t_k làm đặc trưng của tài liệu hay không. Chúng ta chỉ giữ lại những từ có hàm giá trị thông tin không thấp hơn ngưỡng đưa ra, nếu hàm giá trị thông tin của t_k thấp hơn ngưỡng đưa ra thì nó sẽ bị loại bỏ. Bảng 1.2 trình bày, một số hàm tính giá trị thông tin của từ t_k đối với lớp c_i.

Bảng 1.2: Một số hàm tính toán giá trị thông tin của từ trong phân lớp

Tên hàm	Kí hiệu	Công thức toán học

Hệ số liên kết DIA (DIA association factor)	$z(t_k, c_i)$	$P(c_i \mid t_k)$
Lợi nhuận thông tin (Information gain)	$IG(t_k, c_i)$	$\sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \log \frac{P(t, c)}{P(t) P(c)}$
Thông tin liên hệ (Mutual information)	$MI(t_k, c_i)$	$\log \frac{P(t_k, c_i)}{P(t_k) P(c_i)}$
Chi bình phương (Chi-square)	$\lambda(t_k, c_i)$	$\frac{\lvert Tr \rvert \cdot [P(t_k, c_i).P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i).P(\bar{t}_k, c_i)]^2}{P(t_k).P(\bar{t}_k).P(c_i).P(\bar{c}_i)}$
Hệ số NGL (NGL coefficient)	$NGL(t_k, c_i)$	$\frac{\sqrt{\lvert Tr \rvert} \cdot [P(t_k, c_i).P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i).P(\bar{t}_k, c_i)]}{\sqrt{P(t_k).P(\bar{t}_k).P(c_i).P(\bar{c}_i)}}$
Điểm tương quan (Relevancy score)	$RS(t_k, c_i)$	$\log \frac{P(t_k \mid c_i) + d}{P(\bar{t}_k \mid \bar{c}_i) + d}$
Hệ số GSS (GSS coefficient)	$GSS(t_k, c_i)$	$P(t_k, c_i).P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i).P(\bar{t}_k, c_i)$

Các hàm trên cho kết quả lựa chọn từ tốt hơn là phương pháp tần xuất tài liệu. Kết quả thực nghiệm của Yang và Pedersen [12] trên nhiều tập mẫu và phân lớp khác nhau cho thấy rằng, các kỹ thuật như: IG hoặc λ^2 có thể giảm không gian từ tới 100 lần mà không ảnh hưởng (hoặc giảm rất ít) hiệu quả phân lớp.

1.3.2 Phương pháp biểu diễn văn bản bằng mô hình không gian vector

Trong phần này chúng tôi xin trình bày phương pháp biểu diễn văn bản theo không gian vector dựa vào tần suất, đây được coi là một hướng tiếp cận tương đối đơn giản và hiệu quả để biểu diễn văn bản. Và cũng là phương pháp được sử dụng để cài đặt thuật toán phân lớp văn bản theo phương pháp Support Vector Machine và các thuật toán cải tiến của nó được trình bày trong phần sau của luận văn.

Trong mô hình tần xuất, ma trận W=$\{w_{ij}\}$ được xác định dựa trên tần số xuất hiện của thuật ngữ t_i trong văn bản d_j hoặc trong toàn bộ tập văn bản D.

1.3.2.1 Phương pháp dựa trên tần số thuật ngữ (TF-Term Frequency)

Các giá trị w_{ij} được xác định dựa trên tần số (hay số lần) xuất hiện của thuật ngữ trong văn bản. Gọi f_{ij} là tần số của thuật ngữ t_i trong văn bản d_j, thì w_{ij} có thể được tính bởi một trong số các công thức sau:

$$w_{ij} = f_{ij} \qquad (1.1)$$

$$w_{ij} = 1+\log(f_{ij}) \qquad (1.2)$$

$$w_{ij} = \sqrt{f_{ij}} \qquad (1.3)$$

Trong phương pháp này, trọng số w_{ij} tỷ lệ thuận với số lần xuất hiện của thuật ngữ t_i trong văn bản d_j, w_{ij} càng lớn thì văn bản d_j càng phụ thuộc nhiều vào thuật ngữ t_i.

Tuy nhiên, cũng có những từ xuất hiện nhiều lần trong văn bản, nhưng nó lại không có ý nghĩa nhiều đối với văn bản như tần số xuất hiện của nó. Phương pháp IDF trình bày sau đây, phần nào khắc phục nhược điểm của phương pháp TF.

1.3.2.2 Phương pháp dựa trên nghịch đảo tần số văn bản (IDF - Inverse Document Frequency)

Với phương pháp này, giá trị w_{ij} được tính như sau:

$$w_{ij} = \begin{cases} \log\frac{m}{h_i} = \log(m) - \log(h_i) & \text{nếu } t_i \text{ xuất hiện trong } d_j \\ 0 & \text{nếu ngược lại} \end{cases} \qquad (1.4)$$

Trong đó, m là số lượng văn bản, h_i là số các văn bản có chứa thuật ngữ t_i

Trọng số w_{ij} trong công thức này cho ta biết độ quan trọng của thuật ngữ t_i trong văn bản d_j. Nếu w_{ij} càng lớn, thì số lượng văn bản chứa t_i càng ít. Điều đó có nghĩa là t_i là thuật ngữ quan trọng để phân biệt văn bản d_j với các văn bản khác trong cơ sở dữ liệu.

1.3.2.3 Phương pháp TF × IDF

Phương pháp này, thực chất là sự kết hợp của phương pháp TF và phương pháp IDF. Theo phương pháp này, trọng số w_{ij} được tính bằng tần số xuất hiện của thuật ngữ t_i trong văn bản d_j và *khả năng xuất hiện* của thuật ngữ t_i trong các văn bản khác, cụ thể:

$$w_{ij} = \begin{cases} 1+\log(f_{ij}) \log\left(\frac{m}{h_i}\right) & \text{nếu } h_i \geq 1 \\ 0 & \text{nếu ngược lại} \end{cases} \qquad (1.5)$$

1.4 Phương pháp đánh giá hiệu quả phân lớp

Giả sử ta qui định như sau:

Tỷ lệ dương đúng TP (True Positive): là số văn bản được gán nhãn là 1 và việc gán nhãn này là đúng.

Tỷ lệ dương sai FP (False Positive): là số văn bản được gán nhãn là 1 nhưng việc gán nhãn này là sai.

Tỷ lệ âm sai FN (False Negative) : là số văn bản được gán nhãn là –1 nhưng việc gán nhãn này là sai.

Tỷ lệ âm đúng TN (True Negative): là số văn bản được gán nhãn là –1 và việc gán nhãn này là đúng.

Bảng 1.3: Định nghĩa các tỷ lệ để đánh giá hiệu quả phân lớp

Phân lớp		Quyết định của chuyên gia	
		YES	NO
Quyết định của hệ thống	YES	**TP**	**FP**
	NO	**FN**	**TN**

Hiệu quả (hay độ chính xác) của hệ thống phân lớp, được đánh giá bởi các công thức sau:

$$\text{Precision} = \frac{TP}{FP + TP} \qquad (1.6)$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \qquad (1.7)$$

$$F - score = \frac{2 * recall * precision}{recall + precision} \qquad (1.8)$$

Chương này đã trình bày và phân tích các bước tiền xử lý văn bản bao gồm: phương pháp tách từ, phương pháp lựa chọn đặc trưng và biểu diễn văn bản. Kết quả của các bước này là mỗi văn bản d_i đã được biểu diễn bằng một vector $x_i=(w_{i1}, w_{i2}, ..., w_{in})$, w_{ij} là trọng số của từ t_j trong văn bản d_i. Quá trình phân lớp văn bản không thực hiện trực tiếp với các văn bản d_i mà thực hiện trên các vector x_i.

Đồng thời chương này cũng trình bày sơ lược phương pháp đánh giá hiệu quả phân lớp. Tiếp theo, chúng ta sẽ nghiên cứu một số phương pháp phân lớp văn bản phổ biến.

CHƯƠNG 2: CÁC PHƯƠNG PHÁP PHÂN LỚP VĂN BẢN PHỔ BIẾN

Trong chương này chúng ta sẽ nghiên cứu các thuật toán phân lớp phổ biến hiện nay. Nhiều thực nghiệm cho thấy các phương pháp như: K-trung bình (K-means), cây quyết định (Decision tree), K-láng giềng gần nhất (K-nearest neighbors), phương pháp sử dụng các vector hỗ trợ SVM (Support Vector Machines) là những phương pháp có hiệu quả phân lớp tương đối tốt và thường được sử dụng.

2.1 Thuật toán K-Trung bình (K-means)

Ý tưởng

Ý tưởng của thuật toán là chia m phần tử (m mẫu dữ liệu văn bản) thành n nhóm (n lớp) sao cho các phần tử trong cùng một nhóm sẽ gần tâm của nhóm đó nhất.

Sau đây ta sẽ tìm hiểu thuật toán K-Trung bình (K-means) cổ điển và một cải tiến của nó đó là thuật toán K-Trung bình mờ (Fuzzy K-means).

2.1.1 Thuật toán K –Trung bình cổ điển

Giả sử ta muốn gom m mẫu dữ liệu đầu vào có các vector đặc trưng lần lượt là $x_1, x_2, \dots x_m$ vào k nhóm (k<m). Trong đó V_i là vector đặc trưng n chiều của mẫu thứ i.

Đầu tiên ta khởi tạo các giá trị trung bình (hay có thể gọi là tâm) của k nhóm là các vector n chiều $C_1, C_2, \dots, C_k$ (thường là khởi tạo ngẫu nhiên). Sau đó tiến hành tính khoảng cách từ các mẫu đầu vào x_i đối với từng tâm nhóm C_j, hay chính là việc xác định mối quan hệ thành viên của từng mẫu đầu vào x_i bằng cách tính $||x_i - C_j||$. Đối với mỗi mẫu x_i, khoảng cách tối thiểu sẽ xác định được mối quan hệ thành viên đối với nhóm tương ứng.

Thuật toán:

- Bước 1: Khởi tạo tâm của k nhóm: $C_1, C_2, \dots, C_k$
- Bước 2: Lặp lại:

(a) Phân loại m mẫu x_i vào các nhóm có tâm là C_j sao cho khoảng cách $||x_i – C_j||$ là nhỏ nhất.

(b) Tính toán lại tâm C_j (chỉnh tâm)

Đến khi: các tâm C_j không đổi.

Kết quả: Các lớp C_1, C_2, ….,C_k

Đối với thuật toán K-Trung bình cổ điển, việc khởi tạo tâm các nhóm có ảnh hưởng rất lớn đến kết quả phân nhóm.

2.1.2 Thuật toán K-Trung bình mờ

Đây là một cải tiến của thuật toán K-Trung bình cổ điển. Trong mỗi vòng lặp của K-Trung bình cổ điển, giả sử mỗi vector đặc trưng thuộc chính xác một nhóm. Chúng ta giảm nhẹ điều này và giả sử rằng mỗi mẫu x_i có vài mức độ quan hệ thành viên mờ trong nhóm C_j.

Xác xuất của quan hệ thành viên nhóm cho mỗi điểm được chuẩn hoá như sau:

$$\sum_{i=1}^{k} P(\omega_i|x_j) = 1, \text{ với } j = 1,\ldots,n$$

Mỗi C_j được tính lại như sau:

$$C_j = \left(\sum_{j=1}^{n} P(\omega_i|x_j)^b x_j\right) / \left(\sum_{i=1}^{n} P(\omega_i|xj)^b\right)$$

Và mỗi $P(\omega_i|x_j)$ được tính lại như sau:

$$P(\omega_i|x_j) = (1/d_{ij})^{1/(b-1)} / \left(\sum_{r=1}^{c} (1/d_{rj})^{1/(b-1)}\right), \text{ với } dij = \|xj - Ci\|^2$$

Thuật toán:

- Bước 1: Khởi tạo:
  - Các tâm C_1 ,C_2 ,…C_k
  - $P(\omega_i|x_j)$, với i=1, …,k và j=1, …,n
- Bước 2: Chuẩn hoá xác suất quan hệ thành viên nhóm.
- Bước 3: Lặp lại:

(a) Phân n lớp mẫu theo phương pháp người láng giềng gần nhất C_i;

(b) Tính toán lại C_i

(c) Tính toán lại $P(\omega_i|x_j)$.

Đến khi: Không thay đổi trong C_i và $P(\omega_i|x_j)$

Kết quả: các lớp C_1 ,C_2 ,…C_k

Thuật toán K-Trung bình mờ cải tiến sự hội tụ của thuật toán K-Trung bình. Tuy nhiên, phương pháp này vẫn còn hạn chế là việc đặc tả không chính xác số nhóm.

2.2 Thuật toán cây quyết định (Decision tree) [3]

Phương pháp cây quyết định được Mitchell đưa ra vào năm 1996. Nó được sử dụng rộng rãi nhất cho việc học quy nạp từ tập mẫu lớn. Đây là phương pháp học xấp xỉ các hàm mục tiêu có giá trị rời rạc. Mặt khác cây quyết định còn có thể chuyển sang dạng biểu diễn tương đương dưới dạng cơ sở tri thức là các luật *Nếu – Thì.*

Ý tưởng

Bộ phân lớp cây quyết định là một dạng cây mà mỗi nút được gán nhãn là một đặc trưng, mỗi nhánh là giá trị trọng số xuất hiện của đặc trưng trong văn bản cần phân lớp, và mỗi lá là nhãn của phân lớp tài liệu. Việc phân lớp của một tài liệu d_j sẽ được duyệt đệ quy theo trọng số của những đặc trưng có xuất hiện trong văn bản d_j. Thuật toán lặp đệ quy đến khi đạt đến nút lá và nhãn của d_j chính là nhãn của nút lá tìm được. Thông thường việc phân lớp văn bản nhị phân sẽ tương thích với việc dùng cây nhị phân.

Cách thực hiện

Cây quyết định này được tổ chức như sau: Các nút trong được gán nhãn bởi các thuật ngữ, nhãn của các cung tương ứng với trọng số của thuật ngữ trong tài liệu mẫu, nhãn của các lá tương ứng với nhãn của các lớp. Cho một tài liệu d_j, ta sẽ thực hiện so sánh các nhãn của cung xuất phát từ một nút trong (tương ứng với một thuật ngữ nào đó) với trọng số của thuật ngữ này trong d_j, để quyết định nút trong nào sẽ

được duyệt tiếp. Quá trình này được lặp từ nút gốc của cây, cho tới khi nút được duyệt là một lá của cây. Kết thúc quá trình này, nhãn của nút lá sẽ là nhãn của lớp được gán cho văn bản.

Với phương pháp này, phần lớn người ta thường chọn phương pháp nhị phân để biểu diễn văn bản, cũng như cây quyết định.

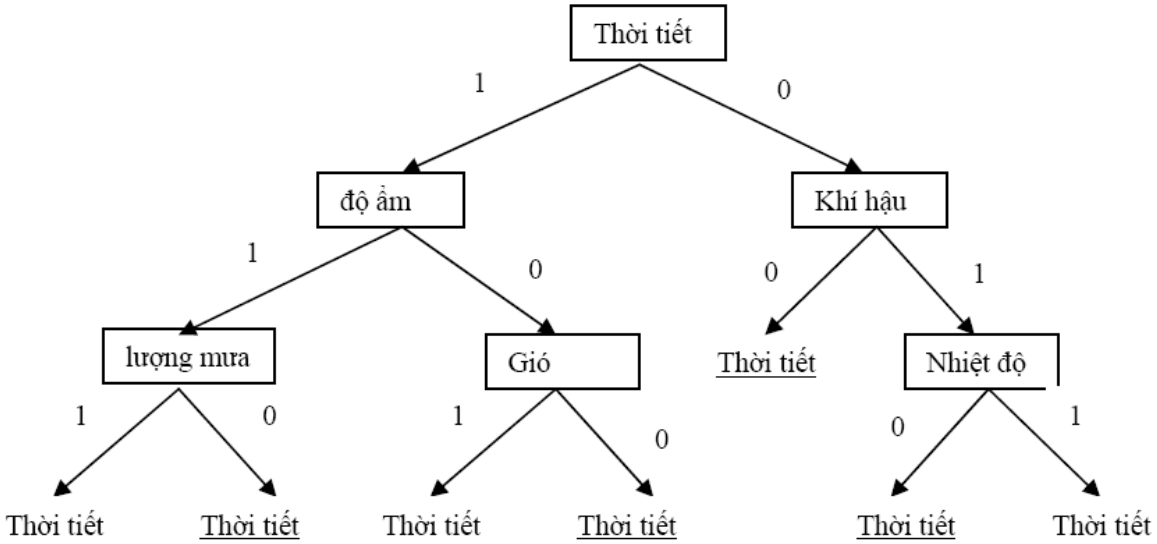
Ví dụ

Ta có bảng dữ liệu gồm 10 tài liệu được mô tả bằng vector nhị phân thông qua 7 thuật ngữ "*thời tiết*", "*độ ẩm*", "*lượng mưa*", "*gió*", "*khí hậu*", "*thuyền*", "*nhiệt độ*". Trong đó cột cuối cùng trong bảng là nhãn được gán cho từng tài liệu với chủ đề ***thời tiết***, giá trị của tài liệu d_i trong cột này bằng 1 tương ứng d_i thuộc chủ đề thời tiết, nếu giá trị này bằng 0 thì d_i không thuộc chủ đề thời tiết.

Bảng 2.1: Biểu diễn văn bản bằng vector nhị phân

Tài liệu	*thời tiết*	*độ ẩm*	*lượng mưa*	*gió*	*khí hậu*	*thuyền*	*nhiệt độ*	***thời tiết***
d_1	1	1	1	0	0	0	0	1
d_2	1	1	0	0	0	1	0	0
d_3	1	1	1	0	0	0	1	1
d_4	1	1	1	0	0	0	0	1
d_5	1	0	0	1	0	0	0	1
d_6	1	0	0	1	1	1	0	1
d_7	1	0	0	0	0	1	0	0
d_8	0	1	0	0	0	1	0	0
d_9	0	0	0	0	1	0	1	1
d_{10}	0	0	0	0	1	0	0	0

Cây quyết định được xây dựng tương ứng với bảng 2.1 là:



Hình 2.1: Xây dựng cây quyết định cho tập mẫu dùng để huấn luyện

Từ cây quyết định trên ta xây dựng được cơ sở tri thức dưới dạng luật *Nếu - Thì* như sau:

Nếu (*thời tiết*=1) và (*lượng mưa*=1) và (*độ ẩm*=1) ***Thì*** class *thời tiết*=1

Nếu (*thời tiết*=1) và (*lượng mưa*=0) và (*độ ẩm*=1) ***Thì*** class *thời tiết*=0

Nếu (*thời tiết*=1) và (*gió*=0) và (*độ ẩm*=0) ***Thì*** class *thời tiết*=0

Nếu (*thời tiết*=1) và (*gió*=1) và (*độ ẩm*=0) ***Thì*** class *thời tiết*=1

Nếu (*thời tiết*=0) và (*khí hậu*=0) ***Thì*** class *thời tiết*=0

Nếu (*thời tiết*=0) và (*khí hậu*=1) và (*nhiệt độ*=0) ***Thì*** class *thời tiết*=0

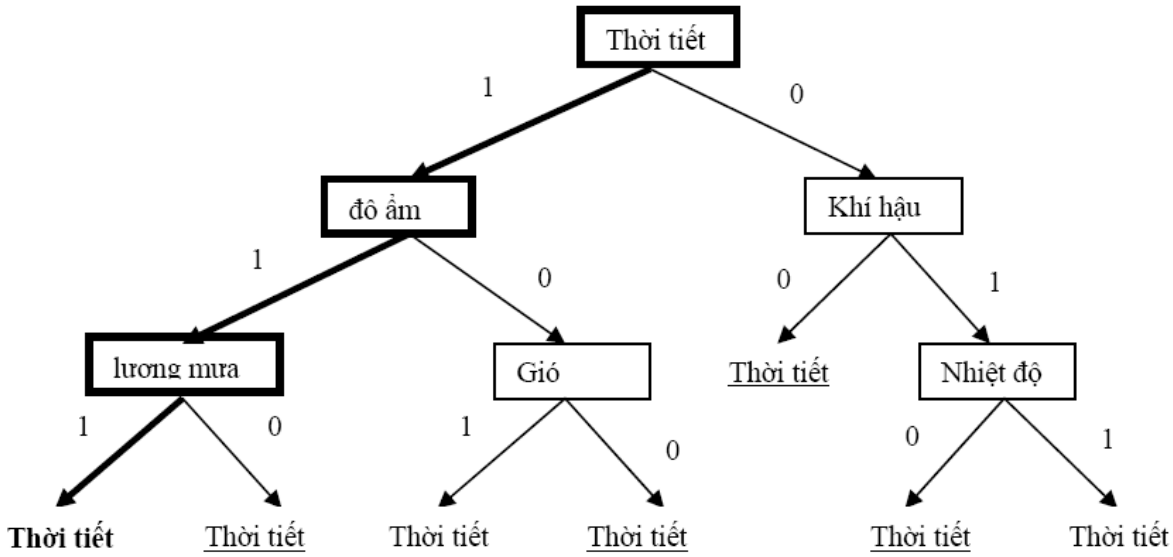
Nếu (*thời tiết*=0) và (*khí hậu*=1) và (*nhiệt độ*=1) ***Thì*** class *thời tiết*=1

Xét tài liệu d, được biểu diễn bởi vector nhị phân như sau:

d = (*thời tiết, lượng mưa, độ ẩm, gió, khí hậu, thuyền, nhiệt độ*)

=(1, 1, 1, 0, 0, 1, 0)

Quá trình tìm kiếm lời giải trên cây quyết định sẽ như sau:



Hình 2.2: Quá trình tìm kiếm lời giải trên cây quyết định

Class *thời tiết*=1, hay nói cách khác văn bản d thuộc lớp văn bản nói về chủ đề thời tiết (lớp thời tiết).

Các thuật toán cây quyết định ngày càng được phát triển và cải tiến. Nhưng hầu hết các thuật toán này đều dựa vào cách tiếp cận từ trên xuống và chiến lược tìm kiếm tham lam trong không gian tìm kiếm của cây quyết định. Trong số các thuật toán học cây quyết định thì thuật toán ID3 và cải tiến của nó là C4.5 được đánh giá là hiệu quả và được sử dụng phổ biến nhất.

2.3 K-láng giềng gần nhất (K-Nearest Neighbor) [3]

Ý tưởng

Ý tưởng chính của thuật toán K-láng giềng gần nhất (K-NN) là *so sánh độ phù hợp của văn bản d với từng nhóm chủ đề, dựa trên k văn bản mẫu trong tập huấn luyện mà có độ tương tự với văn bản d là lớn nhất.*

Có 2 vấn đề cần quan tâm khi phân lớp văn bản bằng thuật toán K- láng giềng gần nhất là xác định khái niệm *gần,* công thức để tính mức độ gần; và làm thế nào để tìm được nhóm văn bản phù hợp nhất với văn bản đó (nói cách khác là tìm được chủ đề thích hợp để gán cho văn bản).

Khái niệm gần ở đây được hiểu là độ tương tự giữa các văn bản. Có nhiều cách để xác định độ tương tự giữa hai văn bản, trong đó công thức Cosine trọng số được coi là hiệu quả để đánh giá độ tương tự giữa hai văn bản. Cho $T=\{t_1, t_2, ..., t_n\}$

là tập hợp các thuật ngữ; $W=\{w_{t1}, w_{t2}, ..., w_{tn}\}$ là vector trọng số, w_{ti} là trọng số của thuật ngữ t_i. Xét hai văn bản $X=\{x_1, x_2, \ldots, x_n\}$ và $Y=\{y_1, y_2, \ldots, y_n\}$, x_i, y_i lần lượt là tần số xuất hiện của thuật ngữ t_i trong văn bản X, Y. Khi đó độ tương tự giữa hai văn bản X và Y được tính theo công thức sau:

$$Sim(X,Y) = \ \cos ine(X,Y,W) = \frac{\sum_{t\in T}(x_t \times w_t) \times (y_t \times w_t)}{\sqrt{\sum_{t\in T}(x_t \times w_t)^2}\sqrt{\sum_{t\in T}(y_t \times w_t)^2}}$$

Trong vector X, Y các thành phần x_i, y_i được chuẩn hoá theo tần số xuất hiện (TF – xem công thức 1.1, 1.2, 1.3) của thuật ngữ t_i trong các văn bản X và Y. Vector W được xác định bằng tay hoặc được tính theo một thuật toán tham lam nào đó. Một đề xuất đưa ra là tính vector W theo nghịch đảo tần suất văn bản IDF (xem công thức 1.4), khi đó văn bản được biểu diễn dưới dạng vector tần xuất TFxIDF (xem công thức 1.5).

Có nhiều đề xuất để tìm nhãn phù hợp gán cho văn bản khi đã tìm được k văn bản gần nhất. Sau đây chúng tôi sẽ trình bày ba cách được sử dụng nhiều nhất:

2.3.1 Gán nhãn văn bản gần nhất

Theo phương pháp này, văn bản d_i được gán cho nhóm chủ đề có chứa văn bản (trong số k văn bản) có độ tương tự cao nhất với d_i. Giải pháp này tương đối đơn giản. Tuy nhiên, nó không được đánh giá cao, vì nó sẽ dẫn đến kết quả sai khi tập mẫu có nhiễu, mặt khác kết quả đưa ra của phương pháp này không mang tính tổng hợp.

Giả sử với k=8, chúng ta tìm được 8 văn bản trong tập huấn luyện gần nhất với văn bản d, độ tương tự của các văn bản này với văn bản d được trình bày trong bảng 2.2:

Bảng 2.2: Ví dụ 1 về độ tương tự giữa văn bản và chủ đề

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
Độ tương tự	0.9	0.5	0.89	0.8	0.75	0.6	0.5	0.4
Chủ đề	chủ đề 1	chủ đề 1	chủ đề 2	chủ đề 2	chủ đề 2	chủ đề 3	chủ đề 3	chủ đề 3

Theo cách gán nhãn này ta sẽ gán nhãn *chủ đề 1* cho văn bản d. Vì văn bản gần với d nhất là văn bản d_1, và văn bản d_1 lại thuộc *chủ đề 1*. Tuy nhiên, trong trường hợp dữ liệu huấn luyện có nhiễu (trong chủ đề 1 chỉ có một văn bản có độ tương tự với d là lớn, các văn bản còn lại của chủ đề 1 có độ tương tự với d là thấp) thì kết quả phân lớp dựa theo cách này không tốt.

2.3.2 Gán nhãn theo số đông

Để dễ hiểu, ta xét ví dụ sau: Giả sử với k=8, chúng ta tìm được 8 văn bản gần nhất với văn bản d, độ tương tự của các văn bản này với văn bản d được trình bày trong bảng sau:

Bảng 2.3: Ví dụ 2 về độ tương tự giữa văn bản và chủ đề

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
Độ tương tự	0.9	0.85	0.89	0.8	0.75	0.6	0.5	0.4
Chủ đề	chủ đề 1	chủ đề 1	chủ đề 2	chủ đề 2	chủ đề 2	chủ đề 3	chủ đề 3	chủ đề 3

Theo phương pháp này, chủ đề 2 sẽ được gán cho văn bản d vì có 3 văn bản thuộc chủ đề này. Dù có 3 văn bản thuộc chủ đề 3 nhưng ta không gán chủ đề 3 cho văn bản d vì độ tương tự của các văn bản d_6, d_7, d_8 với văn bản d quá thấp. Và ta cũng không gán chủ đề 1 cho văn bản d, vì chủ đề 1 chỉ có 2 văn bản. Mặc dù đã khắc phục được phần nào nhược điểm của phương pháp ***gán nhãn văn bản gần nhất***, nhưng nó vẫn chưa được đánh giá cao, xét ví dụ sau:

Bảng 2.4: Ví dụ 3 về độ tương tự giữa văn bản và chủ đề

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
Độ tương tự	0.9	0.85	0.89	0.5	0.45	0.4	0.35	0.8
Chủ đề	chủ đề 1	chủ đề 1	chủ đề 1	chủ đề 2	chủ đề 2	chủ đề 2	chủ đề 2	chủ đề 3

Trong trường hợp này ta nên gán chọn chủ đề 1, mà không nên chọn chủ đề 2 để gán cho văn bản d vì độ tương tự của văn bản d với các văn bản thuộc chủ đề 2 quá thấp.

2.3.3 Gán nhãn theo độ phù hợp của chủ đề

Độ phù hợp giữa một văn bản d và chủ đề c được tính theo công thức sau:

$$sim(d,c) = \sum_{d_i \in c} sim(d,d_i)$$

Theo phương pháp này, ta lần lượt tính độ phù hợp của văn bản d với từng chủ đề từ k văn bản đã lấy ra, sau đó chọn chủ đề có độ phù hợp cao nhất gán cho văn bản d.

Theo phương pháp này, với dữ liệu trong bảng 2.4 thì chủ đề 1 sẽ được chọn để gán nhãn cho văn bản d.

Nhận xét thuật toán K-NN:

- Thuật toán K-NN là dễ hiểu và đơn giản.

- Không cho kết quả phân lớp văn bản tốt trong trường hợp dữ liệu huấn luyện có nhiễu. Mỗi cách để tính độ tương tự của văn bản d với chủ đề c đều có những nhược điểm mà ta không thể khắc phục được một cách trọn vẹn. Chẳng hạn với phương pháp *gán nhãn theo độ phù hợp của chủ đề* như đã trình bày ở trên, mặc dù đã có thể khắc phục được nhược điểm của các phương pháp *gán nhãn văn bản gần nhất* và *gán nhãn theo số đông*, nhưng vẫn không thể cho kết quả chính xác trong trường hợp mẫu huấn luyện có nhiễu.

Ví dụ: Một văn bản d, có độ tương tự với các chủ đề như sau:

Bảng 2.5: Ví dụ 4 về độ tương tự giữa văn bản và chủ đề

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
Độ tương tự	0.98	0.43	0.4	0.7	0.6	0.5	0.6	0.5
Chủ đề	chủ đề 1	chủ đề 1	chủ đề 1	chủ đề 2	chủ đề 2	chủ đề 2	chủ đề 3	chủ đề 3

Từ bảng trên chúng ta có độ tương tự giữa d với các chủ đề là:

Sim(d,chủ đề 1)=0.98+0.43+0.4=1.81

Sim(d,chủ đề 2)=0.7+0.6+0.5=1.8

Sim(d,chủ đề 3)=0.6+0.5=1.1

Theo phương pháp *gán nhãn theo độ phù hợp của chủ đề* thì d được gán nhãn cho chủ đề 1, nhưng bằng trực quan chúng ta thấy d gần với chủ đề 2 hơn là chủ đề 1.

2.4 Support Vector Machines (SVM)

2.4.1 Giới thiệu

SVM là một phương pháp phân lớp xuất phát từ lý thuyết học thống kê, dựa trên nguyên tắc tối thiểu rủi ro cấu trúc (Structural Risk Minimisation). SVM sẽ cố gắng tìm cách phân lớp dữ liệu sao cho có lỗi xảy ra trên tập kiểm tra là nhỏ nhất (Test Error Minimisation). Ý tưởng của nó là ánh xạ (tuyến tính hoặc phi tuyến) dữ liệu vào không gian các vector đặc trưng (space of feature vectors) mà ở đó một siêu phẳng tối ưu được tìm ra để tách dữ liệu thuộc hai lớp khác nhau.

SVM đã được ứng dụng rất nhiều trong việc nhận dạng mẫu như nhận dạng chữ viết tay, nhận dạng đối tượng, nhận dạng khuôn mặt trong ảnh, và phân lớp văn bản.

Kết quả so sánh các phương pháp phân lớp văn bản khác nhau của Thorsten Joachims như trong bảng 2.6 cho thấy sự chính xác vượt trội của SVM trong ứng dụng phân lớp văn bản. Kết quả thử nghiệm trên tập dữ liệu Reuters-21578 gồm 9603 văn bản huấn luyện 3299 văn bản kiểm tra, thuộc 135 chủ đề khác nhau (Độ chính xác được tính theo tỉ lệ Precision/recall) [10]

Bảng 2.6: Kết quả so sánh phương pháp phân lớp sử dụng SVM với K-NN

Lớp văn bản	K-NN	SVM(đa thức) d=				SVM (rbf) γ=			
		1	2	3	4	0.6	0.8	1.0	1.2
earn	97.3	98.2	98.4	98.5	98.4	98.5	98.5	98.4	98.3
acq	92.0	92.6	94.6	95.2	95.2	95.0	95.3	95.3	95.4
Money-fx	78.2	66.9	72.5	75.4	74.9	74.0	75.4	76.3	75.9

Grain	82.2	91.3	93.1	92.4	91.3	93.1	91.9	91.9	90.6
Crude	85.7	86.0	87.3	88.6	88.9	88.9	89.0	88.9	88.2
Trade	77.4	69.2	75.5	76.6	77.3	76.9	78.0	77.8	76.8
Interest	74.0	69.8	63.3	67.9	73.1	74.4	75.0	76.2	76.1
Ship	79.2	82.0	85.4	86.0	86.5	85.4	86.5	87.6	87.1
Wheat	76.6	83.1	84.5	85.2	85.9	85.2	85.9	85.9	85.9
Corn	77.9	86.0	86.5	85.3	85.7	85.1	85.7	85.7	84.5
Độ chính xác trung bình	**82.3**	**84.2**	**85.1**	**85.9**	**86.2**	**86.4**	**86.5**	**86.3**	**86.2**
		Trung bình: **86.0**				Trung bình: **86.4**			

Từ kết quả bảng 2.6 ta thấy phương pháp SVM rất thích hợp cho phân lớp văn bản. Sau đây chúng ta sẽ tìm hiểu về bài toán phân lớp văn bản sử dụng phương pháp SVM.

2.4.2 Bài toán và cách giải quyết [1],[6],[11]

Bài toán

Chúng ta hãy xem xét một bài toán phân lớp văn bản bằng phương pháp Support Vector Machines như sau:

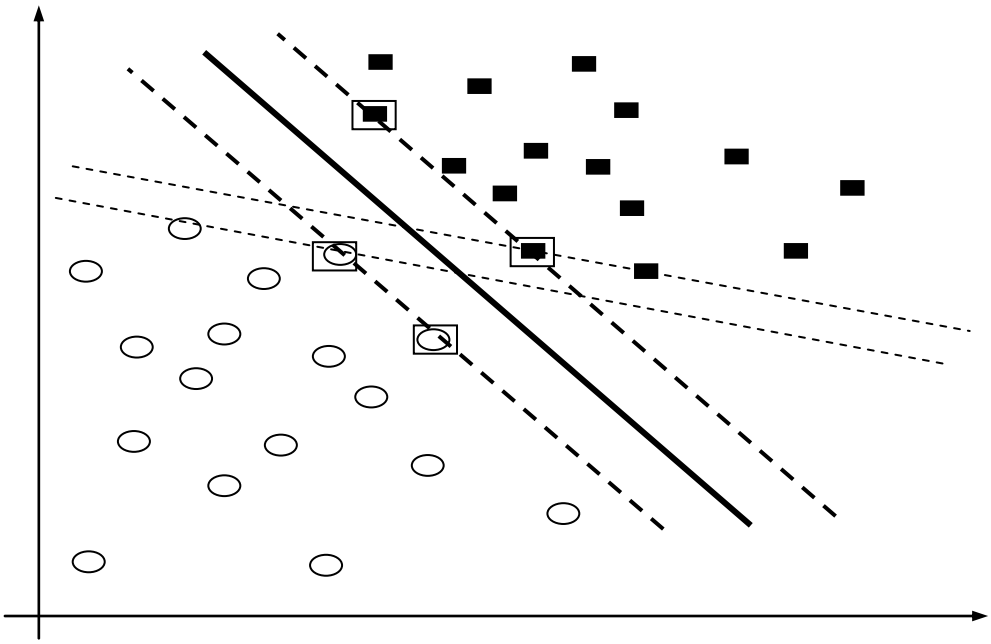
Kiểm tra xem một tài liệu bất kỳ d thuộc hay không thuộc một phân lớp c cho trước? Nếu d$\in$c thì d được gán nhãn là *1*, ngược lại thì d được gán nhãn là *–1*.

Cách giải quyết

Giả sử, chúng ta lựa chọn được tập các đặc trưng là $T=\{t_1, t_2, ..., t_n\}$, thì mỗi văn bản d_i sẽ được biểu diễn bằng một vector dữ liệu $x_i=(w_{i1}, w_{i2}, ..., w_{in})$, $w_{ij} \in R$ là trọng số của từ t_j trong văn bản d_i. Như vậy, tọa độ của mỗi vector dữ liệu x_i tương ứng với tọa độ của một điểm trong không gian R^n. Quá trình phân lớp văn bản sẽ thực hiện xử lý trên các vector dữ liệu x_i chứ không phải là các văn bản d_i. Bởi vậy, trong phần này chúng tôi sẽ sử dụng đồng nhất các thuật ngữ : văn bản, vector dữ liệu, điểm dữ liệu.

Dữ liệu huấn luyện của SVM là tập các văn bản đã được gán nhãn trước $Tr=\{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\}$, trong đó, x_i là vector dữ liệu biểu diễn văn bản d_i

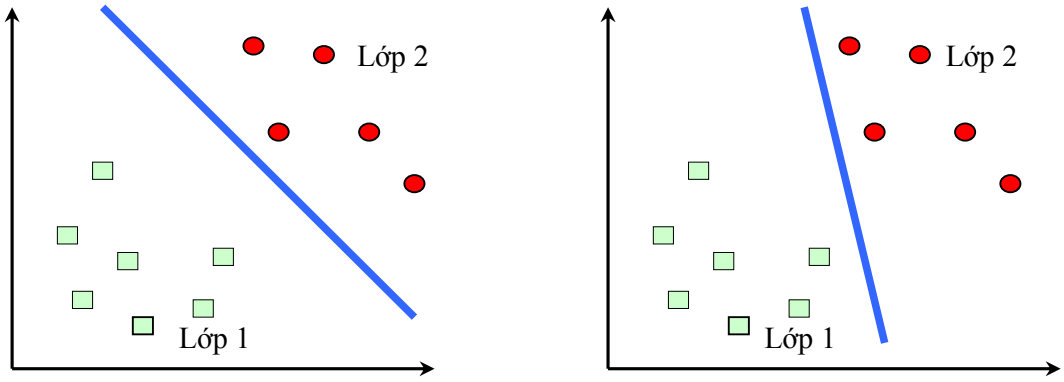
($x_i \in R^n$), $y_i \in \{+1, -1\}$, cặp *(x_i, y_i)* được hiểu là vector x_i (hay văn bản d_i) được gán nhãn là y_i. Nếu coi mỗi văn bản d_i được biểu diễn tương ứng với một điểm dữ liệu trong không gian R^n thì *ý tưởng của SVM là tìm một mặt hình học (siêu phẳng) f(x) "tốt nhất" trong không gian n-chiều để phân chia dữ liệu sao cho tất cả các điểm x_+ được gán nhãn 1 thuộc về phía dương của siêu phẳng ($f(x_+)>0$), các điểm x_- được gán nhãn –1 thuộc về phía âm của siêu phẳng ($f(x_-)<0$).* Với bài toán phân lớp SVM, một siêu phẳng phân chia dữ liệu được gọi là "tốt nhất", nếu khoảng cách từ điểm dữ liệu gần nhất đến siêu phẳng là lớn nhất. Khi đó, việc xác định một tài liệu $x \notin Tr$ có thuộc phân lớp c hay không, tương ứng với việc xét dấu của $f(x)$, nếu $f(x)>0$ thì $x \in c$, nếu $f(x) \leq 0$ thì $x \notin c$.



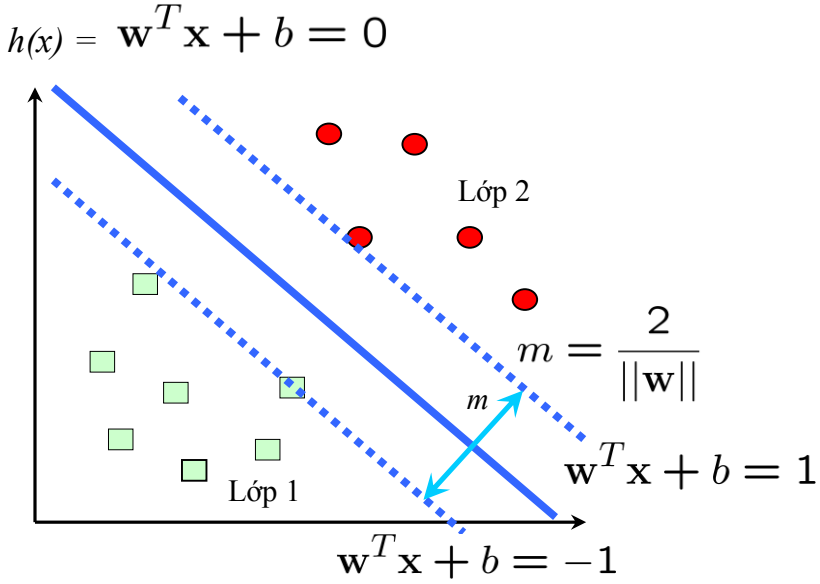
Hình 2.3: Siêu phẳng phân chia tập mẫu huấn luyện

Trong hình 2.3, đường tô đậm là siêu phẳng tốt nhất và các điểm được bao bởi hình chữ nhật là những điểm gần siêu phẳng nhất, chúng được gọi là các vector hỗ trợ (support vector). Các đường nét đứt mà các support vector nằm trên đó được gọi là lề (margin).

Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm gần nhất của mỗi lớp đến mặt phẳng này. Khoảng cách biên càng lớn thì siêu phẳng quyết định càng tốt và việc phân lớp càng chính xác. Mục đích của SVM là tìm được khoảng cách (biên) lớn nhất và lỗi tách sai là bé nhất.



Hình 2.4: Ví dụ về biên không tốt.



Hình 2.5: Ví dụ về biên tối ưu.

Từ đó bài toán đặt ra là tìm siêu phẳng tách $w^T.x+b=$. Đây cũng là bài toán chính của SVM.

Cho tập dữ liệu

$$Tr = \{(x_1, y_1),...,(x_l, y_l)\} \qquad x_i \in R^n, y_i \in \{-1\} \qquad (2.1)$$

Trường hợp 1

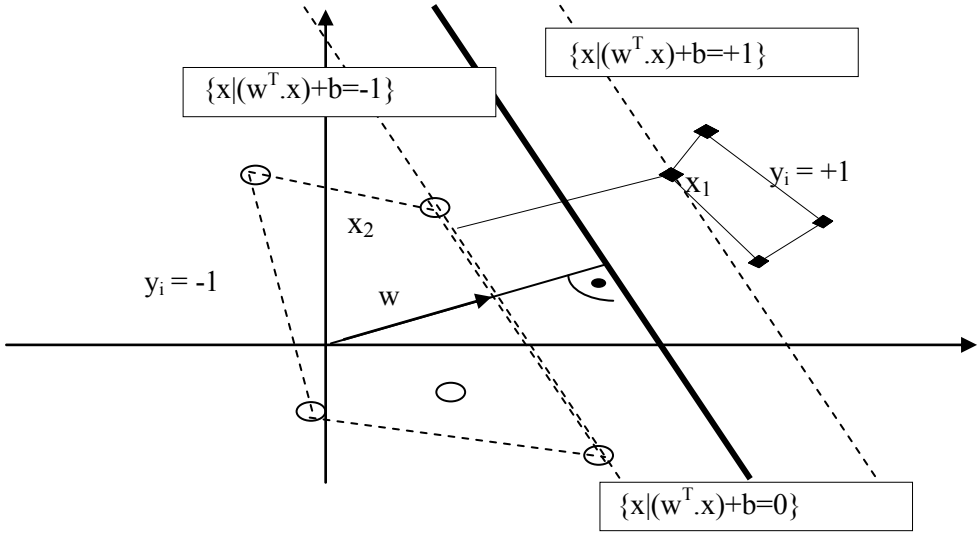
Tập dữ liệu Tr có thể phân chia tuyến tính được mà không có nhiễu (nghĩa là tất cả các điểm được gán nhãn 1 thuộc về phía dương của siêu phẳng, tất cả các điểm được gán nhãn –1 thuộc về phía âm của siêu phẳng) thì chúng ta có thể tìm được một siêu phẳng tuyến tính có dạng (2.2) để phân chia tập dữ liệu này:

$w^T.x+b=$ (2.2)

Trong đó: $w \in R^n$ là vector trọng số (weight vector).

$b \in R$ là hệ số tự do.

sao cho $f(x_i)=sign\{w^T x_i+b\}=$ ⎧+ nếu $y_i=$ ⎩− nếu $y_i=$ $\forall$,y_i) ∈ r (2.3)



Hình 2.6: Siêu phẳng phân chia dữ liệu và các ràng buộc

Giả sử rằng siêu phẳng phân chia dữ liệu (2.6) với các ràng buộc*:*

$\min_i \left|w^T.x_i+b\right|=$ *i=1, .., l* (2.4)

hay y_i $w^T.x_i+$ ≥ , i= , ..., *l* (2.5)

Vấn đề đặt ra bây giờ là xác định các hệ số *w* và *b* như thế nào để siêu phẳng tìm được là tốt nhất? Siêu phẳng tốt nhất là siêu phẳng mà có khoảng cách từ điểm dữ liệu huấn luyện gần nhất đến siêu phẳng là xa nhất. Mà khoảng cách từ một điểm dữ liệu x_i đến siêu phẳng (2.2) là:

$$d(w,b;x_i)=\frac{\left|w^T.x_i+\right|}{\|w\|} \qquad (2.6)$$

$\left|w^T.x_i+\right|$: là giá trị tuyệt đối của biểu thức $w^T.x_i+b$

$\|w\|$: là độ dài Ơcơlit của vector *w*

Giả sử *h(w,b)* là tổng của khoảng cách từ điểm dữ liệu gần nhất của lớp 1 đến siêu phẳng và khoảng cách từ điểm dữ liệu gần nhất của lớp –1 đến siêu phẳng. Ta có:

$$\begin{aligned} h(w,b) &= \min_{x_i, y_i=} d(w,b;x_i) + \min_{x_i, y_i=} d(w,b;x_i) \\ &= \min_{x_i, y_i=} \frac{\left| w^T . x_i + \right|}{\|w\|} + \min_{x_i, y_i=} \frac{\left| w^T . x_i + \right|}{\|w\|} \\ &= \frac{1}{\|w\|} \left(\min_{i, y_i=} \left| w^T . x_i + \right| + \min_{x_i, y_i=} \left| w^T . x_i + \right| \right) \\ &= \frac{2}{\|w\|} \end{aligned} \qquad (2.7)$$

Như vậy, siêu phẳng tối ưu là siêu phẳng có $h(w,b) = 1/\|w\|$ lớn nhất, tương đương với $\|w\|$ là nhỏ nhất.

Tóm lại, việc tìm siêu phẳng tốt nhất tương đương với việc giải bài toán tối ưu sau:

$$\begin{cases} \underset{w}{Min}\ \Phi(w) = \frac{1}{2}\|w\|^2 \\ y_i(w^T . x_i + b) \geq 1, \qquad i = 1, \ldots, l \end{cases} \qquad (2.8)$$

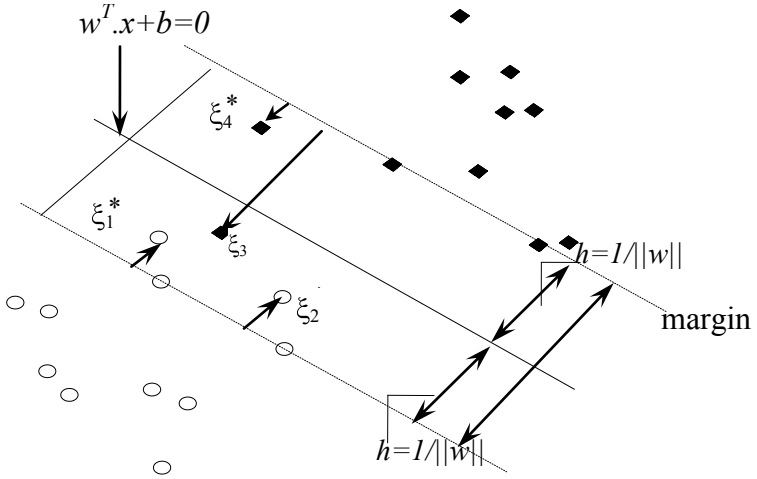
Trường hợp 2

Tập dữ liệu huấn luyện *Tr* có thể phân chia được tuyến tính nhưng có nhiễu (Hình 2.7). Trong trường hợp này, hầu hết các điểm trong tập dữ liệu được phân chia bởi siêu phẳng tuyến tính. Tuy nhiên có một số ít điểm bị nhiễu, nghĩa là điểm có nhãn dương nhưng lại thuộc về phía âm của siêu phẳng, điểm có nhãn âm thuộc về phía dương của siêu phẳng.

Trong trường hợp này, chúng ta thay ràng buộc $y_i(w^T . x_i + b) \geq$ bằng ràng buộc (2.9) .

$$y_i(w^T . x_i +) \geq -\xi \qquad i = , \ldots, l \qquad (2.9)$$

Ở đây, ξ_i gọi là các biến lới lỏng (slack variable) $\xi_i \geq 0$.



Hình 2.7: Trường hợp dữ liệu có nhiễu

Bài toán 2.8 trở thành

$$\begin{cases} \text{Min } \Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_i(w^T.x_i + b) \geq 1 - \xi_i \qquad i = 1,\ldots,l \\ \xi_i \geq 0 \qquad i = 1,\ldots,l \end{cases} \tag{2.10}$$

C là tham số xác định trước, định nghĩa giá trị ràng buộc, C càng lớn thì mức độ vi phạm đối với những lỗi thực nghiệm càng cao.

Trường hợp 3

Tuy nhiên không phải tập dữ liệu nào cũng có thể phân chia tuyến tính được. Trong trường hợp này, chúng ta sẽ ánh xạ các vector dữ liệu *x* từ không gian n-chiều vào một không gian m-chiều (m>n) , sao cho trong không gian m-chiều này tập dữ liệu có thể phân chia tuyến tính được. Giả sử ϕ là một ánh xạ phi tuyến tính từ không gian R^n vào không gian R^m.

$$\phi \quad R^n \rightarrow \quad ^n$$

Khi đó, vector x_i trong không gian R^n sẽ tương ứng với vector $\phi(x_i)$ trong không gian R^m.

Thay $\phi(x_i)$ vào (2.10) ta có (2.11):

$$\begin{cases} \text{Min } \Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_i(w^T.\phi(x_i)+b) \geq 1-\xi_i, & i=1,\ldots,l \\ \xi_i \geq 0 & i=1,\ldots,l \end{cases} \qquad (2.11)$$

2.4.3 Hàm nhân Kernel

Việc tính toán trực tiếp $\phi(x_i)$ là phức tạp và khó khăn. Nếu biết hàm nhân (Kernel function) $K(x_i, x_j)$, để tính tích vô hướng $\phi(x_i)\phi(x_j)$ trong không gian m-chiều, thì chúng ta không cần làm việc trực tiếp với ánh xạ $\phi(x_i)$.

$$K(x_i, x_j) = \phi(x_i)\phi(x_j) \qquad (2.12)$$

Hàm nhân Kernel là những hàm trả về giá trị tích trong giữa các ảnh của các điểm dữ liệu trong một vài không gian. Việc lựa chọn K cũng chính là chọn ϕ. Các hàm kernel có thể được tính toán hiệu quả ngay cả trên không gian có rất nhiều chiều.

Hàm Kernel đóng một vai trò quan trọng trong bài toán phân loại sử dụng phương pháp SVM, nó tạo ra ma trận Kernel tóm tắt tất cả dữ liệu.

Trong thực tế, Hàm nhân đa thức đa thức với số bậc thấp thì khởi tạo tốt cho hầu hết các ứng dụng. Đối với phân loại văn bản thì sử dụng Hàm nhân tuyến tính là tốt nhất vì số chiều đặc trưng của hàm này đã đủ lớn.

Một số hàm nhân hay dùng trong phân lớp văn bản là :

Hàm nhân tuyến tính (linear): $K(x_i, x_j) = x_i^T x_j$ (2.13)

Hàm nhân đa thức (polynomial function) : $K(x_i, x_j)=(x_i x_j+1)^d$ (2.14)

với d : thường là các số tự nhiên từ 1 đến 10

Hàm RBF (radial basis function) : $K(x_i, x_j)=\exp(-\gamma(x_i-x_j)^2), \ \gamma \in R^+$ (2.15)

2.4.4 Thuật toán Sequential Minimal Optimization (SMO)

SMO là thuật toán đơn giản có thể giải quyết nhanh bài toán qui hoạch toàn phương của SVM (Quadratic Programming) mà không cần sử dụng ma trận lưu trữ. SMO phân rã bài toán qui hoạch toàn phương thành dãy các bài toán con, sử dụng

định lý Osuna để đảm bảo sự hội tụ. Đây là một trong những phương pháp tốt nhất hiện nay có thể giải quyết rất tốt bài toán SVM nói trên.

Không giống như các phương thức khác, SMO chọn cách giải quyết tối ưu hóa vấn đề nhỏ nhất có thể tại mỗi bước. Ở mỗi bước SMO chọn hai số nhân Lagrange để cùng tối ưu, tìm những giá trị tối ưu cho các số nhân này và cập nhật SVM tương ứng với giá trị tối ưu mới. Sự thuận tiện của SMO trong thực tế là việc xử lý hai số nhân Lagrange được thực hiện theo phép phân tích. Hơn nữa SMO không phụ thuộc vào ma trận lưu trữ thêm tại tất cả các bước.

Ba thành phần của SMO:

- Phương pháp giải tích để xử lý hai số nhân Lagrange
- Heuristic để chọn số nhân tối ưu
- Phương pháp để tính b tại mỗi bước.

2.5 Đánh giá các thuật toán phân lớp phổ biến

Từ các phân tích ở trên ta thấy rằng:

Thuật toán K-Trung bình

Ưu điểm: K-trung bình mờ cải tiến sự hội tụ của thuật toán K-Trung bình.

Khuyết điểm:

- Việc khởi tạo các nhóm có ảnh hưởng rất lớn đến kết quả phân nhóm.
- Việc đặc tả không chính xác số nhóm.

Thuật toán K-láng giềng gần nhất

Ưu điểm: Dễ hiểu, dễ cài đặt thuật toán.

Khuyết điểm:

- Kết quả phân lớp là không tốt trong trường hợp dữ liệu huấn luyện có nhiễu.
- Rất khó có thể tìm ra k tối ưu.
- Cần nhiều văn bản để huấn luyện.

Thuật toán cây quyết định

Ưu điểm:

- Dễ hiểu, dễ cài đặt thuật toán.

- Có thể chấp nhận trường hợp tập dữ liệu huấn luyện có nhiễu, và cho hiệu quả phân lớp tương đối cao.

Khuyết điểm:

- Cây quyết định sinh ra quá phức tạp, và đôi khi có những nhánh của cây ít (hoặc không) được sử dụng đến khi đưa ra quyết định phân lớp.

- Cần nhiều văn bản cho tập dữ liệu huấn luyện.

Thuật toán Support Vector Machines

Ưu điểm:

Support Vector Machines (SVM) hoàn toàn thích hợp với bài toán phân lớp vì:

- Số lượng các đặc trưng (kích thước không gian đặc trưng) của văn bản được phân loại không ảnh hưởng đến khả năng của hệ thống sử dụng SVM. Nói cách khác, giải thuật SVM có thể xử lý với bất kỳ văn bản nào trong khi đang huấn luyện dù cho số lượng đặc trưng chứa trong nó lớn.

- Hiệu quả phân lớp cao : SVM hoàn toàn có khả năng lựa chọn được một phương án mà có rủi ro là nhỏ nhất. Nguồn gốc của SVM dựa trên sự chắc chắn về lỗi chính xác, có thể phân loại ngẫu nhiên các mẫu văn bản được chọn mà lỗi được giữ sao cho nhỏ nhất. Vì vậy, giải thuật SVM giúp giảm thiểu biên trên các lỗi chính xác và làm cho hệ thống tin cậy hơn.

- Tránh hiện tượng tràn lỗi: Có thể tránh được hiện tượng vector biểu diễn có số chiều lớn, vì việc tối thiểu rủi ro của phân lớp SVM không phụ thuộc vào số chiều của văn bản.

- Số lượng văn bản của tập huấn luyện không quá lớn: Nhiều thực nghiệm cho thấy, SVM vẫn cho hiệu quả phân lớp cao với một số ít mẫu huấn luyện.

Khuyết điểm:

- Trong SVM thông thường thì các điểm dữ liệu đều có giá trị như nhau, mỗi một điểm sẽ thuộc hoàn toàn vào một trong hai lớp. Tuy nhiên trong nhiều trường hợp có một vài điểm sẽ không thuộc chính xác vào một lớp nào đó, những điểm này được gọi là những điểm nhiễu, và mỗi điểm có thể sẽ không có ý nghĩa như nhau đối với mặt phẳng quyết định.

- Để đạt kết quả phân loại tốt cần chọn hàm nhân Kernel phù hợp.

- Yêu cầu phải lặp đi lặp lại quá trình huấn luyện đối với bài toán nhiều lớp vì SVM chỉ giải quyết bài toán phân lớp 2 lớp.

Để khắc phục nhược điểm của các phương pháp trên, trong chương tiếp theo ta sẽ xem xét các thuật toán phân lớp cải tiến dựa phương pháp phân lớp văn bản Support Vector Machines. Đây là những phương pháp phân lớp mới, cho kết quả phân lớp cao hơn so với các phương pháp phổ biến trên. Thậm chí, cho kết quả phân lớp tốt ngay cả trong trường hợp dữ liệu huấn luyện có nhiễu, và số mẫu huấn luyện là ít.

CHƯƠNG 3: CÁC THUẬT TOÁN CẢI TIẾN CỦA PHƯƠNG PHÁP PHÂN LỚP VĂN BẢN BẰNG SUPPORT VECTOR MACHINES

Trong chương này chúng ta sẽ nghiên cứu các thuật toán phân lớp cải tiến dựa trên phương pháp phân lớp văn bản Support Vector Machines (SVM) cụ thể là các thuật toán phân lớp Fuzzy Support Vector Machines (FSVM), Support Vector Machines Nearest Neighbor (SVM-NN). Đồng thời, chúng ta cũng khảo sát các chiến lược phân loại đa lớp OAR, OAA, Fuzzy OAO áp dụng cho các thuật toán phân lớp 2 lớp FSVM, SVM-NN để thực hiện phân loại đa lớp.

3.1 Fuzzy Support Vector Machines (FSVM)

3.1.1 Bài toán và cách giải quyết

Trong SVM thông thường thì các điểm dữ liệu đều có giá trị như nhau, mỗi một điểm sẽ thuộc hoàn toàn vào một trong hai lớp. Tuy nhiên trong nhiều trường hợp có một vài điểm sẽ không thuộc chính xác vào một lớp nào đó, những điểm này được gọi là những điểm nhiễu, và mỗi điểm có thể sẽ không có ý nghĩa như nhau đối với mặt phẳng quyết định. Để giải quyết vấn đề này, Fuzzy Support Vector Machines (FSVM) đã được giới thiệu bằng cách sử dụng một hàm thành viên để xác định giá trị đóng góp của mỗi điểm dữ liệu đầu vào của SVM vào việc hình thành siêu phẳng.

Cho tập dữ liệu huấn luyện $S = \{(x_i, y_i, s_i), i = ,..., l\}$

Với x_i là một mẫu huấn luyện, $x_i \in$ ^{n}, y_i là nhãn của x_i, $y_i \in$ $\{-,+\}$

s_i là một hàm thành viên thỏa $\sigma \le s_i \le$, σ là một hằng số đủ nhỏ > 0.

Bài toán được mô tả như sau:

$$\begin{cases} \text{Min } \Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} s_i\xi_i \\ y_i(w^T\phi(x_i)+b) \ge 1-\xi_i \qquad i=1,\ldots,l \\ \xi_i \ge 0 \qquad\qquad i=1,\ldots,l \end{cases} \qquad (3.1)$$

C là một hằng số. Hàm thành viên s_i thể hiện mức độ ảnh hưởng của điểm x_i đối với một lớp. Giá trị s_i có thể làm giảm giá trị của biến ξ , vì vậy điểm x_i tương ứng với ξ có thể được giảm mức độ ảnh hưởng hơn.

Để giải quyết bài toán tối ưu trên ta xây dựng hàm Lagrangian:

$$L(w,b,\xi,\alpha\ \beta) = \frac{1}{2}w^T.w + C\sum_{i=1}^{l} s_i\xi - \sum_{i=1}^{l}\alpha\ y_i(w^T.z_i + b) - 1 + \xi) - \sum_{i=1}^{l}\beta\ \xi \quad (3.2)$$

trong đó $z_i = \phi(x_i)$ (3.3)

Nghiệm của bài toán tối ưu (3.1) chính là điểm yên ngựa của hàm Lagrangian (3.2) với các tham số thỏa điều kiện sau :

$$\frac{\partial L(w,b,\xi,\alpha\ \beta}{\partial w} = w - \sum_{i=1}^{l}\alpha\ y_i z_i = 0 \quad (3.4)$$

$$\frac{\partial L(w,b,\xi,\alpha\ \beta}{\partial b} = -\sum_{i=1}^{l}\alpha\ y_i = 0 \quad (3.5)$$

$$\frac{\partial L(w,b,\xi,\alpha\ \beta}{\partial \xi} = s_i C - \alpha\ - \beta\ = 0 \quad (3.6)$$

Áp dụng những điều kiện (3.4), (3.5) (3.6) vào công thức (3.1), (3.2) có thể được chuyển thành bài toán:

$$\begin{cases} \max_{\alpha} L_D(\alpha\ = \sum_{i=1}^{l}\alpha\ - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha\ y_i y_j K(x_i,x_j) \\ \sum_{i=1}^{l}\alpha\ y_i = 0 \\ 0 \le \alpha\ \le s_i C \qquad i = 1,2,\ldots,l \end{cases} \quad (3.7)$$

Đây là dạng đơn giản của bài toán QP (Quadratic programming). Và tìm giá trị tối ưu $(\overline{w},\overline{b},\xi,\alpha$ cho những điều kiện Kunhn-Tucker (Cristianini và Shawe-Taylor 2000) được định nghĩa như sau:

$$\overline{w} - \sum_{i\in I_{s.v.}}\alpha\ y_i z_i = 0 \quad (3.8)$$

$$-\sum_{i\in I_{s.v.}}\alpha_{\cdot\ i} = \quad (3.9)$$

$$y_i(\overline{w}.z_i + \bar{\ }) - \ + \xi \geq \ ), \ i = \ ,...l \quad (3.10)$$

$$(s_i C - \alpha\ )\xi = \ ), \ i = \ ,...l \quad (3.11)$$

$$\alpha\ \ y_i(\overline{w}.z_i + \bar{\ }) - \ + \xi\ ) = \ ), \ i = \ ,...l \quad (3.12)$$

Với $I_{s.v.}$ = {j|x_j là support vector}

Theo các điều kiện trên ta sẽ tìm được giá trị tối ưu α . Có được α , các tham số của siêu phẳng tối ưu là:

$$\overline{w} = \sum_{i\in I_{s.v.}}\alpha\ _i z_i \quad (3.13)$$

$\bar{b}$ được tính từ công thức (3.12) bằng cách chọn bất kỳ i sao cho α >0. Điểm x_i tương ứng với với α >0 được gọi là support vector.

Như vậy trong không gian m chiều, việc phân lớp tài liệu x tương đương với việc tính hàm *f(x)* :

$$f(x) = \text{sgn}(\sum_{i=}^{l}\alpha\ _i K(x_i, x) + \bar{b}) \quad (3.14)$$

3.1.2 Hàm thành viên

Việc chọn hàm thành viên s_i thích hợp là rất quan trọng trong FSVM. Theo Chun hàm thành viên s_i dùng để giảm mức độ ảnh hưởng của những điểm dữ liệu nhiễu được mô tả trong [11] là một hàm xác định khoảng cách giữa điểm dữ liệu x_i với trung tâm của nhóm tương ứng với i.

Gọi C^+ là tập chứa các điểm x_i với y_i =1

C^+ ={x_i|$x_i \in$ S và y_i =1}

Tương tự gọi C^- ={x_i|$x_i \in$ S và y_i =-1}

X_+ và X_- là trung tâm của lớp C^+, C^-.

Bán kính của lớp C^+ là:

$$r_+ = max||X_+ - x_i|| \quad với\ x_i \in C^+ \quad (3.15)$$

và bán kính của lớp C^- là:

$$r_- = max||X_- - x_i|| \quad với\ x_i \in C^- \quad (3.16)$$

Hàm thành viên s_i được định nghĩa như sau:

$$s_i = \begin{cases} 1 - \| X_+ - x_i \| / (r_+ + \delta & \text{nếu } x_i \in C^+ \\ 1 - \| X_- - x_i \| / (r_- + \delta & \text{nếu } x_i \in C^- \end{cases} \qquad (3.17)$$

δ là một hằng số để tránh trường hợp $s_i = 0$

Tuy nhiên FSVM với hàm thành viên (3.17) vẫn chưa đạt kết quả tốt do việc tính toán khoảng cách giữa các điểm dữ liệu với trung tâm của nhóm được tiến hành ở không gian đầu vào, không gian n chiều. Trong khi đó trong trường hợp tập dữ liệu không thể phân chia tuyến tính, để hình thành siêu phẳng ta phải đưa dữ liệu về một không gian khác với số chiều m cao hơn gọi là không gian đặc trưng (feature space). Vì vậy để có thể đạt kết quả tốt hơn, Xiufeng Jiang, Zhang Yi và Jian Cheng Lv (2006) [11] đã xây dựng một hàm thành viên khác dựa trên ý tưởng của hàm thành viên (3.17) nhưng được tính toán trong không gian đặc trưng m chiều.

Giả sử ϕ là một ánh xạ phi tuyến tính từ không gian R^n vào không gian R^m.

$$\phi \quad R^n \rightarrow \quad ^n$$

Khi đó, vector x_i trong không gian R^n sẽ tương ứng với vector $\phi(x_i)$ trong không gian R^m.

Định nghĩa ϕ_+ là trung tâm của lớp C^+ trong không gian đặc trưng:

$$\phi_+ = \frac{1}{n_+} \sum_{x_i \in C^+} \phi(x_i) \qquad (3.18)$$

n_+ là số phần tử của lớp C^+

và ϕ_- là trung tâm của lớp C^- trong không gian đặc trưng:

$$\phi_- = \frac{1}{n_-} \sum_{x_i \in C^-} \phi(x_i) \qquad (3.19)$$

n_- là số phần tử của lớp C^-

Định nghĩa bán kính của C^+:

$$r_+ = \max \| \phi_+ - \phi(x_i) \| \qquad với\ x_i \in C^+ \qquad (3.20)$$

và bán kính của C^-:

$$r_- = \max \| \phi_- - \phi(x_i) \| \qquad với\ x_i \in C^- \qquad (3.21)$$

Khi đó,

$$r_+^2 = \max \| \phi(x') - \phi_+ \|^2$$
$$= \max\{ \phi^2(x') - 2\phi(x').\phi_+ + \phi_+^2 \}$$
$$= \max\{\phi(x') - \frac{2}{n_+}\sum_{x_i \in C^+} \phi(x_i)\phi(x') + \frac{1}{n_+^2}\sum_{x_i \in C^+}\sum_{x_j \in C^+} \phi(x_i)\phi(x_j)\}$$
$$= \max\{K(x',x') - \frac{2}{n_+}\sum_{x_i \in C^+} K(x_i,x') + \frac{1}{n_+^2}\sum_{x_i \in C^+}\sum_{x_j \in C^+} K(x_i,x_j)\} \qquad (3.22)$$

Với x' $\in C^+$ và n_+ là số mẫu huấn luyện trong lớp C^+. Tương tự :

$$r_-^2 = \max\{K(x',x') - \frac{2}{n_-}\sum_{x_i \in C^-} K(x_i,x') + \frac{1}{n_-^2}\sum_{x_i \in C^-}\sum_{x_j \in C^-} K(x_i,x_j)\} \qquad (3.23)$$

Với x' $\in C^-$ và n_- là số mẫu huấn luyện trong lớp C^-.

Bình phương khoảng cách giữa $x_i \in C^+$ và trung tâm của lớp trong không gian đặc trưng có thể được tính như sau:

$$d_{i+}^2 = K(x_i,x_i) - \frac{2}{n_+}\sum_{x_j \in C^+} K(x_i,x_j) + \frac{1}{n_+^2}\sum_{x_j \in C^+}\sum_{x_k \in C^+} K(x_j,x_k)\} \qquad (3.24)$$

Tương tự như vậy bình phương khoảng cách giữa $x_i \in C^-$ và trung tâm của lớp trong không gian đặc trưng có thể được tính như sau:

$$d_{i-}^2 = K(x_i,x_i) - \frac{2}{n_-}\sum_{x_j \in C^-} K(x_i,x_j) + \frac{1}{n_-^2}\sum_{x_j \in C^-}\sum_{x_k \in C^-} K(x_j,x_k)\} \qquad (3.25)$$

Với mỗi i (i=1,…,l), hàm thành viên s_i được mô tả như sau:

$$s_i = \begin{cases} 1 - \sqrt{\| d_{i+}^2 \| / (r_+^2 + \delta} & \text{nếu } x_i \in C^+ \\ 1 - \sqrt{\| d_{i-}^2 \| / (r_-^2 + \delta} & \text{nếu } x_i \in C^- \end{cases} \qquad (3.26)$$

Ta thấy s_i là một hàm của trung tâm và bán kính của mỗi lớp trong không gian đặc trưng.

Theo kết quả thử nghiệm của Xiufeng Jiang, Zhang Yi và Jian Cheng Lv hàm thành viên theo công thức (3.26) bằng cách sử dụng hàm nhân để tính toán trong không gian m chiều có thể làm giảm ảnh hưởng của các điểm nhiễu hiệu quả hơn hàm thành viên của Lin CF và Wang SD (2002) và cho kết quả phân lớp tốt hơn [11].

3.1.3 Thuật toán huấn luyện Kernel-Adatron

Quá trình huấn luyện FSVM là quá trình giải bài toán tối ưu (3.7) để tìm được nghiệm α tối ưu. Quá trình huấn luyện này khá phức tạp và đòi hỏi nhiều chi phí cho việc tính toán. Sử dụng thuật toán Kernel-Adatron có thể đơn giản hóa quá trình huấn luyện FSVM.

Thuật toán Kernel-Adatron được mô tả như sau: [6]

Bước 1. Khởi tạo $\alpha = 0$

Bước 2. For i=1,..,l thực hiện bước 3,4

Bước 3. Đối với mỗi điểm (x_i, y_i) ta tính:

$$z_i = \sum_{j=1}^{l} \alpha_j y_j K(x_i, y_j)$$

Bước 4. Tính $\delta\alpha_i = \eta(1 - z_i y_i)$:

4.1. Nếu $(\alpha_i + \delta\alpha_i) \le 0$ thì $\alpha_i = 0$

4.2. Nếu $0 < (\alpha_i + \delta\alpha_i) < C_i$ thì $\alpha_i = \alpha_i + \delta\alpha_i$

4.3. Nếu $(\alpha_i + \delta\alpha_i) \ge C_i$ thì $\alpha_i = C_i$

η là một hằng số điều chỉnh sự hội tụ của thuật toán.

Bước 5. Nếu số lần lặp bị vượt hoặc lề λ xấp xỉ 1 thì ngừng, nếu không thì quay trở lại bước 2 cho lần lặp t = t+1.

$$\lambda = \frac{1}{2}[\min_{\{i|y_i=+1\}}(z_i) - \max_{\{i|y_i=-1\}}(z_i)]$$

3.2 Support Vector Machines Nearest Neighbor (SVM-NN)

Support Vector Machines Nearest Neighbor (SVM-NN) (Blanzieri & Melgani 2006) là một thuật toán phân lớp cải tiến gần đây nhất của phương pháp phân lớp SVM. SVM-NN là một kỹ thuật phân loại văn bản máy học sử dụng kết hợp cách tiếp cận K-láng giềng gần nhất (K-NN) với những luật ra quyết định dựa trên SVM (SVM-based decision rule).

3.2.1 Ý tưởng của thuật toán SVM-NN

Thuật toán phân lớp SVM-NN kết hợp các ý tưởng của thuật toán phân lớp SVM và thuật toán phân lớp K-NN.

Nó hoạt động theo cách sau:

- Cho một mẫu để phân loại, thuật toán xác định k mẫu gần nhất trong các mẫu dữ liệu của tập dữ liệu huấn luyện.
- Một phân loại SVM được huấn luyện trên những mẫu này.
- Sau đó, các bộ phân loại SVM được huấn luyện sẽ được sử dụng để phân loại các mẫu chưa biết.

3.2.2 **Thuật toán SVM-NN[4][5]**

Đầu vào:

Mẫu x để phân lớp;

Bộ dữ liệu huấn luyện $T = \{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$;

Số láng giềng gần nhất k;

Các bước của thuật toán:

Bước 1: tìm k mẫu (xi, yi) với giá trị nhỏ nhất của

$$K(x_i, x_i) - 2 * K(x_i, x)$$

Bước 2: huấn luyện theo mô hình SVM trên k mẫu được lựa chọn.

Bước 3: phân lớp mẫu x dùng mô hình SVM trên, nhận kết quả dưới dạng số thực.

Bước 4: Ra quyết định (sử dụng threshold t).

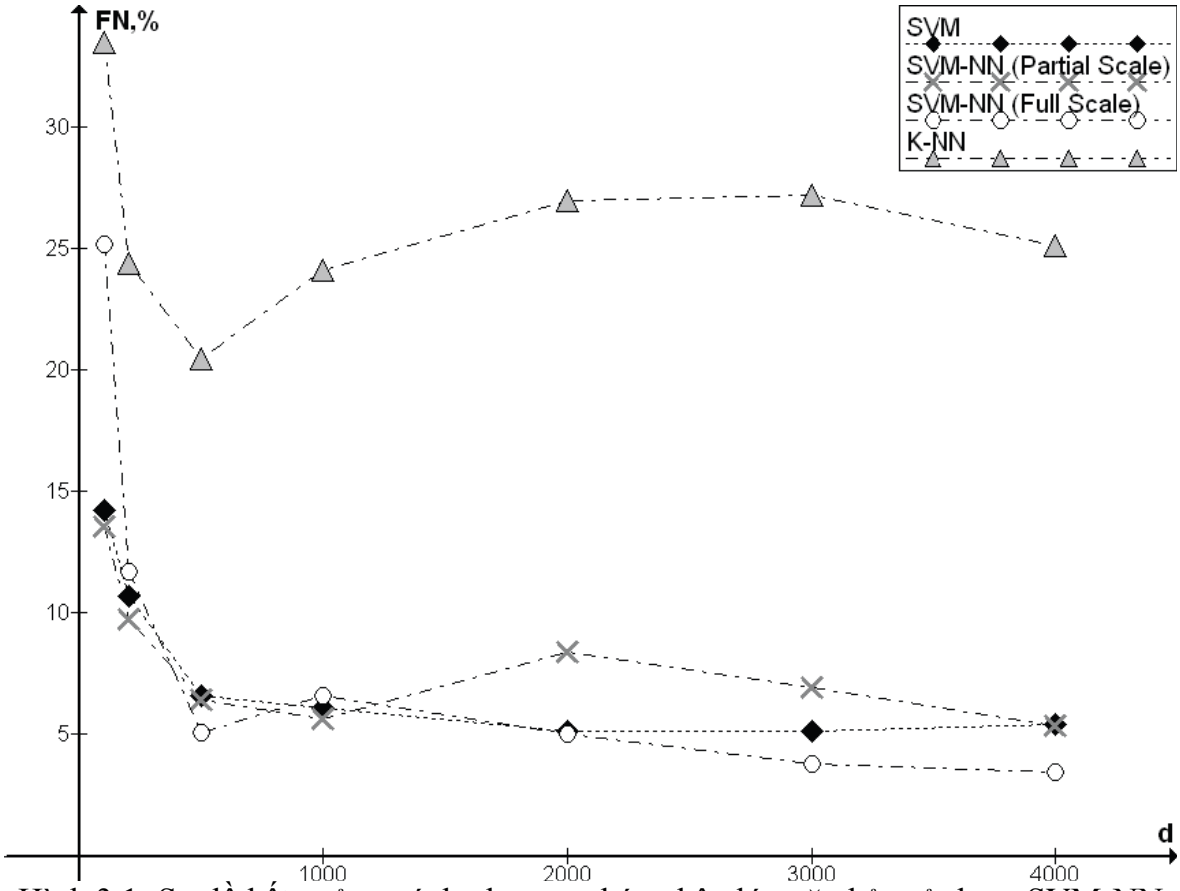
Kết quả:

Quyết định $y \in \{-1, 1\}$

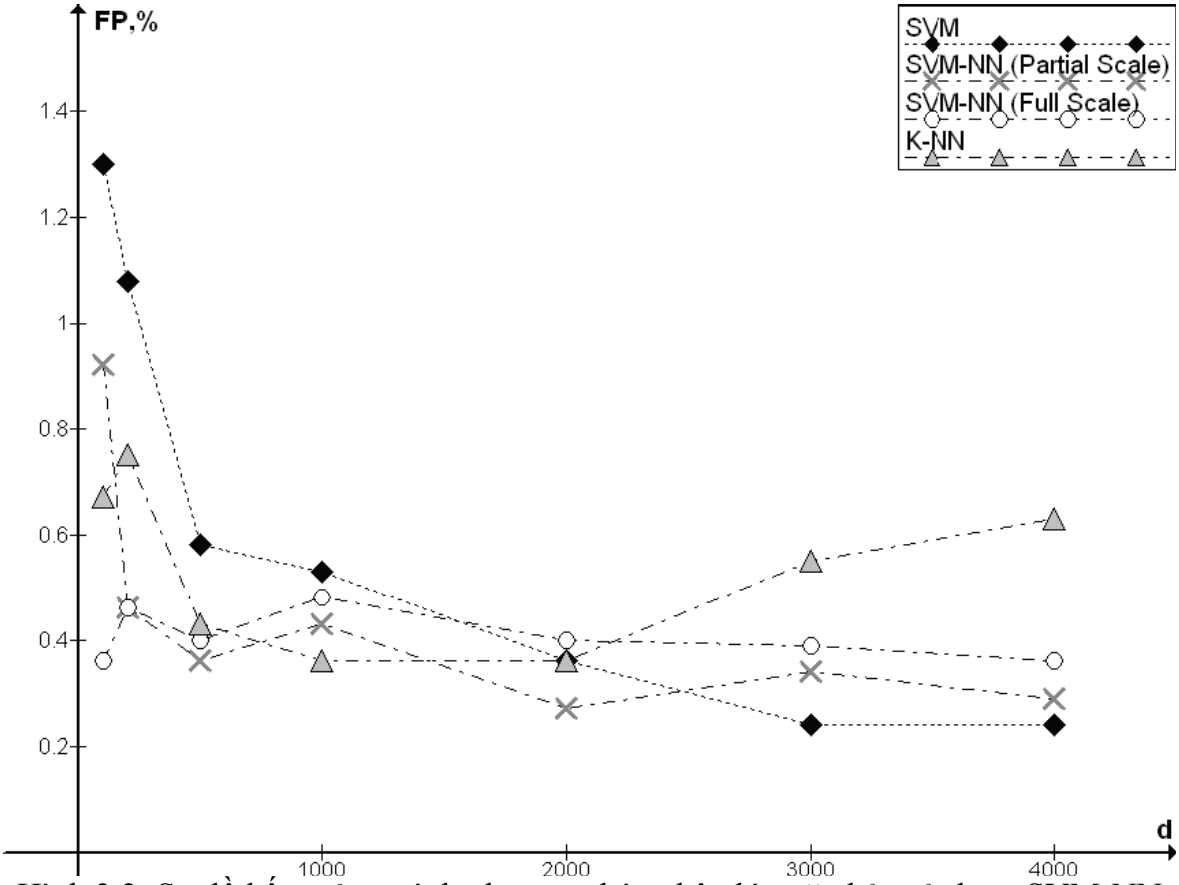
Để đánh giá hiệu quả phân lớp của SVM-NN so với SVM và K-NN, Enrico Blanzieri và Anton Bryl [4][5] đã thực hiện một thử nghiệm để so sánh. Thử nghiệm này sử dụng độ đo Euclide để xác định các láng giềng gần nhất và hàm nhân tuyến tính cho SVM.

Hai biến thể của SVM-NN trong so sánh thử nghiệm: SVM-NN qui mô một phần (Partial Scale SVM-NN) và SVM-NN qui mô đầy đủ (Full Scale). Trong đó,

SVM-NN qui mô một phần (Partial Scale) tìm kiếm các giá trị tối ưu của k chỉ giữa các giá trị tương đối thấp và do đó nó nhanh hơn nhưng ít chính xác hơn SVM-NN qui mô đầy đủ (Full Scale SVM-NN).



Hình 3.1: Sơ đồ kết quả so sánh phương pháp phân lớp văn bản sử dụng SVM-NN với K-NN và SVM (theo tỷ lệ âm sai FN)



Hình 3.2: Sơ đồ kết quả so sánh phương pháp phân lớp văn bản sử dụng SVM-NN với K-NN và SVM (theo tỷ lệ dương sai FP)

Trong 2 sơ đồ trên:

- Tỷ lệ âm sai **FN** (False Negative) là số văn bản được gán nhãn là –1 nhưng việc gán nhãn này là sai).

- Tỷ lệ dương sai **FP** (False Positive) là số văn bản được gán nhãn là 1 nhưng việc gán nhãn này là sai.

- **d** là số lượng các từ đặc trưng.

- **SVM** là một bộ phân lớp SVM, Threshold t được lấy giá trị là 0,48.

- **SVM-NN (Partial Scale)** và **SVM-NN (Full Scale)** là các bộ phân loại SVM-NN. Các giá trị tối ưu của k và t được ước tính bằng cách sử dụng dữ liệu huấn luyện. Giá trị tối ưu của k là nhỏ hơn 25% của tập dữ liệu huấn luyện.

- **K-NN** là một bộ phân loại K-NN.

Trên sơ đồ hình 3.1, hình 3.2, chúng ta có thể thấy rằng, hiệu quả của các phương pháp được thể hiện dựa trên tỷ lệ phân lớp lỗi (Error Rate - tức tỷ lệ phân lớp không chính xác) bao gồm tỷ lệ âm sai FN và tỷ lệ dương sai FP:

- Với số lượng các từ đặc trưng d thấp thì SVM-NN có khả năng phân lớp tốt hơn SVM. Đặc biệt, khi d = 500, tỷ lệ âm sai FN đối với SVM-NN (Full Scale) là thấp hơn đáng kể so với K-NN và SVM, và tỷ lệ dương sai FP là thấp hơn không đáng kể so với SVM-NN (Full Scale) lẫn SVM-NN (Partial Scale), nhưng vẫn thấp hơn đáng kể so với K-NN.

- Với số lượng các từ đặc trưng d cao hơn, lợi thế của SVM-NN là nhỏ hơn, nhưng nó vẫn tồn tại: khi d = 4000, SVM-NN (Full Scale) là tốt hơn đáng kể về tỷ lệ âm sai FN, mặc dù xấu hơn một chút về tỷ lệ dương sai FP.

3.3 Chiến lược phân loại đa lớp

Các thuật toán SVM, FSVM, SVM-NN đã trình bày ở phần trên chỉ áp dụng cho phân lớp hai lớp, tức là xác định một văn bản có hay không thuộc một lớp cho trước. Việc áp dụng trong bài toán phân lớp đa lớp cần kết hợp với các chiến lược phân lớp khác.

Phần này chúng ta sẽ tìm hiểu các chiến lược áp dụng trong bài toán phân lớp văn bản thuộc nhiều loại khác nhau. Ý tưởng của bài toán phân lớp đa lớp là chuyển về bài toán phân lớp hai lớp bằng cách xây dựng nhiều bộ phân lớp hai lớp để giải quyết. Các chiến lược phân lớp đa lớp phổ biến này là One-against-One (OAO) [8], [9] và One-against-Rest (OAR) [7].

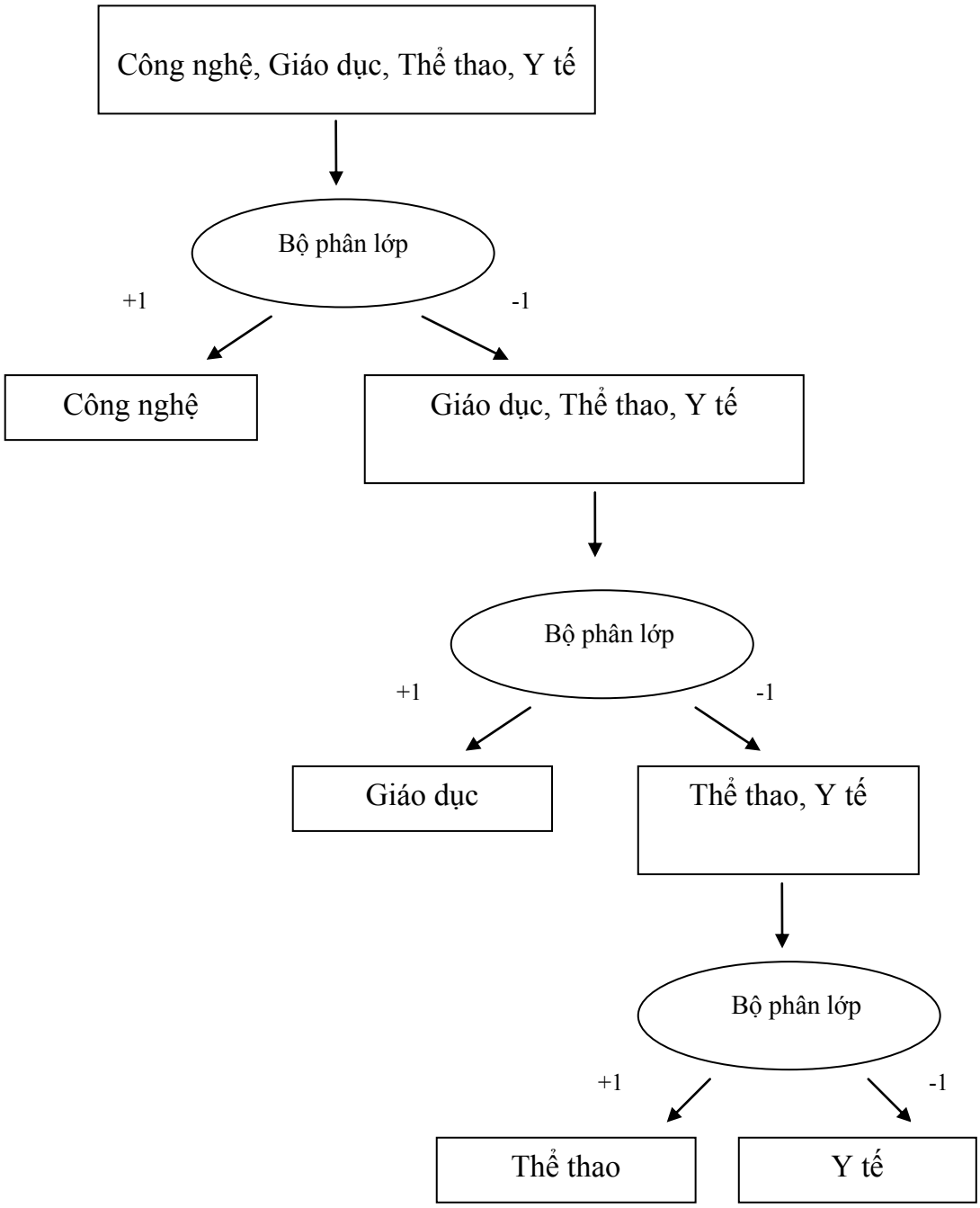
3.3.1 Chiến lược One-against-Rest (OAR)

Trong chiến lược này ta sử dụng (n-1) bộ phân lớp đối với n lớp. Bài toán phân lớp n lớp được chuyển thành n bài toán phân lớp hai lớp. Trong đó bộ phân lớp hai lớp thứ i được xây dựng trên lớp thứ i và tất cả các lớp còn lại. Hàm quyết định thứ i dùng để phân lớp thứ i và những lớp còn lại có dạng:

$$D_i(x) = w_i^t x + b_i \qquad (3.27)$$

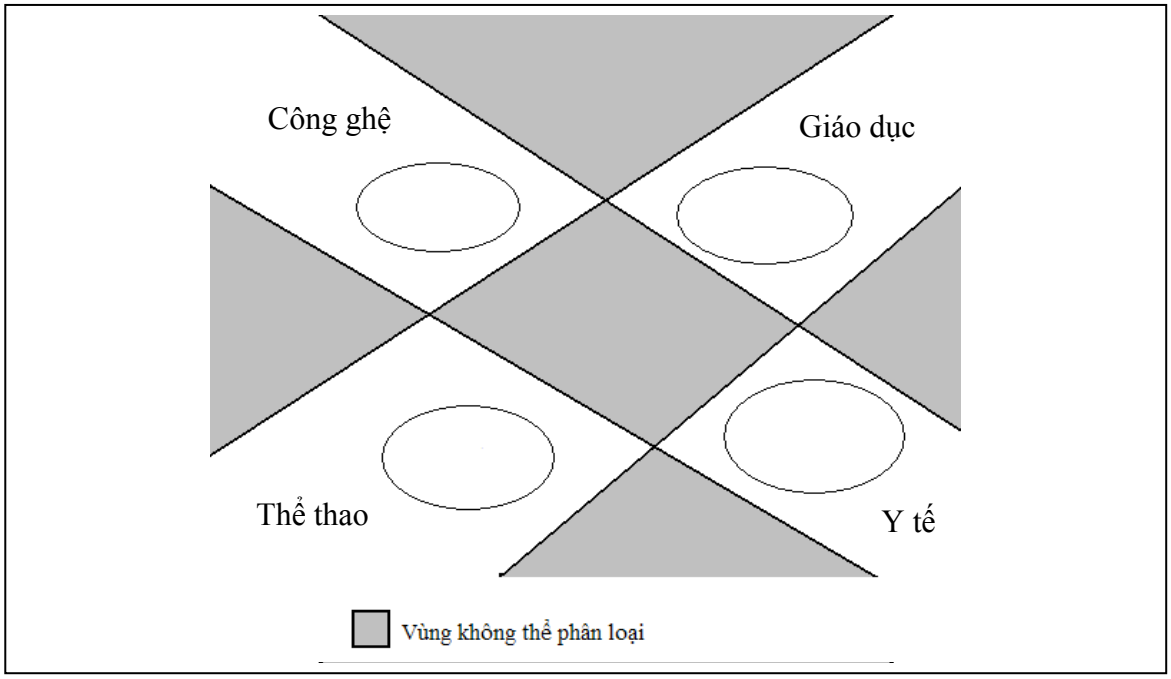
Siêu phẳng $D_i(\vec{x}) = 0$ hình thành siêu phẳng phân chia tối ưu, các support vector thuộc lớp i thỏa $D_i(\vec{x}) =$ và các support vector thuộc lớp còn lại thỏa $D_i(\vec{x}) = -$. Nếu vector dữ liệu x thỏa mãn điều kiện $D_i(\vec{x}) > 0$ đối với duy nhất một i, x sẽ được phân vào lớp thứ i.

Ví dụ phân lớp các văn bản thuộc các chủ đề: Công nghệ, Giáo dục, Thể thao, Y tế Kinh tế, Văn hóa, Xã hội, Thể thao theo chiến lược OAR.



Hình 3.3: Ví dụ phân lớp đa lớp theo chiến lược OAR

Tuy nhiên nếu điều kiện D_i (x) > 0 thỏa mãn đối với nhiều i, hoặc không thỏa đối với i nào thì trong trường hợp này ta không thể phân lớp được vector x. Để giải quyết vấn đề này chiến lược One-against-One (OAO) [7] được đề xuất sử dụng.

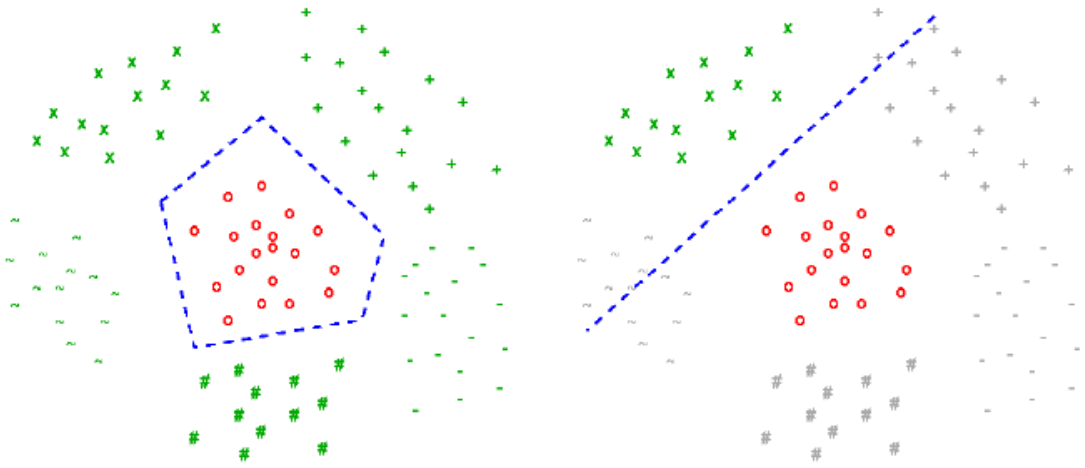


Hình 3.4: Vùng không phân lớp được theo chiến lược OAR

3.3.2 Chiến lược One-against-One (OAO)

Trong chiến lược này ta sử dụng n(n-1)/2 bộ phân lớp hai lớp được xây dựng bằng cách bắt cặp từng hai lớp một nên chiến lược này còn được gọi là pairwise và sử dụng phương pháp lựa chọn theo đa số để kết hợp các bộ phân lớp này để xác định được kết quả phân lớp cuối cùng. Số lượng các bộ phân lớp không bao giờ vượt quá n(n-1)/2.

So với chiến lược OAR thì chiến lược này ngoài ưu điểm giảm bớt vùng không thể phân lớp mà còn làm tăng độ chính xác của việc phân lớp. Trong chiến lược OAR ta phải xây dựng một siêu phẳng để tách một lớp ra khỏi các lớp còn lại, việc này đòi hỏi sự phức tạp và có thể không chính xác. Tuy nhiên trong chiến lược OAO ta chỉ cần phân tách một lớp ra khỏi một lớp khác mà thôi.



Hình 3.5: Ví dụ phân lớp sử dụng chiến lược OAR và OAO

Trong hình 5.3 ta thấy chiến lược OAR (hình bên trái) phải xây dựng siêu phẳng để tách lớp đánh dấu "o" ra khỏi tất cả các lớp khác. Còn chiến lược OAO (hình bên phải) chỉ cần tách lớp "o" ra khỏi lớp đánh dấu "x"

Chiến lược OAR chỉ cần n-1 bộ phân lớp cho n lớp. Trong khi đó chiến lược OAO lại cần đến n(n-1)/2 bộ phân lớp. Nhưng số mẫu huấn luyện cho từng bộ phân lớp trong OAO lại ít hơn và việc phân lớp cũng đơn giản hơn. Vì vậy chiến lược OAO có độ chính xác cao hơn nhưng chi phí để xây dựng lại tương đương với chiến lược OAR.

Hàm quyết định phân lớp của lớp i đối với lớp j trong chiến lược OAO là:

$$D_{ij}(x) = w_{ij}^t x + b_{ij} \qquad (3.27)$$

$$D_{ij}(x) = -D_{ji}(x) \qquad (3.28)$$

Đối với một vector x ta tính :

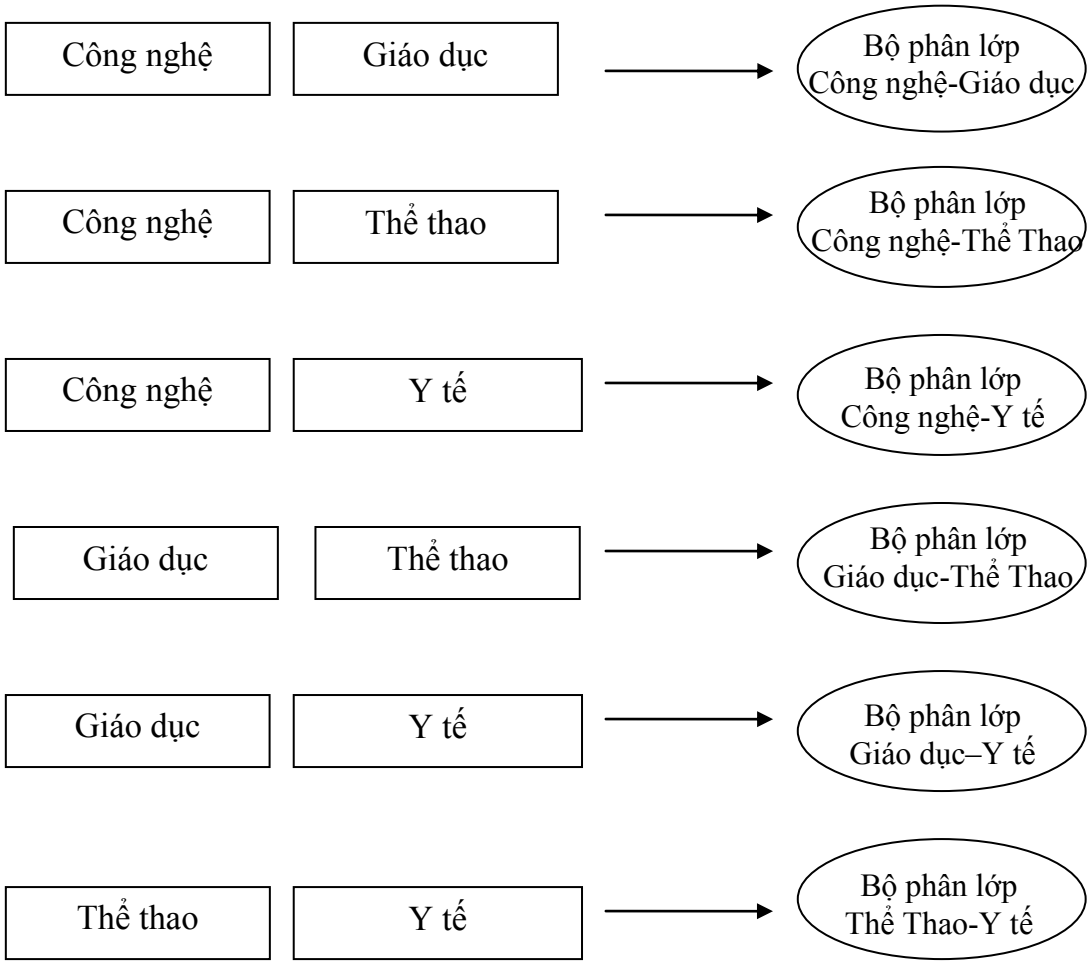
$$D_i(x) = \sum_{j \neq i, j=1}^{n} sign(D_{ij}(x)) \qquad (3.29)$$

với $$sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \qquad (3.30)$$

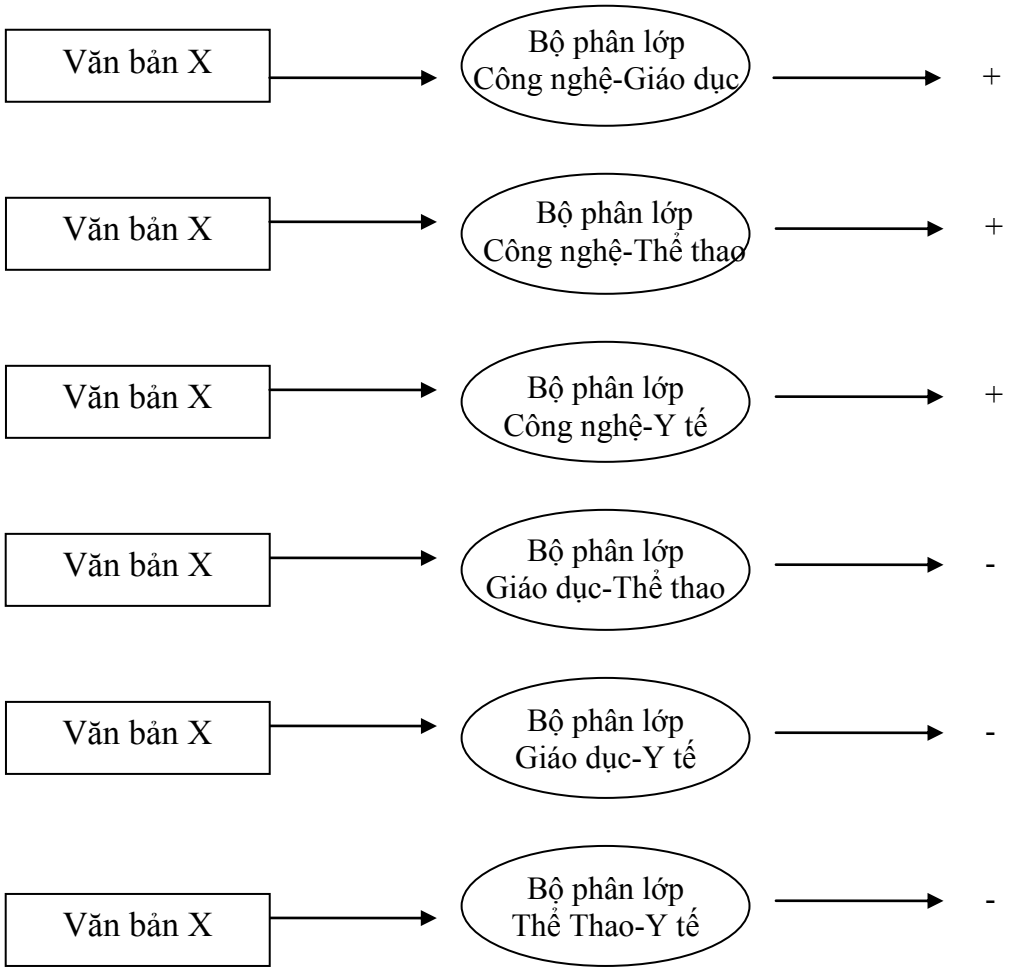
Và x được phân vào lớp i sao cho:

$$\arg\max_{i=\ ,\ldots,n} D_i \quad (3.31)$$

Ví dụ phân lớp các văn bản thuộc các chủ đề: Kinh tế, Văn hóa, Xã hội, Thể thao theo chiến lược OAO.



Xây dựng các bộ phân lớp bằng cách bắt cặp các lớp như sau:



Hình 3.6: Ví dụ phân lớp đa lớp theo chiến lược OAO

Văn bản X sau khi qua các bộ phân lớp có kết quả như sau:

$D_{\text{Công nghệ}}(X) = 3$

$D_{\text{Giáo dục}}(X) = 0$

$D_{\text{Thể thao}}(X) = 1$

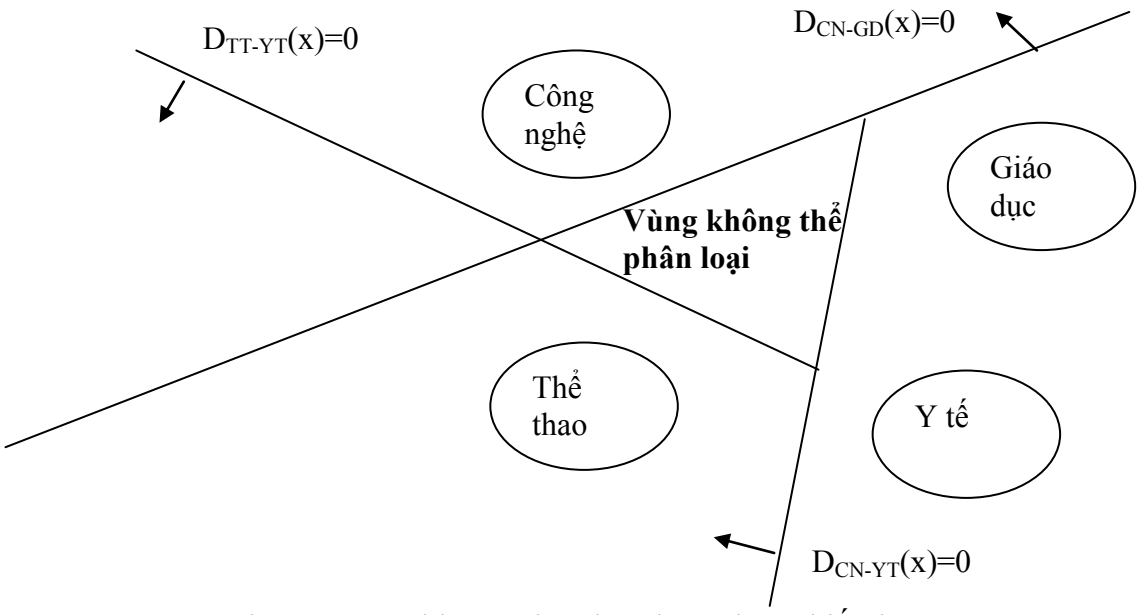
$D_{\text{Y tế}}(X) = 2$

Suy ra:

$X = \text{argmax}\ (D_{\text{Công nghệ}}(X),\ D_{\text{Giáo dục}}(X), D_{\text{Thể thao}}(X), D_{\text{Y tế}}(X)) = \text{Công nghệ}$

Vậy theo ví dụ ở hình 3.6 thì X được phân lớp vào nhóm văn bản Công nghệ.

Tuy nhiên nếu điều kiện $\arg\max_{i=,\ldots,n} D_i(x)$ được thỏa mãn đối với nhiều i thì trong trường hợp này cũng không thể xác định được x thuộc lớp nào.



Hình 3.7: Vùng không phân lớp được theo chiến lược OAO

Trong hình trên vùng không thể phân lớp thỏa $D_{Công\ nghệ}(x)=1$, $D_{Y\ tế}(x)=1$, $D_{Giáo\ dục}(x)=1$. Vì vậy, nó chứa các vector x không thể phân lớp.

Để giải quyết vấn đề này Shigeo Abe và Takuya Inoue đã giới thiệu Phân lớp đa lớp mờ [7].

3.3.3 Phân lớp đa lớp mờ (Fuzzy OAO)

Phương pháp phân lớp đa lớp mờ được xây dựng trên phương pháp phân lớp đa lớp OAO kết hợp với việc sử dụng một hàm thành viên để xác định kết quả phân lớp khi vector x nằm trong những vùng không thể phân lớp được tô đậm ở hình 5.5.

Đối với siêu phẳng tối ưu $D_{ij}(x)=0 (\neq i)$ chúng ta định nghĩa các hàm thành viên như sau:

$$m_{ij}(x)=\begin{cases}1 & \text{với } D_{ij}(x)\geq \\ D_{ij}(x) & \text{còn lại}\end{cases}, \qquad (3.32)$$

Từ các $m_{ij}(x), i \neq j = 1,..., n$, chúng ta định nghĩa hàm thành viên thứ i của vector x như sau:

$$m_i(x) = \min_{j=1,...,n} m_{ij}(x) \qquad (3.33)$$

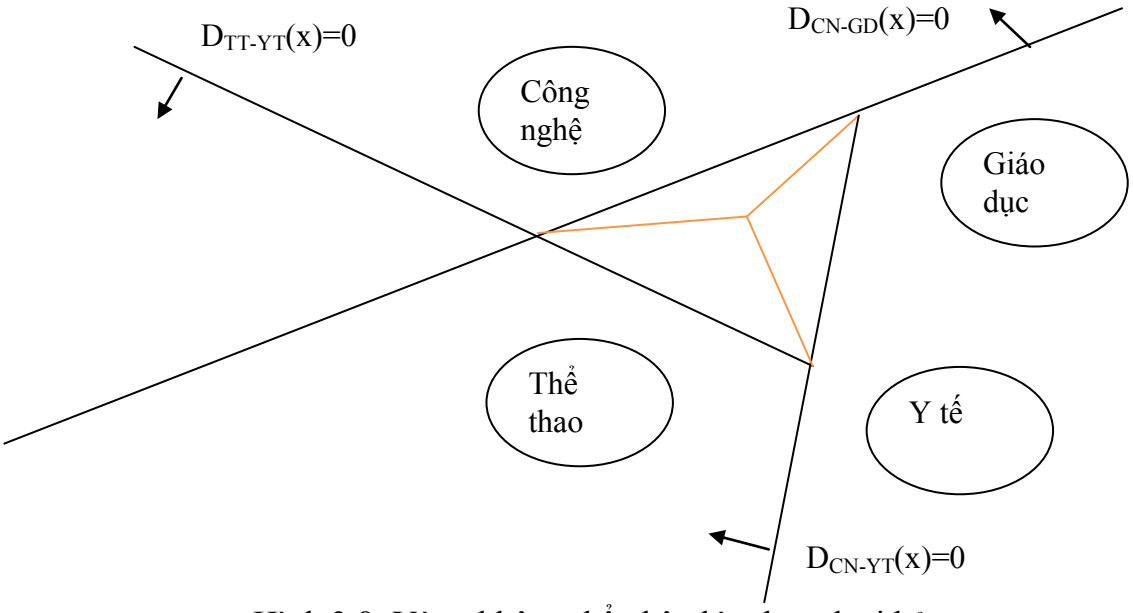
Công thức (3.33) trên tương đương với

$$m_i(x) = \min_{j \neq i, j=1,...n} D_{ij}(x) \qquad (3.34)$$

Bây giờ x được phân lớp vào lớp i theo công thức

$$\arg\max_{i=1,...,n} m_i(x) \qquad (3.35)$$

Vì vậy vùng không phân lớp được của hình 3.7 được giải quyết như trong hình 3.8 dưới đây



Hình 3.8: Vùng không thể phân lớp được loại bỏ

Kết quả thử nghiệm của Shigeo Abe và Takuya Inoue [7] khi so sánh với phương pháp phân lớp pairwise trong bảng kết quả 3.9 cho thấy sự chính xác hơn trong việc phân lớp.

Trong bảng kết quả 3.9, PW và MFSVM là phân lớp pairwise và phương pháp phân lớp đa lớp mờ của tác giả. Bộ dữ liệu thử nghiệm của tác giả gồm dữ liệu

về Blood Cell, Thyroid và Hiragana (dữ liệu 50 mẫu và 13 mẫu). So sánh dựa trên % số mẫu nhận dạng đúng.

Bảng 3.1: Kết quả so sánh phương pháp phân lớp đa lớp mờ

Dữ liệu	**Hàm nhân**	**Tham số**	**PW(%)**	**MFSVM(%)**
Blood cell	Poly	4	91.26	**92.35**
		5	91.03	**92.19**
		6	90.74	**91.74**
	RBF	10	92.52	**91.74**
Thyroid	Poly	4	96.27	**96.62**
	RBF	10	95.10	**95.16**
Hiragana-50	Poly	1	98.00	98.24
		2	98.89	98.94
		4	98.87	**98.94**
	RBF	0.1	99.02	98.02
		0.01	98.81	**98.96**
Hiragana-13	Poly	2	99.46	**99.63**
		3	99.47	**99.57**
		4	99.49	**99.57**
	RBF	1	99.76	99.76
		0.1	99.56	**99.70**

Theo thử nghiệm của tác giả phương pháp phân lớp đa lớp mờ cho kết quả tốt hơn các phương pháp khác, đặc biệt ở những dữ liệu về blood cell là những dữ liệu khó phân lớp.

Như vậy phương pháp phân lớp đa lớp mờ đã giải quyết được tình trạng các văn bản không thể phân lớp được khi phân lớp theo chiến lược OAO.

3.4 Đánh giá các thuật toán phân lớp cải tiến

Thuật toán Fuzzy Suport Vectot Machines (FSVM)

FSVM là một bước cải tiến của SVM bằng cách sử dụng hàm thành viên để làm giảm ảnh hưởng của những điểm dữ liệu nhiễu. Một số kết quả thực nghiệm

cho thấy FSVM có kết quả phân lớp tốt hơn SVM, đặc biệt trong trường hợp dữ liệu có nhiễu.

Thuật toán Suport Vectot Machines Nearest Neighbor (SVM-NN)

- Ưu điểm: các bộ phân lớp SVM-NN thể hiện khả năng phân lớp tốt hơn đáng kể so với bộ phân lớp SVM trong trường hợp số lượng từ đặc trưng thấp. Trong trường hợp số lượng từ đặc trưng lớn, khả năng phân lớp tốt hơn bộ phân lớp SVM là chưa rõ ràng, nhưng vẫn tốt hơn nhiều so với bộ phân lớp K-NN.

- Nhược điểm: tốc độ phân lớp khá chậm, tiêu tốn nhiều tài nguyên, đặc biệt là khi k láng giềng gần có giá trị lớn.

CHƯƠNG 4: TỔNG QUAN VỀ BÀI TOÁN TRUY TÌM VĂN BẢN

Trong chương này chúng ta sẽ khảo sát cơ sở lý thuyết về hệ truy tìm văn bản, các mô hình của hệ truy tìm văn bản. Sau cùng, chúng ta sẽ tìm hiểu chi tiết về hệ truy tìm văn bản theo mô hình không gian vector.

4.1 Hệ truy tìm văn bản

Việc tìm kiếm thông tin văn bản theo truyền thống thì được thực hiện nhân công, ví dụ như, cách nhanh nhất để tìm thông tin trong một quyển sách là đọc và tìm trong bảng mục lục của quyển sách đó.

Đến khi có sự xuất hiện của máy tính thì việc tìm kiếm thông tin nói chung cũng như văn bản nói riêng đã thay đổi hoàn toàn, thậm chí đã có một cuộc cách mạng lớn. Đó là sự xuất hiện của hệ truy tìm thông tin nói chung và hệ truy tìm thông tin văn bản nói riêng. Ngày nay, hệ truy tìm thông tin có một vai trò tối quan trọng không những đối với cuộc sống, công việc hàng ngày của chúng ta mà còn đối với sự phát triển của khoa học công nghệ. Các hệ truy tìm thông tin điển hình được người dùng quan tâm nhiều nhất hiện nay là google, yahoo, …

Định nghĩa

Hệ truy tìm văn bản là một hệ thống giải quyết việc truy tìm những văn bản trong tập văn bản của hệ thống liên quan đến thông tin mà người sử dụng hệ thống cần. Những thông tin được người dùng đưa vào hệ thống bởi các câu truy vấn. Những văn bản liên quan với câu truy vấn sẽ được hệ thống trả về.

Nguyên lý hoạt động

Nguyên lý hoạt động cốt lõi của hệ truy tìm văn bản là tự động quy trình kiểm tra tài liệu bằng cách tính độ đo tương quan giữa câu truy vấn và tài liệu.

Quy trình

Quy trình của hệ truy tìm thông tin văn bản như sau:

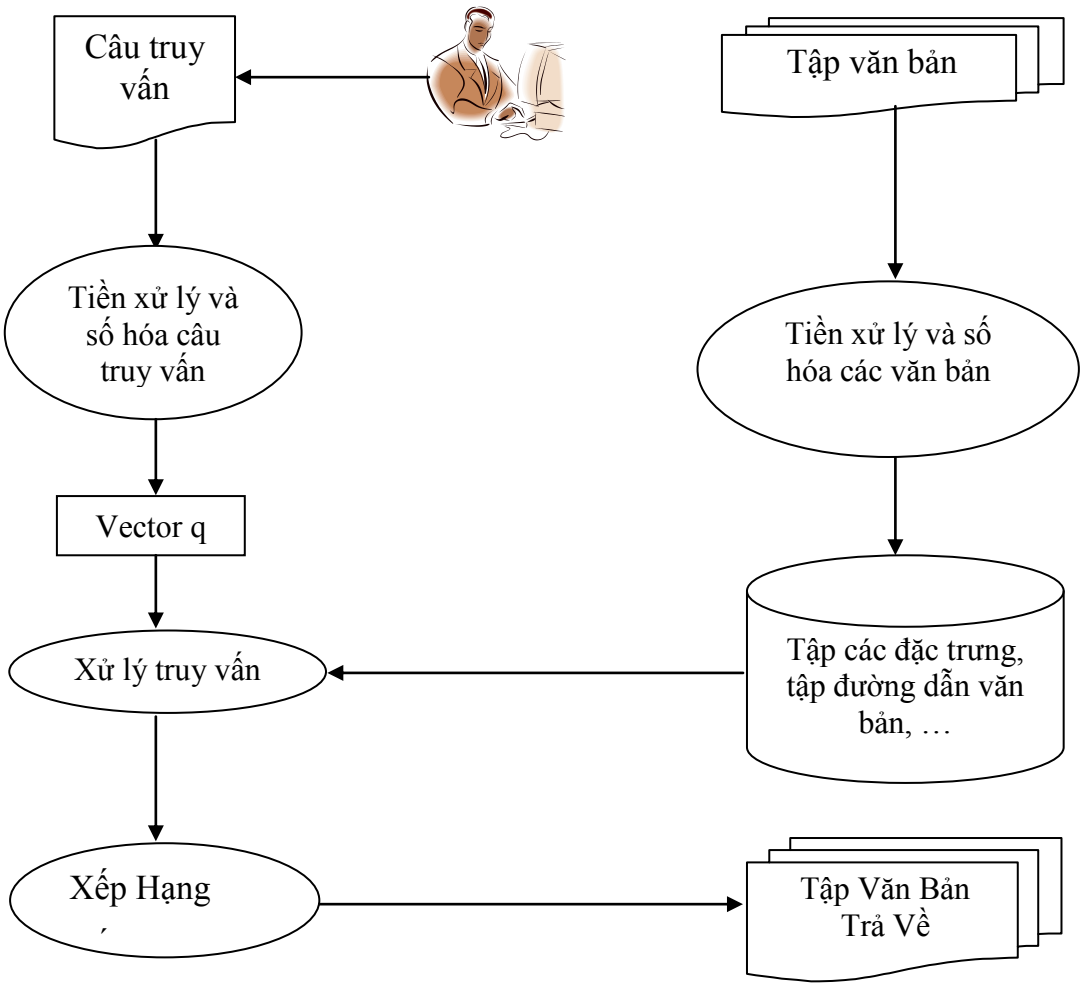
- Người dùng muốn tìm một văn bản liên quan đến một chủ đề nào đó thì người dùng cung cấp một mô tả chủ đề đó dưới dạng câu truy vấn.

- Từ câu truy vấn này, hệ truy tìm sẽ lọc ra những từ đặc trưng.

- Những từ đặc trưng này sẽ được so khớp với những từ đặc trưng của kho văn bản đã được xử lý.

- Hệ thống sẽ trả về những văn bản có độ liên quan cao nhất với câu truy vấn.

Kiến trúc



Hình 4.1: Kiến trúc của hệ truy tìm văn bản

Thành phần chính của kiến trúc trên là việc tiền xử lý và số hóa văn bản, thành phần này có nhiệm vụ chuyển tập văn bản ở ngôn ngữ tự nhiên thành tập các từ đặc trưng có cấu trúc.

4.2 Các mô hình của hệ truy tìm văn bản

Mô hình Boolean

Mô hình Boolean là mô hình cổ điển và đơn giản đã được sử dụng trước đây và cho đến nay vẫn còn được sử dụng trong các hệ thống truy tìm. Mô hình Boolean dựa trên lý thuyết tập hợp (set theory) và đại số Boolean (Boolean algebra). Mô hình Boolean phổ biến bởi vì cả lý thuyết tập hợp và đại số Boolean có mối quan hệ

đơn giản và dễ hiểu, vì vậy các hệ truy tìm được xây dựng trên mô hình này, người dùng dễ dàng sử dụng.

Với mô hình Boolean văn bản được biểu diễn bởi một vector nhị phân, tức là các vector có các phần tử thuộc {0, 1}. Từ đặc trưng thứ k_i xuất hiện trong văn bản d_j thì trọng số $w_{ij} = 1$, ngược lại $w_{ij} = 0$.

Tất cả các truy vấn được biểu diễn bởi các biểu thức Boolean, sử dụng ba phép toán cơ bản: not, and, or.

Văn bản truy vấn sử dụng mô hình này được xem như: hoặc liên quan đến nội dung truy vấn hoặc không, ở đây không có cách để để tìm các văn bản chỉ liên quan cục bộ hay còn gọi là liên quan một phần của câu truy vấn.

Mô hình xác suất

Cho câu truy vấn của người dùng q và văn bản d trong tập văn bản. Mô hình xác suất tính xác suất mà văn bản d liên quan đến câu truy vấn của người dùng. Mô hình giả thiết xác suất liên quan của một văn bản với câu truy vấn phụ thuộc cách biểu diễn chúng. Tập văn bản kết quả được xem là liên quan và có tổng xác suất liên quan với câu truy vấn lớn nhất.

Mô hình không gian vector

Mô hình không gian vector khắc phục những nhược điểm của mô hình boolean là việc sử dụng trọng số cho từ đặc trưng khác trọng số nhị phân (non-binary). Trọng số từ đặc trưng không giới hạn bởi hai trị 0 hoặc 1, các trọng số này được sử dụng để tính toán độ đo tương tự của mỗi văn bản với câu truy vấn.

Với mô hình không gian vector, các văn bản, câu truy vấn và từ đặc trưng được biểu diễn thành các vector trong không gian vector. Sử dụng các phép toán trên không gian vector để tính toán *độ đo tương tự* giữa câu truy vấn và các văn bản hoặc các từ đặc trưng, kết quả sau khi tính toán có thể được xếp hạng theo độ đo tương tự với vector truy vấn.

Ngoài ra, mô hình không gian vector còn hướng dẫn người dùng biết được những văn bản độ tương tự cao hơn có nội dung gần với nội dung họ cần hơn so với các văn bản khác.

So sánh các mô hình

Bảng 4.1: So sánh ưu khuyết của các mô hình truy tìm văn bản

Mô hình	*Ưu điểm*	*Khuyết điểm*
Boolean	Đơn giản, dễ dùng	- Số lượng văn bản trả về tùy thuộc vào số từ xuất hiện của câu truy vấn có liên quan hay không. - Văn bản trả về không được quan tâm đến thứ tự quan hệ với câu truy vấn. - Vì dựa trên phép toán logic nhị phân nên một văn bản được tìm kiếm chỉ xác định hai trạng thái: liên quan hoặc không với câu truy vấn. - Việc chuyển một câu truy vấn của người dùng sang dạng biểu thức Boolean không đơn giản.
Xác suất	- Các văn bản được sắp xếp dựa vào xác suất liên quan đến câu truy vấn.	- Mô hình không quan tâm đến số lần xuất hiện của từ chỉ mục trong văn bản - Việc tính toán xác suất khá phức tạp và tốn nhiều chi phí.
Không gian vector	- Đơn giản, dễ dùng. - Có quan tâm đến việc xếp hạng các văn bản theo mức độ liên quan. - Khắc phục các hạn chế trên mô hình Boolean	- Các văn bản trả về tuy cải thiện hơn mô hình boolean nhưng vẫn không có quan hệ về nghĩa với câu truy vấn. - Số chiều ma trận có thể rất lớn nên hạn chế về mặt lưu trữ và thời gian.

4.3 Hệ truy tìm văn bản theo mô hình không gian vector

4.3.1 Giới thiệu hệ truy tìm văn bản theo mô hình không gian vector

Mô hình tổng quát của hệ truy tìm văn bản theo không gian vector là một bộ [D, Q, F, R(q_i, d_j)]. Trong đó:

- D là tập văn bản.
- Q là các câu truy vấn.
- F là mô hình biểu diễn tập văn bản, câu truy vấn và các quan hệ của chúng.
- R(q_i, d_j) là hàm xếp hạng theo đo độ tương tự giữa câu truy vấn $q_i \in$ và văn bản $d_j \in$. Hàm xếp hạng xác định một thứ tự về mức độ liên quan của các văn bản với câu truy vấn q_i.

Mô hình không gian vector sẽ mô tả tất cả các văn bản trong tập văn bản thành một tập các từ đặc trưng sau khi đã loại bỏ các từ ít có ý nghĩa. Các từ đặc trưng này cũng chính là các từ chứa nội dung chính của tập văn bản.

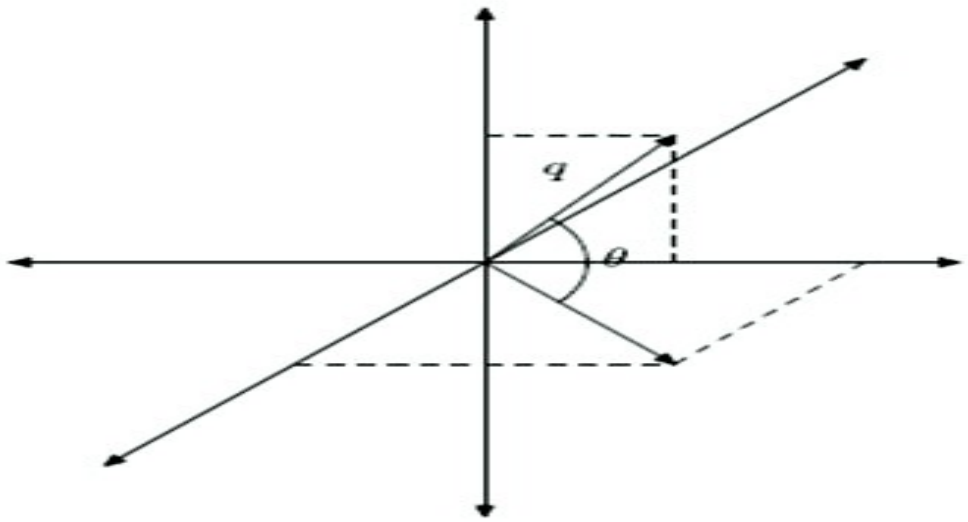
Mô hình không gian vector dựa trên giả thiết là nội dung của văn bản có thể được hiểu như sự kết hợp của các từ đặc trưng. Một văn bản d được biểu diễn như một vector của các từ đặc trưng $\mathbf{d} = (t_1, t_2 \ldots, t_n)$ với t_i là trọng số của từ đặc trưng thứ i với $1 \leq i \leq n$. Mỗi từ đặc trưng này được gán một trọng số (có thể là số lần xuất hiện của từ đặc trưng t_i trong văn bản d), trọng số của một từ đặc trưng nói lên sự liên quan của nó đến nội dung của một văn bản. Mỗi từ đặc trưng trong văn bản biểu diễn một chiều (dimension) trong không gian.

Tương tự, câu truy vấn cũng được biểu diễn như một vector $\mathbf{q} = (\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_n)$.

Sau khi đã biểu diễn tập văn bản và câu truy vấn thành các vector trong không gian vector, ta có thể sử dụng các phép toán trên không gian vector để tính toán độ đo Cosine (độ đo tương tự) giữa các vector văn bản và vector truy vấn. Kết quả sau khi tính toán có thể được xếp hạng theo các độ đo tương tự trên.

Giả sử, mỗi văn bản d được biểu diễn bằng một vector một chiều của các từ đặc trưng $\mathbf{d} = (t_1, t_2 \ldots, t_n)$ với t_i là từ chỉ mục thứ i (1=<i<=n) trong văn bản d.

Tương tự, câu truy vấn cũng được biểu diễn bằng một vector $q = (\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_n)$. Lúc đó độ đo tương tự của văn bản d và câu truy vấn q chính là độ đo Cosine của chúng.



Hình 4.2 Góc giữa vector truy vấn và vector văn bản

4.3.2 Tiền xử lý và số hóa văn bản theo mô hình không gian vector

Hệ truy tìm văn bản cùng sử dụng các phương pháp tách từ tiếng Việt, các kỹ thuật lựa chọn từ đặc trưng, các phương pháp biểu diễn văn bản của hệ phân lớp văn bản đã trình bày ở chương 1, vì cả 2 hệ này đều dựa trên mô hình không gian vector.

4.3.3 Ma trận biểu diễn tập văn bản

Trong mô hình không gian vector một tập có n văn bản được biểu diễn bởi m từ đặc trưng được vector hóa thành một ma trận gọi là *ma trận từ đặc trưng – văn bản*. Trong đó, n văn bản trong tập văn bản được biểu diễn thành n vector cột, m từ đặc trưng được biểu diễn thành m dòng. Do đó phần tử d_{ij} của ma trận A chính là trọng số của từ đặc trưng i xuất hiện trong văn bản j. Thông thường, trong một tập văn bản số từ đặc trưng lớn hơn rất nhiều so với văn bản m >> n.

$$A = \begin{pmatrix} d_{11} & d_{21} & \bullet & \bullet & \bullet & d_{1n} \\ d_{12} & d_{22} & \bullet & \bullet & \bullet & d_{2n} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ d_{m1} & d_{m2} & \bullet & \bullet & \bullet & d_{mn} \end{pmatrix}$$

Hình 4.3 Ma trận từ đặc trưng – văn bản

Ví dụ 1: Giả sử

Ta có n = 5 văn bản như sau:

D1: How to Bake Bread without Recipes

D2: The Classic Art of Viennese Pastry

D3: Numerical Recipes: The Art of Scientific Computing

D4: Breads, Pastries, Pies and Cakes : Quantity Baking Recipes

D5: Pastry: A Book of Best French Recipes

Ta có m = 6 từ chỉ mục cho các văn bản trên – các từ gạch chân

T1: bak(e, ing)

T2: recipe(s)

T3: bread(s)

T4: cake(s)

T5: pastr(y, ies)

T6: pie(s)

Với 5 văn bản và 6 từ chỉ mục ta biểu diễn ma trân từ đặc trưng – văn bản A_{6x5} như sau:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

4.3.4 Truy vấn văn bản theo mô hình không gian vector

Trong mô hình không gian vector, việc truy vấn tập dữ liệu văn bản để tìm những văn bản liên quan với câu truy vấn dựa vào các kỹ thuật tính toán trên mô hình không gian vector. Một câu truy vấn được xem như tập các từ đặc trưng và được biểu diễn như các văn bản trong tập văn bản.Vì câu truy vấn rất ngắn nên có rất nhiều từ đặc trưng của tập văn bản không xuất hiện trong câu truy vấn, có nghĩa là hầu hết các thành phần của vector truy vấn là zero. Thủ tục truy vấn chính là tìm các văn bản trong tập văn bản liên quan với câu truy vấn hay còn gọi là các văn bản có độ đo tương tự "cao" với câu truy vấn. Theo cách biểu diễn hình học, các văn bản được chọn là các văn bản gần với câu truy vấn nhất theo một độ đo (measure) nào đó.

Độ đo thường được sử dụng nhất là độ đo Cosine của góc giữa vector truy vấn và vector văn bản. Nếu ma trận từ đặc trưng – văn bản có các cột được ký hiệu là d_j , j = 1, …, n thì n độ đo Cosine của vector truy vấn q với n văn bản trong tập văn bản được tính theo công thức:

$$\cos\theta_j = \frac{d_j^T q}{\|d_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^{m} d_{ij} q_i}{\sqrt{\sum_{i=1}^{m} d_{ij}^2} \sqrt{\sum_{i=1}^{m} q_i^2}} \qquad (4.1)$$

Sử dụng tập văn bản trong ví dụ 1 ở trên để ví dụ cho thủ tục truy vấn, dựa trên công thức (4.1) tính góc của các vector trong không gian vector 6 chiều ($\Re$). Giả sử người sử dụng cần tìm thông tin 'baking bread'. Với câu truy vấn trên tương ứng với vector truy vấn là: $q^{(1)} = (\ 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1\ )^T$

với các phần tử khác 0 cho hai từ baking và bread. Việc tìm kiếm các văn bản liên quan được thực hiện bằng cách tính Cosine của các góc θ_j giữa vector truy vấn $q^{(1)}$ với các vector văn bản d_j bằng công thức (4.1). Một văn bản được xem như liên quan (relevant) và được trả về nếu Cosine của góc được tạo bởi vector truy vấn và vector văn bản đó lớn hơn một ngưỡng (threshold) cho trước. Trong cài đặt thực tế ngưỡng được kiểm nghiệm và quyết định bởi người xây dựng hệ thống. Nhưng đối với ví dụ nhỏ này chỉ sử dụng ngưỡng là 0.5.

Với vector truy vấn $q^{(1)}$, chỉ có giá trị Cosine của các góc khác zero: $\cos\theta = 0.8165$ và $\cos\theta = 0.5774$. Vậy các văn bản liên quan đến baking và bread là D1 và D4 được trả về, các văn bản D2, D3 và D5 không liên quan và được bỏ qua.

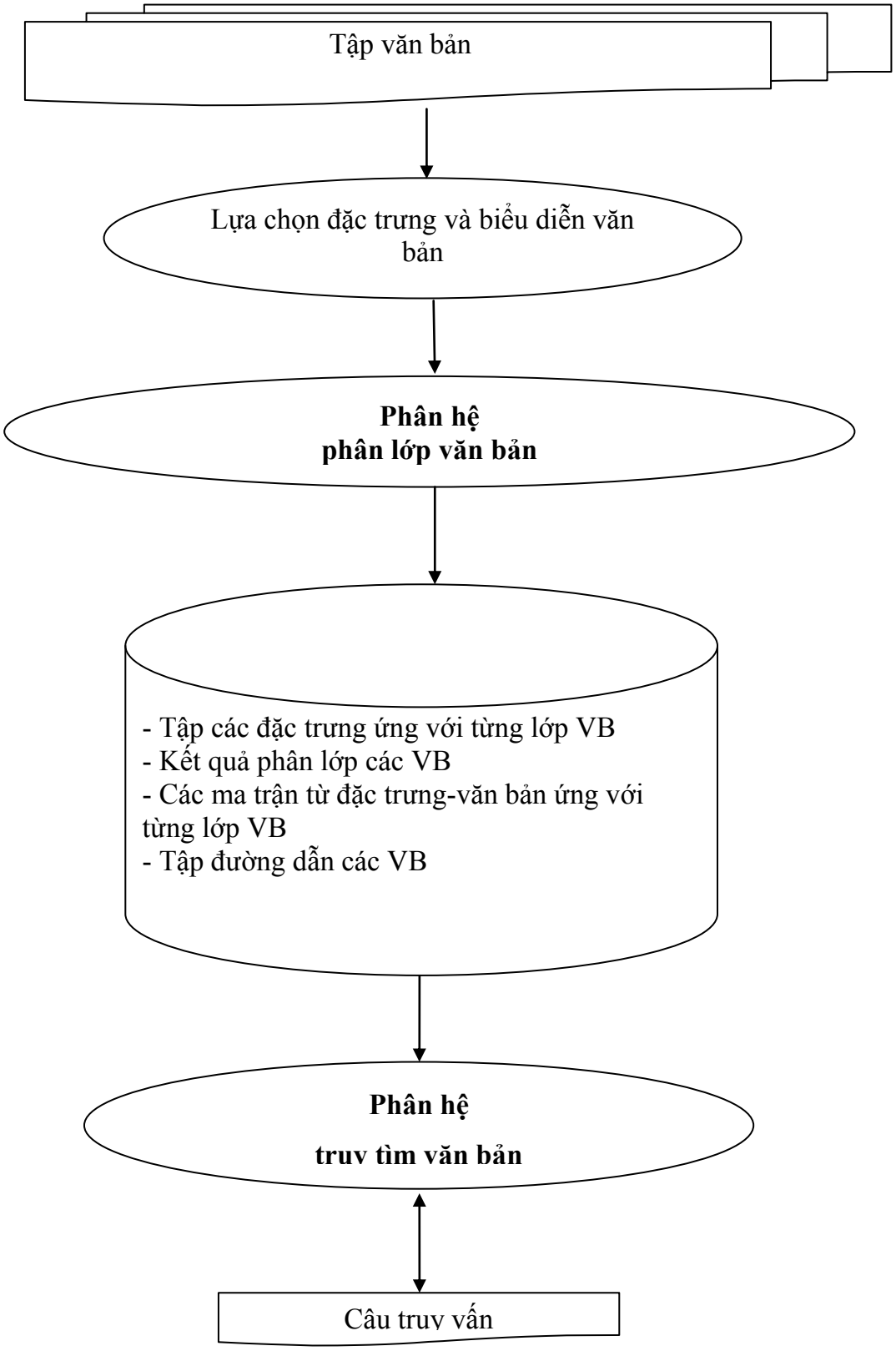
CHƯƠNG 5: XÂY DỰNG THỬ NGHIỆM HỆ PHÂN LỚP VÀ TRUY TÌM VĂN BẢN

Nội dung chương này sẽ trình bày các bước xây dựng hệ phân lớp và truy tìm văn bản. Đầu tiên, trình bày việc xây dựng các thành phần của phân hệ phân lớp văn bản với tập văn bản thử nghiệm thuộc các lĩnh vực: giáo dục, y tế, công nghệ, thể thao. Sau đó tiếp tục trình bày việc xây dựng các thành phần của phân hệ truy tìm văn bản trên tập văn bản đã được phân lớp bên trên.

Hệ thống gồm 2 phân hệ chính đó là phân hệ phân lớp văn bản và phân hệ truy tìm văn bản. Hệ thống cài đặt áp dụng phương pháp phân lớp văn bản 2 lớp cải tiến của SVM là SVM-NN kết hợp chiến thuật phân loại đa lớp OAO, Fuzzy OAO cho phân hệ phân lớp văn bản, cài đặt áp dụng phương pháp truy tìm văn bản theo mô hình không gian vector cho phân hệ truy tìm văn bản.

Tập văn bản sẽ được phân hệ phân lớp phân ra thành các nhóm văn bản. Sau đó, phân hệ truy tìm sẽ đáp ứng việc truy tìm văn bản dựa vào kết quả phân lớp trên tập văn bản. Bằng việc kết hợp với phân hệ phân lớp, phân hệ truy tìm sẽ cải thiện đáng kể tốc độ, hiệu quả truy tìm vì không phải thực hiện truy tìm trên toàn bộ tập văn bản mà chỉ thực hiện truy tìm trên một hoặc vài nhóm văn bản có liên quan.

Sơ đồ thực hiện của hệ phân lớp và truy tìm văn bản được mô tả như sau:

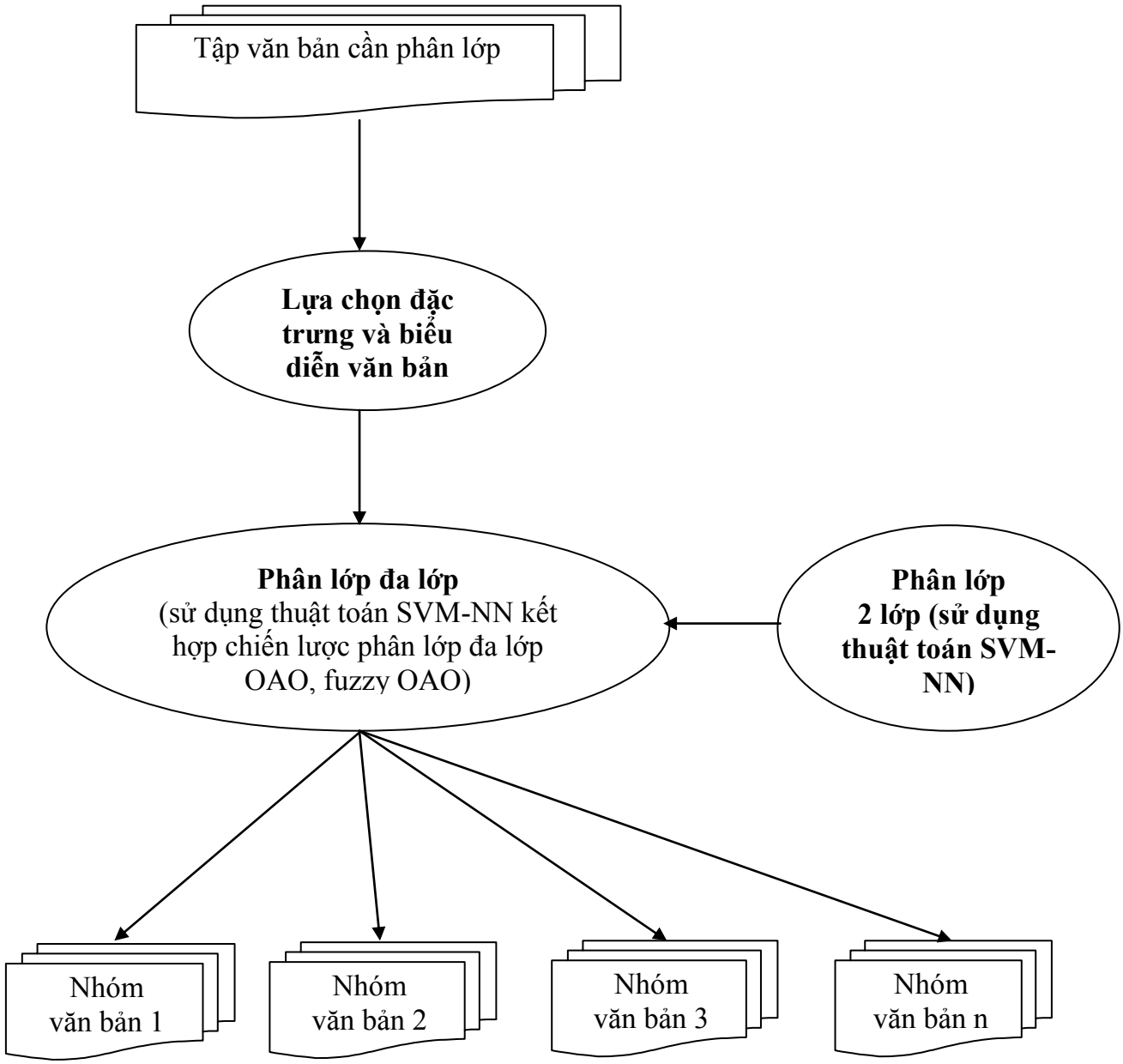


Hình 5.1: Sơ đồ thực hiện của hệ phân lớp và truy tìm văn bản

5.1 Phân hệ phân lớp văn bản

5.1.1 Thiết kế phân hệ phân lớp văn bản

Kiến trúc của phân hệ phân lớp văn bản



Hình 5.2: Kiến trúc của phân hệ phân lớp văn bản

Các modul của phân hệ phân lớp văn bản

Phân hệ phân lớp văn bản bao gồm các modul chính như sau:

- Module lựa chọn các từ đặc trưng và biểu diễn văn bản tiếng Việt.
- Module phân lớp 2 lớp sử dụng thuật toán SVM-NN.

- Module phân lớp đa lớp (sử dụng thuật toán SVM-NN kết hợp chiến lược phân lớp đa lớp OAO hoặc Fuzzy OAO).

5.1.2 Module lựa chọn các từ đặc trưng và biểu diễn văn bản tiếng Việt

Luận văn thực hiện tách từ bằng phương pháp xây dựng các ôtômát để tách các văn bản tiếng Việt thành các từ là đầu vào của bài toán.

Số các từ tách được từ tất cả các văn bản trong một nhóm là rất lớn. Tuy nhiên chỉ có một số từ là có ảnh hưởng đến việc phân lớp, những từ này gọi là những từ đặc trưng. Vì vậy ta phải tìm những từ đặc trưng này để giảm số chiều biểu diễn văn bản và tăng độ chính xác và tốc độ phân lớp. Cách thực hiện như sau:

- Trước tiên ta loại bỏ những từ không quan trọng trong văn bản như là các dấu câu, các con số, các hư từ. Những từ này xuất hiện thường xuyên trong tất cả các văn bản nên không thể xem là từ đặc trưng.

- Những từ đặc trưng là những từ xuất hiện nhiều trong nhóm và có mức độ ảnh hưởng đến nhóm nhiều nhất. Tập đặc trưng được lựa chọn như sau: $T = \{t \in Dr \mid \#t \geq k \text{ và } IG(t,c) \geq \theta\}$ trong đó *Dr* là tập từ điển ban đầu, #t là số lần xuất hiện của t trong toàn bộ tập dữ liệu huấn luyện, IG(t,c) là lợi nhuận thông tin của từ t đối với phân lớp c (tính theo công thức Information Gain), *k là ngưỡng chỉ số lần xuất hiện của t trong tập dữ liệu huấn luyện, θ là ngưỡng để đánh giá lợi nhuận thông tin của từ t đối với phân lớp c*. Tập các từ trong T, được coi là các đặc trưng để biểu diễn các văn bản của tập dữ liệu huấn luyện cũng như tập dữ liệu kiểm tra.

Quá trình xây dựng Module lựa chọn các từ đặc trưng và biểu diễn văn bản gồm 3 Module con: Module tạo các tập tin tách từ, Module tạo tập tin đặc trưng, Module tạo vector trọng số W của các từ đặc trưng.

Sau khi lựa chọn được các đặc trưng ta sẽ biểu diễn các văn bản dưới dạng một vector $x_i=(w_{i1}, w_{i2}, ..., w_{i|T|})$, trong đó w_{ij} là trọng số của từ t_j trong văn bản d_i, $(t_j \in T)$ được tính bằng phương pháp TFxIDF.

5.1.3 Module phân lớp 2 lớp sử dụng thuật toán SVM-NN

Ta phải xây dựng một bộ phân lớp 2 lớp sử dụng thuật toán SVM-NN để phân lớp văn bản thuộc 2 nhóm, giữa một chủ đề cần quan tâm và một chủ đề khác.

Quá trình xây dựng một bộ phân lớp 2 lớp sử dụng thuật toán SVM-NN gồm 2 Module con: Modul huấn luyện theo mô hình SVM và Modul đưa ra quyết định phân lớp SVM-NN.

Module huấn luyện theo mô hình SVM:

Input :

- Tập các vector biểu diễn văn bản huấn luyện V_{Tr} đã được gán nhãn.
- Giá trị của tham số C.
- Tham số d của hàm nhân $K(x_i, x_j)$

Thuật toán :

Bước 1 : Sử dụng thuật toán SMO để thực hiện quá trình huấn luyện. Kết thúc thuật toán này chúng ta tìm được giá trị tối ưu của α

Bước 2 : Lưu lại các giá trị α tối ưu sử dụng trong module ra quyết định phân lớp.

Output :

Các hệ số α tối ưu của siêu phẳng tối ưu, *b*.

Module đưa ra quyết định phân lớp SVM-NN:

Input :

- Tập các vector biểu diễn văn bản huấn luyện V_{Tr} đã được gán nhãn.
- Vector cần phân lớp x.
- Giá trị của tham số k láng giềng gần nhất.
- Tham số d của hàm nhân $K(x_i, x_j)$ (dùng làm đối số cho *modul huấn luyện theo mô hình SVM*)

Thuật toán :

- Tìm k mẫu (xi, yi) thuộc V_{Tr} với giá trị nhỏ nhất của $K(x_i, x_i) - 2 * K(x_i, x)$
- Thực hiện huấn luyện SVM trên tập văn bản huấn luyện $V_{Tr\text{-}K}$ (tập k mẫu tìm được bên trên - tập con của tập V_{Tr}) bằng cách sử dụng *modul huấn luyện theo mô hình SVM*. Kết quả huấn luyện ta thu được: các hệ số α tối ưu của siêu phẳng tối ưu, giá trị b.

Với vector *x,* tính giá trị của hàm $f(x) = \sum \alpha_i K(x_i, x) + b$, với x_i thuộc $V_{Tr\text{-}K}$

- Nếu *f(x)>0* thì *x* được gán nhãn là 1.
- Ngược lại *x* được gán nhãn là –1.

Output :

Đưa ra quyết định phân lớp cho văn bản x.

5.1.4 Modul phân lớp đa lớp

Ta sử dụng chiến lược phân lớp OAO, Fuzzy OAO (chiến lược phân lớp đa lớp mờ) do ưu điểm của nó so với chiến lược OAR như đã trình bày ở phần trên.

Quá trình xây dựng một phân lớp đa lớp gồm 2 Module con: Modul xây dựng các bộ phân lớp 2 lớp và Modul xây dựng bộ phân lớp đa lớp.

Module xây dựng các bộ phân lớp 2 lớp

Xây dựng các bộ phân lớp 2 lớp SVM-NN từ các cặp nhóm văn bản và văn bản cần phân lớp. Ta có 04 nhóm văn bản: Y tế, Giáo dục, Công nghệ, Thể thao thì theo OAO sẽ có 4(4-1)/2 = 6 bộ phân lớp 2 lớp SVM-NN: (Công nghệ – Giáo dục), (Công nghệ - Thể thao), (Công nghệ - Y tế), (Giáo dục – Thể thao), (Giáo dục – Y tế), (Thể thao – Y tế).

Input:

- Các nhóm văn bản Y tế, Giáo dục, Công nghệ, Thể thao.
- Văn bản cần phân lớp
- Giá trị của tham số k láng giềng gần nhất (làm đối số cho *Module phân lớp 2 lớp sử dụng thuật toán SVM-NN*).

Thuật toán:

Đối với mỗi cặp nhóm văn bản (Công nghệ – Giáo dục), (Công nghệ - Thể thao), (Công nghệ - Y tế), (Giáo dục – Thể thao), (Giáo dục – Y tế), (Thể thao – Y tế) ta sử dụng *Module phân lớp 2 lớp sử dụng thuật toán SVM-NN* được xây dựng bên trên để xây dựng bộ phân lớp cho mỗi cặp nhóm văn bản này.

Output:

Các bộ phân lớp 2 lớp SVM-NN: (Công nghệ – Giáo dục), (Công nghệ - Thể thao), (Công nghệ - Y tế), (Giáo dục – Thể thao), (Giáo dục – Y tế), (Thể thao – Y tế).

Module 2: Xây dựng bộ phân lớp đa lớp

Ta xây dựng một bộ phân lớp đa lớp dành cho việc phân lớp văn bản thuộc các nhóm Y tế, Giáo dục, Công nghệ, Thể thao.

Input:

- Các văn bản cần phân lớp.

Thuật toán:

Đối với mỗi văn bản cần phân lớp:

- Ta sẽ xây dựng các bộ phân lớp 2 lớp SVM-NN (Công nghệ – Giáo dục), (Công nghệ - Thể thao), (Công nghệ - Y tế), (Giáo dục – Thể thao), (Giáo dục – Y tế), (Thể thao – Y tế).
- Sau đó áp dụng chiến thuật phân lớp đa lớp OAO để phân lớp văn bản này. Trường hợp không phân lớp văn bản này được ta sẽ áp dụng chiến thuật phân lớp đa lớp mờ Fuzzy OAO.

Output:

- Các văn bản được phân vào các lớp thích hợp.

5.1.5 Cài đặt phân hệ phân lớp văn bản

Phân hệ phân lớp văn bản được cài đặt như thiết kế trình bày ở hình 5.2.

Các bước thực hiện như sau:

Bước 1: Huấn luyện

- Chuẩn bị huấn luyện:

+ Các văn bản huấn luyện của từng nhóm văn bản được đưa vào từng thư mục con Y tế, Giáo dục, Công nghệ, Thể thao trong thư mục Nhóm Văn Bản ở thư mục gốc của chương trình.

+ Chạy *module Tokenizer* để tạo các tập tin tách từ của các nhóm văn bản trên.

+ Chạy *module SelectTerm* để tạo tập tin đặc trưng của các nhóm văn bản.

+ Chạy *module CalWVector* để tạo vector trọng số W của các từ đặc trưng của từng nhóm văn bản.

- Huấn luyện:

+ Chạy *modul SVM-NN* để huấn luyện các bộ phân lớp 2 lớp SVM-NN cho từng cặp nhóm văn bản: (Công nghệ – Giáo dục), (Công nghệ - Thể thao), (Công nghệ - Y tế), (Giáo dục – Thể thao), (Giáo dục – Y tế), (Thể thao – Y tế). Chương trình sẽ tạo ra các tập tin kết quả huấn luyện nằm trong thư mục *resource*.

Cấu trúc các thư mục dữ liệu

+ Cho bước chuẩn bị huấn luyện được tổ chức như sau:

Thư mục *Nhóm văn bản* chứa các thư mục con:

- *Y te:* chứa các văn bản huấn luyện lĩnh vực Y tế.
- *Giao duc:* chứa các văn bản huấn luyện lĩnh vực Giáo dục.
- *Cong nghe:* chứa các văn bản huấn luyện lĩnh vực Công nghệ.
- *The thao:* chứa các văn bản huấn luyện lĩnh vực Thể thao.

+ Cho bước sau khi chuẩn bị huấn luyện được tổ chức như sau:

Trong mỗi thư mục của các nhóm văn bản có hai thư mục con:

- *Parse:* chứa các tập tin tách từ khi chạy *module Tokenizer*.
- *DacTrung:* chứa tập tin *"dac trung.txt"* là tập tin chứa các từ đặc trưng của nhóm văn bản khi chạy *modul SelectTerm*, tập tin *"Wvector.txt"* chứa trọng số của các từ đặc trưng tính theo phương pháp nghịch đảo tần số văn bản (IDF) khi chạy *module CalWVector*.

+ Cho bước sau khi huấn luyện được tổ chức như sau:

Thư mục *resource* chứa các tập tin dữ liệu cần cho quá trình huấn luyện. Trong thư mục *resource* có thư mục con *svm-nn* chứa các tập tin kết quả sau khi huấn luyện, gồm các tập tin:

- *"svm-nn_ congnghe_giaoduc.txt"*: Bộ phân lớp giữa lĩnh vực công nghệ và giáo dục.
- *"svm-nn_ congnghe_ thethao.txt"*: Bộ phân lớp giữa lĩnh vực công nghệ và thể thao.
- *"svm-nn_ congnghe_yte.txt"*: Bộ phân lớp giữa lĩnh vực công nghệ và y tế.
- *"svm-nn_giaoduc_thethao.txt"*: Bộ phân lớp giữa lĩnh vực giáo dục và thể thao.
- *"svm-nn_giaoduc_yte.txt"*: Bộ phân lớp giữa lĩnh vực giáo dục và y tế.
- *"svm-nn_thethao_yte.txt"*: Bộ phân lớp giữa lĩnh vực thể thao và y tế.

Bước 2: Tiến hành phân lớp các văn bản

- Chạy *module Tokenizer* để tạo các tập tin tách từ của các văn bản cần phân lớp.

- Sau đó, đối với từng văn bản cần phân lớp:

+ Chạy *modul SVM-NN* trên từng bộ phân lớp 2 lớp đã được tạo ra trong quá trình huấn luyện, để thực hiện các phân lớp 2 lớp SVM-NN cho từng văn bản đó.

+ Chạy *modul Classify* để thực hiện phân lớp đa lớp cho từng văn bản đó.

- Kết quả phân lớp của toàn bộ các văn bản cần phân lớp được lưu trong tập tin chứa kết quả phân lớp *ketquaphanlop.txt*.

Lưu ý: Tập tin chứa kết quả phân lớp *ketquaphanlop.txt* sẽ được sử dụng làm dữ liệu đầu vào cho phân hệ truy tìm văn bản.

5.1.6 Kết quả thử nghiệm của phân hệ phân lớp văn bản

Bảng 5.1 dưới đây sẽ trình bày kết quả thử nghiệm phân hệ phân lớp văn bản sử dụng phương pháp phân lớp cải tiến SVM-NN kết hợp chiến thuật phân loại đa lớp OAO, Fuzzy OAO. Tập văn bản thử nghiệm gồm 820 văn bản huấn luyện, 120 văn bản kiểm tra thuộc 4 lĩnh vực (công nghệ, giáo dục, thể thao, y tế). Thuật toán SVM-NN với tham số k láng giềng gần được chọn là 50, tham số C là 20, tham số d của hàm nhân đa thức là 2. Kết quả thử nghiệm cho thấy độ chính xác của phương pháp phân lớp trên là khá cao.

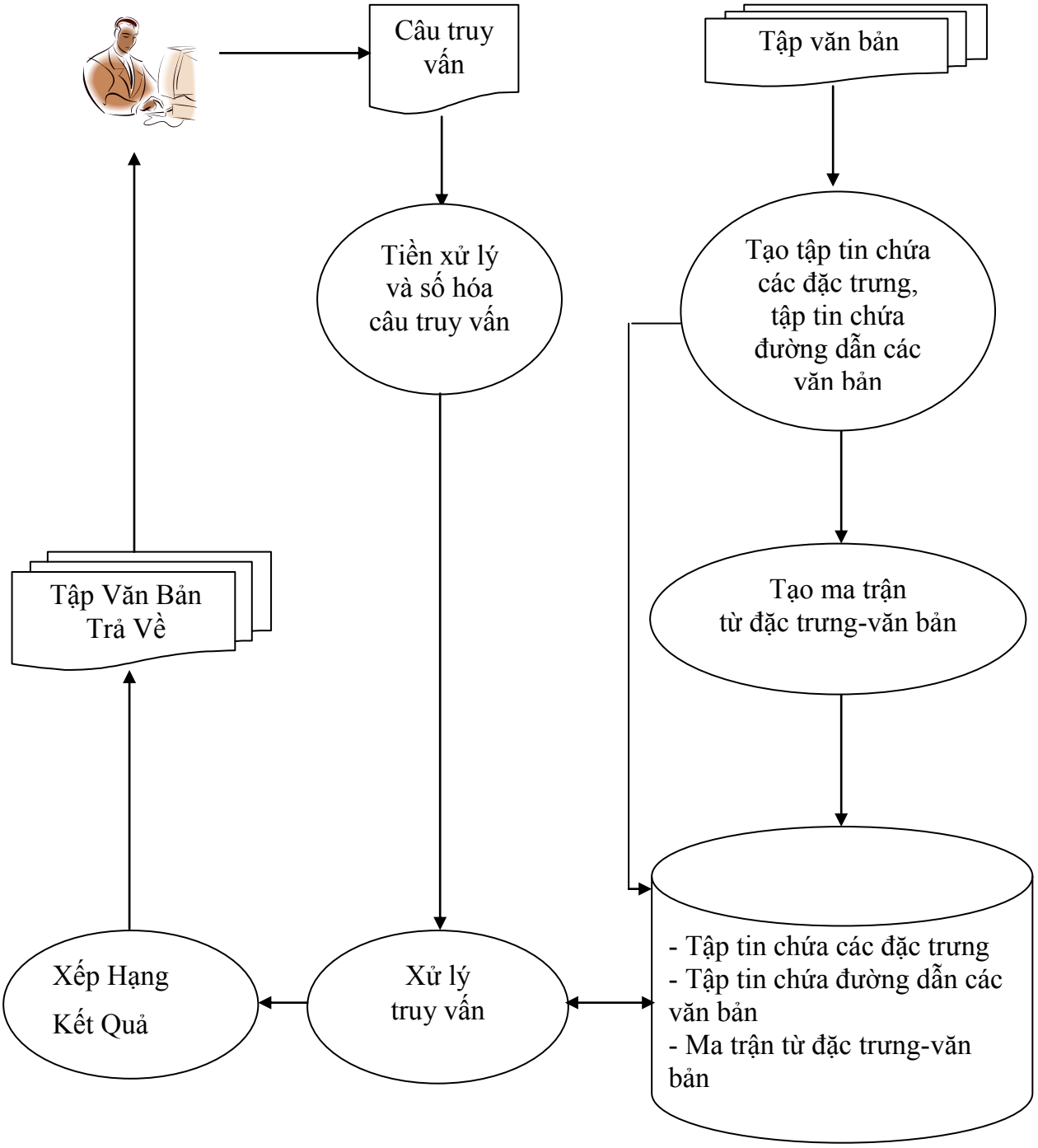
Bảng 5.1: Kết quả thử nghiệm phân hệ phân lớp văn bản

STT	Nhóm	Số VB được phân loại	Số VB được phân loại đúng	Tỷ lệ % VB được phân loại đúng
1.	Công nghiệp	30	26	86,66
2.	Giáo dục	30	24	80
3.	Thể thao	30	26	86,66
4.	Y tế	30	25	83,33

5.2 Phân hệ truy tìm văn bản VSM

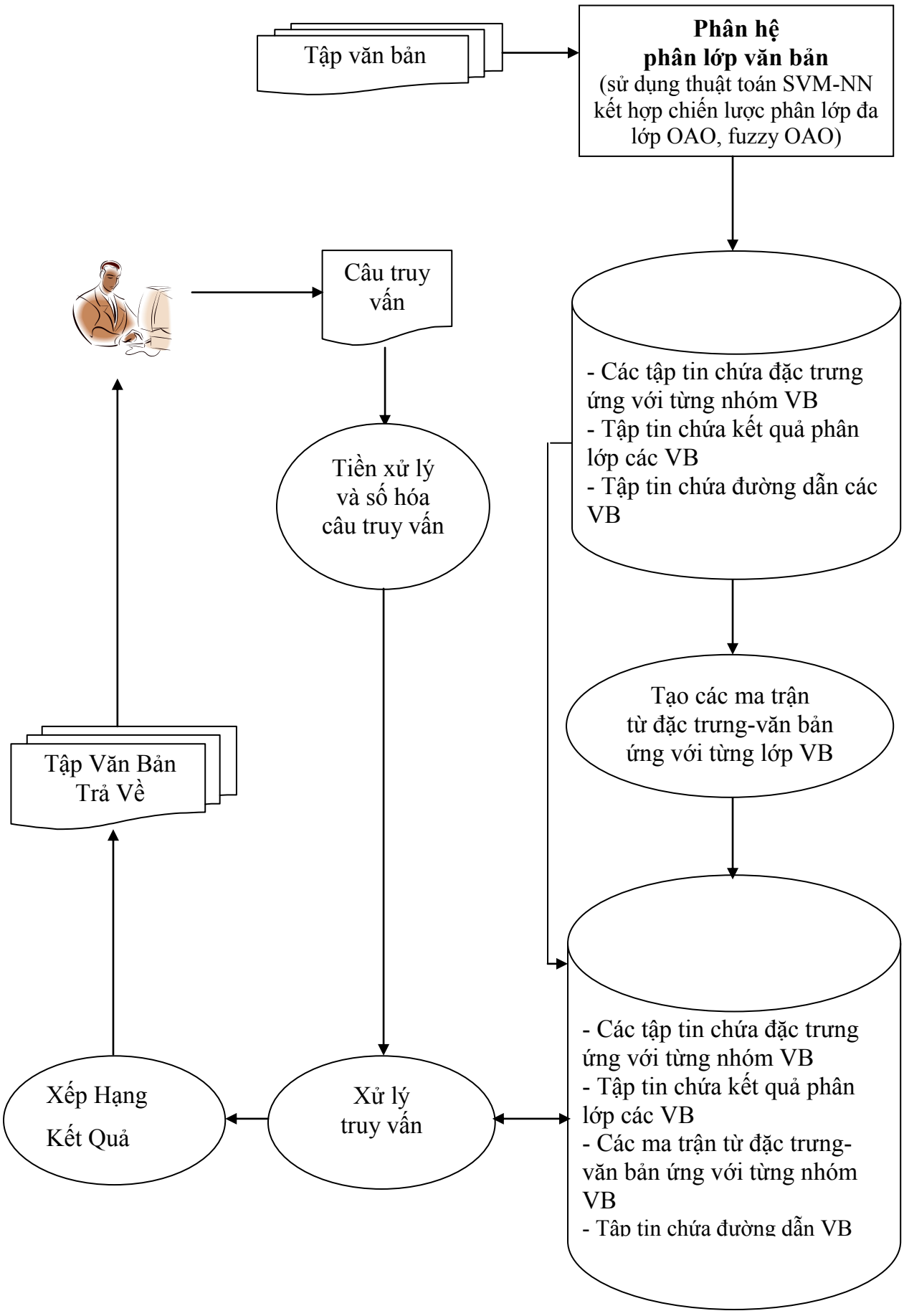
5.2.1 Thiết kế phân hệ truy tìm văn bản VSM

Kiến trúc của phân hệ truy tìm văn bản VSM



Hình 5.3: Kiến trúc cơ bản của phân hệ truy tìm văn bản VSM

Kiến trúc trên chỉ là kiến trúc cơ bản của phân hệ truy tìm văn bản. Mục tiêu của luận văn là sau khi nghiên cứu các phương pháp phân lớp cải tiến, chúng ta sẽ ứng dụng kết quả phân lớp của các phương pháp đó vào phân hệ truy tìm văn bản nhằm mục đích cải thiện tốc độ, hiệu quả truy tìm. Bằng việc kết hợp với phân hệ phân lớp văn bản sử dụng phương pháp SVM-NN và chiến lược phân lớp đa lớp OAO; Fuzzy OAO, chúng ta xây dựng được một mô hình truy tìm văn bản mới có kiến trúc được cải tiến như sau:



Hình 5.4: Kiến trúc cải tiến của phân hệ truy tìm văn bản VSM

Các modul của phân hệ truy tìm văn bản VSM

Phân hệ truy tìm văn bản bao gồm 2 modul chính như sau:

- Modul tạo ma trận từ đặc trưng-văn bản.

- Modul xử lý truy tìm bao gồm các chức năng:

    + Tính các độ đo Cosin.

    + Xếp hạng kết quả truy tìm.

    + Giao diện thực hiện truy vấn và hiển thị kết quả trả về.

Modul tạo ma trận từ đặc trưng-văn bản

Phân hệ phân lớp văn bản sau khi thực hiện sẽ cung cấp dữ liệu đầu vào cho phân hệ truy tìm văn bản: các tập tin chứa đặc trưng ứng với từng nhóm văn bản đã được phân lớp, tập tin chứa kết quả phân lớp của các văn bản, tập tin chứa đường dẫn các văn bản. Từ các tập tin chứa đặc trưng ứng với từng nhóm văn bản đã được phân lớp, mỗi văn bản được vector hoá thành một vector và mỗi nhóm văn bản sẽ được biểu diễn thành mỗi ma trận ứng với nhóm văn bản đó. Mỗi cột của ma trận biểu diễn vector của mỗi văn bản. Mỗi ma trận ứng với từng nhóm văn bản sẽ được lưu trong một tập tin.

Module xử lý truy tìm

Chức năng tính các độ đo Cosin

Modul này thực hiện truy tìm các văn bản trong tập văn bản liên quan với câu truy vấn (các văn bản có độ đo Cosine "cao" với câu truy vấn) bằng cách tính độ đo Cosine của từng vector cột (của ma trận từ đặc trưng-văn bản) với vecor truy vấn. Một văn bản được xem như liên quan và được trả về nếu độ đo Cosine của vector truy vấn với vector văn bản đó lớn hơn một ngưỡng (threshold). Trong cài đặt của module này, ngưỡng được chọn là 0.04.

Các bước thực hiện cơ bản:

- Thực hiện lọc ra tất cả các từ đặc trưng trong câu truy vấn bằng cách so sánh nó với các tập tin chứa đặc trưng ứng với từng nhóm văn bản.

- Nếu các từ đặc trưng trong câu truy vấn thuộc nhóm văn bản nào thì mới thực hiện tính toán các độ đo Cosine của từng vector văn bản thuộc nhóm đó (từng vector cột của ma trận từ đặc trưng-văn bản ứng với nhóm văn bản đó) với vecor truy vấn. Nhóm văn bản này tạm gọi là nhóm văn bản có liên quan. Nếu các từ đặc trưng trong câu truy vấn không thuộc một nhóm văn bản, chúng ta sẽ không thực hiện tính toán các độ đo Cosine trên nhóm văn bản đó, cũng không thực hiện các xử lý tiếp theo trên nhóm văn bản đó (không truy tìm trên nhóm văn bản đó).

- Thực hiện so sánh các độ đo Cosin đã tính toán được (giữa vector truy vấn và vector văn bản thuộc các nhóm văn bản có liên quan) với ngưỡng (threshold) để trả về các văn bản có liên quan với câu truy vấn.

Chức năng xếp hạng kết quả truy tìm

Các văn bản trả về sẽ được hiển thị theo thứ tự độ liên quan với câu truy vấn từ cao đến thấp. Việc xếp hạng kết quả trả về được thực hiện theo thứ tự giảm dần của các độ đo Cosine đã tính toán được.

Chức năng giao diện thực hiện truy vấn và hiển thị kết quả trả về

Để mang tính ứng dụng thực tiễn cao, giao diện thực hiện truy vấn văn bản được thiết kế theo dạng ứng dụng web.

5.2.2 Cài đặt phân hệ truy tìm văn bản VSM

Phân hệ truy tìm văn bản được cài đặt như thiết kế trình bày ở hình 5.4.

Dữ liệu đầu vào

Hệ truy tìm văn bản được cài đặt thử nghiệm trên tập 120 văn bản thuộc 4 lĩnh vực (công nghệ, giáo dục, thể thao, y tế) đã được phân lớp bởi phân hệ phân lớp văn bản SVM-NN. Sau khi phân lớp với tập 120 văn bản trên, ta có các tập tin dữ liệu đầu ra được dùng làm dữ liệu đầu vào cho phân hệ truy tìm văn bản như sau:

- Các tập tin chứa đặc trưng ứng với từng nhóm văn bản có đường dẫn tương đối như sau: *dactrung/congnghe.txt, dactrung/giaoduc.txt, dactrung/thethao.txt, dactrung/yte.txt*.

- Tập tin chứa kết quả phân lớp các văn bản: *dactrung/ketquaphanlop.txt*.

- Tập tin chứa đường dẫn các văn bản: *dactrung/path.txt*.

Các bước thực hiện

- Chạy *module tạo ma trận đặc trưng-văn bản* : tạo các tập tin chứa ma trận từ đặc trưng-văn bản ứng với từng nhóm văn bản. Ta có các tập tin: *matrix/congnghe.txt, matrix/giaoduc.txt, matrix/thethao.txt, matrix/yte.txt*.

- Chạy *module xử lý truy tìm* : thực hiện nhập câu truy vấn, kết quả truy tìm trả về được hiển thị như sau:

+ Hiển thị thông tin về các nhóm văn bản không liên quan (không thực hiện truy tìm trên các văn bản thuộc nhóm đó)

+ Hiển thị các văn bản cần truy tìm, xếp hạng giảm dần theo độ đo Cosin.

+ Mỗi văn bản trả về hiển thị kết quả phân lớp, độ đo Cosin của văn bản đó.

Giao diện thực hiện truy vấn và hiển thị kết quả trả về như sau:

Hình 5.5: Giao diện thực hiện truy vấn và hiển thị kết quả trả về

5.2.3 Đánh giá kết quả cải tiến của phân hệ truy tìm văn bản VSM

Đối với hệ truy tìm văn bản có kiến trúc cơ bản, *module xử lý truy tìm* sẽ thực hiện tính toán các độ đo Cosin và các xử lý khác trên toàn bộ tập văn bản. Điều này làm mất rất nhiều thời gian và tiêu tốn rất nhiều không gian lưu trữ, tài nguyên tính toán, tốc độ truy tìm sẽ rất chậm, nếu số lượng văn bản lớn (hoặc số lượng từ đặc trưng lớn).

Đối với hệ truy tìm văn bản có cải tiến bằng cách sử dụng các tập tin kết quả của quá trình phân lớp làm dữ liệu đầu vào, *module xử lý truy tìm* sẽ không thực hiện tính toán các độ đo Cosin trên tất cả các văn bản mà chỉ thực hiện trên các văn bản thuộc nhóm có từ đặc trưng liên quan với câu truy vấn. Điều này làm tiết kiệm rất nhiều thời gian, không gian lưu trữ, tài nguyên tính toán, qua đó làm tăng đáng kể tốc độ truy tìm.

Chúng ta xem xét cụ thể kết quả truy tìm ở hình 5.5. Tập 120 văn bản thuộc 4 lĩnh vực (công nghệ, giáo dục, thể thao, y tế) đã được phân hệ phân lớp phân ra thành 4 nhóm văn bản tương ứng. Phân hệ truy tìm văn bản có cải tiến bằng cách sử dụng kết quả phân lớp bên trên đã không phải thực hiện xử lý truy tìm văn bản trên 4 nhóm, mà chỉ xử lý truy tìm trên 2 nhóm văn bản (y tế và thể thao). Điều này làm tăng tốc độ truy tìm khoảng 2 lần so với hệ truy tìm cơ bản mà không kết hợp với phân hệ phân lớp văn bản (do phải xử lý truy tìm trên toàn bộ 4 nhóm văn bản).

Tóm lại, bằng việc kết hợp với phân hệ phân lớp văn bản, phân hệ truy tìm văn bản sẽ cải thiện đáng kể tốc độ, hiệu quả truy tìm vì không phải thực hiện xử lý truy tìm trên toàn bộ tập văn bản mà chỉ thực hiện truy tìm trên một hoặc vài nhóm văn bản có liên quan với câu truy vấn.

CHƯƠNG 6: KẾT LUẬN

6.1 Đánh giá kết quả

Đối với các kỹ thuật phân lớp văn bản, luận văn đã tìm hiểu kỹ thuật phân lớp văn bản Support Vector Machines (SVM). Đồng thời luận văn cũng đã có một số nghiên cứu các thuật toán phân lớp văn bản cải tiến dựa trên kỹ thuật SVM để giải quyết bài toán phân lớp:

- Nghiên cứu thuật toán Fuzzy SVM cho phép loại bỏ các dữ liệu nhiễu trong quá trình huấn luyện và cải thiện độ chính xác của quá trình phân lớp.

- Nghiên cứu, cài đặt áp dụng thuật toán SVM Nearest Neighbor với việc kết hợp ý tưởng của thuật toán K-Nearest Neighbor và thuật toán SVM để cải thiện hiệu quả phân lớp.

- Nghiên cứu,cài đặt áp dụng các chiến lược phân lớp văn bản đa lớp OAR (One - against - Rest), OAO (One - against - One) và kỹ thuật cải tiến việc phân lớp đa lớp này là phân lớp đa lớp mờ Fuzzy OAO (Fuzzy One - against - One).

Đối với các kỹ thuật phục vụ truy tìm văn bản, luận văn đã tìm hiểu sử dụng mô hình truy tìm văn bản theo mô hình không gian vector VSM (Vector Space Model).

Từ kết quả nghiên cứu trên, luận văn đã xây dựng thử nghiệm được một hệ thống tự động phân lớp và phục vụ truy tìm thông tin văn bản thực tế theo mô hình không gian vector VSM có cải tiến so với hệ thống truy tìm theo mô hình VSM cơ bản. Việc cải tiến hệ thống truy tìm thông tin văn bản VSM được thực hiện bằng cách kết hợp sử dụng các kết quả phân lớp trên kho văn bản trước khi thực hiện các kỹ thuật xử lý truy tìm. Kết quả của việc cải tiến này là phân hệ truy tìm văn bản đã cải thiện đáng kể tốc độ, hiệu quả truy tìm vì không phải thực hiện xử lý truy tìm trên toàn bộ kho văn bản mà chỉ thực hiện truy tìm trên một hoặc vài nhóm văn bản có liên quan với câu truy vấn.

Kết quả cài đặt thực nghiệm của hệ thống là khá tốt, cho thấy tính khả thi tương đối khi triển khai áp dụng vào thực tế.

Tuy nhiên, luận văn vẫn còn một số hạn chế sau cần giải quyết:

- Chưa thực hiện tự động cập nhật kết quả phân lớp và xử lý truy tìm khi thêm vào một văn bản mới vào kho văn bản.

- Thuật toán cải tiến SVM Nearest Neighbor được cài đặt có tốc độ thực thi còn chậm.

- Chưa có chức năng thu thập thông tin tự động trên các website.

6.2 Hướng phát triển

Để luận văn có thể áp dụng vào thực tế tốt hơn, cần phải tiếp tục nghiên cứu, cải tiến một số vấn đề sau:

- Cho phép thực hiện tự động phân lớp và xử lý phục vụ việc truy tìm khi thêm vào một văn bản mới vào kho văn bản.

- Nghiên cứu cải tiến tốc độ thực thi của thuật toán SVM Nearest Neighbor.

- Nghiên cứu các kỹ thuật rút trích thông tin văn bản tự động. Từ đó áp dụng xây dựng hệ thống tự động thu thập thông tin văn bản trên các website, phân loại và phục vụ truy tìm thông tin văn bản.

- Thực hiện phân lớp văn bản vào nhiều nhóm khác nhau (Multi-Categorization).

- Phát triển thêm các ứng dụng như tóm tắt văn bản, dịch tự động các văn bản sau khi thu thập và phân lớp.

Hiện nay, bài toán phân lớp và bài toán truy tìm thông tin nói chung cũng như thông tin văn bản nói riêng vẫn còn nhiều vấn đề chưa được giải quyết triệt để. Do đó, tác giả mong muốn được góp ý thêm để có thể hoàn thiện hơn nữa những tồn tại của luận văn.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Nguyễn Kim Anh, Nguyễn Thị Kim Ngân (2006), "*Phân lớp văn bản tiếng Việt sử dụng phương pháp Support Vector Machines"*, Khoa Công nghệ thông tin, ĐHBK Hà Nội.

[2] Nguyễn Thị Minh Huyền, Vũ Xuân Lương, Lê Hồng Phương (2003), *"Sử dụng bộ gán nhãn từ loại xác suất QTAG cho văn bản tiếng Việt"*, *Kỷ yếu Hội thảo ICT.rda'03,* trang 22-23.

[3] Trang Nhật Quang (2007), *"Đề xuất một công cụ hỗ trợ thu thập và phân loại thông tin tiếng Việt trên internet"*, Luận văn Thạc sĩ, Đại học Khoa học Tự nhiên TP.HCM, TP.HCM.

Tiếng Anh

[4] Enrico Blanzieri, Anton Bryl (2007), *"Evaluation of the Highest Probability SVM Nearest Neighbor Classifier With Variable Relative Error Cost"*, University of Trento, Italy.

[5] Enrico Blanzieri, Anton Bryl (2007), *"Instance-Based Spam Filtering Using SVM Nearest Neighbor Classifier"*, University of Trento, Italy.

[6] Li-Cheng Jin (2004), "*Application of Fuzzy Support Vector Machines in Medical Engineering and Bioinformatics"*, Master Thesis, Institute of Electronics and Information Engineering National Kaohsiung University of Applied Sciences, Taiwan.

[7] Shigeo Abe and Takuya Inoue (2002), *"Fuzzy Support Vector Machines for Multiclass Problems"*, ESANN'2002 proceedings, pp. 113-118.

[8] Shigeo Abe and Takuya Inoue (2001), *"Fuzzy Support Vector Machines for Pattern Classification"*, In Proceeding of International

Joint Conference on Neural Networks (IJCNN ’01), volume 2, pp. 1449-1454.

[9] Tsui-Feng Hu (2004), “*Fuzzy Correlation and Support Vector Learning Approach to Multi-Categorization of Documents”*, Master Thesis, Institute of Information Management I-Shou University, Taiwan.

[10] T.Joachims (1998), *“Text Categorization with Support Vector Machines: Learning with Many Relevant Features”* in Proceedings of ECML-98, 10th European Conference on Machine Learning, number 1398, pp. 137–142.

[11] Xiufeng Jiang, Zhang Yi and Jian Cheng Lv (2006)*, “Fuzzy SVM with a new fuzzy membership function”,* Neural Computing and Applications, Volume 15(3), pp. 268-276.

[12] Yiming Yang, Jan O. Pedersen (1997), *"A comparative Study on Feature Selection in Text Categorization"*, Proceedings of {ICML}-97, 14th International Conference on Machine Learning, pp. 412-420.