**MỤC LỤC**

**PHẦN MỞ ĐẦU 02**

**CHƯƠNG 1: TỔNG QUAN VỀ KHAI PHÁ VĂN BẢN 03**

1. **Khai phá dữ liệu văn bản 03**
2. **Phân loại văn bản 03**
3. **Quy trình phân loại văn bản 05**

**CHƯƠNG 2: BÀI TOÁN PHÂN LOẠI VĂN BẢN THEO CHỦ ĐỀ 10**

1. **Phát biểu bài toán 10**
2. **Các quá trình xử lý 10**

**CHƯƠNG 3: TÌM HIỂU SVM VÀ XÂY DỰNG CHƯƠNG TRÌNH 25**

1. **Phân loại văn bản Tiếng Việt theo chủ đề định sẵn 25**
2. **Phân tích thiết kế hệ thống chương trình 30**
3. **Giới thiệu chương trình 32**

**KẾT LUẬN 36**

**TÀI LIỆU THAM KHẢO 37**

**PHẦN MỞ ĐẦU**

1. **Tên đề tài**

Tìm hiểu phương pháp phân loại văn bản bằng SVM

1. **Nhóm sinh viên thực hiện**

Phạm Tuấn Tài - 2001181302

Đỗ Minh Trí - 2001180141

Nguyễn Võ Minh Nhật – 2001181237

1. **Giáo viên hướng dẫn**

Thầy Bùi Công Danh

1. **Lý do chọn đề tài**

Trong thời đại bùng nổ công nghệ thông tin hiện nay, hệ thống dữ liệu số hoá trở nên khổng lồ để phục vụ cho việc lưu trữ trao đổi thông tin, dữ liệu số hoá này rất đa dạng - nó có thể là các dữ liệu dưới dạng tập tin văn bản text, tập tin văn bản MS Word, tập tin văn bản PDF,... . Các tập tin văn bản cũng được lưu trữ trên máy tính cục bộ hoặc được truyền tải trên Intenet, cùng với thời gian và số lượng người dùng tăng nhanh thì các tập tin này ngày càng nhiều và đến một thời điểm nào đó thì số lượng tập tin này sẽ vượt quá tầm kiểm soát, do đó khi muốn tìm kiếm lại một văn bản nào đó việc tìm kiếm sẽ rất khó khăn và phức tạp, đặc biệt là trong trường hợp người cần tìm kiếm không nhớ rõ các câu cần tìm chính xác trong văn bản, do đó khi người sử dụng muốn sắp xếp các thông tin tìm được theo thể loại (nhóm văn bản) thì thời gian thực hiện sẽ mất rất nhiều thời gian và công sức bỏ ra cũng không phải nhỏ.

Từ các nhu cầu trên mà yêu cầu về một hệ thống nhận dạng và phân loại văn bản để đáp ứng yêu cầu phân loại văn bản sau đó mới thực hiện tìm kiếm được ra đời nhằm đáp ứng yêu cầu thực tế của người dùng. Do vậy việc nghiên cứu đề tài: “Tìm hiểu phương pháp phân loại văn bản bằng SVM” là rất cần thiết.

1. **Mục tiêu của đề tài**

* Tìm hiểu về bài toán xử lý văn bản, phân loại văn bản theo chủ đề.
* Xây dựng một bản báo cáo về bài toàn xử lý văn bản, phân loại văn bản theo chủ đề, SVM
* Xây dựng chương trình demo phân loại văn bản bằng SVM.

1. **Đối tượng nghiên cứu**

* Bài toán phân loại văn bản
* Phương pháp SVM
* Ngôn ngữ lập trình C#

1. **Phạm vi nghiên cứu**

* Đề tài tập trung nghiên cứu về bài toán phân loại văn bản, phương pháp SVM

1. **Nội dung nghiên cứu**

* Xử lý văn bản
* Bài toán phân loại văn bản theo chủ đề
* Phân loại văn bản tiếng Việt theo chủ đề định sẵn

**CHƯƠNG 1: TỔNG QUAN VỀ KHAI PHÁ VĂN BẢN**

1. **Khai phá dữ liệu văn bản**

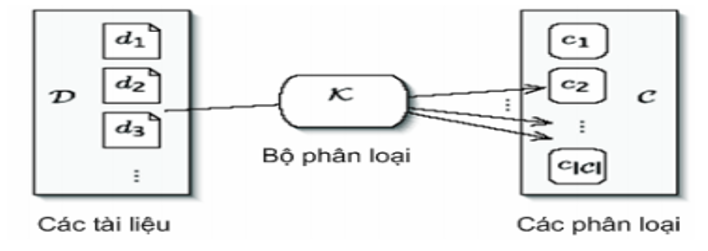
* Văn bản là các tài liệu ở dạng không có cấu trúc hoặc bán cấu trúc. Ví dụ như các các file dạng đuôi .txt, .doc.
* Khai phá dữ liệu văn bản là việc trích ra, lấy ra các thông tin có ích, chưa được biết đến còn tiềm ẩn trong các kho dữ liệu văn bản lớn.
* Khai phá dữ liệu văn bản là việc thu thập và phân tích dữ liệu bằng các công cụ tự động hoặc bán tự động từ các nguồn tài liệu đã có khác nhau để có được các tri thức mới, chưa được biết đến trước đó.
* Khai phá văn bản là phát hiện ra các mô tả chung của các lớp đối tượng, các từ khoá, các mối liên quan về mặt nội dung, sự phân loại của các đối tượng văn bản…

1. **Phân loại văn bản**

Để hiểu một cách đơn giản thì phân loại văn bản là việc gán các tài liệu vào trong các phân loại dựa trên nội dung của chúng. Sử dụng học máy, mục tiêu là để học các bộ phân loại từ các mẫu mà văn bản chưa thấy có thể được tự động phân loại.

Phân loại văn bản là một bài toán xử lý văn bản cổ điển, đó là ánh xạ một văn bản vào một chủ đề đã biết trong tập hữu hạn các chủ đề dựa trên ngữ nghĩa của văn bản. Ví dụ: một bài viết trong một tờ báo có thể thuộc một (hoặc một vài) chủ đề nào đó (như thể thao, sức khỏe, kinh tế…). Việc tự động phân loại văn bản vào một chủ đề nào đó giúp cho việc sắp xếp, lưu trữ và truy vấn tài liệu dễ dàng hơn về sau.

Về mặt hình thức, cho trước một tập các nhãn (các phân loại) *C = {c1, ... , c|C|}* và một tập các tài liệu chưa thấy trước đó *D = {d1, d2, ...}*, một bộ phân loại là một hàm *K* ánh xạ từ *D* tới tập của tất cả các tập con của *C*.



Các cách tiếp cận hiện đại chuẩn để tạo các hàm phân loại mới là để xây dựng chúng sử dụng các kỹ thuật học máy từ một tập các tài liệu huấn luyện *Tr.* Đây là một tập các tài liệu do người dùng cung cấp và được gán nhãn trước mà cho phép một phân loại phân bổ tương tự với sự phân bổ của *D*, và có các nối dụng cung cấp thông tin phần nào về các tài liệu sẽ được ánh xạ tới các phân loại. Các giải thuật sau đó có thể được phát triển để tạo ra các sự tổng quát hoá về quan hệ giữa nội dung tài liệu và phân loại tài liệu, mã hoá các sự tổng quát hoá đó trong hàm học *K.*

Đặc điểm nổi bật của bài toán này là sự đa dạng của chủ đề văn bản và tính đa chủ đề của văn bản. Tính đa chủ đề của văn bản làm cho sự phân loại chỉ mang tính tương đối và có phần chủ quan, nếu do con người thực hiện và dễ bị nhập nhằng khi phân loại tự động. Rõ ràng một bài viết về giáo dục cũng có thể xếp vào kinh tế nếu như bài viết bàn về tiền nong đầu tư cho giáo dục và tác động của đầu tư này đến kinh tế - xã hội. Về bản chất, một văn bản là một tập những từ ngữ có liên quan với nhau tạo nên nội dung ngữ nghĩa của văn bản. Từ ngữ của một văn bản là đa dạng do tính đa dạng của ngôn ngữ (đồng nghĩa, đa nghĩa, từ vay mượn nước ngoài...) và số lượng từ cần xét là lớn. Ở đây cần lưu ý rằng, một văn bản có thể có số lượng từ ngữ không nhiều nhưng số lượng từ ngữ cần xét là rất nhiều vì phải bao hàm tất cả các từ của ngôn ngữ đang xét.

Trên thực tế công việc phân loại văn bản là một công việc bắt buộc ở bất cứ cơ quan, văn phòng nào giúp xử lý công việc một cách thuận lợi, chính xác, khoa học hơn. Tùy theo nhu cầu của mỗi cơ quan, văn phòng mà văn bản được phân loại theo nhiều nội dung và hình thức khác nhau. Vì thế, đây là một công việc phức tạp đòi hỏi sự chính xác cao và mất nhiều thời gian, công sức nhất là với số lượng văn bản lớn.

1. **Quy trình phân loại văn bản**

Để huấn luyện một bộ phân loại theo cách trên, người dùng phải bắt đầu với một một huấn luyện, và được gọi là *Tr*. Đây là một tập tài liệu được gán nhãn trước với các phân loại mà được xem là đúng hoàn toàn - chúng được gán thủ công bởi một chuyên gia về lĩnh vực đó, một người biết rõ kiểu tài liệu chứa trong các văn bản và biết làm thế nào để gán các phân loại tới từng tài liệu dựa trên nội dung tài liệu

Phác thảo cơ bản về việc tạo các ứng dụng phân loại văn bản là khá đơn giản: các tài liệu trong *Tr* được đưa tới hệ thống phân loại văn bản, hệ thống xử lý nội dung các tài liệu và hàm phân loại xác định *K* được sinh ra có thể được sử dụng để phân loại các tài liệu tương lai từ tập *D*. Trong một ứng dụng, nhiều điểm của quy trình này cần được quản lý trong nhiều cách xác định. Từ phần a. tới i. miêu tả các giai đoạn cuả quá trình này.

* 1. **Lưu trữ tài liệu**

Trong một tổ chức mà cần một ứng dụng phân loại văn bản, các tài liệu có thể có từ nhiều nguồn. Chúng có thể xuất phát từ các văn bản thuần tuý (chỉ có các ký tự) hoặc các thông điệp thư điện tử đã định dạng, chúng có thể là các trang web đã được xử lý và còn thô, chúng có thể là các bộ dữ liệu từ một CSDL (phần d.), hoặc chúng không thể có một biểu diễn ngoài hệ thống phân loại văn bản. Bởi vậy quan trọng để nhận dạng khái niệm phương tiện lưu trữ tài liệu và tiến hành chuyển đổi từ các phương tiện đó tới một trung gian có thể truy cập tới hệ thống phân loại văn bản, là một phần quan trọng của quy trình phân loại văn bản.

* 1. **Định dạng văn bản**

Tuy hầu hết các phân loại văn bản xem xét một tài liệu là một văn bản thuần tuý chuỗi dữ liệu, nhưng điều này là khó gặp trong một môi trường ứng dụng. Các tài liệu có thể được lưu trong nhiều định dạng tài liệu, bao gồm văn bản thuần tuý (plain text), HyperText Markup Language (HTML), Adobe’s Portable Document format (PDF), Extensible Markup Language (XML), Microsoft Word (DOC), các thư điện tử mã hoá MIME… Dữ liệu bên trong mỗi tài liệu cũng có thể được xem xét một phần định dạng của nó khi khối lượng thông tin trích dẫn hoặc các biến đổi khác cần được áp dụng tới dữ liệu tài liệu để làm chúng có khả năng có thể truy cập được tới hệ thống phân loại văn bản. Ví dụ, các chuỗi số trong một số tập tài liệu có thể là các thuật ngữ đáng xem xét khi phân loại, nhưng ngược lại trong các bộ dữ liệu khác có thể chỉ có dữ liệu nhiễu. Vì lý do tương tự như đã đề cập trong phần trước, có thể mong muốn với một hệ thống phân loại văn bản giải quyết với các vấn đề trực tiếp, hoặc để cung cấp một kỹ thuật để mở rộng hệ thống để nhận dạng các chuẩn mới, hoặc là chuyển đổi dữ liệu của tất cả các dữ liệu tới một chuẩn nhận dạng bởi hệ thống.

* 1. **Cấu trúc hóa tài liệu**

Trong phân loại văn bản, một ít là tạo bởi cấu trúc văn bản, ngoại trừ rằng một hệ thống phân loại văn bản có thể gán các trọng số quan trọng tới các thuật ngữ trong một tài liệu theo các trọng số quan trọng được thiết lập trước theo các phần trong các thuật ngữ đó được tìm thấy. Ví dụ, một thuật ngữ tìm thấy trong một tiêu đề của một tài liệu có thể được xem xét 2 lần quan trọng hơn một thuật ngữ tìm thấy trong nội dung. Tuy nhiên, khi nghiên cứu vào sự phân loại của việc thực hiện các tài liệu có cấu trúc, có thể là một lĩnh vực rất lớn để xem xét việc xây dựng các hệ thống phân loại văn bản.

* 1. **Tách dữ liệu**

Để chuyển đổi văn bản của một tài liệu thành dữ liệu mà có thể được phân tích bởi một giải thuật học máy, cần chia văn bản thành các đơn vị rời, mỗi đơn vị tương ứng với một từ hoặc cụm từ trong văn bản. Việc này được gọi là tách từ (tokenization). Trong phần thảo luận này, từ thuật ngữ ám chỉ tới thực thể ngôn ngữ chính xác khi nó xuất hiện trong văn bản nguồn, token là một chuỗi được trích ra bởi hệ thống phân loại văn bản. Việc phân đoạn dữ liệu văn bản vào trong các khoanh (chunk) biểu diễn các từ riêng biệt có thể giống với bài toán không phức tạp, nhưng thực tế có nhiều biến đổi quy trình này sẽ được thực hiện. Việc phân tách các từ bằng khoảng trống (dấu cách, tab...) là không đủ vì chưa thực hiện với các dấu chấm hay các từ ghép, việc tách từ này thường cần được tạo với một vài kiến trức về tập tài liệu D. Hơn nữa, nhiều ngôn ngữ như tiếng Hàn hay tiếng Nhật không chứa các khoảng trắng để chỉ ra sự phân tách giữa các từ. Quy trình tách từ có thể xoá dữ liệu có trong danh sách từ dừng (stop-list) định nghĩa trước đó. Từ dừng là các thường xuất hiện trong lĩnh vực (như “the” hay “and” trong các văn bản tiếng Anh) và được cho rằng chứ một ít hoặc không liên quan tới vấn đề thuộc phân loại bằng tay.

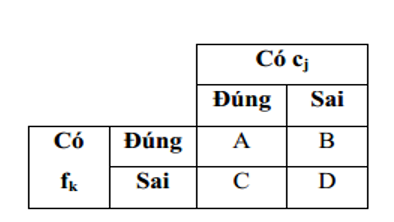
Để đưa ra các vấn đề đó, một hệ thống phân loại văn bản cần cho phép các biến đổi trong tiến trình phân loại văn bản. Việc này có thể bao gồm các điều chỉnh một tập các tham số điều khiển hoặc một vài trường hợp người phát triển ứng dụng có thể viết thêm mã điều khiển trong trong các trường hợp xác định.

* 1. **Giảm chiều**

Giống với nhiều lĩnh vực nghiên cứu xử lý ngôn nhữ, nhiều bộ phân loại văn bản phải giải quyết vấn đề tính đa chiều cao. Số chiều của không gian trong đó giải thuật học máy thực hiện có thể là lớn hơn nhiều tổng số các thuật ngữ trong *Tr*.

Tính đa chiều cao có thể đưa ra hai vấn đề. Đầu tiên, một vài giải thuật học máy có thể hiệu quả trên dữ liệu ít chiều, nhưng chúng yêu cầu nhiều thời gian hoặc bộ nhớ hơn thực tế khi tính đa chiều của tập dữ liệu là cao. Thứ hai, dữ liệu *Tr* trong không gian đa chiều lớn có thể là quá rải rác, mà không đủ các điểm dữ liệu khác 0 để tạo ra bất kỳ một biến đổi quy nạp nào trong quá trình huấn luyện. Điều này đặc biệt đúng trong các ngôn ngữ hình thái học cao như Phần Lan, trong đó một từ đơn có thể có hàng ngàn hoặc hàng triệu kiểu biến tố, và hầu hết các kiểu có thể chỉ thấy một lần trong toàn bộ *Tr*, làm chúng hầu như không có ích trong quá trình học quy nạp.

Một cách nhắm vào vấn đề tính đa chiều cao là áp dụng một giải thuật chặn ngôn ngữ thành các thuật ngữ tìm thấy trong *Tr* và *D*. Các giải thuật đó biến đổi các từ bằng cách xoá các tiền tố và hậu tố. Một cách khác để giảm tính đa chiều là trong một hệ thống phân loại áp dụng lựa chọn đặc trưng hoặc trích chọn đặc trưng.



**Ngẫu nhiên cho phân loại cj và thuật ngữ fk**

* 1. **Mô hình hóa không gian vector**

Như đã trình bày ở trên, tác giả đã đề xuất mỗi tài liệu có thể được xem như một vector trong một không gian vector chung có số chiều được biểu diễn là tập các đặc trưng duy nhất từ *Tr*. Ý tưởng này tạo thành các cơ sở cho một vài kỹ thuật học máy, gồm có các bộ phân loại máy học vector hỗ trợ (SVM) và k láng giềng gần nhất (KNN). Nó cũng cho phép các giải thuật xử lý vector tuỳ ý trên dữ liệu văn bản tận dụng các kết quả phân loại. Một tập các giải thuật thường được sử dụng cho mục đích thu nhận thông tin là sơ đồ trọng số thuật ngữ TF/IDF của Salton và Buckley, cho phép một vài cách khác nhau để biểu diễn một tài liệu như một vector trong không gian vector chung. Các thuật ngữ có thể được đánh trọng số bởi tần xuất của chúng trong tài liệu, tính logarit các tần xuất đó hoặc bởi một biểu diễn minh họa có hay không có thuật ngữ. Trọng số thuật ngữ cũng có thể được làm giảm bằng một yếu tố biểu diễn sự xuất hiện của thuật ngữ trong các tài liệu khác, trên học thuyết bất cứ thuật ngữ nào đưa ra trong hầu hết các tài liệu có sự liên quan giữa các phân loại. Cuối cùng, độ dài toàn bộ của vector tài liệu có thể được giảm bớt theo nhiều cách khác nhau

* 1. **Giải thuật học máy**

Nhiều giải thuật học máy khác nhau đã được nghiên cứu trong lĩnh vực nghiên cứu phân loại văn bản và các giải thuật mới hoặc sự biến đổi trên hệ thống hiện tại vẫn tiếp tục đang được phát triển. Hơn nữa, lựa chọn giải thuật có thể phụ thuộc vào các giải thuật - ứng dụng khác không chỉ trong khả năng thực hiện chính xác trên các tập dữ liệu khác nhau, nhưng cũng trong các nguồn chung có thể yêu cầu chọn một giải thuật để hợp nhất vào trong một framework. Trong một hệ thống phân loại văn bản, cần cho phép chọn theo một vài giải thuật chuẩn và thống nhất các giải thuật đã được phát triển bởi các nhà nghiên cứu.

* 1. **Thiết lập cấu hình học máy**

Thậm chí trong một giải thuật học máy cũng có thể có một vài tham số có ảnh hưởng tới quy trình huấn luyện và phân loại. Ví dụ, giải thuật k láng giềng gần nhất có một tham số điều chỉnh k và một giải thuật SVM cho phép một vài biến về kiểu kernel sử dụng và hầu hết các giải thuật cho vào một vài kiểu điều khiển việc cân bằng precision và recall với nhau. Để thu nhận sự thực hiện tương ứng với một bài toán đã cho, những người ừng dụng cần các cách đơn gian tới các tham số đó. Thực tế, vấn đề này không chỉ duy nhât với thành phần học máy của phân loại văn bản. Một vài hướng đã thảo luận trước đó của bài toán phân loại văn bản như chọn lựa chọn đặc trưng, giảm chiều và biến đổi không gian vector có thể được điều khiển bởi các tham số mà người kiểm soát có thể muốn thay đổi. Tính nhất quán trong việc điều khiển các tham số của hệ thống có thể là một thành phần quan trọng trong thiết kế.

* 1. **Học tăng cường**

Một vài ứng dụng phân loại văn bản có thể được mong muốn kết hợp lại với người dùng về có các quyết định phân loại của hệ thống có thể là đúng hoặc không đúng. Việc này có thể cho phép với một số nhỏ các tập huấn luyện ban đầu Tr hoặc với việc phân loại trên các khái niệm mà có thể thay đổi theo thời gian. Quá trình này được gọi là học trực tuyến hay hoặc tăng cường. Tuy nhiên, việc học tăng cường là không thể với tất cả các phương pháp học máy, một vài giải thuật (mạng nơron) không thể hợp nhất với dấu hiệu mới vào trong mô hình của chúng mà không thực hiện lại quy trình huấn luyện. Với các giải thuật mà có thể được hỗ trợ, việc sử dụng học tăng cường có thể được xem là quan trọng trong việc xây dựng một ứng dụng phân loại văn bản.

**CHƯƠNG 2: BÀI TOÁN PHÂN LOẠI VĂN BẢN THEO CHỦ ĐỀ**

1. **Phát biểu bài toán**

Bài toán phân loại văn bản có thể được phát biểu như sau: cho trước một tập văn bản D={d1,d2,..., dn} và tập chủ đề được định nghĩa C={c1,c2,…,cn}.

Nhiệm vụ của bài toán là gán lớp di thuộc về cj đã được định nghĩa. Hay nói cách khác, mục tiêu của bài toán là đi tìm hàm f:

f: DxC  Boolean



+ f (d,c)= true nếu d thuộc về lớp c.

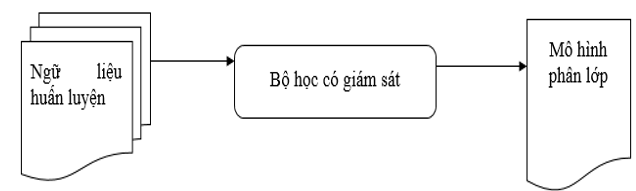
+ f (d,c)= false nếu d không thuộc về lớp c.

1. **Các quá trình xử lý**
2. **Mô hình tổng quát**

\*Giai đoạn huấn luyện

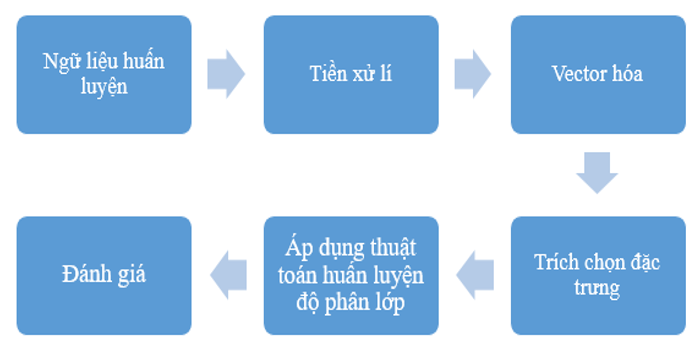
Chúng ta có một tập huấn luyện, mỗi phần tử trong tập huấn luyện được gán vào một hoặc nhiều lớp mà chúng ta sẽ thể hiện chúng bằng một mô hình mã hoá. Thông thường, mỗi phần tử trong tập huấn luyện được thể hiện theo dạng (,c). Trong đó,  là vector biểu diễn cho văn bản trong tập huấn luyện.

Sau đó, chúng ta định nghĩa một lớp mô hình và một thủ tục huấn luyện. Lớp mô hình là họ các tham số của bộ phân loại, thủ tục huấn luyện là một giải thuật (hay thuật toán) để chọn ra một họ các tham số tối ưu cho bộ phân loại. Ta xây dựng mô hình huấn luyện để dùng cho việc phân loại văn bản:



**Sơ đồ huấn luyện**

* Đầu vào: ngữ liệu huấn luyện và thuật toán huấn luyện
* Đầu ra: mô hình phân lớp (bộ phân lớp - classifier)



**Mô hình huấn luyện**

Trong đó:

- Ngữ liệu huấn luyện: kho ngữ liệu thu thập từ nhiều nguồn khác nhau.

- Tiền xử lý: chuyển đổi tài liệu trong kho ngữ liệu thành một hình thức phù hợp để phân loại.

- Vector hoá: mã hoá văn bản bởi một mô hình trọng số.

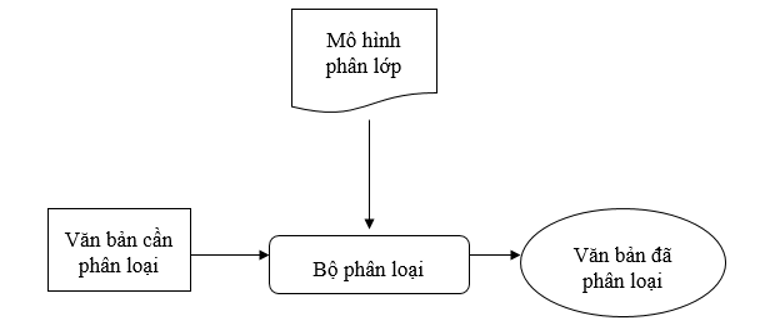
- Trích chọn đặc trưng: loại bỏ những từ (đặc trưng) không mang thông tin khỏi tài liệu nhằm nâng cao hiệu suất phân loại và giảm độ phức tạp của thuật toán huấn luyện.

- Áp dụng thuật toán huấn luyện độ phân lớp: áp dụng thuật toán SVM.

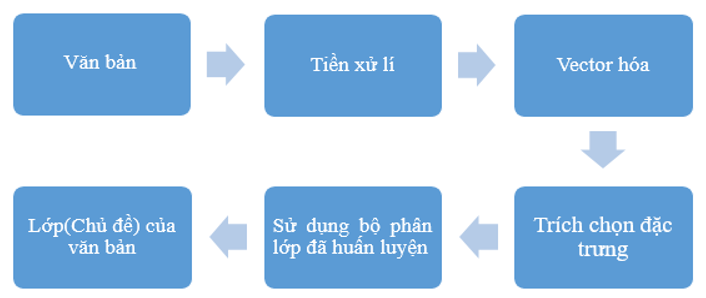
- Đánh giá: bước đánh giá hiệu suất (chất lượng) của bộ phân lớp.

Thủ tục huấn luyện sẽ được thực thi lặp nhiều lần để tìm họ các tham số tối ưu sau mỗi lần lặp. Tuy nhiên, do ban đầu họ các tham số được gán với một giá trị khởi tạo, do đó nếu giá trị khởi tạo ban đầu được gán sai thì kết quả tối ưu của họ các tham số có thể chỉ là tối ưu cục bộ.

\*Giai đoạn phân lớp

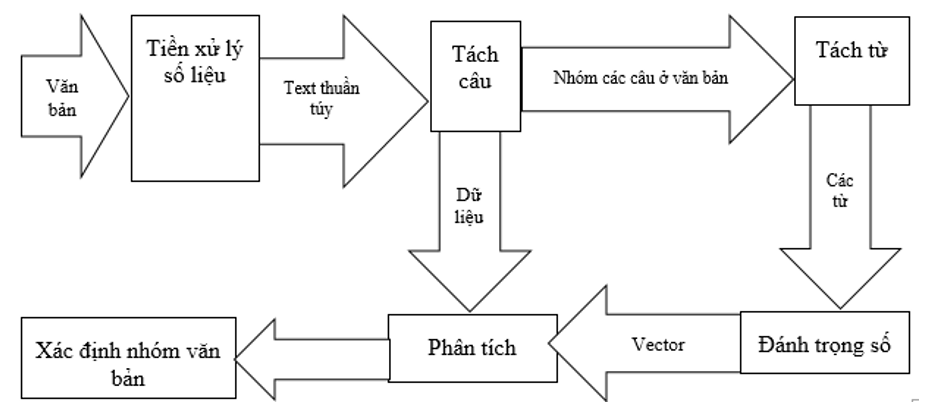


**Sơ đồ phân lớp**



**Mô hình phân lớp**

1. **Các quá trình xử lý văn bản**



**b.1. Khái niệm trọng số**

Một vấn đề quan trọng nữa trong việc biểu diễn một văn bản là tính trọng số cho vector đặc trưng của văn bản. Có nhiều cách khác nhau để tính trọng số này:

+ Word frequency weighting.

+ Boolean weighting.

+ tf\*idf weighting.

+ Entropy weighting.

Tuy nhiên, để đơn giản cho vấn đề này chúng ta sẽ chỉ xem xét cách tính Word frequency weighting (trọng số tần suất từ) và tf\*idf, một cách đơn giản đó là đếm số từ đó trong văn bản.

**Ba giá trị trong cách tính trọng số thuật ngữ (từ) thường dùng**

|  |  |  |
| --- | --- | --- |
| **Quanlity** | **Symbol** | **Definition** |
| Term frequency | Tfi,j | Number of occurrences of wi in dj |
| Document frequency | dfi | Number of document in the collection that wi occurs in |
| Collection frequency | Cfi | Total number of occurrences of wi in the collection |

Có ba thông tin được sử dụng trong cách tính trọng số bằng tần suất từ là: term frequency (tfij số lần suất hiện của từ wi trong văn bản dj), document frequency (dfi số văn bản có chứa từ wi), collection frequency (cfi số lần suất hiện của từ wi trong cả tập ngữ liệu). Trong đó, dfi cfi và Σjtfij = cfi.

Thông tin được nắm bắt bởi term frequency là sự nổi bật của thông tin (hay từ) trong một văn bản. Term frequency càng cao (số lần xuất hiện càng nhiều trong văn bản) thì đó là từ miêu tả tốt cho nội dung văn bản. Giá trị thứ hai, document frequency, có thể giải thích như là một bộ chỉ định nội dung thông tin. Một từ được tập trung ngữ nghĩa thường xảy ra nhiều lần trong một văn bản nếu nó cũng xuất hiện trong tất cả các văn bản khác. Nhưng từ không được tập trung ngữ nghĩa trải đều đồng nhất trong tất cả các văn bản.

Hãy xem xét một ví dụ sau, kho ngữ liệu của báo New York Times, hai từ try và insurance được thống kế như sau:

**Kho ngữ liệu của báo New York Times**

|  |  |  |
| --- | --- | --- |
| **Word** | **Collection frequency** | **Document frequency** |
| Insurance | 10440 | 3997 |
| Try | 10422 | 8760 |

Hai từ try và insurance có giá trị cf gần như nhau. Nhưng ngược lại, với giá trị df từ insurance chỉ xuất hiện trong hầu như chỉ một nửa kho dữ liệu. Điều này giải thích là bởi vì, từ try có thể được sử dụng trong hầu hết các chủ đề nhưng từ insurance chỉ được dùng để ám chỉ đến một khái niệm nhỏ mà chỉ liên quan đến một số lượng nhỏ các chủ đề. Một tính chất nữa của từ được tập trung ngữ nghĩa đó là nếu chúng xuất hiện trong một văn bản thì chúng sẽ xuất hiện vài lần.

Để thể hiện trọng số phản ánh hết thông tin của từ, thường ta sẽ kết hợp cả hai loại trọng số là tf và df trong một đơn vị chung. Dạng biểu diễn trọng số này được gọi là tf \* idf. Công thức kết hợp hai giá trị trọng số:



nếu tfij=0

Weight=

nếu tfij≥1

Trong đó, N là tổng số văn bản. Biểu thức thứ nhất áp dụng cho các từ có xuất hiện trong văn bản còn biểu thức thứ hai cho các từ không xuất hiện trong văn bản.

**b.2. Đánh giá bộ phân lớp**

Sau khi đã tìm được họ các tham số tối ưu cho bộ phân lớp (hay có thể nói là bộ phân lớp đã được huấn luyện xong), nhiệm vụ tiếp theo là cần phải đánh giá (kiểm tra) bộ phân lớp đó cho kết quả như thế nào. Tuy nhiên, quá trình kiểm tra phải được thực hiện trên một tập ngữ liệu khác với tập ngữ liệu huấn luyện, còn được gọi với cái tên là tập ngữ liệu kiểm tra (a test set). Việc kiểm tra bộ phân lớp là một sự đánh giá trên một tập ngữ liệu chưa được biết vì thế đó là sự đo lường, đánh giá duy nhất cho biết khả năng thực sự của một bộ phân lớp.

Để đơn giản, ta sẽ xem xét một bộ phân lớp nhị phân (phân hai lớp). Những bộ phân lớp thường được đánh giá bằng cách lập một bảng thống kê sau:

**Bảng thống kê những bộ phân lớp**

|  |  |  |
| --- | --- | --- |
|  | **YES is correct** | **No is correct** |
| **YES was assigned** | a | b |
| **No was assigned** | c | d |

Trong đó:

a: là số lượng đối tượng thuộc về lớp đang xét và bộ phân lớp gán vào lớp.

b: là số lượng đối tượng không thuộc về lớp đang xét nhưng được bộ phân lớp gán vào lớp.

c: là số lượng đối tượng thuộc về lớp đang xét nhưng được bộ phân lớp loại khỏi lớp.

d: là số lượng đối tượng không thuộc về lớp đang xét và được bộ phân lớp loại khỏi lớp.

Để đánh giá chất lượng bộ phân lớp, hai đơn vị đo lường quan trọng đó là độ đúng đắn (accuracy) được đo bằng công thức  và độ sai lỗi (Error) được tính bằng công thức  . Độ đo này phản ánh đầy đủ chất lượng của bộ phân lớp. Tuy nhiên, khi đánh giá bộ phân lớp, thường người ta chỉ xem xét những đối tượng thuộc về lớp và được phân lớp đúng, còn những đối tượng không thuộc về lớp thường sẽ ít được quan tâm.

Do đó, một số độ đo khác đã được định nghĩa. Các độ đo bao gồm:

+ Precision (độ chính xác): 

+ Recall (độ bao phủ, độ đầy đủ): 

+ Fallout (độ loại bỏ): 

**b.3. Mô hình không gian vector**

Mô hình không gian vector là một trong những mô hình được sử dụng rộng rãi nhất cho việc tìm kiếm (truy hồi) thông tin. Nguyên nhân chính là bởi vì sự đơn giản của nó.

Trong mô hình này, các văn bản được thể hiện trong một không gian có số chiều lớn, trong đó mỗi chiều của không gian tương ứng với một từ trong văn bản. Phương pháp này có thể biểu diễn một cách hình tượng như sau: mỗi văn bản D được biểu diễn dưới dạng (vector đặc trưng cho văn bản D). Trong đó,  = (x1, x2, ..., xn) và n là số lượng đặc trưng hay số chiều của vector văn bản, là trọng số của đặc trưng thứ i (với 1 i  n).

Như vậy, nếu trong kho ngữ liệu của quá trình huấn luyện nhiều văn bản, ta kí hiệu Dj, là văn bản thứ j trong tập ngữ liệu và vector j = (x1j, x2j, ..., xnj), là vector đặc trưng cho văn bản Dj, và xij là trọng số thứ i của vector văn bản j.

**b.4.** **Phương pháp biểu diễn văn bản**

Một trong những nhiệm vụ đầu tiên trong việc xử lý phân loại văn bản là chọn được một mô hình biểu diễn văn bản thích hợp. Một văn bản ở dạng thô (dạng chuỗi) cần được chuyển sang một mô hình khác để tạo thuận lợi cho việc biểu diễn và tính toán. Tuỳ thuộc vào từng thuật toán phân loại khác nhau mà chúng ta có mô hình biểu diễn riêng. Một trong những mô hình đơn giản và thường được sử dụng trong nhiệm vụ này là mô hình không gian vector. Một văn bản trong nhiệm vụ này được biểu diễn theo dạng , với Rn là một vector n chiều để đo lường giá trị của phần tử văn bản.

**b.5. Tiền xử lý số liệu**

Mục đích của bước này là xử lý tương đối sạch dữ liệu đọc vào để các bước sau sẽ xử lý tốt hơn, do đó công việc của bước này sẽ chỉ là chuyển thành chuỗi ký tự thuần túy (text), do đó nó sẽ có yêu cầu như sau:

+ Đầu vào: tệp văn bản cần phải phân tích (File PDF, TXT, DOC, HTML).

+ Đầu ra: chuỗi ký tự thuần túy (text only).

Nhiệm vụ: đọc nội dung các tập tin số liệu cần đọc, chuyển các văn bản cần các tag, các thông tin định dạng,..

- Để thống nhất khuôn dạng của văn bản thì tất cả các văn bản phải có cùng một phông chữ duy nhất, phông chữ được chọn là font Unicode, do đó trước khi thực hiện việc chuyển thành chuỗi ký tự (text) thì việc đầu tiên phải làm là chuyển tất cả các văn bản có font chữ khác với font chữ Unicode về thành font chữ Unicode. Do đa phần các tệp văn bản hiện nay đều đã sử dụng font Unicode và việc nhận dạng font tiếng Việt sử dụng trong tệp văn bản là khá khó khăn do đó phần chuyển đối này sẽ được làm bằng tay. Các số liệu phải được làm sạch các thông tin không phải là text, các thông tin này có thế là hình ảnh, âm thanh, định dạng văn bản... Việc tách này phụ thuộc vào từng kiểu tập tin dữ liệu đầu vào:

+ Nếu dữ liệu đầu vào là tệp văn bản dạng text (.txt) thì lấy tất cả số dữ liệu.

+ Nếu dữ liệu đầu vào là tệp văn bản dạng MS word (.doc) thì sẽ sử dụng Microsoft.Oficce.Core để chuyển đổi với công cụ này việc chuyển đối một file dạng Microsoft word sang text chỉ là một hàm.

+ Nếu dữ liệu đầu vào là tệp văn bản dạng PDF thì sẽ sử dụng control PDFbox để đọc và loại bỏ các thuộc tính không cần thiết cho chương trình như hình ảnh, ẩm thanh, định dạng và chỉ lấy giá trị text.

+ Nếu dữ liệu đầu vào là các tệp văn bản (htm) hay (html) thì việc loại bỏ các dữ liệu là loại bỏ các đoạn tag định dạng, các link liên kết, các link hình ảnh.

\* Quy trình thực hiện loại bỏ ở bước tiền xử lý:

- Loại bỏ các thông tin định dạng trang web: các trang web hiện nay được thiết kế theo chuẩn HTML bao gồm các thẻ (tag) định dạng cho các thành phần nội dung trong trang web, em có thể liệt kê theo nhóm như sau:

+ Tag định dạng thông tin chung của trang web: <TITLE>, <!DOCTYPE>, <HEAD>, <HTML>,...

+ Tag phân vùng, chia dòng, chia cột,....: <BR>, <DIV>, <TABLE>, <TR>…

+ Tag liệt kê đề mục: <LI>, <MENU>, <OL>,..

+ Tag định dạng chữ, hiệu ứng: <FONT>, <B>, <I>, <A>,..

+ Tag xử lý: <SCRIPT>, <APPLET>, <CODE>,...

Các thông tin định dạng, xử lý này cần được loại bỏ, chỉ giữ lại cho những phần thông tin bằng lời mà trang Web muốn thông báo cho người xem.

Loại bỏ các vùng văn bản phụ không cần thiết: sau khai đã loại bỏ thông tin định dạng, thông tin xử lý và trích ra phần thông tin bằng lời, trong thông tin này vẫn còn những thông tin phụ, không cần thiết mà ta cần tiếp tục loại bỏ. Trong trang web, ngoài thông tin chính của trang web, thường chứa nhiều thông tin phụ khác như: thông tin quảng cáo, thông báo phụ, các đề mục, menu,... Do đó, cần có một cách phù hợp để bỏ qua những phần nội dung không cần thiết và chỉ giữ lại phần nội dung chính để tách và thực hiện theo hai cách sau:

+ Cách 1: dùng các Heuristic hoặc học máy để rút ra các luật tách lấy những phần văn bản “tường thuật”.

+ Cách 2: áp dụng dùng đặc tính tính ngữ pháp để loại bỏ các phần văn bản không tạo thành câu, chẳng hạn những đoạn không có chứa động từ hoặc chứa nhiều hơn bốn từ không thể xác định từ loại.

- Làm sạch số liệu tiếp theo bao gồm:

+ Loại bỏ các khoảng trắng nhiều hơn 1 khoảng trắng.

+ Các dấu xuống dòng.

+ Cách dòng trống.

+ Các ký tự lạ.

Văn bản trước khi được vector hoá, tức là trước khi sử dụng, cần phải được tiền xử lý. Quá trình tiền xử lý sẽ giúp nâng cao hiệu suất phân loại và giảm độ phức tạp của thuật toán huấn luyện.

**b.6. Tách câu**

- Mục đích của bước này là tách một văn bản text thuần túy thành các câu:

+ Đầu vào: chuỗi ký tự văn bản thuần túy.

+ Đầu ra: vecto chứa các câu được tách trong văn bản.

- Đoạn văn bản sẽ được duyệt tuần tự và sẽ được cho ngắt câu khi gặp các ký tự ngắt câu như “.’ (chấm), “!” (chấm than), ”?” (chấm hỏi) với điều kiện: ký tự tiếp theo (có thể có các ký tự “khoảng trắng” ở giữa) là ký tự viết in. Loại bỏ được các trường hợp không phải ngắt câu như:

+ Dấu “.” không phải là ngắt câu mà là dấu trong một chuỗi số. Có được điều này vì nếu là “dấu chấm” trong chuỗi số thì ký tự tiếp theo phải là số, không phải ký tự viết in.

+ Dấu “.” trong một loạt “dấu ba chấm” bên trong câu chưa phải là cuối câu.

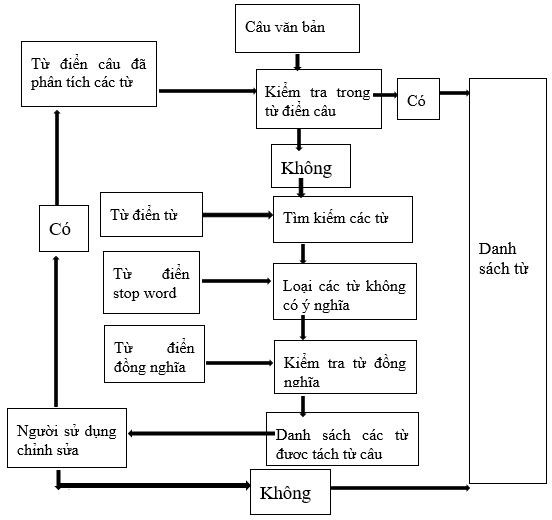
**b.7. Tách từ**

Tách các từ từ các câu đã được lấy ra, từ ở đây là từ tiếng Việt do đó đây là điều phải lưu ý:

+ Đầu vào: câu văn bản.

+ Đầu ra: vecto chứa các từ có nghĩa trong câu.

Tách từ là vấn đề quan trọng nhất của chương trình, nó quyết định chương trình có thế thực hiện đúng và chính xác việc phân loại hay không là nhờ kết quả của việc tách từ đúng hay sai. Do đặc điểm tiếng Việt đặc biệt là chú ý việc tiếng Việt không thể tách từ bằng khoảng trắng nên việc chọn phương pháp để tách từ cũng khá khó khăn. Do đó, ta sử dụng mô hình tách từ như sau:



**Sơ đồ tách từ trong văn bản**

Bước 1: Đối với một câu văn bản đưa vào sẽ kiếm tra trong dữ liệu có sẵn đã có mẫu câu này hay chưa, nếu đã có sẽ lấy các mẫu tách từ của mẫu câu này.

Bước 2: Nếu chưa có mẫu câu này thì chương trình sẽ đọc chữ đầu tiên và xem tiếp chữ kế tiếp, nếu chữ đầu tiên và chữ kế tiếp có trong cơ sở dữ liệu thì chương trình sẽ được chữ tiếp theo, cứ như vậy cho đến khi đọc chữ tiếp theo mà dãy chữ đó không có trong dữ liệu thì sẽ dừng lại và lấy từ là dãy chữ đã đọc được, tức là chương trình sẽ duyệt một ngữ hoặc câu từ trái sang phải và chọn từ có nhiều âm tiết nhất có mặt trong từ điển, rồi cứ thế tiếp tục cho từ kế tiếp cho đến hết câu.

Bước 3: Sau khi thực hiện xong bước 2 chương trình sẽ kiểm tra và loại bỏ các từ có tính chất kết nối, mô tả không có ý nghĩa trong câu và chỉ giữ lại những từ có ý nghĩa nhất.

Bước 4: Các từ này trước khi được đưa vào phân tích cần phải qua bước kiểm tra từ đồng nghĩa nhưng khác âm như: túc cầu – bóng đá, địa cầu – trái đất,... Đệ quy tất cả các từ này về chung một mẫu thống nhất.

Bước 5: Trong trường hợp đang lấy mẫu thì sau khi thực hiện bước 2 xong chương trình sẽ đưa kết quả để người sử dụng có thể tự xác định xem việc tác tử là đúng hay sai, trong trường hợp người sử dụng phải xác định lại thì mẫu câu này sẽ được lưu trong dữ liệu để lần sau khi gặp mẫu câu này thì chương trình sẽ tự động tách như ở bước 1.

Đây là quy trình chung nhưng khi thực hiện thì để tăng tốc độ (khi kiểm tra từ theo phương pháp khớp tối đa thì tốc độ tìm kiếm khá chậm) thì sẽ có 2 quy trình khác nhau:

- Quy trình thực hiện mẫu thử: thực hiện đúng như quy trình đã nêu ở trên trong đó:

+ Từ điển tiếng Việt là kết hợp từ hai từ điển: từ điển 78000 từ của Đinh Điền và từ điển của chương trình Vikass (chương trình phân loại tin từ điện tử).

+ Từ điển các từ hư (Stop word) cũng sử dụng của chương trình ViKass.

+ Từ điển từ đồng nghĩa tự xây dựng.

- Quy trình khi thực hiện chương trình phân loại văn bản thì dễ tăng tốc độ các bước sẽ hơi khác một chút như sau:

+ Từ điển tiếng Việt sẽ là những từ được dùng để đánh giá phân loại văn bản kết hợp với từ điển đồng nghĩa. Bỏ qua bước lại bỏ các từ hư (stop word) vì các hư từ thực ra là đã bị loại bỏ khi lấy các từ chỉ thuộc nhóm các từ mẫu.

**b.8. Gán nhãn – Đánh trọng số**

Gán nhãn loại là định lượng các từ trong văn bản:

+ Đầu vào: vector các từ.

+ Đầu ra: xác định nhóm của văn bản.

Việc gán nhãn – đánh trọng số là để lượng hóa các từ trong văn bản, nhờ việc lượng hóa này mà chương trình có thể xác định được các văn bản đang chọn thuộc nhóm văn bản nào. Việc đánh nhãn cũng có tính chất quyết định đến kết quả phân loại văn bản. Việc gán nhãn – đánh trọng số sẽ được thực hiện như sau, các từ của văn bản sau khi đọc vào sẽ được sắp xếp vào một bảng có các thông tin sau:

+ Từ đặc trưng.

+ Số lần xuất hiện trong văn bản: đây là số đếm lần xuất hiện trong văn bản.

+ Khoảng cách lớn nhất: đây là khoảng cách tính bình quân gia quyền lớn nhất của khoảng cách của từ đặc trưng đến đầu câu.

+ Khoảng cách nhỏ nhất: đây là khoảng cách tính bình quân gia quyền nhỏ nhất của khoảng cách của từ đặc trưng đến đầu câu.

+ Khoảng cách trung bình: đây là khoảng cách tính bình quân gia quyền trung bình của khoảng cách của từ đặc trưng đến đầu câu.

Như vậy là ta đã có bảng đánh trọng số của một từ trong văn bản, tuy nhiên ta cũng thấy rõ việc gán nhãn – đánh trọng số nếu thực hiện trên tất cả các từ có nghĩa trong văn bản thì sẽ dẫn đến việc vector từ phố biến trong văn bản sẽ có chiều rất lớn và điều này sẽ làm cho việc tính toán cần phải có máy tính rất mạnh. Để tăng tốc độ của chương trình thì phải giảm khối lượng của các vecto này, nhưng việc giảm các vecto này thì lại làm cho việc nhận dạng văn bản không được chính xác. Do đó để thực hiện được công việc đánh giá chính xác và giảm chiều cao các vecto văn bản này thì sẽ xác định như sau:

\*Đối với việc huấn luyện thì với văn bản lấy mẫu sẽ lấy:

+ Một văn bản chỉ lấy các từ có nghĩa được xuất hiện tương đối phổ biển nhất, nghĩa là các từ có số lần xuất hiện nhiều nhất do vậy lấy các từ làm mẫu thì chương trình sắp xếp số lần xuất hiện của các từ đặc trưng theo mức độ từ nhiều đến ít và chỉ lấy 1/3 tổng số từ (xét từ nhiều đến ít) và cách lấy là lấy nhiều nhất.

+ Nếu trong những từ đặc trưng được lấy ra như trên mà những từ có số lần xuất hiện ít hơn 1/3 so với từ có số lần xuất hiện nhiều nhất thì từ đó sẽ được loại ra.

+ Chương trình chấp nhận sai số là nếu hai từ có số lần xuất hiện như nhau nhưng phải loại bỏ một thì chương trình sẽ bỏ ngẫu nhiên một từ và giữ lại một từ mà không quan tâm đến ngữ nghĩa của từ bị bỏ và từ giữ lại.

\*Đối với văn bản phân loại thì:

+ Chỉ lấy các từ có nghĩa trong danh sách các từ đặc trưng được lựa chọn của văn bản mẫu.

+ Các từ có khoảng cách không giao với khoảng cách của từ trong danh sách mẫu từ đặc trưng sẽ bị loại bỏ, có nghĩa là các từ có khoảng cách lớn nhất và nhỏ nhất không nằm trong khoảng cách lớn nhất và nhỏ nhất của bất cứ từ mẫu đặc trưng nào thì sẽ bị loại bỏ khỏi danh sách từ dùng để phân loại.

Với việc gán nhãn – đánh trọng số này thì các số liệu văn bản mẫu sẽ trở thành các vecto:

VMi(M0(x0,y0),M1(x1,y1),M2(x2,y2),......Mm(xm,ym))

Và các văn bản cần phân nhóm sẽ chuyển thành các vecto:

Vi(C0(x0,y0),C1(x1,y1),C2(x2,y2),....Cn(xn,yn))

Trong đó:

+ VM là vector văn bản mẫu.

+ Vi là vector văn bản cần phân loại.

+ Mi là ký tự đặc trưng thứ i của văn bản mẫu.

+ Ci là ký tự đặc trưng thứ i của văn bản cần phân loại.

+ xi là giá trị khoảng cách nhỏ nhất của từ đặc trưng thứ i trong văn bản.

+ yi là giá trị khoảng cách lớn nhất của từ đặc trưng thứ i trong văn bản.

Từ việc so sánh các vecto này bằng phương pháp máy hỗ trợ vector (Support Vector Machine - SVM) sẽ nhận biết được văn bản cần phân loại thuộc nhóm văn bản nào.

**b.9. Sử dụng thuật toán để phân loại văn bản**

Đây là bước chính yếu của chương trình:

+ Đầu vào: vector các từ, dữ liệu chuẩn của các nhóm văn bản.

+ Đầu ra: xác định nhóm của văn bản.

Văn bản sẽ thực hiện các quy trình như tiền xử lý – tách câu – tách từ, sau khi được tách từ sẽ được biểu diễn được biểu diễn với dạng vecto với các thành phần vecto này là các trọng số của các từ:

V(C0(x0,y0), C1(x1,y1), C2(x2,y2),... Cn(xn,yn))

Trong đó:

+ V là vector văn bản cần phân loại.

+ Ci là ký tự đặc trưng thứ i của văn bản cần phân loại.

+ xi là giá trị khoảng cách nhỏ nhất của từ đặc trưng thứ i trong văn bản.

+ yi là giá trị khoảng cách lớn nhất của từ đặc trưng thứ i trong văn bản.

Với mỗi số đặc trưng được chọn, các tài liệu được biểu diễn dưới dạng vecto thưa dùng cách định trọng số từ TFIDF. Mỗi vector thưa gồm 2 mảng:

+ Một mảng số nguyên lưu chỉ số của các giá trị khác 0 của ký tự đặc trưng, số này được lấy từ số chỉ mục của từ đặc trưng mẫu trong cơ sở dữ liệu văn bản mẫu.

+ Một mảng số thực lưu các giá trị khác 0 tương ứng với ký tự này, nó là khoảng cách lớn nhất và nhỏ nhất của ký tự đặc trưng xuất hiện trong văn bản.

Sở dĩ dùng các vector thưa là do số từ xuất hiện trong mỗi tài liệu là rất nhỏ so với tổng số từ được sử dụng, điều này một mặt tiết kiệm bộ nhớ, mặt khác làm tăng tốc độ tính toán lên đáng kể. Thực hiện phân loại văn bản bằng phương pháp SVM, trong đề tài đã sử dụng các hàm để làm thư viện, các hàm ở đây giúp cho việc tính toán các vector của số liệu đọc được, các vector của tập số liệu huấn luyện và các kết quả đưa ra sẽ ở dạng ma trận kết quả và chương trình sẽ nhận dạng ma trận kết quả này để phân loại văn bản thành các nhóm văn bản, đối với các kết quả cho kết quả xác định không xác định rõ ràng thì chương trình sẽ đánh dấu là không xác định được và đưa ra tất cả các kết quả gần giống nhất để phân loại thành văn bản cụ thể.

**CHƯƠNG 3: TÌM HIỂU SVM VÀ XÂY DỰNG CHƯƠNG TRÌNH**

1. **Phân loại văn bản Tiếng Việt theo chủ đề định sẵn**
2. **Tổng quan về SVM**

SVM là phương pháp phân loại rất hiệu quả được Vapnik giới thiệu năm 1995. Ý tưởng của phương pháp là cho trước một tập huấn luyện được biểu diễn trong không gian vector, trong đó mỗi một văn bản được xem như một điểm trong không gian này. Phương pháp này tìm ra một siêu mặt phẳng h quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng, tạm gọi là lớp + (cộng) và lớp – (trừ). Chất lượng của siêu mặt phẳng này được quyết định bởi một khoảng cách (được gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khoảng cách biên càng lớn thì càng có sự phân chia tốt các điểm ra thành hai lớp, nghĩa là sẽ đạt được kết quả phân loại tốt. Mục tiêu của thuật toán SVM là tìm được khoảng cách biên lớn nhất để tạo kết quả phân loại tốt. Xét bài toán phân loại đơn giản nhất - phân loại hai phân lớp với tập dữ liệu mẫu: {(xi, yi)| i = 1, 2, ..., N, xi ∈Rm}. Trong đó mẫu là các vector đối tượng được phân loại thành các mẫu dương và mẫu âm.

Các mẫu dương là các mẫu xi thuộc lĩnh vực quan tâm và được gán nhãn yi = 1.

Các mẫu âm là các mẫu xi không thuộc lĩnh vực quan tâm và được gán nhãn yi = −1.

**Mô hình phân loại SVM**

Mặt siêu phẳng Lề tối ưu

Các mẫu dương

Lề

Các mẫu âm

Trong trường hợp này, bộ phân loại SVM là mặt siêu phẳng phân tách các mẫu dương khỏi các mẫu âm với độ chênh lệch cực đại, trong đó độ chênh lệch – còn gọi là lề (margin) xác định bằng khoảng cách giữa các mẫu dương và các mẫu âm gần mặt siêu phẳng nhất. Mặt siêu phẳng này được gọi là mặt siêu phẳng lề tối ưu. Các mặt siêu phẳng trong không gian đối tượng có phương trình là wTx + b = 0, trong đó w là vector trọng số, b là độ dịch. Khi thay đổi w và b, hướng và khoảng cách từ gốc tọa độ đến mặt siêu phẳng thay đổi. Bộ phân loại SVM được định nghĩa như sau:



Trong đó:

 nếu z ≥ 0

 nếu z < 0

Nếu f(x) = +1 thì x thuộc về lớp dương (lĩnh vực được quan tâm) và ngược lại, nếu f(x) = −1 thì x thuộc về lớp âm (các lĩnh vực khác). Máy học SVM là một họ các mặt siêu phẳng phụ thuộc vào các tham số w và b. Mục tiêu của phương pháp SVM là ước lượng w và b để cực đại hóa lề giữa các lớp dữ liệu dương và âm. Các giá trị khác nhau của lề cho ta các họ mặt siêu phẳng khác nhau và lề càng lớn thì năng lực của máy học càng giảm. Như vậy, cực đại hóa lề thực chất là việc tìm một máy học có năng lực nhỏ nhất. Quá trình phân loại là tối ưu khi sai số phân loại là cực tiểu. Nếu tập dữ liệu huấn luyện là khả tách tuyến tính, có các ràng buộc sau:

nếu

nếu

Hai mặt siêu phẳng có phương trình là:  được gọi là các siêu mặt phẳng hỗ trợ (các đường nét đứt trên hình).

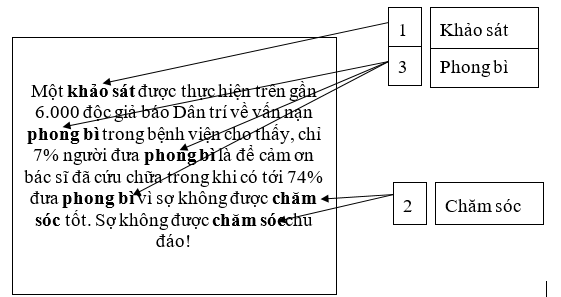
1. **Tiền xử lý văn bản**

Văn bản phải được tiền xử lý trước khi đưa huấn luyện hoặc phân loại. Tiền xử lý văn bản bao gồm: tách văn bản cần xử lý thành các câu, các đoạn theo dấu chấm câu, ứng với mỗi đoạn ta tiến hành tách thành các từ theo ngữ nghĩa, biểu diễn văn bản thành dạng vector.

Đối với ngôn ngữ đa nghĩa như tiếng Việt việc tách các từ trong câu đúng theo ngữ nghĩa là vô cùng khó, hiện nay có nhiều công trình nghiên cứu căn cứ vào xác xuất thống kê, hay là dựa vào các kết quả tìm kiếm google,… tách từ cho ra kết quả khá chính xác. Trong giới hạn thời gian hạn hẹp của đề tài em đã sử dụng phương pháp tách từ dựa trên từ điển và ưu tiên “từ ghép” trước. Với phương pháp này các từ ghép có chiều dài lớn sẽ được ưu tiên tách trước, việc tách như vậy trong một số trường hợp cá biệt sẽ không đúng về mặt ngữ nghĩa (dẫn đến sai số khi phân loại) tuy nhiên kết quả tổng quan chấp nhận được (đúng khoảng 60% - 70% các trường hợp).

Ví dụ: “Học sinh học sinh học” sẽ được tách thành

“Học sinh” + “học sinh” + “học” → sai



**Lọc từ trong văn bản**

Sau khi tách các từ trong văn bản xong, ta sẽ tiến hành thống kê lại các từ xuất hiện trong văn bản, tần suất. Các từ vô nghĩa hoặc sai sẽ được loại bỏ, chỉ lưu lại các từ ngữ đặc trưng cho văn bản.

……….

1

Khảo sát

3

Phong bì

2

Chăm sóc

544:2 656:3 876:1

……….

544. Chăm sóc

……….

656. Phong bì

……….

876. Khảo sát

**Quá trình tiền xử lý**

Ta thấy việc lấy tần suất của từ khá đơn giản bằng cách tách từ và thống kê. Sau khi phân tách toàn bộ văn bản và tính tần suất các từ ta lưu kết quả theo cấu trúc như sau:

… [mã từ 1]:[tần suất từ 1] [mã từ 2]:[tần suất từ 2]….

Trong đó mã từ chính là chỉ mục của từ trong bộ từ điển rút gọn. Các từ được lưu theo thứ tự ưu tiên mã từ trong từ điển.

1. **Phân loại văn bản và SVM**

Phân loại văn bản là một tiến trình đưa các văn bản chưa biết chủ đề vào các lớp văn bản đã biết (tương ứng với các chủ đề hay lĩnh vực khác nhau). Mỗi lĩnh vực được xác định bởi một số tài liệu mẫu của lĩnh vực đó. Để thực hiện quá trình phân loại, các phương pháp huấn luyện được sử dụng để xây dựng bộ phân loại từ các tài liệu mẫu, sau đó dùng bộ phân loại này để dự đoán lớp của những tài liệu mới (chưa biết chủ đề).

Trong quá trình phân loại, các văn bản được biểu diễn dưới dạng vector với các thành phần (chiều) của vector này là các trọng số của các từ. Ở đây, chúng ta bỏ qua thứ tự giữa các từ cũng như các vấn đề ngữ pháp khác. Dưới đây là một số phương pháp định trọng số từ thông dụng:

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

- TFIDF: trọng số từ là tích của tần suất từ TF và tần suất tài liệu nghịch đảo của từ đó và được xác định bằng công thức IDF = log(N / DF) + 1 trong đó:

+ N: kích thước của tập tài liệu huấn luyện.

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

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

Thông thường những từ này là những từ có số lần xuất hiện quá ít hoặc quá nhiều. Tuy vậy việc loại bỏ những từ này có thể không làm giảm đáng kể số lượng các đặc trưng. Với số lượng các đặc trưng lớn bộ phân loại sẽ học chính xác tập tài liệu huấn luyện, tuy vậy nhiều trường hợp cho kết quả dự đoán kém chính xác đối với các tài liệu mới. Để tránh hiện tượng này, ta phải có một tập tài liệu mẫu đủ lớn để huấn luyện bộ phân loại.

Tuy vậy, thu thập được tập mẫu đủ lớn tương ứng với số lượng đặc trưng thường khó thực hiện được trong thực tế. Do đó để cho bài toán phân loại có hiệu quả thực tiễn, cần thiết phải làm giảm số lượng đặc trưng. Có nhiều phương pháp chọn đặc trưng hiệu quả, trong phạm vi đề tài em chọn phương pháp thống kê và loại trừ các từ đơn vô nghĩa, các liên từ hoặc các từ đơn có tần suất lặp lại quá nhiều trong quá trình tiền xử lý văn bản (các từ đơn thường không mang nhiều ý nghĩa đặc trưng cho nhóm bằng từ ghép).

Khi sử dụng các phương pháp chọn đặc trưng, ta có thể loại bỏ đi nhiều từ quan trọng, dẫn đến mất mát nhiều thông tin, điều đó làm cho độ chính xác phân loại sẽ giảm đi đáng kể. Trong thực tế, theo thí nghiệm của Joachims, rất ít đặc trưng không có liên quan và hầu hết đều mang một thông tin nào đó, vì vậy một bộ phân loại tốt nên được huấn luyện với nhiều đặc trưng nhất nếu có thể. Điều này làm cho SVM trở thành một phương pháp thích hợp cho phân loại văn bản, bởi vì giải thuật SVM có khả năng điều chỉnh năng lực phân loại tự động đảm bảo hiệu suất tổng quát hóa tốt, thậm chí cả trong không gian dữ liệu có số chiều cao (số đặc trưng rất lớn) và lượng tài liệu mẫu là có hạn.

Trong những thí nghiệm phân loại văn bản tiếng Việt được thực hiện, em cũng nhận thấy dữ liệu văn bản tiếng Việt nói chung là khả tách. Khi dữ liệu là khả tách thì giải thuật SVM chỉ cần tập trung vào cực đại hóa lề, do đó có thể dẫn tới một hiệu suất tổng quát hóa tốt.

Một điểm đáng chú ý nữa khi huấn luyện SVM cho phân loại văn bản là ta có thể xây dựng được nhiều bộ phân loại khác nhau bằng cách chọn những hàm nhân phù hợp. Nhưng không như các phương pháp khác, mô hình của máy học (các tham số w, b tối ưu) được học một cách tự động trong quá trình huấn luyện SVM. Những phân tích trên đây cho thấy SVM có nhiều điểm phù hợp cho việc ứng dụng trong phân loại văn bản. Và trên thực tế, các thí nghiệm phân loại văn bản tiếng Anh chỉ ra rằng SVM đạt được độ chính xác phân loại cao và tỏ ra xuất sắc hơn so với các phương pháp phân loại văn bản khác.

1. **Phân tích thiết kế hệ thống chương trình**

**Mô tả bài toán**

Công việc phân loại văn bản là một quy trình bao gồm nhiều bước: đọc văn bản, xử lý thông tin sơ bộ, đối chiếu với mẫu văn bản phân loại, phân loại văn bản theo mẫu, thống kê văn bản đã phân loại.

Đây là một quy trình phức tạp, dễ gây nhầm lẫn. Ngoài ra với số lượng văn bản nhiều việc phân loại sẽ gặp nhiều khó khăn. Công tác lưu trữ và thống kê văn bản đã phân loại chậm ảnh hưởng đến quy trình hoạt động của hệ thống.

Vì vậy việc áp dụng công nghệ thông tin vào quy trình này là cần thiết. Chương trình phân loại văn bản sẽ giải quyết được những khó khăn cũng như tăng tốc độ xử lý văn bản, thuận tiện hơn với khối lượng văn bản nhiều.

**Xác định quy trình làm việc của văn bản**

+ Nạp dữ liệu văn bản và chuẩn hóa.

+ Xây dựng từ điển từ rút gọn tạm thời.

+ Đối chiếu và phân loại văn bản.

+ Thống kê văn bản phân loại.

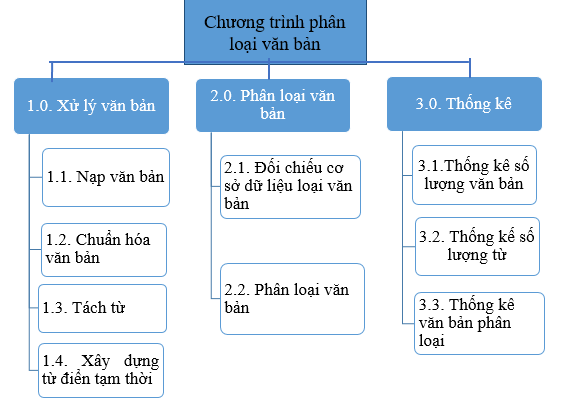
**Yêu cầu của chương trình**

+ Chương trình nhập văn bản xử lý theo kiểu nhập file text có sẵn trong máy.

+ Chương trình hoạt động theo từng bước theo yêu cầu của người dùng: xử lý văn bản, phân loại văn bản.

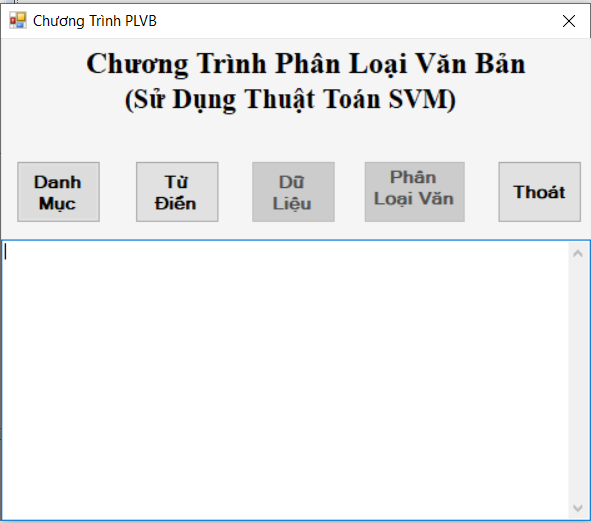
+ Chương trình hiện ra cho người dùng thấy được từ điển đầy đủ, từ điển rút gọn của từng thể loại văn bản, phiên dịch.

+ Chương trình hiện danh sách những văn bản đã xử lý, những văn bản đã phân loại.



**Sơ đồ chức năng**

**Giới thiệu chương trình**



**Khung giao diện chính của chương trình**

Gồm:

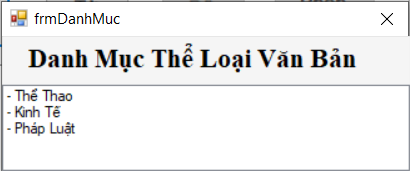
+ Tên chương trình: chương trình phân loại văn bản.

+ Năm phím chức năng: Danh mục, Từ điển, Dữ liệu, Phân loại Văn bản, Thoát.

+ Khung TextBox để hiển thị ra kết quả thực hiện chương trình.

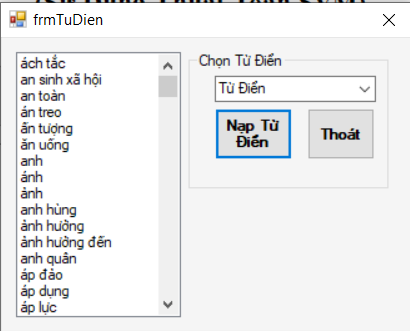
Thực hiện chương trình theo từng nút (bước) của chương trình Demo cụ thể như sau:

**\* Bước 1: Nút danh mục**



Nút danh mục giới thiệu những chủ đề mà chương trình phân loại. Em nghiên cứu giới thiệu 3 chủ đề phân loại văn bản mà chương trình phân loại là : Thể thao, Kinh tế, Pháp luật.

**\* Bước 2: Nút từ điển**



Nút từ điển mở ra khung từ điển của chương trình, khung từ điển là khung kiển thị ra các từ điển của chương trình, tại đây có một ComboBox để lựa chon hiển thị các từ điển của chương trình lên ListBox bên trái, bao gồm: từ điển gốc, từ điển tiếng Việt được nạp sẵn (tìm kiếm từ nguồn Internet) và kết hợp với các bộ từ điển:

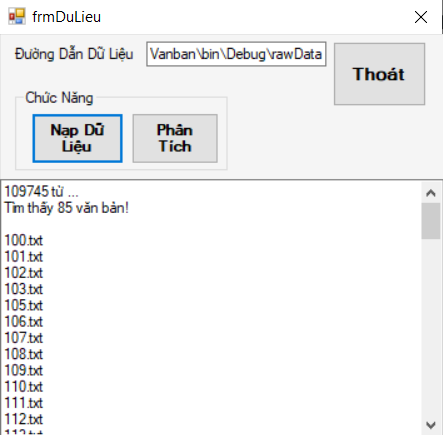
+ Từ điển tiếng Việt là kết hợp từ hai từ điển: từ điển 78000 từ của Đinh Điền và từ điển của chương trình Vikass (chương trình phân loại tin từ điện tử).

+ Từ điển các từ hư (Stop word) cũng sử dụng của chương trình ViKass.

+ Từ điển từ đồng nghĩa tự xây dựng là những từ đặc trưng của chủ đề phân loại nghiên cứu tìm kiếm và thực hiện.

Ngoài ra khung từ điển còn có thêm nút chức năng nạp từ điển để thực hiện việc nạp từ điển đã chọn vào ListBox.

**\* Bước 3: Nút dữ liệu**



Gồm :

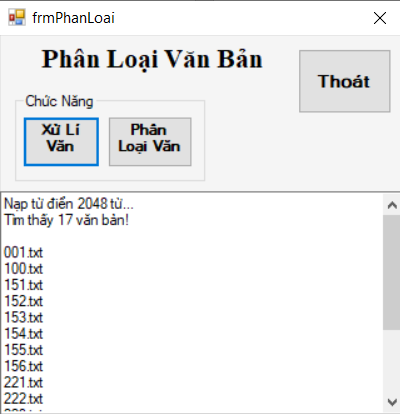
+ Đường dẫn dữ liệu: là đường dẫn trỏ đến file chứa các văn bản cần phân loại, định dạng văn bản mà chương trình phân loại được là file text (.txt).

+ Các phím chức năng bao gồm:

Nạp dữ liệu – ra lệnh nạp dữ liệu văn bản phân loại và chương trình theo đường dẫn ở trên.

Phân tích – thực hiện tách từ của văn bản phân loại.

**\* Bước 4: Nút phân loại văn bản**



Gồm :

+ Nút xử lý văn bản thực hiên chức năng xử lý văn bản bao gồm: Chuẩn hóa, tách từ.

+ Nút phân loại văn bản thực hiện chức năng đối chiếu từ điển, dựa theo thuật toán đã được nạp sẵn để tiến hành phân loại văn bản đã cho theo 3 chủ đề

**\* Bước 5: Nút thoát**

Thoát khỏi chương trình

**KẾT LUẬN**

**1. Kết quả đạt được**

Xây dựng chương trình phân loại văn bản đã đưa ra những cơ sở lý thuyết, các phương pháp phân loại văn bản và xây dựng chương trình Demo phân loại một cách chính xác sử dụng phương pháp hỗ trợ vector (SupportvectorMachine-SVM) theo đúng mục đích đã đặt ra, mặc dù kết quả không thật sự là nổi trội hơn các chương trình khác tương tự nhưng với số liệu dùng để huấn luyện chưa được đa dạng nên vẫn còn có sự nhập nhằng thì kết quả này là hoàn toàn chấp nhận được. Qua chương trình phân loại văn bản này ta cũng thấy để tăng hiệu quả của một chương trình phân loại văn bản thì việc ứng dụng cơ sở lý thuyết, lựa chọn phương pháp và xây dựng bộ dữ liệu là điều quan trọng để kết hợp với nhau để tăng hiệu quả phân loại một cách chính xác nhất.

**2. Hướng phát triển**

- Các hướng cải tiến của chương trình:

+ Sử dụng phương pháp phân tích câu có mức độ chính xác hơn.

+ Sử dụng bộ dữ liệu từ điển dùng để so sánh, phân loại văn bản một cách chính xác nhất.

+ Số liệu mẫu nhiều hơn thì có thể đảm bảo độ chính xác cao hơn.

+ Số liệu mẫu khi lấy vào cần phải có chọn lọc chính xác, tránh sự nhập nhằng giữa các mẫu thử làm kết quả bị hạn chế.

- Các hướng nghiên cứu trong tương lai:

+ Bổ sung thêm bộ phân tích ngữ nghĩa tiếng Việt để tăng mức độ chính xác.

+ Nghiên cứu thêm các thuật toán để bổ sung cho cơ sở dữ liệu phân tích.

+ Nghiên cứu thêm cơ chế dùng kiểm tra được các trang web trên mạng để hỗ trợ cơ chế tìm kiếm và phân loại trực tuyến.

**TÀI LIỆU THAM KHẢO**

<https://nguyenvanhieu.vn/phan-loai-van-ban-tieng-viet/>

<https://github.com/search?l=C%23&q=text+classifier&type=Repositories>

<https://github.com/search?l=C%23&q=svm&type=Repositories>

<https://www.slideshare.net/hoangdaithoqb/luan-van-90720703>

<https://topdev.vn/blog/top-nhung-thuat-toan-machine-learning-ma-bat-cu-data-scientist-nao-cung-can-phai-biet-phan-2/>

<https://devmaster.edu.vn/10-thuat-toan-machine-learning-ma-lap-trinh-vien-can-biet.html>