**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**---------------------------------------**



**NGUYỄN KIM ANH**

**ỨNG DỤNG THUẬT TOÁN MÁY VECTOR HỖ TRỢ TRONG PHÂN LOẠI THÔNG TIN VĂN BẢN TRÊN HỆ THỐNG WEBSITE TUYỂN DỤNG**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

***(Theo định hướng ứng dụng)***

HÀ NỘI - 2016

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**---------------------------------------**



**NGUYỄN KIM ANH**

**ỨNG DỤNG THUẬT TOÁN MÁY VECTOR HỖ TRỢ TRONG PHÂN LOẠI THÔNG TIN VĂN BẢN TRÊN HỆ THỐNG WEBSITE TUYỂN DỤNG**

**CHUYÊN NGÀNH : KHOA HỌC MÁY TÍNH**

**MÃ SỐ: 0 60.48.01.01**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

***(Theo định hướng ứng dụng)***

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. PHAN THỊ HÀ

HÀ NỘI - 2016

**LỜI CAM ĐOAN**

Luận văn này là thành quả của quá trình học tập nghiên cứu của tôi cùng sự giúp đỡ, khuyến khích của các quý thầy cô sau 2 năm tôi theo học chương trình đào tạo Thạc sĩ, chuyên ngành Khoa học máy tính của trường Học viện Công nghệ Bưu chính Viễn thông.

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận văn có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn sách, tạp chí được liệt kê trong danh mục các tài liệu tham khảo và được trích dẫn hợp pháp.

|  |  |
| --- | --- |
|  | TÁC GIẢ  **Nguyễn Kim Anh** |

**LỜI CÁM ƠN**

Tôi xin gửi lời cảm ơn và tri ân tới các thầy cô giáo, cán bộ của Học viện Công nghệ Bưu chính Viễn thông đã giúp đỡ, tạo điều kiện tốt cho tôi trong quá trình học tập và nghiên cứu chương trình Thạc sĩ.

Tôi xin gửi lời cảm ơn sâu sắc tới Tiến sĩ Phan Thị Hà đã tận tình hướng dẫn, giúp đỡ và động viên tôi để hoàn thành tốt nhất Luận văn “ỨNG DỤNG THUẬT TOÁN MÁY VECTOR HỖ TRỢ TRONG PHÂN LOẠI THÔNG TIN VĂN BẢN TRÊN HỆ THỐNG WEBSITE TUYỂN DỤNG“

Do vốn kiến thức lý luận và kinh nghiệm thực tiễn còn ít nên luận văn không tránh khỏi những thiếu sót nhất định. Tôi xin trân trọng tiếp thu các ý kiến của các thầy, cô để luận văn được hoàn thiện

Trân trọng cám ơn.

Tác giả.

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc472150174)

[1.1 Tính cấp thiết của đề tài 1](#_Toc472150175)

[1.2 Tổng quan về vấn đề nghiên cứu 1](#_Toc472150176)

[1.3 Mục đích, đối tượng, phạm vi và phương pháp nghiên cứu 2](#_Toc472150177)

[1.4 Cấu trúc luận văn 3](#_Toc472150178)

[Chương 1 - BÀI TOÁN PHÂN LOẠI VĂN BẢN 5](#_Toc472150179)

[1.1. Phát biểu bài toán phân loại văn bản 5](#_Toc472150180)

[1.2. Một số phương pháp phân loại văn bản 7](#_Toc472150181)

[***1.2.1.*** ***Thuật toán K–Nearest Neighbor (kNN)*** 7](#_Toc472150182)

[***1.2.2.*** ***Thuật toán cây quyết định (Decision tree)*** 8](#_Toc472150183)

[***1.2.3.*** ***Thuật toán SVM (Support vector machine)*** 10](#_Toc472150184)

[***1.2.4.*** ***Thuật toán Naïve Bayes*** 11](#_Toc472150185)

[**1.3.** **Các mô hình biểu diễn văn bản** 13](#_Toc472150186)

[***1.3.1.*** ***Mô hình Boolean*** 13](#_Toc472150187)

[***1.3.2.*** ***Mô hình không gian vector*** 14](#_Toc472150188)

[***1.3.3.*** ***Mô hình xác suất (Mô hình túi các từ)*** 16](#_Toc472150189)

[**1.4.** **Lựa chọn đặc trưng trong biểu diễn văn bản** 17](#_Toc472150190)

[***1.4.1.*** ***Thu gọn đặc trưng biểu diễn*** 17](#_Toc472150191)

[***1.4.2.*** ***Định luật Zipf*** 18](#_Toc472150192)

[**1.5.** **Phương pháp đánh trọng số cho từ khóa** 18](#_Toc472150193)

[***1.5.1.*** ***Phương pháp boolean*** 19](#_Toc472150194)

[***1.5.2.*** ***Phương pháp dựa trên tần số*** 19](#_Toc472150195)

[Chương 2 - ÁP DỤNG SVM VÀO BÀI TOÁN PHÂN LOẠI VĂN BẢN 21](#_Toc472150196)

[2.1. Thu thập và tiền xử lý văn bản 21](#_Toc472150199)

[***2.1.1.*** ***Thu thập địa chỉ trang web URL*** 21](#_Toc472150200)

[***2.1.2.*** ***Lọc nội dung chính của trang web*** 21](#_Toc472150201)

[***2.1.3.*** ***Tiền xử lý văn bản*** 24](#_Toc472150202)

[**2.2.** **Trình bày chi tiết thuật toán Máy vector hỗ trợ SVM** 33](#_Toc472150203)

[***2.2.1.*** ***Giới thiệu thuật toán*** 33](#_Toc472150204)

[***2.2.2.*** ***Ý tưởng thuật toán*** 33](#_Toc472150205)

[***2.2.3.*** ***Thuật toán*** 34](#_Toc472150206)

[1.5 Phân loại đa lớp nội dung văn bản với SVM 42](#_Toc472150207)

[Chương 3 - THỬ NGHIỆM VÀ ĐÁNH GIÁ 47](#_Toc472150208)

[3.1. Xây dựng bộ dữ liệu và tiền xử lý văn bản 47](#_Toc472150212)

[**3.2.** **Giai đoạn phân lớp** 50](#_Toc472150213)

[***3.2.1.*** ***URL phục vụ cho ứng dụng*** 50](#_Toc472150214)

[***3.2.2.*** ***Triển khai ứng dụng*** 50](#_Toc472150215)

[KẾT LUẬN 52](#_Toc472150216)

[DANH MỤC TÀI LIỆU THAM KHẢO 54](#_Toc472150217)

[PHỤ LỤC 56](#_Toc472150218)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Ý nghĩa tiếng Anh** | **Ý nghĩa tiếng Việt** |
| 1 | SVM | Support vector machine | Máy vector hỗ trợ | |
| 2 | NB | Naïve Bayes |  | |
| 3 | kNN | K–Nearest Neighbor | K – Láng giềng gần nhất | |
| 4 | TF | Term Frequency | Tần số xuất hiện của 1 từ | |
| 5 | IDF | Inverse Document Frequency | Tần số nghịch của 1 từ trong tập văn bản | |
| 6 | RSS | Really Simple Syndication | Định dạng tài liệu | |

**DANH MỤC CÁC BẢNG BIỂU**

**DANH MỤC CÁC HÌNH VẼ**

# MỞ ĐẦU

## Tính cấp thiết của đề tài

Trong cuộc sống hiện đại ngày nay, nhu cầu cập nhật và sử dụng thông tin trên internet là một phần thiết yếu trong hoạt động hàng ngày của mỗi người. Tại Việt Nam, với sự bùng nổ của công nghệ thông tin, nhu cầu đọc báo, tìm kiếm những thông tin trên internet đã trở thành một thói quen thường nhật, trong đó, tìm việc làm là một trong những nhu cầu thiết yếu nhất hiện nay. Một yêu cầu được đặt ra là: Làm sao tìm kiếm thông tin việc làm một cách có hiệu quả nhất? Với một khối lượng thông tin quá lớn và đòi hỏi phải xử lý nhanh thì việc phân loại thủ công là điều không thể. Hướng giải quyết được đưa ra là xây dựng một giải pháp cho phép tự động phân loại các thông tin trên theo từng nhóm nghề nghiệp. Kỹ thuật Support Vector Machines (SVM) được đánh giá là công cụ phân loại phổ biến hiện nay cho những bài toán phân lớp phi tuyến. Nhiều ứng dụng đã và đang được xây dựng dựa trên kỹ thuật SVM rất hiệu quả.

Xuất phát từ yêu cầu thực tế, nhằm hướng tới việc phân loại nội dung văn bản tiếng Việt, luận văn có tựa đề: **“*Ứng dụng thuật toán Máy vector hỗ trợ trong phân loại thông tin văn bản trên hệ thống Website tuyển dụng*”**

## Tổng quan về vấn đề nghiên cứu

Sự ra đời và phát triển nhanh chóng của World Wide Web đã tạo điều kiện thuận lợi cho việc phân phối và chia sẻ thông tin trên Internet, do đó dẫn đến bùng nổ thông tin cả về số lượng, chất lượng và các chủ đề thông tin trên đó. Trong luận văn này, em tập trung nghiên cứu về phương pháp phân loại văn bản tiếng Việt, cụ thể là thu thập tự động nội dung của Website, thực hiện và tối ưu các bước tiền xử lý, sau đó phân loại nội dung theo một số lĩnh vực nghề nghiệp bằng một kỹ thuật phân loại khá tốt hiện nay đó là máy vector hỗ trợ SVM.

Công tác phân loại luôn được các thư viện và cơ quan thông tin trên thế giới hết sức quan tâm. Theo Yang & Xiu, 1999, “Việc phân loại văn bản tự động là việc gán các nhãn phân loại lên một văn bản mới dựa trên mức độ tương tự của văn bản đó so với các văn bản đã được gán nhãn trong tập huấn luyện”.

Từ trước đến nay, phân loại văn bản tự động trong tiếng Anh đã có rất nhiều công trình nghiên cứu và đạt được kết quả đáng khích lệ. Dựa trên các thống kê của Yang & Xiu (1999), một số phương pháp phân loại thông dụng hiện nay là: Support Vector Machine - Joachims, 1998; k-Nearest Neighbor - Yang, 1994; Linear Least Squares Fit - Yang and Chute, 1994; Neural Network - Wiener et al, 1995; Naïve Bayes - Baker and Mccallum, 2000. Các phương pháp trên đều dựa vào xác suất thống kê hoặc thông tin về trọng số của từ trong văn bản.

## Mục đích, đối tượng, phạm vi và phương pháp nghiên cứu

***Mục đích nghiên cứu:***

* Tìm hiểu về bài toán phân loại văn bản
* Tìm hiểu về xử lý và phân loại văn bản
* Tìm hiểu và áp dụng thuật toán SVM trong việc phân loại các thông tin việc làm được trích rút từ website

***Đối tượng nghiên cứu:***

* Bài toán tiền xử lý văn bản
* Bài toán phân lớp văn bản
* Phương pháp SVM trong phân loại văn bản

***Phạm vi nghiên cứu***

* Nghiên cứu tổng quan về xử lý ngôn ngữ tự nhiên và bài toán phân loại văn bản.
* Sau đó, luận văn đi sâu tìm hiểu ứng dụng phương pháp học máy SVM trong việc phân loại thông tin tuyển dụng trên website timviecnhanh.com

***Phương pháp nghiên cứu:***

Kết hợp nghiên cứu lý thuyết và thực hành

* Lý thuyết: Tìm hiểu về bài toán xử lý và phân tích văn bản tiếng Việt, phương pháp học máy.
* Phương pháp thực nghiệm: Cài đặt phương pháp học máy SVM cho việc phân lớp thông tin việc làm trên website.

## Cấu trúc luận văn

Với mục tiêu đặt ra như vậy, nội dung và kết quả của luận văn được trình bày qua 3 chương như sau:

CHƯƠNG I: BÀI TOÁN PHÂN LOẠI VĂN BẢN

Giới thiệu chương:

Chương này giới thiệu về bài toán phân loại văn bản, các phương pháp phân loại văn bản, cách biểu diễn văn bản. Chương 1 cũng so sánh phương pháp SVM với các phương pháp phân loại khác từ đó đưa ra lý do tại sao luận văn lại chọn sử dụng SVM.

Nội dung chương:

* Phát biểu bài toán phân loại văn bản
* Một số phương pháp phân loại văn bản
* Các mô hình biểu diễn văn bản
* Lựa chọn đặc trưng trong biểu diễn văn bản
* Phương pháp đánh trọng số cho từ khóa

Kết luận chương:

Chương này đã giới thiệu tổng quan về bài toán phân loại văn bản, các phương pháp phân loại, một số kỹ thuật để biểu diễn văn bản và đưa ra lý do chọn thuật toán SVM trong luận văn.

CHƯƠNG II: ÁP DỤNG SVM VÀO BÀI TOÁN PHÂN LOẠI VĂN BẢN

Giới thiệu chương:

Chương 2 trình bày chi tiết về phương pháp máy vector hỗ trợ SVM trong phân loại văn bản. Đồng thời, chương này cũng giới thiệu việc thu thập nội dung văn bản từ website, sau đó thực hiện tiền xử lý văn bản và cuối cùng là phân loại văn bản.

Nội dung chương:

* Thu thập và tiền xử lý văn bản
* Trình bày chi tiết thuật toán Máy vector hỗ trợ SVM
* Phân loại đa lớp nội dung văn bản với phương pháp SVM

Kết luận chương:

Chương này đã giới thiệu chi tiết thuật toán Máy vector hỗ trợ SVM và trình bày cách thu thập, trích rút và xử lý một cách tự động các tin tức tuyển dụng. Sau đó, luận văn sẽ sử dụng SVM để thực hiện việc huấn luyện bộ phân loại cho bài toán phân loại thông tin tuyển dụng trên hệ thống website tuyển dụng.

**CHƯƠNG III: THỬ NGHIỆM VÀ ĐÁNH GIÁ**

Giới thiệu chương:

Chương này xây dựng bộ dữ liệu huấn luyện và thử nghiệm, sau đó dùng bộ dữ liệu thử nghiệm để thực hiện phân lớp và đưa ra đánh giá.

Nội dung chương:

* Xây dựng bộ dữ liệu huấn luyện và thử nghiệm
* Giai đoạn phân lớp
* Kết luận và hướng phát triển

Kết luận chương:

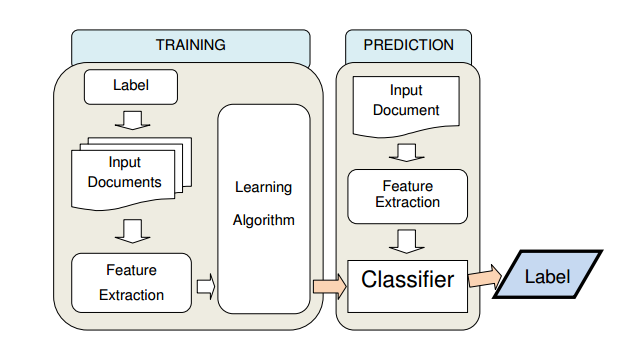
Dữ liệu huấn luyện và kiểm thử được xây dựng bán tự động. Sau đó tiến hành giai đoạn phân lớp và đánh giá kết quả.

# Chương 1 - BÀI TOÁN PHÂN LOẠI VĂN BẢN

## Phát biểu bài toán phân loại văn bản

Phân loại văn bản (hay Text Categorization hoặc Document Classificant) là quá trình gán các văn bản vào một hay nhiều chủ đề đã biết trong một tập hữu hạn các chủ đề đã được xác định từ trước. Ví dụ một bài báo trong một trang web có thể thuộc một hoặc một vài chủ đề nào đó (như thể thao, giáo dục, pháp luật, công nghệ thông tin,…). Việc phân loại có thể được tiến hành một cách thủ công: đọc nội dung của từng văn bản và gán nó vào một lớp nào đó. Tuy nhiên, đối với hệ thống gồm rất nhiều văn bản thì phương pháp này sẽ tốn rất nhiều thời gian và công sức. Do vậy cần phải có phương pháp tự động để phân loại văn bản. Phương pháp này giúp cho việc lưu trữ và truy vấn tài liệu dễ dàng hơn.

Dưới đây là hình vẽ mô tả quy trình của bài toán phân loại văn bản:



Hình 1.1 Quy trình phân loại văn bản

Để tiến hành phân loại văn bản nói chung, chúng ta sẽ thực hiện các bước như sau:

* Bước 1: Xây dựng bộ dữ liệu huấn luyện dựa vào tài liệu văn bản đã được phân loại sẵn. Tiến hành học cho bộ dữ liệu, xử lý và thu thập được dữ liệu của quá trình học là các đặc trưng riêng biệt cho từng chủ đề.
* Bước 2: Dữ liệu cần phân loại được xử lý, rút ra đặc trưng kết hợp với đặc trưng được học trước đó để phân loại và rút ra kết quả.

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 có thể dễ bị nhập nhằng. Ví dụ có bài báo về tuyển dụng nhân viên bán thuốc tại một công ty dược phẩm, bài báo này có thể xếp vào chủ đề tuyển dụng lĩnh vực Y dược hoặc cũng có thể xếp vào chủ đề lĩnh vực Bán hàng.

Về bản chất, một văn bản là một tập hợp 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 thế giới đã có nhiều công trình nghiên cứu đạt những kết quả khả quan, nhất là đối với phân loại văn bản tiếng Anh. Tuy vậy, các nghiên cứu và ứng dụng đối với văn bản tiếng Việt còn nhiều hạn chế do khó khăn về tách từ và câu. Có thể liệt kê một số công trình nghiên cứu trong nước với các hướng tiếp cận khác nhau cho bài toán phân loại văn bản, bao gồm: phân loại với máy học vectơ hỗ trợ, cách tiếp cận sử dụng lý thuyết tập thô, cách tiếp cận thống kê hình vị, cách tiếp cận sử dụng phương pháp học không giám sát và đánh chỉ mục, cách tiếp cận theo luật kết hợp. Theo các kết quả trình bày trong các công trình đó thì những cách tiếp cận nêu trên đều cho kết quả khá tốt. Tuy nhiên khó có thể so sánh các kết quả ở trên với nhau vì tập dữ liệu thực nghiệm của mỗi phương pháp là khác nhau.

Trong những năm gần đây, phương pháp phân loại sử dụng Máy vector hỗ trợ (SVM) được quan tâm và sử dụng nhiều trong những lĩnh vực nhận dạng và phân loại. SVM là một họ các phương pháp dựa trên cơ sở các hàm nhân (kernel) để tối thiểu hóa rủi ro ước lượng. Phương pháp SVM ra đời từ lý thuyết học thống kê do Vapnik và Chervonenkis xây dựng và có nhiều tiềm năng phát triển về mặt lý thuyết cũng như ứng dụng trong thực tiễn. Các thử nghiệm thực tế cho thấy, phương pháp SVM có khả năng phân loại khá tốt đối với bài toán phân loại văn bản cũng như trong nhiều ứng dụng khác (như nhận dạng chữ viết tay, phát hiện mặt người trong các ảnh, ước lượng hồi quy, ...). So sánh với các phương pháp phân loại khác, khả năng phân loại của SVM là tương đương hoặc tốt hơn đáng kể. Vì những lý do đó mà tôi đã chọn phương pháp này cho việc phân loại văn bản tiếng Việt, cụ thể thuật toán và ứng dụng sẽ được trình bày trong các chương sau.

## Một số phương pháp phân loại văn bản

Hiện nay trên thế giới đã có rất nhiều công trình nghiên cứu về các phương pháp phân loại văn bản. Một số phương pháp cần kể đến là: Naïve Bayes, Support Vector Machine, K–Nearest Neighbor, Linear Least Squares Fit, Neural Network, Naïve Bayes, Centroid–Based… Điểm chung của các phương pháp này đều dựa vào xác suất thống kê hoặc dựa vào trọng số của các từ, cụm từ trong văn bản. Trong mỗi phương pháp đều có cách tính toán khác nhau, tuy nhiên các phương pháp này đều phải thực hiện một số bước chung như: mỗi phương pháp sẽ dựa vào thông tin về sự xuất hiện của các từ trong văn bản (tần số xuất hiện trong tập văn bản…) để biểu diễn thành dạng vector, sau đó tùy từng bài toán cụ thể mà chúng ta sẽ quyết định chọn áp dụng phương pháp nào, công thức tính toán nào cho phù hợp để phân loại tập văn bản dựa trên tập các vector đã xây dựng được ở bước trên, nhằm mục đích đạt được kết quả phân loại tốt nhất.

* + 1. ***Thuật toán K–Nearest Neighbor (kNN)***

1. Giới thiệu

Đây là phương pháp truyền thống khá nổi tiếng về hướng tiếp cận dựa trên thống kê đã được nghiên cứu trong nhận dạng mẫu hơn bốn thập kỷ qua (theo tài liệu của Dasarathy năm 1991). kNN được đánh giá là một trong những phương pháp tốt nhất (áp dụng trên tập dữ liệu Reuters phiên bản 21450), được sử dụng từ những thời kỳ đầu của việc phân loại văn bản (được trình bày bởi Marsand năm 1992, Yang năm 1994, Iwayama năm 1995)

1. Ý tưởng

Khi cần phân loại một văn bản mới, thuật toán sẽ tính khoảng cách (khoảng cách Euclide, Cosine ...) của tất cả các văn bản trong tập huấn luyện đến văn bản này để tìm ra k văn bản gần nhất (gọi là k “láng giềng”), sau đó dùng các khoảng cách này đánh trọng số cho tất cả chủ đề. Trọng số của một chủ đề chính là tổng tất cả khoảng cách ở trên của các văn bản trong k láng giềng có cùng chủ đề, chủ đề nào không xuất hiện trong k láng giềng sẽ có trọng số bằng 0. Sau đó các chủ đề sẽ được sắp xếp theo mức độ trọng số giảm dần và các chủ đề có trọng số cao sẽ được chọn là chủ đề của văn bản cần phân loại.

(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)

1. Thuật toán

* Xác định giá trị tham số K (số láng giềng gần nhất)
* Tính khoảng cách giữa đối tượng cần phân lớp (Query Point) với tất cả các đối tượng trong training data (thường sử dụng khoảng cách Euclidean)
* Sắp xếp khoảng cách theo thứ tự tăng dần và xác định K láng giềng gần nhất với Query Point
* Lấy tất cả các lớp của K láng giềng gần nhất đã xác định
* Dựa vào phần lớn lớp của láng giềng gần nhất để xác định lớp cho Query Point

(Dựa vào k văn bản mẫu này đánh trọng số cho chủ đề. Chủ đề của các văn bản là chủ đề có trọng số lớn nhất)

* + 1. ***Thuật toán cây quyết định (Decision tree)***

1. Giới thiệu

Cây quyết định là một cấu trúc cây với:

* Mỗi nút trong (internal node) ứng với một phép kiểm tra trên một thuộc tính.
* Mỗi nhánh biểu diễn một kết quả của phép kiểm tra.
* Các nút lá (leaf node) biểu diễn các lớp hay các phân bố lớp. Nút cao nhất trong cây là nút gốc (root node).

1. Ý tưởng

Cây quyết định được mô tả bằng cách tính toán xác suất có điều kiện. Cây quyết định cũng có thể được mô tả như là một kỹ thuật tính toán và hỗ trợ toán học, kỹ thuật này hỗ trợ việc mô tả, phân loại và khái quát tập dữ liệu đưa vào. Dữ liệu đưa vào có dạng: (x, y) = (x1, x2, … , xk, y )

Biến phụ thuộc y là biến mà chúng ta cố gắng để biết, phân lớp hay tổng quát hóa, còn các biến x1, x2,… là các biến giúp ta thực hiện công việc đó. Để xây dựng được cây quyết định của tập dữ liệu nào đó chúng ta phải hiểu được khái niệm độ đo Entropy và Information Gain (Lợi ích thông tin). Độ đo Entropy: đặc trưng cho độ hỗn tạp (lộn xộn) của một tập bất kỳ các mẫu thử.



Lợi ích thông tin:

* Gain(S, A) là lợi ích thông tin mà thuộc tính A mang lại cho sự phân lớp tập S. A có m giá trị v1, v2, … , vm
* Ký hiệu: Svi = {x ∈ S | x có giá trị thuộc tính A là vi}
* |S| là số phần tử của tập S



1. Thuật toán

Cho tập ví dụ huấn luyện D. Tìm cây quyết định phù hợp với D

Bước 1:

Khởi tạo cây một đỉnh gốc

Toàn bộ tập ví dụ huấn luyện D đều đi vào đỉnh này.

Bước 2:

Repeat

Chọn một đỉnh lá chưa gán nhãn để phát triển gọi là đỉnh hiện thời

Giả sử tập ví dụ huấn luyện đi vào đỉnh này là S

If (S = rỗng)

Then (gán nhãn chung nhất trong D)

Else

If (tất cả các ví dụ trong S đều được gán cùng một nhãn c)

Then (đỉnh hiện thời được gán nhãn c)

Else

Đỉnh hiện thời được gán nhãn là thuộc tính A trong đó

A = argmax Gain (S, Ai)

Ai: ứng viên là nhãn của đỉnh hiện thời và mỗi giá trị v của A được gán nhãn cho nhánh đi từ A tới đỉnh mới.

Tập ví dụ huấn luyện đi tới đỉnh mới đó là Sv

trong đó Sv = {s ∈ S | s có giá trị của thuộc tính A là v}

Until (tất cả các đỉnh của cây đều được gán nhãn)

* + 1. ***Thuật toán SVM (Support vector machine)***

1. Giới thiệu

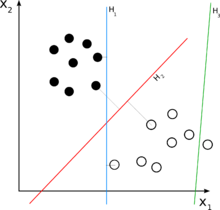
Support vector machine (SVM) – Máy vector hỗ trợlà một phương pháp phân lớp xuất phát từ lý thuyết học thống kê.

1. Ý tưởng

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.

1. Thuật toán

Cho trước một tập huấn luyện được biểu diễn trong không gian vector trong đó mỗi tài liệu là một điểm, 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 lớp + và lớp –. Chất lượng của siêu mặt phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt đồng thời việc phân loại càng chính xác. Mục đích thuật toán SVM là tìm được khoảng cách biên lớn nhất. Thuật toán này sẽ được trình bày cụ thể ở chương sau.



Hình 1.0.1: Phân lớp nhị phân

* + 1. ***Thuật toán Naïve Bayes***

1. Giới thiệu

Naïve Bayes (NB) là phương pháp phân loại dựa vào xác suất được sử dụng rộng rãi trong lĩnh vực máy học (Mitchell trình bày năm 1996, Joachims trình bày năm 1997 và Jason năm 2001) được sử dụng lần đầu tiên trong lĩnh vực phân loại bởi Maron vào năm 1961, sau đó trở nên phổ biến dùng trong nhiều lĩnh vực như trong các công cụ tìm kiếm (được mô tả năm 1970 bởi Rijsbergen), các bộ lọc mail (mô tả năm 1998 bởi Sahami)

1. Ý tưởng

Ý tưởng cơ bản của cách tiếp cận Naïve Bayes là sử dụng xác suất có điều kiện giữa từ và chủ đề để dự đoán xác suất chủ đề của một văn bản cần phân loại. Điểm quan trọng của phương pháp này chính là ở chỗ giả định rằng sự xuất hiện của tất cả các từ trong văn bản đều độc lập với nhau. Với giả định này NB không sử dụng sự phụ thuộc của nhiều từ vào một chủ đề, không sử dụng việc kết hợp các từ để đưa ra phán đoán chủ đề và do đó việc tính toán NB chạy nhanh hơn các phương pháp khác với độ phức tạp theo hàm số mũ.

1. Thuật toán

*Công thức*

* Dựa trên định lý Bayes:
* Trong đó:
* H (Hypothesis) là giả thuyết và E (Evidence) là chứng cứ hỗ trợ cho giả thuyết H
* P(E|H): xác xuất E xảy ra khi H xảy ra (xác suất có điều kiện, khả năng của E khi H đúng).
* P(H|E): xác suất hậu nghiệm của H nếu biết E.

*Áp dụng trong bài toán phân loại*

* Các dữ kiện cần có:
* Một tập dữ liệu D, trong đó mỗi ví dụ học x đã được vector hóa dưới dạng một vector n chiều (x1, x2, ..., xn)
* Một tập xác định các nhãn lớp: C={c1, c2, ..., cm}
* Với một ví dụ (mới) z\*, z\* sẽ được phân vào lớp nào?

🡪 Áp dụng định lý Bayes để xác định phân lớp phù hợp nhất đối với z:

cMAP =argmax P(ci|z) = argmax P(ci | z1, z2,..., zn) = argmax

Vì P(z1, z2,…, zn) là như nhau với các lớp nên chỉ cần tìm phân lớp với:

cMAP =argmax P(z1, z2,..., zn | ci).P(ci)

Giả sử trong phương pháp phân loại Naïve Bayes, các thuộc tính là độc lập có điều kiện đối với các lớp thì:

P(z1,z2,...,zn|ci)=

🡪 cMAP = argmax P(ci).

* 1. **Các mô hình biểu diễn văn bản**

**Bài toán**

Input: Cho tập văn bản miền ứng dụng D = {dj}, tập đặc trưng được chọn biểu diễn văn bản V = {wi}, ma trân trọng số W = (wi,j) .

Output: Tìm biểu diễn của các văn bản dj ∈D.

Một số mô hình: Mô hình Boolean, Mô hình không gian vector, Mô hình túi các từ (Mô hình xác suất)

* + 1. ***Mô hình Boolean***

Một mô hình biểu diễn vector với hàm f cho ra giá trị rời rạc với duy nhất hai giá trị đúng và sai (true và false, hoặc 0 và 1) gọi là mô hình Boolean.

Hàm f tương ứng với từ khóa ti sẽ cho ra giá trị đúng nếu và chỉ nếu từ khóa ti xuất hiện trong văn bản đó.

Mô hình Boolean được xác định như sau:

Giả sử có một tập gồm m văn bản D = {d1, d2, d3, ..., dm}, T là một tập từ vựng gồm n từ khóa T = {t1, t2,......tn}. Gọi w = (wịj) là ma trận trọng số, trong đó wij là trọng số của từ khóa ti trong văn bản dj và được xác định như sau:

Ta xét 2 văn bản sau:

VB1: Life is not only life

VB2: To life is to fight

Áp dụng mô hình Boolean sau khi tiền xử lý văn bản ta có biểu diễn sau:

Bảng 1.1 Biểu diễn văn bản mô hình Boolean



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

Cách biểu diễn văn bản thông dụng nhất là thông qua vector biểu diễn theo mô hình không gian vector (Vector Space Model). Đây là một cách biểu diễn tương đối đơn giản và hiệu quả.

Theo mô hình này, mỗi văn bản được biểu diễn thành một vector. Mỗi thành phần của vector là một từ khóa riêng biệt trong tập văn bản gốc và được gán một giá trị là hàm f chỉ mật độ xuất hiện của từ khóa trong văn bản.



Hình 1.0.2: Biểu diễn vector văn bản trong không gian 2 chiều

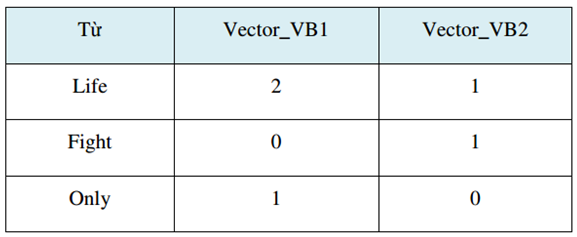
Giả sử ta có một văn bản và nó được biểu diễn bởi vector V(v1, v2,…,vn). Trong đó, vi là số lần xuất hiện của từ khóa thứ i trong văn bản. Ta xét 2 văn bản sau:

VB1: Life is not only life

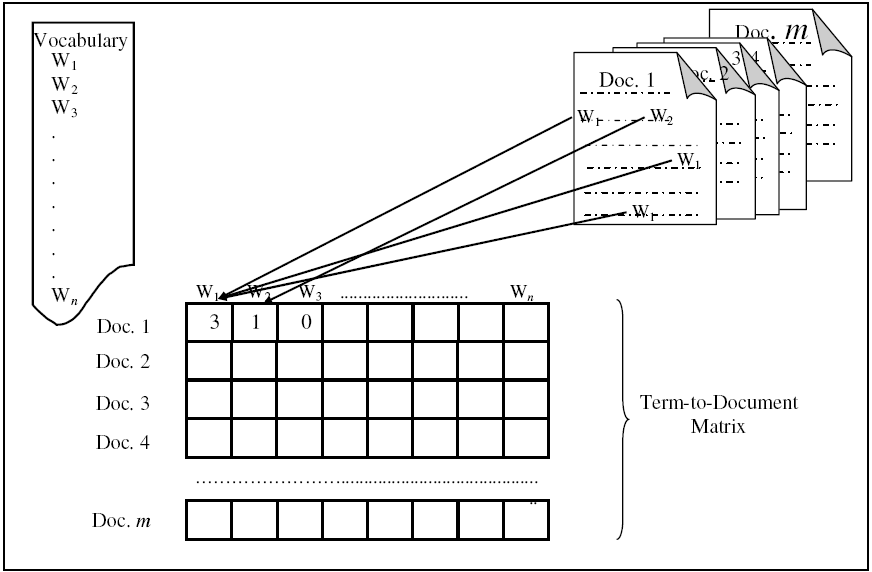
VB2: To life is to fight

Sau khi qua bước tiền xử lý văn bản, ta biểu diễn chúng như sau:

Bảng 2: Biểu diễn văn bản theo mô hình Vector



Trong các cơ sở dữ liệu văn bản, mô hình vector là mô hình biểu diễn văn bản được sử dụng phổ biến nhất hiện nay. Mối quan hệ giữa các trang văn bản được thực hiện thông qua việc tính toán trên các vector biểu diễn vì vậy được thi hành khá hiệu quả. Đặc biệt, nhiều công trình nghiên cứu về mối quan hệ "tương tự nhau" giữa các trang web (một trong những quan hệ điển hình nhất giữa các trang web) dựa trên mô hình biểu diễn vector.



Hình 1.0.3: Biểu diễn văn bản theo mô hình vector

* + 1. ***Mô hình xác suất (Mô hình túi các từ)***

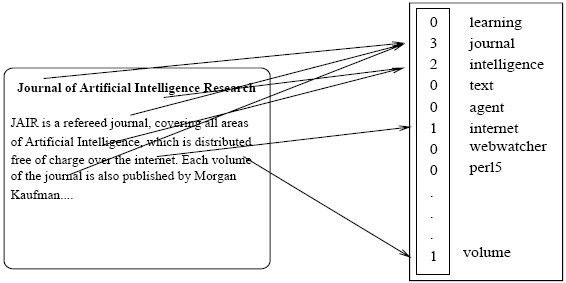
Mô hình xác suất là mô hình toán học làm việc với các biến ngẫu nhiên và phân bố xác xuất của nó. Theo thuật ngữ toán học, một mô hình xác suất có thể coi như một cặp (Y, P), trong đó Y là tập các quan sát (biến ngẫu nhiên) và P là tập các phân bố xác suất trên Y. Khi đó, sử dụng suy diễn xác suất sẽ cho ta kết luận về các phần tử của tập Y.

Văn bản trong mô hình xác suất được coi như một quan sát trong tập Y, trong đó:

* Các từ xuất hiện trong văn bản thể hiện nội dung văn bản
* Sự xuất hiện của các từ là độc lập lẫn nhau và độc lập ngữ cảnh
* Dạng đơn giản: chỉ liệt kê từ
* Dạng chi tiết: liệt kê từ và số lần xuất hiện

*Lưu ý*: Các giả thiết về tính độc lập không hoàn toàn đúng (độc lập lẫn nhau, độc lập ngữ cảnh) song mô hình thi hành hiệu quả trong nhiều trường hợp.

Khi đó văn bản sẽ gồm các từ mà nó chứa trong đó, chính vì vậy phương pháp này được gọi là biểu diễn túi - các - từ (bag-of-word)



Hình 1.0.4: Biểu diên văn bản theo mô hình túi các từ

* 1. **Lựa chọn đặc trưng trong biểu diễn văn bản**
     1. ***Thu gọn đặc trưng biểu diễn***

Một trong những giải pháp để khắc phục những vấn đề trên là thu gọn số lượng các từ để biểu diễn văn bản hay là thu gọn số lượng các đặc trưng bằng cách lựa chọn các đặc trưng có khả năng ảnh hưởng đến chất lượng phân lớp của các giải thuật phân lớp, còn các đặc trưng khác có thể bỏ qua. Việc thu gọn này cần đảm bảo sao cho các đặc trưng còn lại vẫn có khả năng "đại diện" cho toàn bộ văn bản, không làm giảm chất lượng phân lớp.

Lựa chọn đặc trưng là tiến trình lựa chọn một tập các đặc trưng (hay còn gọi là tập phổ biến) xuất hiện trong tập đào tạo và chỉ sử dụng các tập này như là các đặc trưng để biểu diễn văn bản.

* + 1. ***Định luật Zipf***

Luhn đưa ra một phương pháp đơn giản cho việc lựa chọn các từ để biểu diễn văn bản (lựa chọn đặc trưng) như sau:

* Cho một tập gồm n văn bản, tính tần số của mỗi từ duy nhất (xuất hiện một lần) trong mỗi văn bản.
* Tính tần số xuất hiện của mỗi từ trong toàn bộ tập n văn bản
* Sắp xếp tần số các từ giảm dần. Chọn một giá trị ngưỡng trên để loại bỏ các từ có tần số cao hơn ngưỡng đó. Việc này sẽ loại bỏ các từ có tần số cao không tác động tới đặc trưng cho những chủ đề cần phân loại đồng thời chuyển từ đó vào danh sách stopword
* Cũng như vậy, chọn một giá trị ngưỡng dưới để loại bỏ các từ có tần số thấp. Các từ còn lại là các từ được dùng trong quá trình đánh chỉ số văn bản. Việc chọn các từ để đánh chỉ số văn bản hay còn gọi là lựa chọn đặc trưng.
  1. **Phương pháp đánh trọng số cho từ khóa**

**Bài toán**

Input: Cho một tập văn bản miền ứng dụng D và tập từ được chọn biểu diễn văn bản V.

Output: Đánh trọng số từ trong mỗi văn bản ⇨ Xây dựng ma trận {wi,j} là trọng số của từ wi ∈ V trong văn bản dj ∈ D.

Giải pháp: Boolean, dựa theo tần số xuất hiện

* + 1. ***Phương pháp boolean***

Giả sử có một tập gồm m văn bản D = {d1, d2, d3, ..., dm}, T là một tập từ vựng gồm n từ khóa T = {t1, t2,......tn}. Gọi w = (wịj) là ma trận trọng số, trong đó wij là trọng số của từ khóa ti trong văn bản dj và được xác định như sau:

* + 1. ***Phương pháp dựa trên tần số***

1. Phương pháp dựa trên tần số từ khóa TF – Term Frequency

Các giá trị wij được tính dựa trên tần số xuất hiện của từ khóa trong văn bản. Gọi fij là số lần xuất hiện của thuật ngữ ti trong văn bản dj, khi đó wij được tính bởi một trong 3 công thức sau:

wij = fij

Hoặc wij = 1 +

Hoặc wij =

Trong phương pháp này, trọng số wij tỷ lệ thuận với số lần xuất hiện của từ ti trong văn bản dj. Khi số lần xuất hiện từ khóa ti trong văn bản dj càng nhiều thì điều đó có nghĩa là văn bản dj càng phụ thuộc vào từ khóa ti, hay nói cách khác từ khóa ti mang nhiều thông tin trong văn bản dj. Ví dụ: khi văn bản xuất hiện nhiều từ khóa “phần mềm”, điều đó có nghĩa là văn bản đang xét chủ yếu liên quan đến lĩnh vực công nghệ thông tin.

1. Phương pháp dựa trên nghịch đảo tần số văn bản IDF

Trong phương pháp này, giá trị wij được tính theo công thức sau:

wij =

Trong đó: m là số lượng văn bản và hi là số văn bản mà từ khóa ti xuất hiện.

1. Phương pháp TF × IDF

Phương pháp này là tổng hợp của hai phương pháp TF và IDF, giá trị của ma trận trọng số được tính như sau:

wij =

Phương pháp này kết hợp được ưu điểm của cả 2 phương pháp trên. Trọng số wij được tính bằng tần số xuất hiện của từ khóa ti trong văn bản dj và độ hiếm của từ khóa ti trong toàn bộ cơ sở dữ liệu.

**Kết luận chương:**

Chương này tôi đã giới thiệu tổng quan về bài toán phân loại văn bản, một số phương pháp phân loại và các kỹ thuật liên quan đến biểu diễn văn bản. Bên cạnh đó, chương 1 còn đưa ra lý do tại sao lại chọn thuật toán SVM trong luận văn. Ngoài ra cần lưu ý đến 3 yếu tố quan trọng tác động đến kết quả phân loại văn bản:

1) Cần một tập dữ liệu huấn luyện chuẩn và đủ lớn để cho thuật toán học phân loại. Nếu chúng ta có được một tập dữ liệu chuẩn và đủ lớn thì quá trình huấn luyện sẽ tốt và khi đó chúng ta sẽ có kết qủa phân loại tốt sau khi đã được học.

2) Các phương pháp trên hầu hết đều sử dụng mô hình vector để biểu diễn văn bản, do đó phương pháp tách từ trong văn bản đóng vai trò quan trọng quá trình biểu diễn văn bản bằng vector. Yếu tố này rất quan trọng, vì có thể đối với một số ngôn ngữ như tiếng Anh thì thao tác tách từ trong văn bản đơn giản chỉ là dựa vào các khoảng trắng, tuy nhiên trong các ngôn ngữ đa âm tiết như tiếng Việt và một số ngôn ngữ khác thì sử dụng khoảng trắng khi tách từ là không chính xác, do đó phương pháp tách từ là một yếu tố quan trọng.

3) Thuật toán sử dụng để phân loại phải có thời gian xử lý hợp lý, thời gian này bao gồm: thời gian học, thời gian phân loại văn bản, ngoài ra thuật toán này phải có tính tăng cường (incremental function) nghĩa là không phân loại lại toàn bộ tập văn bản khi thêm một số văn bản mới vào tập dữ liệu mà chỉ phân loại các văn bản mới mà thôi, khi đó thuật toán phải có khả năng giảm độ nhiễu (noise) khi phân loại văn bản.

# Chương 2 - ÁP DỤNG SVM VÀO BÀI TOÁN PHÂN LOẠI VĂN BẢN



## Thu thập và tiền xử lý văn bản

* + 1. ***Thu thập địa chỉ trang web URL***

1. Thu thập URL phục vụ cho dữ liệu huấn luyện

Một trong những cách hữu hiệu nhất để thu thập data phục vụ cho việc huấn luyện đó là sử dụng RSS (Really Simple Syndication – định dạng tài liệu). Thông qua RSS chúng ta có thể lấy được những tin chính mới nhất, tiêu đề, tóm tắt và cả đường link để xem toàn bộ tin.

Định dạng RSS cung cấp nội dung web và tóm lược nội dung web cùng với các liên kết đến phiên bản đầy đủ của nội dung tin đó và các [siêu-dữ-liệu](https://vi.wikipedia.org/wiki/Si%C3%AAu_d%E1%BB%AF_li%E1%BB%87u) ([meta-data](https://vi.wikipedia.org/w/index.php?title=Meta_data&action=edit&redlink=1)) khác. Thông tin này được cung cấp dưới dạng một tập tin [XML](https://vi.wikipedia.org/wiki/XML) được gọi là một RSS feed, webfeed, RSS stream hay RSS channel. Cùng với việc hỗ trợ cung cấp chia sẻ thông tin, RSS cho phép những độc giả thường xuyên của một website có thể theo dõi các cập nhật của site đó.

Cụ thể việc lấy tin RSS sẽ được trình bày ở chương 3 của luận văn.

1. Thu thập URL phục vụ cho ứng dụng

Các URL phục vụ cho ứng dụng được thu thập từ các website việc làm không hỗ trợ RSS. Từ các URL đó sẽ được đưa vào ứng dụng để tiến hành phân loại. Cụ thể các website sẽ được trình bày tại chương 3 của luận văn.

* + 1. ***Lọc nội dung chính của trang web***

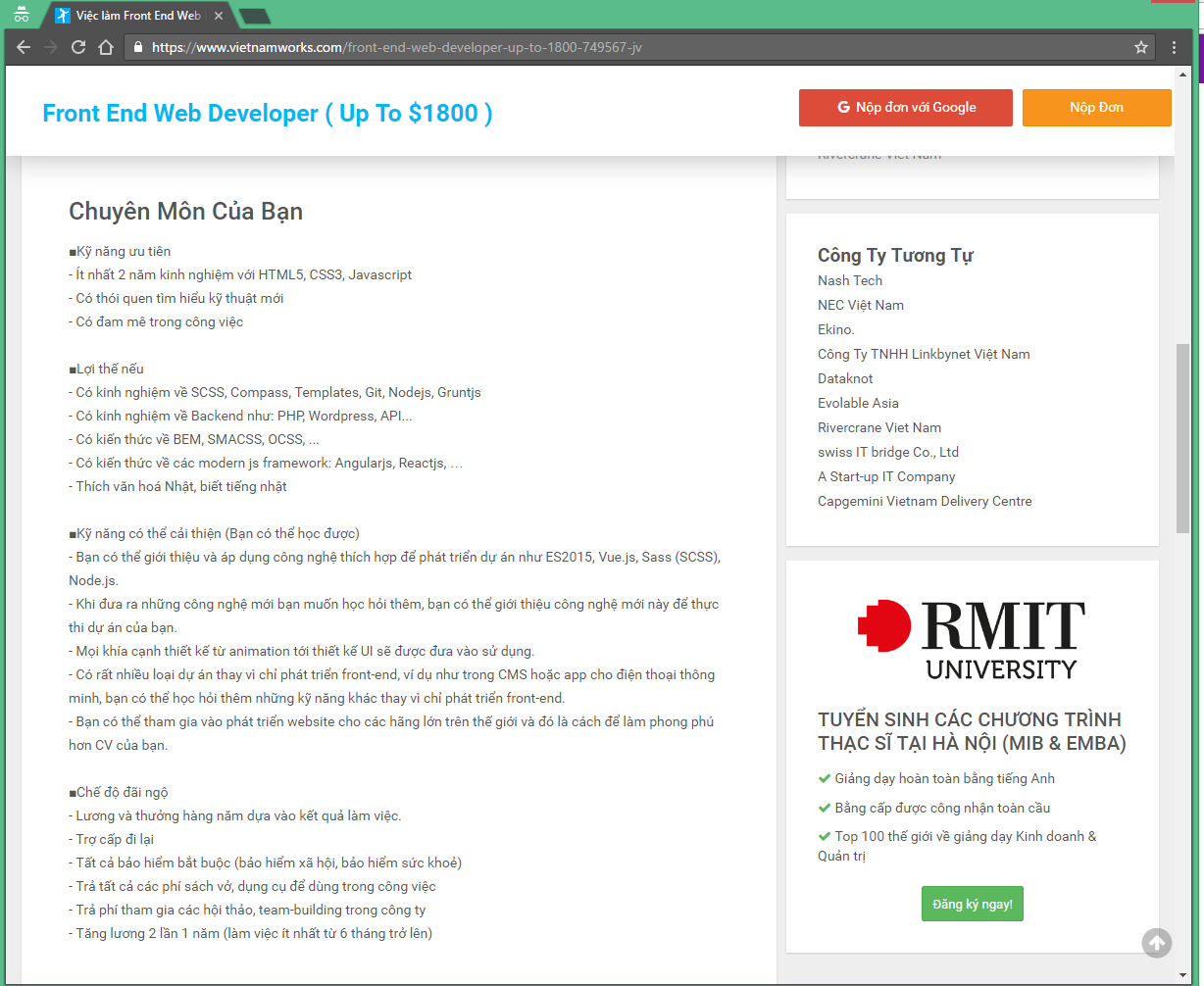
Những trang được tải về bao hàm cả các dấu HTML, văn bản ‘boilerplate’ – các thanh menu, quảng cáo. Đó là các phần tài liệu không hợp lệ, tạm gọi là phần đánh dấu. Trong luận văn, em sử dụng JSOUP (Java HTML Parser) là một thư viện được dùng để phân tích tài liệu HTML. Jsoup cung cấp các API dùng để lấy dữ liệu và thao tác dữ liệu từ URL hoặc từ file HTML. Nó sử dụng các phương thức giống với DOM, CSS , JQuery để lấy dữ liệu và thao tác với dữ liệu. Các trang Web điển hình thường có mục tài liệu mà phần bắt đầu và kết thúc là nhiều ‘boilerplate’ và dấu HTML, còn phần ở giữa là phần nội dung của văn bản trong đó phần đánh dấu tương đối ít, đó là ngôn ngữ và là mục tài liệu cần sử dụng. JSOUP sẽ giúp loại bỏ những phần đánh dấu và chỉ trích rút những văn bản được kết nối.

Phần lớn các trang web được mã hóa ở định dạng HTML (Hyper-Text Mark-up Language). Trong đó, mỗi tệp HTML của trang Web chứa rất nhiều thông tin phụ khác ngoài phần tin chính (main content), như: các pop-up quảng cáo, link liên kết đến các trang, bài khác, hoặc các nhà tài trợ, nhà phát triển, phần ghi quyền tác giả, các thông báo, cảnh báo… Việc làm sạch các văn bản đầu vào ở đây được hiểu chính là quá trình xác định phần nội dung chính của trang web và loại bỏ các thành phần phụ không liên quan. Với các trang web có cấu trúc trang thay đổi thường xuyên, việc bóc tách phần nội dung chính từ các trang web trở nên khó khăn hơn.

Luận văn đã sử dụng Maven cho thư viện này như sau:

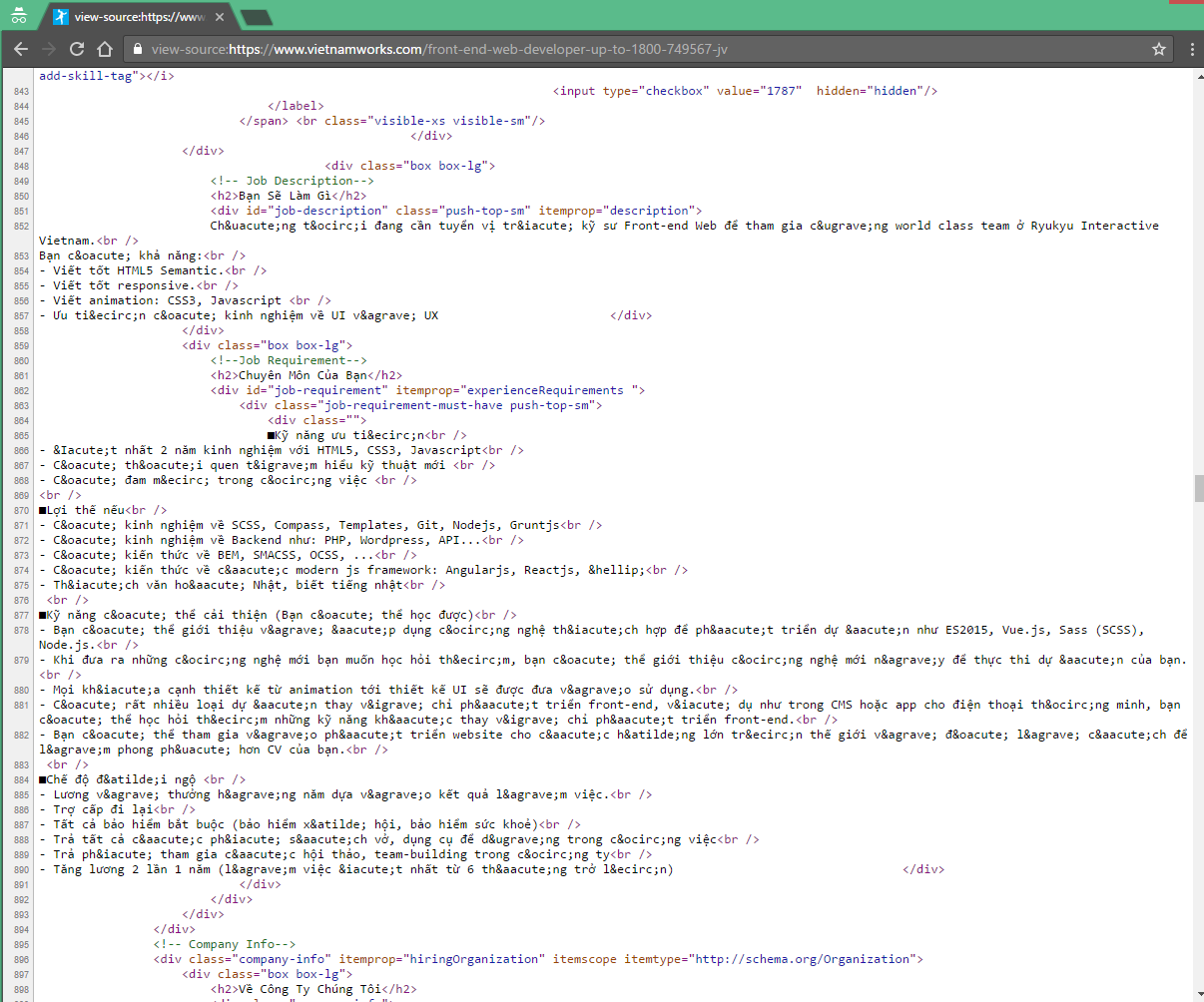


**Hình 2.1: Sử dụng Maven cho thư viên JSOUP**

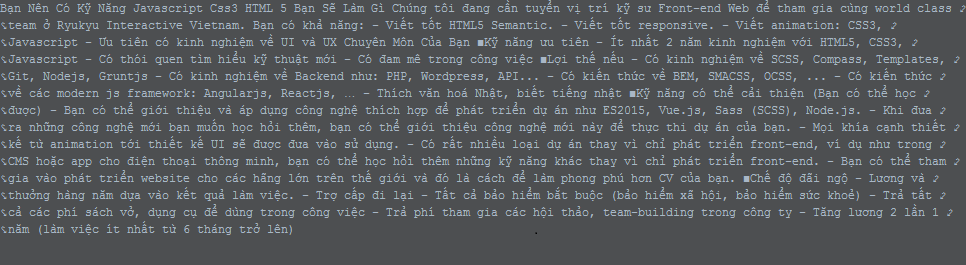


**Hình 2.2: Một website việc làm điển hình**

Hình 2.2 mô tả một bài tuyển dụng việc làm thông thường với phần thân chính của văn bản được đánh dấu. Bên trên cùng là phần header với banner và mục menu; bên dưới là footer, mục liên quan, thông báo bản quyền; bên phải là các liên kết của các nhà quảng cáo và phần không liên quan đến nội dung tuyển dụng.



**Hình 2.3: Nội dung đầu vào là mã html**



**Hình 2.4: Nội dung đầu ra là văn bản đã được trích xuất**

* + 1. ***Tiền xử lý văn bản***

1. Đặc điểm của từ trong tiếng Việt

Tiếng Việt là ngôn ngữ đơn lập. Đặc điểm này bao quát tiếng Việt cả về mặt ngữ âm, ngữ nghĩa, ngữ pháp. Khác với các ngôn ngữ châu Âu, mỗi từ là một nhóm các ký tự có nghĩa được cách nhau bởi một khoảng trắng. Còn tiếng Việt và các ngôn ngữ đơn lập khác thì khoảng trắng không phải là căn cứ để nhận diện từ.

Tiếng:

* Trong tiếng Việt trước hết cần chú ý đến đơn vị xưa nay vẫn quan gọi là tiếng. Về mặt ngữ nghĩa, ngữ âm, ngữ pháp, đều có giá trị quan trọng.
* Sử dụng tiếng để tạo từ có hai trường hợp:
* Trường hợp một tiếng: đây là trường hợp một tiếng được dùng làm một từ, gọi là từ đơn. Tuy nhiên không phải tiếng nào cũng tạo thành một từ.
* Trường hợp hai tiếng trở lên: đây là trường hợp hai hay nhiều tiếng kết hợp với nhau, cả khối kết hợp với nhau gắn bó tương đối chặt chẽ, mới có tư cách ngữ pháp là một từ. Đây là trường hợp từ ghép hay từ phức.

Từ:

Có rất nhiều quan niệm về từ trong tiếng Việt, từ nhiều quan niệm về từ tiếng Việt khác nhau đó chúng ta có thể thấy đặc trưng cơ bản của "từ" là sự hoàn chỉnh về mặt nội dung, từ là đơn vị nhỏ nhất để đặt câu. Người ta dùng "từ" kết hợp thành câu chứ không phải dùng "tiếng", do đó quá trình tách câu thành các "từ" cho kết quả tốt hơn là tách câu bằng “tiếng”.

1. Tách từ

*Bài toán tách từ*

Tách từ là một trong những bước xử lý thông tin đầu tiên trong quá trình xử lý ngôn ngữ tự nhiên, làm đầu vào cơ bản cho các chương trình ứng dụng xử lý ngôn ngữ, tách từ cũng là một nhiệm vụ khó khăn chính trong việc phân loại văn bản, gán nhãn từ loại đối với các ngôn ngữ châu Á như tiếng Hoa, tiếng Thái, tiếng Việt... Bài toán tách từ đặt ra nhằm phân tích một dãy các các từ đưa vào máy tính tách ra thành một tập các từ có nghĩa. Đối với các ngôn ngữ thuộc hệ Ấn – Âu việc tách từ khá đơn giản dựa vào vị trí của các ký tự đặc biệt để chia tách từ (Ví dụ như trong tiếng Anh là dấu cách). Tuy nhiên, với các ngôn ngữ châu Á như tiếng Hoa, tiếng Thái… thì để giải quyết bài toán tách từ lại gặp những vấn đề phức tạp hơn nhiều, sự phức tạp này chủ yếu là do các ngôn ngữ nói trên không có ký tự đặc biệt để phân biệt ranh giới giữa các từ như các ngôn ngữ thuộc hệ Ấn – Âu. Để giải quyết bài toán tách từ cho tiếng Việt cũng gặp phải các vấn đề tương tự nói trên, do đó việc ứng dụng các kết quả nghiên cứu của những nước phương tây cho bài toán tách từ tiếng Việt là hết sức hạn chế và không đem lại hiệu quả.

Trong các năm gần đây các công trình nghiên cứu về bài toán tách từ tiếng Việt thường tập trung vào 2 hướng tiếp cận chính để giải quyết bài toán cho tiếng Việt đó là:

* Hướng tiếp cận giải quyết bài toán tách từ tiếng Việt dựa trên từ (Word - based): Từ điển, thống kê,…
* Hướng tiếp cận giải quyết bài toán tách từ tiếng Việt dựa trên âm tiết (Character-based): Unigram, bigram, trigram, n-gram…

*Một số phương pháp tách từ*

**Phương pháp Maximum Matching: Forward / Backward**

Phương pháp so khớp tối đa (MM-Maximum Matching) hay còn gọi là LRMM - Left Right Maximum Matching. Ở phương pháp này, chúng ta sẽ duyệt một ngữ hoặc câu từ trái sang phải và chọn từ có nhiều âm tiết nhất có mặt trong từ điển và cứ thực hiện lặp lại như vậy cho đến hết câu.

Dạng đơn giản của phương pháp dùng để giải quyết nhập nhằng từ đơn. Giả sử chúng ta có một chuỗi ký tự C1, C2, …, Cn. Chúng ta sẽ áp dụng phương pháp từ đầu chuỗi. Đầu tiên kiểm tra xem C1có phải là từ hay không, sau đó kiểm tra xem C1C2 có phải là từ hay không. Tiếp tục thực hiện như thế cho đến khi tìm được từ dài nhất .

Dạng phức tạp: Quy tắc của dạng này là phân đoạn từ. Thông thường người ta chọn phân đoạn ba từ có chiều dài tối đa. Thuật toán bắt đầu từ dạng đơn giản, cụ thể là nếu phát hiện ra những cách tách từ gây nhập nhằng, giả sử C1 là từ và C1C2 cũng là một từ, khi đó chúng ta kiểm tra ký tự kế tiếp trong chuỗi C1, C2, ..., Cn để tìm tất cả các đoạn ba từ có bắt đầu với C1 hoặc C1C2.

Ví dụ: Giả sử chúng ta có được các đoạn sau:

- C1 C2 C3 C4

- C1C2 C3C4 C5

- C1C2 C3C4 C5C6

Khi đó chuỗi dài nhất sẽ là chuỗi thứ ba. Do đó từ đầu tiên của chuỗi thứ ba (C1C2) sẽ được chọn. Thực hiện các bước cho đến khi được chuỗi từ hoàn chỉnh.

Nhận xét :

Phương pháp này thực hiện tách từ đơn giản, nhanh và chỉ cần dựa vào từ điển để thực hiện. Tuy nhiên, khuyết điểm của phương pháp này cũng chính là từ điển, nghĩa là độ chính xác khi thực hiện tách từ phụ thuộc hoàn toàn vào tính đủ, tính chính xác của từ điển.

**Phương pháp Transformation – based Learning**

Phương pháp này tiếp cận dựa trên tập ngữ liệu đã đánh dấu. Theo cách tiếp cận này để cho máy tính có thể nhận biết ranh giới giữa các từ để có thể tách từ chính xác, chúng ta sẽ cho máy học các câu mẫu trong tập ngữ liệu đã được đánh dấu ranh giới giữa các từ đúng. Chúng ta thấy phương pháp rất đơn giản, vì chỉ cần cho máy học các tập câu mẫu và sau đó máy sẽ tự rút ra qui luật của ngôn ngữ và để từ đó sẽ áp dụng chính xác khi có những câu đúng theo luật mà máy đã rút ra. Và để tách từ được hoàn toàn chính xác trong mọi trường hợp thì đòi hỏi phải có một tập ngữ liệu tiếng Việt thật đầy đủ và phải được huấn luyện lâu để có thể rút ra các luật đầy đủ.

**Phương pháp tách tách từ tiếng Việt dựa trên thống kê từ Internet và thuật giải di truyền**

Phương pháp tách từ tiếng Việt dựa trên thống kê từ Internet và thuật giải di truyền – IGATEC (Internet and Genetics Algorithm based Text Categorization for Documents in Vietnamese) do H. Nguyễn đề xuất năm 2005 như một hướng tiếp cận mới trong tách từ với mục đích phân loại văn bản mà không cần dùng đến một từ điển hay tập ngữ liệu học nào. Trong hướng tiếp cận này, tác giả kết hợp giữa thuật toán di truyền với dữ liệu thống kê được lấy từ Internet .

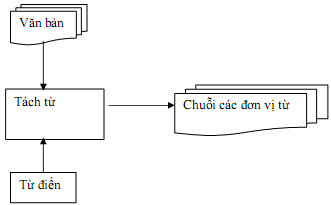
*Công cụ vnTokenizer*

Đặt bài toán: Đối với một văn bản tiếng Việt, dấu cách được đặt giữa các âm tiết chứ không phải giữa các từ. Một từ có thể có một, hai hoặc nhiều âm tiết nên có nhiều cách phân chia các âm tiết thành các từ, gây ra nhập nhằng. Việc phân giải nhập nhằng này gọi là bài toán tách từ.

Bài toán: Cho 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).

Giới thiệu công cụ vnTokenizer: vnTokenizer là công cụ tách từ tiếng Việt được nhóm tác giả Nguyễn Thị Minh Huyền, Vũ Xuân Lương và Lê Hồng Phương phát triển dựa trên phương pháp so khớp tối đa (Maximum Matching) với tập dữ liệu sử dụng là bảng âm tiết tiếng Việt và từ điển từ vựng tiếng Việt. Công cụ được xây dựng bằng ngôn ngữ Java, mã nguồn mở. Có thể dễ dàng sửa đổi nâng cấp và tích hợp vào các hệ thống phân tích văn bản tiếng Việt khác.

Quy trình thực hiện tách từ theo phương pháp so khớp tối đa:



**Hình** 2.5: **Quy trình tách từ**

Đầu vào của công cụ tách từ vnTokenizer là một câu hoặc một văn bản được lưu dưới dạng tệp.

Đầu ra là một chuỗi các đơn vị từ được tách.

Các đơn vị từ bao gồm các từ trong từ điển cũng như các chuỗi số, chuỗi kí từ nước ngoài, các hình vị ràng buộc (gồm các phụ tố), các dấu câu và các chuỗi kí tự hỗn tạp khác trong văn bản (ISO, 2008). Các đơn vị từ không chỉ bao gồm các từ có trong từ điển, mà cả các từ mới hoặc các từ được sinh tự do theo một quy tắc nào đó (như phương thức thêm phụ tố hay phương thức láy) hoặc các chuỗi kí hiệu không được liệt kê trong từ điển.

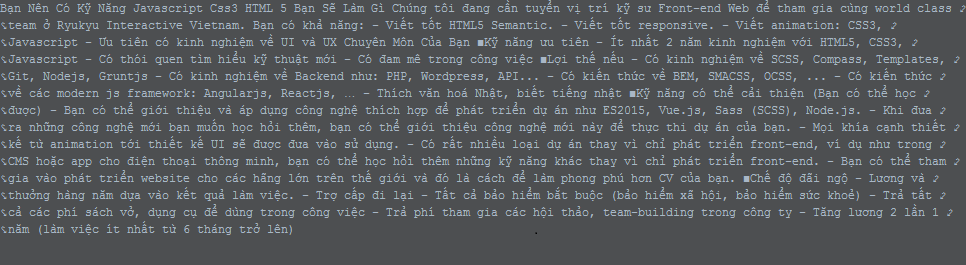
Công cụ sử dụng tập dữ liệu đi kèm là tập từ điển từ vựng tiếng Việt, danh sách các đơn vị từ mới bổ sung, được biểu diễn bằng ôtômat tối tiểu hữu hạn trạng thái, tệp chứa các biểu thức chính quy cho phép lọc các đơn vị từ đặc biệt (xâu dạng số, ngày tháng,…) và các tệp chứa các thống kê unigram và bigram trên kho văn bản tách từ mẫu.

Với các đơn vị từ đã có trong từ điển, khi thực hiện tách từ cũng được xử lý hiện tượng nhập nhằng bằng cách kết hợp với các thống kê unigram và bigram. Chẳng hạn trong tiếng Việt thường gặp các trường hợp nhập nhằng như:

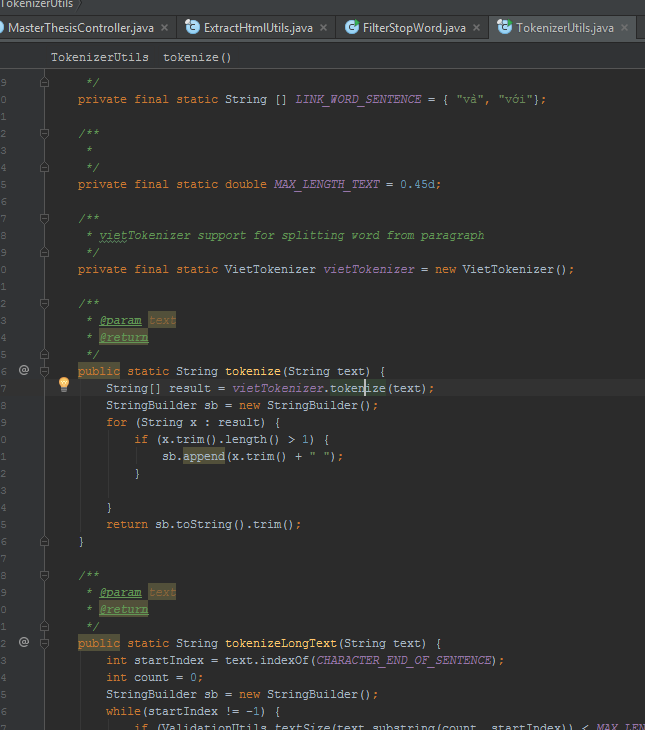
Xâu AB vừa có thể hiểu là 1 đơn vị từ, vừa có thể là chuỗi 2 đơn vị từ A-B.

Xâu ABC có thể tách thành 2 đơn vị AB-C hoặc A-BC.

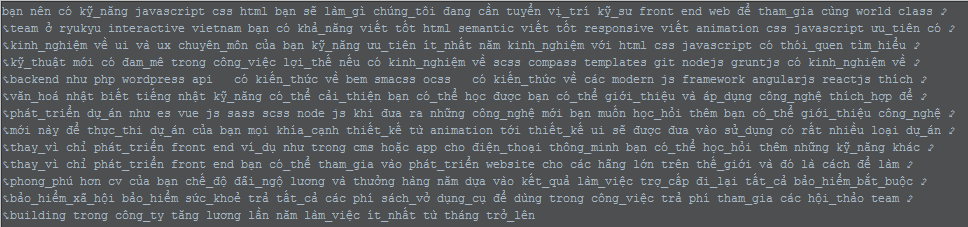
Minh họa sử dụng công cụ vnTokenizer:



**Hình** 2.6**: Đầu vào của thuật toán tách từ**



**Hình** 2.7**: Quá trình xử lý của công cụ vnTokenizer**



**Hình** 2.8**: Đầu ra là văn bản đã được tách từ**

Đánh giá kết quả:

Kết quả đánh giá của công cụ được cho là ổn định đối với nhiều loại văn bản/ văn phong khác nhau. Độ chính xác trung bình đạt được là khoảng 94%.

Lưu ý: sau bước tách từ, văn bản sẽ được đưa về dạng chữ thường để loại bỏ stopword.

1. Loại bỏ stop-word

Từ dừng (stop-word) dùng để chỉ các từ xuất hiện quá nhiều trong các văn bản của toàn tập kết quả, thường thì không giúp ích gì trong việc phân biệt nội dung của các tài liệu văn bản. Ví dụ, những từ “và”, “thì”, “là”, “mà” thường xuất hiện hầu hết trong các văn bản thì được gọi là stop-word. Vì đặc điểm của từ dừng nên chúng được loại bỏ mà không ảnh hưởng đến các việc biểu diễn văn bản tiếp theo.

Danh sách các stop-word tiếng Việt sẽ được đính kèm trong phần phụ lục của quyển luận văn.

Dưới đây là một số stop-word tiếng Việt cụ thể:

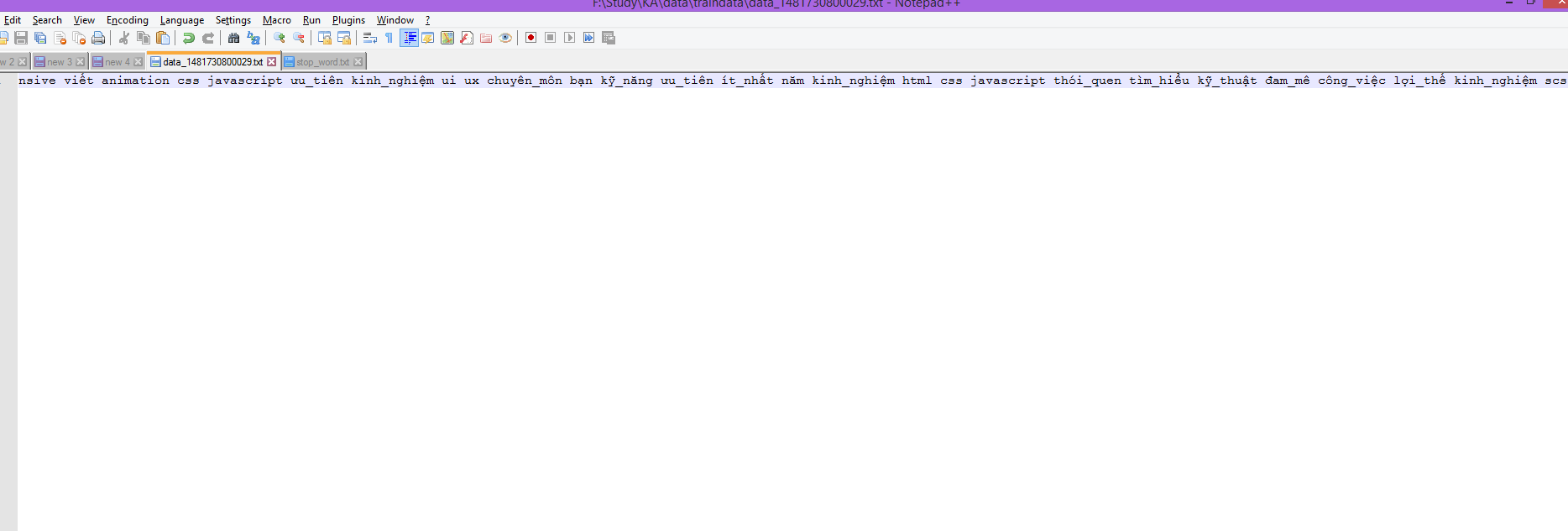
Bảng2**.**1**: Danh sách stopword tiếng Việt**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a\_ha  a-lô  à\_ơi  á  à | á\_à  ạ  ạ\_ơi  ai  ai\_nấy | chính\_là  chính\_thị  cho\_biết  cho\_hay  cho\_phép | cho\_đến  cho\_đến\_khi  cho\_nên  cho\_tới  cho\_tới\_khi | thế\_à  thế\_là  thế\_mà  thế\_nào  thế\_nên | thế\_ra  thế\_thì  thếch  thì  thi\_thoảng |

1. Làm sạch văn bản

Tiếp theo, ta cần phải “làm sạch” văn bản để tạo ra một tập dữ liệu thiết thực hơn. Ví dụ:

* Loại bỏ dấu câu: “.”, “,”,“?”,“!”,“-”,“(”,“)”,“:”,“[”,“]”
* Loại bỏ các kí tự đặc biệt: [ ],[.], [,], [:], [“], [”], [;], [/], [[]], [~], [`], [!], [@], [#], [$],[%],[^],[&],[\*],[(],[)]
* Các chữ số, phép tính toán số học: các ký tự số cũng thường được loại ra vì không mang lại nhiều ý nghĩa cho văn bản (trừ một số trường hợp đặc biệt, ví dụ thông tin trong lĩnh vực lịch sử, tài chính – ngân hàng).
* Ngoài ra, cần phải loại bỏ các dòng trống, các khoảng trắng có nhiều hơn một khoảng trắng, các ký tự đặc biệt (@, #, $, %, ^, /, \,…)



Hình 2.9: Văn bản sau khi loại stopwords và làm sạch

Sau khi loại bỏ các stopword và làm sạch, văn bản có thể xem như là một tập hợp các đặc trưng, đó là tập hợp các từ “quan trọng” còn lại để biểu diễn văn bản. Việc phân loại văn bản sẽ dựa trên các đặc trưng này. Tuy nhiên, có thể thấy rằng, số đặc trưng của một văn bản là lớn và không gian các đặc trưng (tất cả đặc trưng) của tất cả các văn bản đang xem xét là rất lớn, về nguyên tắc, nó bao gồm tất cả các từ trong một ngôn ngữ. Chính vì vậy, phân loại dựa trên các đặc trưng này cần phải có cách xử lí, lựa chọn đặc trưng nhằm rút ngắn số chiều của không gian đặc trưng. Luận văn đã sử dụng trọng số của một từ được tính theo tần suất xuất hiện của từ trong văn bản (TF) và tần suất nghịch đảo của từ (IDF). Cụ thể phần này đã được trình bày ở chương 1, mục 1.5.

* 1. **Trình bày chi tiết thuật toán Máy vector hỗ trợ SVM**
     1. ***Giới thiệu thuật toán***

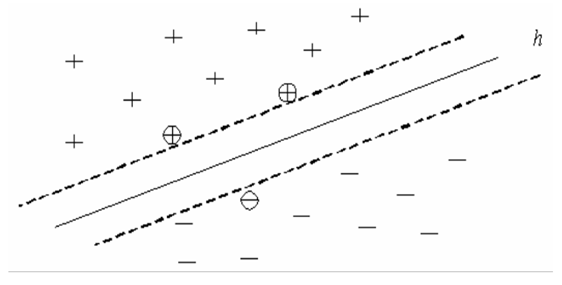
Máy vectơ hỗ trợ (SVM - viết tắt tên tiếng Anh support vector machine) là một khái niệm trong [thống kê](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_Th%E1%BB%91ng_k%C3%AA) và [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh) cho một tập hợp các phương pháp [học có giám sát](https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_c%C3%B3_gi%C3%A1m_s%C3%A1t) liên quan đến nhau để [phân loại](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_lo%E1%BA%A1i_%28h%E1%BB%8Dc_m%C3%A1y%29&action=edit&redlink=1) và [phân tích hồi quy](https://vi.wikipedia.org/wiki/Ph%C3%A2n_t%C3%ADch_h%E1%BB%93i_quy).

SVM được định nghĩa như sau:

* Là phương pháp dựa trên nền tảng của lý thuyết thống kê nên có một nền tảng toán học chặt chẽ để đảm bảo rằng kết quả tìm được là chính xác.
* Là thuật toán học giám sát (supervied learning) được sử dụng cho phân lớp dữ liệu.
* Là 1 phương pháp thử nghiệm, đưa ra 1 trong những phương pháp mạnh và chính xác nhất trong số các thuật toán nổi tiếng về phân lớp dữ liệu.
  + 1. ***Ý tưởng thuật toán***

Cho trước một tập huấn luyện, được biểu diễn trong không gian vector, trong đó mỗi tài liệu là một điểm, phương pháp này tìm ra một siêu phẳng h quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng là lớp + và lớp -. Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, đồng thời việc phân loại càng chính xác.

Mục đích của phương pháp SVM là tìm được khoảng cách biên lớn nhất để tạo kết quả phân lớp tốt, điều này được minh họa như sau:



**Hình 2.10: Siêu phẳng h phân chia dữ liệu huấn luyện thành 2 lớp + và – với khoảng cách biên lớn nhất. Các điểm gần h nhất là các vector hỗ trợ (Support Vector được khoanh tròn)**

* + 1. ***Thuật toán***

Ban đầu, thuật toán SVM được thiết kế cho bài toán phân lớp nhị phân. Xét một ví dụ của bài toán phân lớp như hình vẽ; ở đó ta phải tìm một đường thẳng sao cho bên trái nó toàn là các điểm đỏ, bên phải nó toàn là các điểm xanh. Bài toán mà dùng đường thẳng để phân chia này được gọi là phân lớp tuyến tính (linear classification).



Hình 2.11: Phân lớp nhị phân

Hàm tuyến tính phân biệt hai lớp như sau:

(1)

Trong đó:

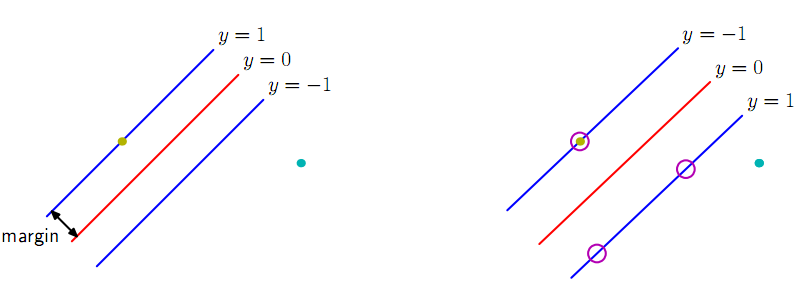
* là vector trọng số hay vector chuẩn của siêu phẳng phân cách, T là kí hiệu chuyển vị.
* là độ lệch
* là vector đặc trưng, làm hàm ánh xạ từ không gian đầu vào sang không gian đặc trưng.

Tập dữ liệu đầu vào gồm N mẫu input vector {x1, x2,...,xN}, với các giá trị nhãn tương ứng là {t1,…,tN} trong đó .

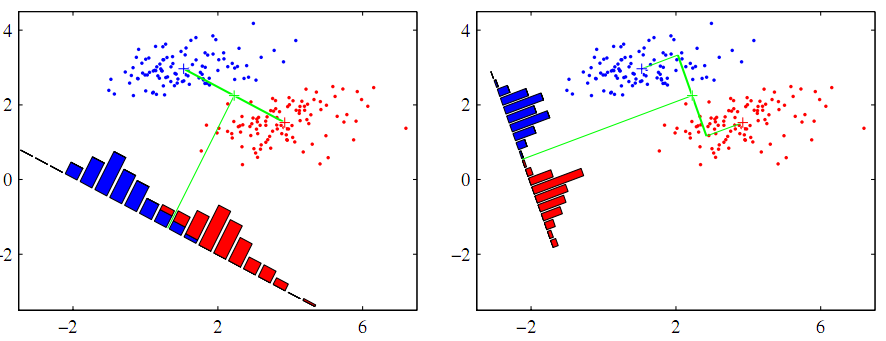
Lưu ý cách dùng từ ở đây: điểm dữ liệu, mẫu… đều được hiểu là input vector xi; nếu là không gian 2 chiều thì đường phân cách là đường thẳng, nhưng trong không gian đa chiều thì gọi đó là siêu phẳng.

Giả sử tập dữ liệu của ta có thể phân tách tuyến tính hoàn toàn (các mẫu đều được phân đúng lớp) trong không gian đặc trưng (feature space), do đó sẽ tồn tại giá trị tham số w và b theo (1) thỏa cho những điểm có nhãn và cho những điểm có , vì thế mà cho mọi điểm dữ liệu huấn luyện.

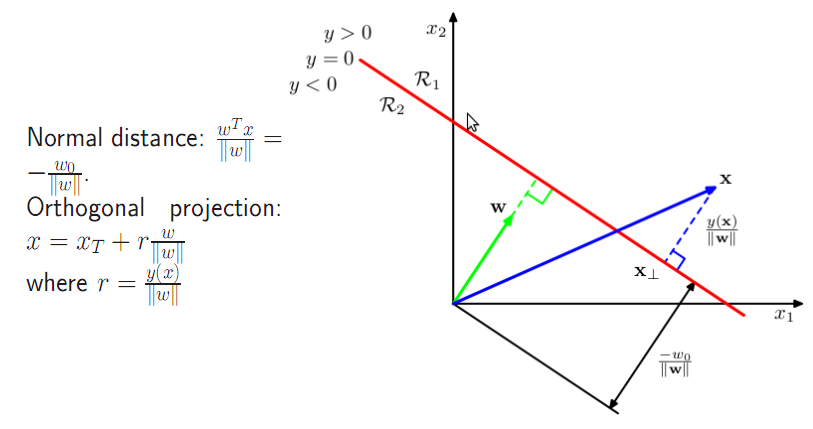
SVM tiếp cận giải quyết vấn đề này thông qua khái niệm gọi là lề, đường biên… (margin). Lề được chọn là khoảng cách nhỏ nhất từ đường phân cách đến mọi điểm dữ liệu hay là khoảng cách từ đường phân cách đến những điểm gần nhất.



Trong SVM, đường phân lớp tốt nhất chính là đường có khoảng cách margin lớn nhất (tức là sẽ tồn tại rất nhiều đường phân cách xoay theo các phương khác nhau, và ta chọn ra đường phân cách mà có khoảng cách margin là lớn nhất).



Ta có công thức tính khoảng cách từ điểm dữ liệu đến mặt phân cách như sau:



Do ta đang xét trong trường hợp các điểm dữ liệu đều được phân lớp đúng nên cho mọi n. Vì thế khoảng cách từ điểm xn đến mặt phân cách được viết lại như sau:

(2)

Lề là khoảng cách vuông góc đến điểm dữ liệu gần nhất xn từ tập dữ liệu, và chúng ta muốn tìm giá trị tối ưu của w và b bằng cách cực đại khoảng cách này. Vấn đề cần giải quyết sẽ được viết lại dưới dạng công thức sau:

(3)

Chúng ta có thể đem nhân tử ra ngoài bởi vì w không phụ thuộc n. Giải quyết vấn đề này một cách trực tiếp sẽ rất phức tạp, do đó ta sẽ chuyển nó về một vấn đề tương đương dễ giải quyết hơn. Ta sẽ scale và cho mọi điểm dữ liệu, từ đây khoảng cách lề trở thành 1, việc biến đổi này không làm thay đổi bản chất vấn đề.

(4)

Từ bây giờ, các điểm dữ liệu sẽ thỏa ràng buộc:

(5)

Vấn đề tối ưu yêu cầu ta cực đại được chuyển thành cực tiểu , ta viết lại công thức:

(6)

Việc nhân hệ số ½ sẽ giúp thuận lợi cho lấy đạo hàm về sau.

Lý thuyết Nhân tử Lagrange:

Vấn đề cực đại hàm f(x) thỏa điều kiện sẽ được viết lại dưới dạng tối ưu của hàm Lagrange như sau:

Trong đó x và λ phải thỏa điều kiện Karush-Kuhn-Tucker (KKT) như sau:

Nếu là cực tiểu hàm f(x) thì hàm Lagrange sẽ là:

Để giải quyết bài toán trên, ta viết lại theo hàm Lagrange như sau:

(7)

Trong đó là nhân tử Lagrange.

Lưu ý dấu (–) trong hàm Lagrange, bởi vì ta cực tiểu theo biến w và b, và là cực đại theo biến a.

Lấy đạo hàm L(w,b,a) theo w và b ta có:

(8)

(9)

Loại bỏ w và b ra khỏi L(w,b,a) bằng cách thế (8), (9) vào. Điều này sẽ dẫn ta đến vấn đề tối ưu:

(10)

Thỏa các ràng buộc:

(11)

(12)

Ở đây hàm nhân (kernel function) được định nghĩa là .

Vấn đề tạm thời gác lại ở đây, ta sẽ thảo luận kỹ thuật giải quyết (10) thỏa (11), (12) này sau.

Để phân lớp cho 1 điểm dữ liệu mới dùng mô hình đã huấn luyện, ta tính dấu của y(x) theo công thức (1), nhưng thế w trong (8) vào:

(13)

Thỏa các điều kiện KKT sau:

(14)

(15)

(16)

Vì thế với mọi điểm dữ liệu, hoặc là hoặc là . Những điểm dữ liệu mà có sẽ không xuất hiện trong (13) và do đó mà không đóng góp trong việc dự đoán điểm dữ liệu mới.

Những điểm dữ liệu còn lại được gọi là support vector, chúng thỏa , đó là những điểm nằm trên lề của siêu phẳng trong không gian đặc trưng.

Support vector chính là cái mà ta quan tâm trong quá trình huấn luyện của SVM. Việc phân lớp cho một điểm dữ liệu mới sẽ chỉ phụ thuộc vào các support vector.

Giả sử rằng ta đã giải quyết được vấn đề (10) và tìm được giá trị nhân tử a, bây giờ ta cần xác định tham số b dựa vào các support vector xn có . Thế (13) vào:

(17)

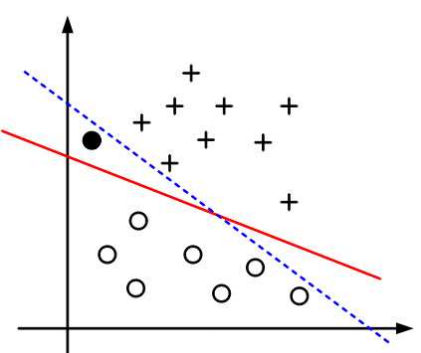
Trong đó S là tập các support vector. Mặc dù ta chỉ cần thế một điểm support vector xn vào là có thể tìm ra b, nhưng để đảm bảo tính ổn định của b ta sẽ tính b theo cách lấy giá trị trung bình dựa trên các support vector.

Đầu tiên ta nhân tn vào (17) (lưu ý , và giá trị b sẽ là:

(18)

Trong đó Ns là tổng số support vector.

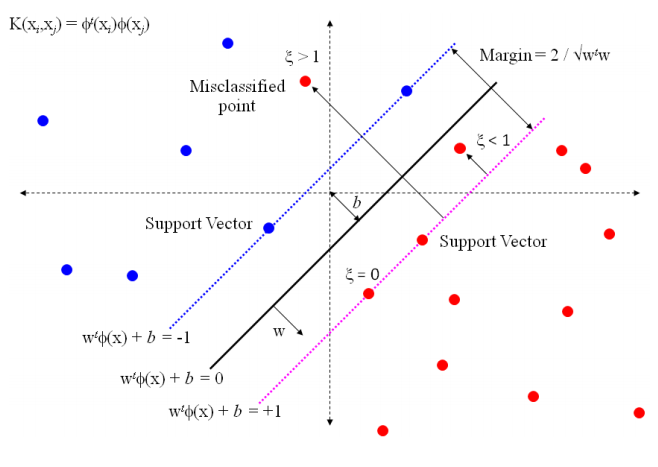
Ban đầu để dễ trình bày thuật toán ta đã giả sử là các điểm dữ liệu có thể phân tách hoàn toàn trong không gian đặc trưng . Nhưng việc phân tách hoàn toàn này có thể dẫn đến khả năng tổng quát hóa kém, vì thực tế một số mẫu trong quá trình thu thập dữ liệu có thể bị gán nhãn sai, nếu ta cố tình phân tách hoàn toàn sẽ làm cho mô hình dự đoán quá khớp.



Để chống lại sự quá khớp, chúng ta chấp nhận cho một vài điểm bị phân lớp sai.

Để làm điều này, ta dùng các biến slack variables cho mọi điểm dữ liệu.

* cho những điểm nằm trên lề hoặc phía trong của lề
* cho những điểm còn lại.
* Do đó những điểm nằm trên đường phân cách sẽ có
* Còn những điểm phân lớp sai sẽ có



Công thức (5) sẽ viết lại như sau:

(20)

Mục tiêu của ta bây giờ là cực đại khoảng cách lề, nhưng đồng thời cũng đảm bảo tính mềm mỏng cho những điểm bị phân lớp sai. Ta viết lại vấn đề cần cực tiểu:

(21)

Trong đó C > 0 đóng vai trò quyết định đặt tầm quan trọng vào biến hay là lề.

Bây giờ chúng ta cần cực tiểu (21) thỏa ràng buộc (20) và . Theo Lagrange ta viết lại:

(22)

Trong đó và là các nhân tử Lagrange.

Các điều kiện KKT cần thỏa là:

(23)

(24)

(25)

(26)

(27)

(28)

Với n = 1,…,N

Lấy đạo hàm (22) theo w, b và {}:

(29)

(30)

(31)

Thế (29), (30), (31) vào (22) ta được:

(32)

Từ (23), (26) và (31) ta có:

Vấn đề cần tối ưu giống hệt với trường hợp phân tách hoàn toàn, chỉ có điều kiện ràng buộc khác biệt như sau:

(33)

(34)

Thế (29) vào (1), ta sẽ thấy để dự đoán cho một điểm dữ liệu mới tương tự như (13).

Như trước đó, tập các điểm có không có đóng góp gì cho việc dự đoán điểm dữ liệu mới.

Những điểm còn lại tạo thành các support vector. Những điểm có và theo (25) thỏa:

(35)

Nếu theo (31) có , từ (28) suy ra và đó là những điểm nằm trên lề.

Những điểm có có thể là những điểm phân lớp đúng nằm giữa lề và đường phân cách nếu hoặc có thể là phân lớp sai nếu

Để xác định tham số b trong (1) ta sẽ dùng những support vector mà có vì thế :

(36)

Lần nữa, để đảm bảo tính ổn định của b ta tính theo trung bình:

(37)

Trong đó M là tập các điểm có

Để giải quyết (10) và (32) ta dùng thuật toán Sequential Minimal Optimization (SMO) do Platt đưa ra vào 1999. Thuật toán này sử dụng tập dữ liệu huấn luyện (còn gọi là tập làm việc) có kích thước nhỏ nhất bao gồm hai hệ số Lagrange. Bài toán quy hoạch toàn phương nhỏ nhất phải gồm hai hệ số Lagrange vì các hệ số Lagrange phải thỏa mãn ràng buộc đẳng thức (34). Phương pháp SMO cũng có một số heuristic cho việc chọn hai hệ số Lagrange để tối ưu hóa ở mỗi bước. Mặc dù có nhiều bài toán quy hoạch toàn phương con hơn so với các phương pháp khác, mỗi bài toán con này được giải rất nhanh dẫn đến bài toán quy hoạch toàn phương tổng thể cũng được giải một cách nhanh chóng.

## Phân loại đa lớp nội dung văn bản với SVM

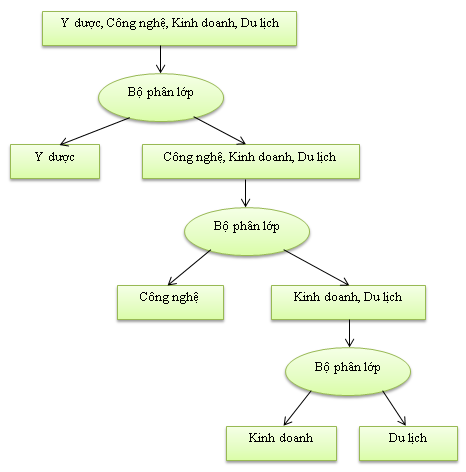
Bây giờ xét đến trường hợp phân nhiều lớp K > 2. Chúng ta có thể xây dựng việc phân K-class dựa trên việc kết hợp một số đường phân 2 lớp.

Sau đây ta tìm hiểu các chiến lược áp dụng trong bài toán phân lớp văn bản thuộc nhiều chủ đề 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), và One-against-Rest (OAR).

*Chiến lược One-against-Rest*

Trong chiến lược OAR, ta sẽ dùng K-1 bộ phân lớp nhị phân để xây dựng K-class. Bài toán phân lớp K lớp được chuyển thành K-1 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:

Siêu phẳng yi(x) = 0 hình thành siêu phẳng phân chia tối ưu, các support vector thuộc lớp i thỏa yi(x) = 1 và các support vector thuộc lớp còn lại thỏa yi(x) = -1. Nếu vector dữ liệu x thỏa mãn điều kiện yi(x)>0 đối với duy nhất một I, x sẽ được phân vào lớp thứ i.



**Hình** 2.12**: Ví dụ phân lớp văn bản thuộc các chủ đề chiến lược OAR**

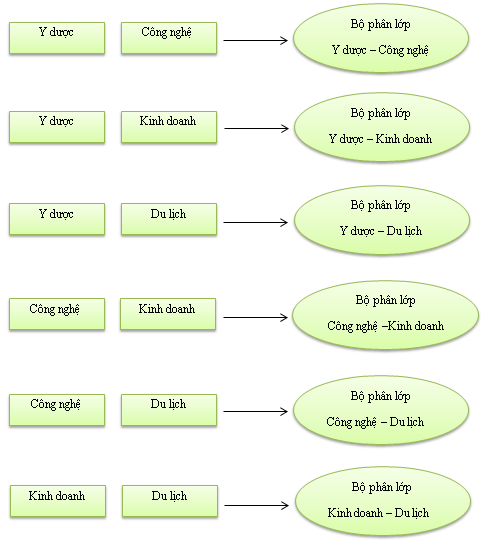
*Chiến lược One-against-One*

Chiến lược OAO, dùng K(K-1)/2 bộ phân lớp nhị phân được xây dựng bằng cách bắt cặp hai lớp một nên chiến lược này còn được gọi là pairwise và sử dụng theo phương pháp đa số kết lợp các bộ phận lớp này để xác định được kết quả phân lớp cuối cùng. Số lượng các bộ phân lớp không bao giờ vượt quá K(K-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 tách một lớp ra khỏi một lớp khác mà thôi.

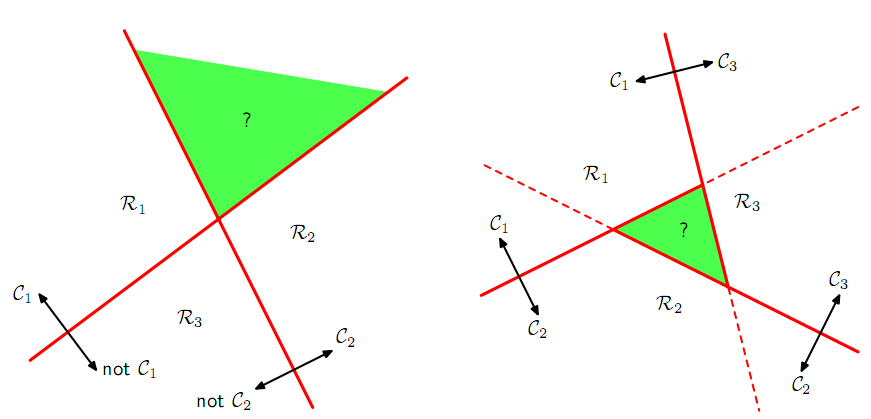
Chiến lược OAR chỉ cần K-1 bộ phân lớp cho K lớp. Trong khi đó chiến lược OAO lại cần đến K(K-1)/2 bộ phân lớp. Nhưng số mẫu huấn luyện cho từng bộ phân lớp trong OAO lại ít hơn và việc phân lớp cũng đơn giản hơn. Vì vậy chiến lược OAO có độ chính xác cao hơn nhưng chi phí để xây dựng lại tương đương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

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



**Hình** 2.13**: Ví dụ phân lớp văn bản thuộc các chủ đề chiến lược OAO**

Tuy nhiên cả hai chiến lược trên đều dẫn đến vùng mập mờ trong phân lớp (như hình 2.14).



**Hình 2.14: Vùng mập mờ trong phân lớp**

Ta có thể tránh được vấn đề này bằng cách xây dựng K-Class dựa trên K hàm tuyến tính có dạng:

Và một điểm x được gán vào lớp Ck khi với mọi .

**Kết luận chương:**

Chương này đã giới thiệu chi tiết thuật toán Máy vector hỗ trợ SVM và trình bày cách thu thập, trích rút và xử lý một cách tự động các tin tức tuyển dụng. Sau đó, luận văn sẽ sử dụng SVM để thực hiện việc huấn luyện bộ phân loại cho bài toán phân loại thông tin tuyển dụng trên hệ thống website tuyển dụng.

# Chương 3 - THỬ NGHIỆM VÀ ĐÁNH GIÁ

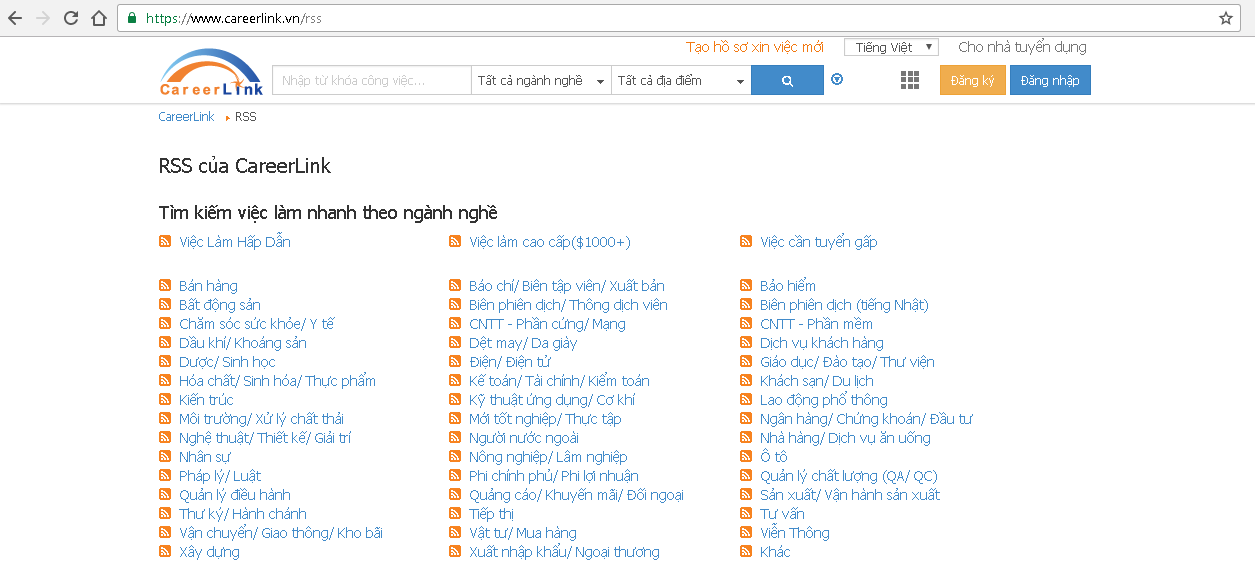


## Xây dựng bộ dữ liệu và tiền xử lý văn bản

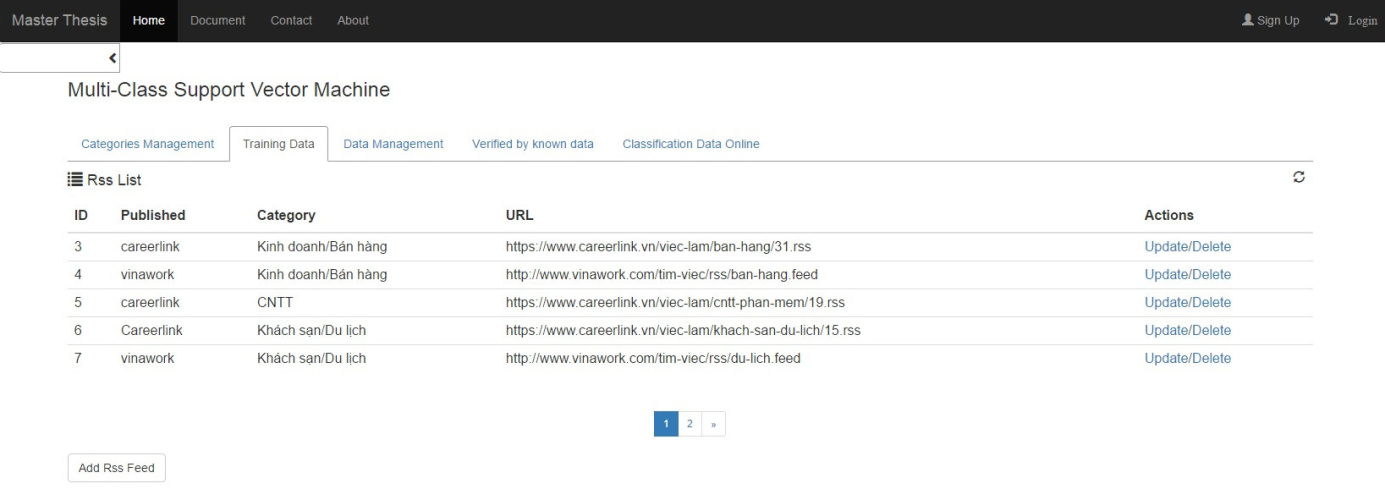
RSS (viết tắt từ Really Simple Syndication) là một tiêu chuẩn định dạng tài liệu dựa trên XML nhằm giúp người sử dụng dễ dàng cập nhật và tra cứu thông tin một cách nhanh chóng và thuận tiện nhất bằng cách tóm lược thông tin vào trong một đoạn dữ liệu ngắn gọn, hợp chuẩn.

Công nghệ của RSS cho phép người dùng Internet có thể đặt mua thông tin từ các websites có cung cấp khả năng RSS (RSS feeds); chúng thường là các site có nội dung thay đổi và được thêm vào thường xuyên. Để có thể dùng công nghệ này, các người quản trị site đó tạo ra hay quản lí một phần mềm chuyên dụng (như là một hệ thống quản lí nội dung - content management system - CMS) mà với định dạng XML máy có thể đọc được, có thể biểu diễn các bài tin mới thành một danh sách, với một hoặc hai dòng cho mỗi bài tin và một liên kết đến bài tin đầy đủ đó. Khác với việc mua nhiều ấn bản của các tờ báo hay tạp chí in giấy, hầu hết việc mua RSS là miễn phí.

Trong phạm vi luận văn, em đã sử dụng lấy tin tự động từ RSS để cập nhật các việc làm hằng ngày nhằm xây dựng bộ dữ liệu huấn luyện. Dữ liệu được lấy từ 2 trang web: <https://www.careerlink.vn/> và <http://www.vinawork.com/>



Hình 3.1: Thông tin RSS trên website <https://www.careerlink.vn/>



Hình 3.2: Danh sách các RSS được thu thập phục vụ giai đoạn huấn luyện



Hình 3.3: Các category được sử dụng

Các bài báo việc làm được thu thập được thông qua RSS từ tháng 8/2016. Các bài báo này được lấy về sau đó tiến hành tiền xử lý, cuối cùng được lưu dưới dạng file phục vụ cho huấn luyện và kiểm tra. Dữ liệu thu thập được gồm có 1409 bài báo, trong đó chọn 800 bài dùng cho huấn luyện và 609 bài dùng cho việc kiểm tra. Cụ thể số liệu như sau:

Bảng 3.1: Số lượng tập dữ liệu huấn luyện và kiểm tra

|  |  |  |
| --- | --- | --- |
| Chủ đề | Tập dữ liệu huấn luyện | Tập dữ liệu kiểm tra |
| Kinh doanh/Bán hàng | **200** | **233** |
| Công nghệ thông tin (CNTT) | **200** | **179** |
| Nhân sự | **200** | **117** |
| Khách sạn/Du lịch | **200** | **80** |

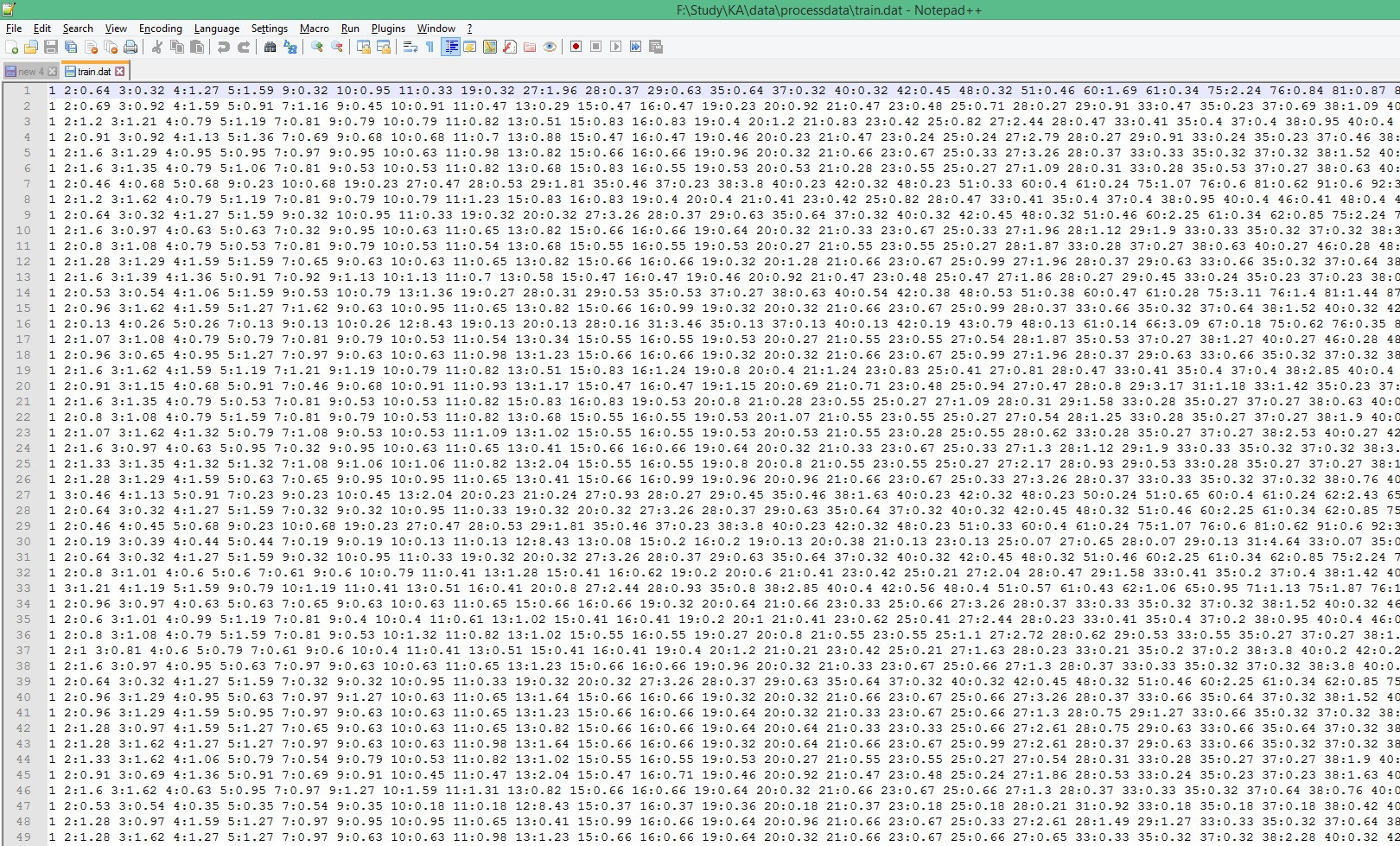
Sau các bước tiền xử lý, văn bản được biểu diễn dưới dạng:

<classi> <label1>:<value1> <label2>:<value2> ... <labeln>:<valuen>

Trong đó:

* classi là nhãn phân loại của mỗi chủ đề.
* labeli là chỉ số của từ đặc trưng thứ i trong không gian đặc trưng có xuất hiện ở phản hồi đó.
* valuei là trọng số của indexi được tính bởi công thức TF.IDF, nếu valuei = 0 thì không cần phải ghi đặc trưng đó.

Định dạng này tuân theo định dạng dữ liệu đầu vào của chương trình SVMMulticlass.



Hình 3.4: Biểu diễn văn bản dưới dạng vector

*Đánh giá độ chính xác của thuật toán:*

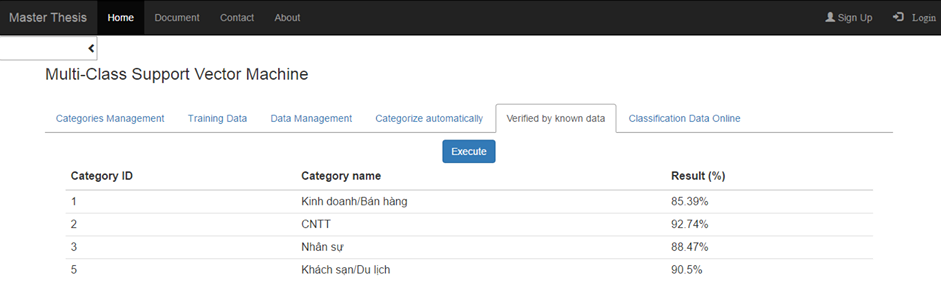
Chúng ta không thể khẳng định một phương pháp phân lớp văn bản cụ thể nào là chính xác hoàn toàn. Vì vậy việc đưa ra độ đo để đánh giá hiệu quả của thuật toán phân lớp giúp chúng ta có thể xác định được độ chính xác của thuật toán, từ đó áp dụng thuật toán đó vào việc phân lớp.

Độ chính xác có thể được tính theo các công thức:

Công thức đánh giá số 1:

Công thức đánh giá số 2:

Sau đây là kết quả thực nghiệm trên bộ test gồm 609 văn bản:



Hình 3.5: Kết quả độ chính xác của việc phân loại

* 1. **Giai đoạn phân lớp**
     1. ***URL phục vụ cho ứng dụng***

Mục đích của ứng dụng tìm kiếm thông tin việc làm đó là lấy những thông tin việc làm từ các website đưa vào hệ thống, sau đó sẽ tiến hành phân loại các việc làm đó theo chủ đề đã có. Danh sách các URL em sử dụng để phục vụ cho ứng dụng như sau:

[http://worklink.vn](http://worklink.vn/?gclid=CjwKEAiAjvrBBRDxm_nRusW3q1QSJAAzRI1tTwJW1a4LIhZQT60REsSbsTqh9dG44LdT6XBYtO2n7hoC5T_w_wcB)

<http://careerbuilder.vn/tim-viec-lam.html>

<https://www.timviecnhanh.com/>

<https://www.vietnamworks.com/tim-viec-lam>

<https://mywork.com.vn/>

* + 1. ***Triển khai ứng dụng***

Mục đích của ứng dụng là xây dựng một danh sách các việc làm được lấy từ các website được liệt kê trong phần 3.2.1 và tiến hành phân loại chúng vào 4 category có sẵn là Kinh doanh/Bán hàng, Công nghệ thông tin, Nhân sự và Khách sạn/Du lịch

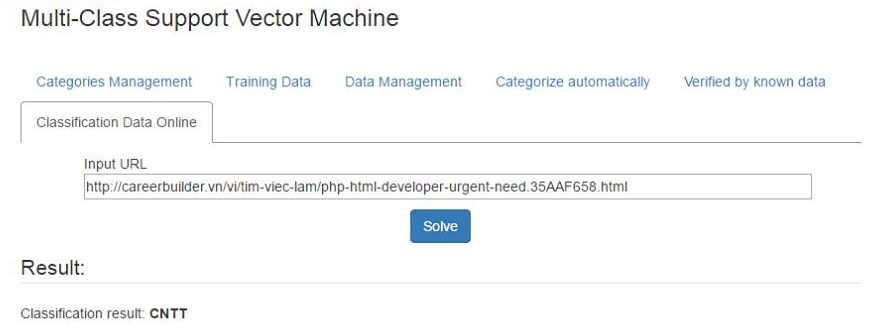
**Bước 1**: Input URL từ các danh sách các website việc làm không support RSS

**Bước 2**: Trích xuất văn bản từ các URL

**Bước 3**: Tiền xử lý văn bản

* Tách từ
* Loại bỏ từ dừng
* Làm sạch văn bản
* Vector hóa văn bản theo định dạng đầu vào của thuật toán SVM.

**Bước 4**: Thực hiện phân loại theo bốn chủ đề



Hình 3.6: Phân loại URL được input vào ứng dụng

**Kết luận chương**

Chương này đã trình bày cụ thể về việc xây dựng bộ dữ liệu huấn luyện và thử nghiệm, từ đó tính toán được độ chính xác của thuật toán; và đưa ra ứng dụng về việc phân loại URLs của các website việc làm vào nhãn tương ứng.

**KẾT LUẬN**

1. **Kết quả đạt được**

Luận văn tiến hành nghiên cứu giải quyết bài toán phân loại văn bản tiếng Việt dựa vào đặc trưng. Bài toán là nền tảng cho nhiều ứng dụng quan trọng thực tế như lọc thư rác, hệ thống khuyến cáo/quan điểm người dùng,…

Những kết quả chính mà đồ án đạt được:

* Nghiên cứu và tìm hiểu về bài toán phân lớp, trình bày một số thuật toán phân lớp tiêu biểu
* Tìm hiểu một số đặc điểm của văn bản tiếng Việt phục vụ cho quá trình tiền xử lý
* Tìm hiểu và áp dụng các công cụ tiền xử lý dữ liệu đầu vào
* Nghiên cứu và tìm hiểu về thuật toán Support Vector Machine trên hai lớp và nhiều lớp
* Xây dựng chương trình lấy tin RSS hàng ngày
* Xây dựng chương trình huấn luyện và kiểm thử với bộ dữ liệu lấy từ chương trình lấy tin RSS.
* Xây dựng chương trình phân loại việc làm vào các category có sẵn từ các website không hỗ trợ RSS

1. **Hạn chế:**

* Một văn bản đầu vào sau khi phân loại chỉ thuộc về một loại nhãn.
* Hạn chế số lượng và chất lượng của kho dữ liệu tin tức ảnh hưởng đến chất lượng phân loại của hệ thống.
* Cần xác định giá trị chuẩn để một văn bản thuộc vào 1 hoặc nhiều thể loại, hoặc không thuộc thể loại nào.

**3. Hướng phát triển**

* Xây dựng bộ dữ liệu lớn hoàn chỉnh, phong phú về các lĩnh vực việc làm và chia các lĩnh vực càng nhỏ càng tốt
* Cần xác định giá trị chuẩn để một văn bản thuộc vào 1 hoặc nhiều thể loại, hoặc không thuộc thể loại nào
* Cải thiện hiệu xuất, tăng tốc độ xử lý dữ liệu
* Kiểm soát được thông tin lấy về theo ngày tháng năm mong muốn đảm bảo tính cập nhật của việc làm

# DANH MỤC TÀI LIỆU THAM KHẢO

**Tài liệu Tiếng Việt**

|  |  |
| --- | --- |
| [1] | Phan Thị Hà, Hà Hải Nam (2012), “Automatic main text extraction from web pages” *Tạp chí Khoa học Công nghệ, Viện Hàn lâm Khoa học Việt Nam*. |
| [2] | Phạm Nguyên Khang, Trần Cao Đệ (2012), “Phân loại văn bản với máy học vector hỗ trợ và cây quyết định”*,* *Tạp chí Khoa học* |
| [3] | Lê Hồng Phương “Nghiên cứu từ vựng Tiếng Việt với hệ thống SKETCH ENGINE” |
| [4] | Từ Minh Phương (2010), “Bài giảng Nhập môn trí tuệ nhân tạo”*, Học Viện Công nghệ bưu chính viễn thông* |
| [5] | Trần Thị Thu Thảo, Vũ Thị Chinh. “Xây dựng hệ thống phân loại tài liệu tiếng Việt” |
| [6] | Nguyễn Minh Thành (2011), “Đồ án Môn học Xử lý ngôn ngữ tự nhiên”  **Tài liệu Tiếng Anh** |
| [7] | Aidan Finn, Nicholas Kushmerick & Barry Smyth: Content classification for digital libraries |
| [8] | Adam L. Berger & Stephen A. Della Pietra & Vincent J. Della Pietra (1996), “A Maximum Entropy Approach to Natural Language Processing” |
| [9] | G. Piatetsky Shapiro, W. Frawley (1991), “Knowledge Discovery in Databases*”*, MIT Cambridge, MA. |
| [10] | Ian H.Witten, Eibe Frank (2005), “Data Mining: Practical Machine Learning Tools and Techniques” |
| [11] | Joachims T. (1998), “Text categorization with Support Vector Machines: Learning with many relevant features”, in Proc. of the European Conference on Machine Learning (ECML), pages 137–142 |
|  | **Website tham khảo** |
| [12] | <http://en.wikipedia.org/wiki/Support_vector_machine> |
| [13] | <http://svmlight.joachims.org/> |
| [14] | <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#libsvm_data_sets> |
| [15] | <http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html> |
| [16] | <http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTokenizer> |
| [17] | <http://z7.invisionfree.com/minhtanone/ar/t46.htm> |
| [18] | <http://seo4b.com/thuat-ngu-SEO/stop-words-la-gi.html> |

# PHỤ LỤC

**Danh sách stop-word tiếng Việt**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a\_ha | bập\_bà\_bập\_bõm | bỗng | chết\_nỗi | có |
| a-lô | bập\_bõm | bỗng\_chốc | chết\_tiệt | có chăng là |
| à\_ơi | bất\_chợt | bỗng\_dưng | chết\_thật | có dễ |
| á | bất\_cứ | bỗng\_đâu | chí\_chết | có thể |
| à | bất\_đồ | bỗng\_không | chín | có\_thể |
| á\_à | bất\_giác | bỗng\_nhiên | chỉn | có thế |
| ạ | bất\_kể | bỏ\_bố | chính | có vẻ |
| ạ\_ơi | bất\_kì | bỏ\_mẹ | chính\_là | cóc khô |
| ai | bất\_kỳ | bớ | chính\_thị | coi bộ |
| ai\_ai | bất\_luận | bởi | chỉ | coi mòi |
| ai\_nấy | bất\_nhược | bởi\_chưng | chỉ do | con |
| ái | bất\_quá | bởi\_nhưng | chỉ là | còn |
| ái\_chà | bất\_thình\_lình | bởi\_thế | chỉ tại | cô mình |
| ái\_dà | bất\_tử | bởi\_vậy | chỉ vì | cổ lai |
| ái\_khanh | bây\_bẩy | bởi\_vì | chiếc | công nhiên |
| alô | bây\_chừ | bức | cho đến | cơ |
| amen | bây\_giờ | cả | cho đến khi | cơ chừng |
| anh\_ta | bây\_giờ | cả\_thảy | cho nên | cơ hồ |
| áng | bây\_nhiêu | cái | cho tới | cơ mà |
| ào | bấy | các | cho tới khi | cơn |
| ăn | bấy\_giờ | cả\_thảy | choa | cu cậu |
| ắt | bấy\_chầy | cả\_thể | chốc chốc | của |
| ắt\_hẳn | bấy\_chừ | càng | chớ | cùng |
| ắt\_là | bấy\_giờ | cần | chớ chi | cùng cực |
| âu\_là | bấy\_lâu | căn | chợt | cùng nhau |
| ầu\_ơ | bấy\_lâu\_nay | căn\_cắt | chú | cùng với |
| ấy | bấy\_nay | cật\_lực | chu cha | cũng |
| ba | bấy\_nhiêu | cật\_sức | chú mày | cũng như |
| bài | bèn | cây | chú mình | cũng vậy |
| bàn | béng | cha\_chả | chui cha | cũng vậy thôi |
| bán\_mạng | bển | chành\_chạnh | chùn chùn | cứ |
| bản | bệt | chao\_ôi | chùn chũn | cứ việc |
| bao\_giờ | bị | chắc | chủn | cực kì |
| bao\_lăm | biết\_bao | chắc\_hẳn | chung cục | cực kỳ |
| bao\_lâu | biết\_bao\_nhiêu | chăn\_chắn | chung qui | cực lực |
| bao\_nả | biết\_chừng\_nào | chăng | chung quy | cuộc |
| bao\_nhiêu | biết\_đâu | chằn\_chặn | chung quy lại | cuốn |
| bay\_biến | biết\_đâu\_chừng | chẳng\_lẽ | chúng mình | dành |
| bằng | biết\_đâu\_đấy | chẳng\_những | chúng ta | dào |
| bằng\_ấy | biết\_mấy | chẳng\_nữa | chúng tôi | dạ |
| bằng\_không | bộ | chẳng\_phải | chứ | dần dà |
| bằng\_nấy | bội\_phần | chậc | chứ lại | dần dần |
| bắt\_đầu\_từ | bông | chầm\_chập | chứ lị | dầu sao |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| dẫu | để cho | ha | hồi lùng | không thể |
| dẫu sao | đêm đêm | hả | hốt nhiên | không\_thể |
| dễ sợ | đến | hà rằm | hơi | không thể nào |
| dễ thường | đến cùng | hà rầm | hỡi | không trách |
| dĩ chí | đến đây | hàng | hỡi ôi | khư khư |
| do | đến nỗi | hay là | hòn | khứ hồi |
| do vì | đều | hãy | hơn | khướt |
| do đó | đều đều | hắn | hung | kia |
| do vậy | đi | hẳn là | húng hắng | kia mà |
| dở chừng | đích thị | hằng | huống | kì cùng |
| dù cho | đó | hẵng | huống chi | kì thật |
| dù rằng | đó đây | hiện\_nay | huống gì | kì thực |
| dùng | đoá | hầu như | huống hồ | kì tình |
| duy | đôi | hcm | huống nữa | kìa |
| dữ | đôi khi | hé | hừ | kìn kìn |
| dưới | đối với | hèn chi | hử | kỳ thật |
| đã | đổng | hèn gì | hứ | kỳ thực |
| đại để | đột nhiên | hèn nào | hừm | kỳ tình |
| đại loại | đơ | hén | ít | lá |
| đại nhân | đúng | hê | kém | là |
| đại phàm | đúng là | hề | kẻo mà | lạ lùng |
| đạt | đúng như | hễ | kẻo nữa | lại |
| đoạn | đúng thật | hềnh hệch | kẻo rồi | lại thế nữa |
| đang | đúng thật là | hết | kế đến | làm |
| đáng lẽ | đúng vậy | hết mình | kế tiếp | làm sao |
| đáng lí | đùng đùng | hết mực | kha khá | lắm |
| đáng lý | đưa | hết nước | khá | lắm lắm |
| đành đạch | được | hết nước hết cái | khả dĩ | lắm lúc |
| đánh đùng | eo ôi | hết sảy | khẩu | lắm khi |
| đáo để | êu | hết sức | khi nãy | lẳng lặng |
| đặc cách | gặp | hết thảy | khi đó | lầm lụi |
| đăm đắm | gì | hết ý | khi ấy | làn |
| đằng ấy | gia dĩ | hình | khi | lần hồi |
| đâu | giả như | hình như | khi không | lần lượt |
| đâu đâu | giả phỏng | hiếm | khiến | lập tức |
| đâu đây | giả tỉ như | hiện | khó | lâu lâu |
| đâu đấy | giả tỷ như | hiện nay | khôn cùng | lẻn |
| đâu đó | giá mà | hiện tại | khôn xiết | lén |
| đầu | giá mà | hò khoan | không biết | leo lẻo |
| đầu tiên | giá như | hoài của | không có | lẽo đẽo |
| đây | giá phỏng | hoặc | không dưng | lêu |
| đây đó | giờ | hoặc giả | không khéo | lí láu |
| đấy | giờ đây | hoặc là | không mấy khi | lịa |
| để | giữa | hồ dễ | không những | lóc cóc |
| để mà | giữa chừng | hộc tốc | không phải | lọc cọc |
| long lóc | năng | nhỉ | phải chi | rất mực |
| lông lốc | nằng nặc | nhiều | phải chăng | ren rén |
| lũ lượt | nẫy | nhiên hậu | phăn phắt | rén |
| lui lủi | nấy | nhiệt liệt | phắt | rích |
| lùi lũi | nên | nhóm | phè | riệt |
| lùi lụi | nên chi | nhón nhén | phỉ phui | riu ríu |
| lủi thủi | nền | nhỡ ra | pho | rón rén |
| luôn | nếu | nhung nhăng | phóc | rồi |
| luôn luôn | nếu như | như | phỏng | rốt cục |
| luôn thể | ngay | như chơi | phỏng như | rốt cuộc |
| luôn tiện | ngay cả | như không | phót | rút cục |
| lý láu | ngay lập tức | như quả | phốc | rứa |
| mãi | ngay lúc | như thể | phụt | sa sả |
| mãi mãi | ngay khi | như tuồng | phương chi | sạch |
| mặc dầu | ngay từ | như vậy | phứt | sao |
| mặc dù | ngay tức khắc | nhưng | qua quít | sau chót |
| mặc dù vậy | ngày càng | nhưng mà | qua quýt | sau cùng |
| mặc nhiên | ngày ngày | những | quả | sau cuối |
| mặc sức | ngày xưa | những ai | quả đúng | sau đó |
| mặc tình | ngày xửa | những như | quả là | sắp |
| mặt | ngăn ngắt | nhược bằng | quả tang | sất |
| màn | nghe chừng | nlđ | quả thật | sẽ |
| mất | nghe đâu | nó | quả tình | sì |
| mấy | nghen | nóc | quả vậy | so |
| mầy | nghiễm nhiên | nọ | quá | so\_sánh |
| mậy | nghỉm | nổi | quá chừng | song le |
| mèm | ngõ hầu | nớ | quá độ | số là |
| miễn là | ngoải | nữa | quá đỗi | sống |
| miễn sao | ngoài | nức nở | quá lắm | sốt sột |
| min | ngôi | oai oái | quá sá | sở dĩ |
| mình | ngọn | oái | quá thể | suýt |
| mọi | ngọt | ô hay | quá trời | sự |
| món | ngộ nhỡ | ô hô | quá ư | tà tà |
| mô phật | ngươi | ô kê | quá xá | tại |
| mô tê | nhau | ô kìa | quý hồ | tại vì |
| một | nhân dịp | ồ | quyển | tấm |
| một mực | nhân tiện | ôi chao | quyết | tấn |
| một phép | nhất | ôi thôi | quyết nhiên | tự |
| một số | nhất đán | ối dào | ra | tự vì |
| mỗi | nhất định | ối giời | ra phết | tanh |
| mới | nhất loạt | ối giời ơi | ra trò | tăm tắp |
| muốn | nhất luật | ôkê | ráo | tắp |
| mựa | nhất mực | ổng | ráo trọi | tắp lự |
| nào | nhất nhất | ơ | rày | tất cả |
| nào là | nhất quyết | ơ hay | răng | tất tần tật |
| nay | nhất sinh | ơ kìa | rằng | tất tật |
| này | nhất tâm | ờ | rằng là | tất thảy |
| nãy | nhất tề | ớ | rất | tênh |
| nãy giờ | nhất thiết | ở | rất chi là | tha hồ |
| năm thì mười hoạ | nhé | ơi | rất đỗi | thà |
| thà là | thỏm | trên | úi dào | vô luận |
| thà rằng | thọt | trển | ư | vô vàn |
| thái quá | thốc | trệt | ứ hự | vốn dĩ |
| than ôi | thốc tháo | trếu tráo | ứ ừ | với lại |
| thanh | thộc | trệu trạo | ử | vở |
| thành ra | thôi | trong | ừ | vung tàn tán |
| thành thử | thốt | trỏng | và | vung tán tàn |
| thảo hèn | thốt nhiên | trời đất ơi | vả chăng | vung thiên địa |
| thảo nào | thuần | trừ phi | vả lại | vụt |
| thậm | thục mạng | tp | vạn nhất | vừa |
| thậm chí | thúng thắng | tphcm | văng tê | vừa mới |
| thật lực | thửa | tù tì | vẫn | xa xả |
| thật vậy | thực ra | tuần | vâng | xăm xăm |
| thật ra | thực vậy | tuần tự | vậy | xăm xắm |
| thẩy | thương ôi | tuốt luốt | vậy là | xăm xúi |
| thêm | tiện thể | tuốt tuồn tuột | vậy thì | xềnh xệch |
| thế | tiếp đó | tuốt tuột | veo | xệp |
| thế à | tiếp theo | tuy | veo veo | xin |
| thế là | tít mù | tuy nhiên | vèo | xiết bao |
| thế mà | tỏ ra | tuy rằng | về | xoành xoạch |
| thế nào | tỏ vẻ | tuy thế | vì | xoẳn |
| thế nên | tò te | tuy vậy | vì chưng | xoét |
| thế ra | toà | tuyệt nhiên | vì thế | xoẹt |
| thế thì | toé khói | từ | vì vậy | xon xón |
| thếch | toẹt | từ\_tốn | ví bằng | xuất kì bất ý |
| thi thoảng | tọt | từng | ví dù | xuất kỳ bất ý |
| thì | tốc tả | tức thì | ví phỏng | xuể |
| thình lình | tôi | tức tốc | ví thử | ý chừng |
| thỉnh thoảng | tối ư | tựu trung | vị tất | ý da |
| thoạt | tông tốc | ủa | vn |  |
| thoạt nhiên | tột | úi | vô hình trung |  |
| thoắt | tràn cung mây | úi chà | vô kể |  |