ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

KHÓA LUẬN TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG LỌC RÁC CHO HỆ THỐNG PHÁT HIỆN TIN NÓNG TỪ CÁC TRANG TIN TỨC

Giảng viên hướng dẫn: **TS. Huỳnh Ngọc Tín**

Sinh viên 1: **Hoàng Anh Minh**

MSSV: **13520505**

Lớp: **CNPM08**

Khóa: **2013-2017**

Sinh viên 2: **Lâm Tuấn Anh**

MSSV: **13520020**

Lớp: **CNPM08**

Khóa: **2013-2017**

TP HỒ CHÍ MINH – Tháng 6, năm 2017

ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

KHÓA LUẬN TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG LỌC RÁC CHO HỆ THỐNG PHÁT HIỆN TIN NÓNG TỪ CÁC TRANG TIN TỨC

Giảng viên hướng dẫn: **TS. Huỳnh Ngọc Tín**

Sinh viên 1: **Hoàng Anh Minh**

MSSV: **13520505**

Lớp: **CNPM08**

Khóa: **2013-2017**

Sinh viên 2: **Lâm Tuấn Anh**

MSSV: **13520020**

Lớp: **CNPM08**

Khóa: **2013-2017**

Đơn vị đồng hành: **VCCorp**

TP HỒ CHÍ MINH – Tháng 12, năm 2017

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số …… ngày …….. của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. ……. – Chủ tịch.
2. ……. – Thư ký.
3. ……. – Ủy viên.
4. ……. – Ủy viên.

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn chân thành đến TS. Huỳnh Ngọc Tín, người đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn em trong suốt quá trình thực hiện khóa luận tốt nghiệp này. Những kinh nghiệm, lời nhận xét và chia sẻ của thầy truyền đạt cho em thật sự rất quý báu và ý nghĩa.

Em cũng gửi lời cảm ơn đến quý thầy cô đang công tác tại Trường Đại học Công Nghệ Thông Tin nói chung và Khoa Khoa học máy tính nói riêng đã dạy dỗ, truyền đạt những kiến thức, kinh nghiệm vô cùng quý báu, giúp em vận dụng trong quá trình thực tập.

Ngoài ra, xin cảm ơn sự hỗ trợ quý giá của các bạn trong nhóm AdTech tại công ty VCCorp trong quá trình xây dựng hệ thống.

Cuối cùng, em xin chúc thầy luôn mạnh khỏe và đạt nhiều thành công trong cuộc sống.

Tp. HCM, tháng 12 năm 2017

Sinh viên thực hiện

Hoàng Anh Minh

Lâm Tuấn Anh

Mục lục

Danh mục các ký hiệu, thuật ngữ và chữ viết tắt

Topic Detection and Tracking	: Phát hiện và theo dõi sự kiện
First Story	: Bài viết đầu tiên về một sự kiện
First Story Detection	: Phát hiện bài viết đầu tiên
Nearest Neighbor Search	: tìm láng giềng gần nhất
Locality Sensitive Hashing	: thuật toán Locality Sensitive Hashing
Document	: một bài viết/điểm dữ liệu trong hệ thống
Tweet	: một bài đăng bởi bất kỳ người dùng nào trên Twitter
Merge Threshold	: Giá trị ngưỡng dùng khi xét hai document có đủ tương đồng để gom vào một cụm hay không

Danh sách bảng

Danh sách hình vẽ

TÓM TẮT KHÓA LUẬN

Cùng với sự phát triển nhanh chóng của internet là sự tăng trưởng mạnh mẽ của ngành truyền thông báo chí. Cụ thể, tính đến ngày 31/12/2015, cả nước Việt Nam có 105 cơ quan báo điện tử, 207 trang thông tin điện tử tổng hợp của các cơ quan báo chí [1], và cứ mỗi giờ có khoảng 2,118 bài được đăng trên các trang báo mạng Việt Nam [2]. Cùng với sự tăng trưởng lớn về thông tin là nhu cầu phân tích và xử lý các thông tin đó để phục vụ cho nhiều mục tích khác nhau, và ngày nay nhiều doanh nghiệp đã và đang xây dựng các hệ thống phân tích dữ liệu tin tức để phục vụ tốt hơn cho người dùng và biên tập viên.

Việc phân tích dữ liệu tin tức đòi hỏi hệ thống phải thu thập dữ liệu từ nhiều website tin tức khác nhau để có được một nguồn dữ liệu đa dạng, phong phú và đủ lớn. Tuy nhiên, dữ liệu thu thập từ các website có bản chất không phù hợp với mục đích của hệ thống phân tích có thể gây ra nhiễu và khiến cho kết quả phân tích, xử lý không chính xác. Do đó, việc sàn lọc dữ liệu thu thập được trước khi đưa vào phân tích và xử lý là cần thiết.

Hiểu được nhu cầu trên, chúng em thực hiện khóa luận này với mục tiêu xây dựng một hệ thống lọc "rác" cho dữ liệu tin tức từ các nguồn báo chính thống Việt Nam, cụ thể là lọc "rác" cho hệ thống "Phát hiện tin nóng".

Trong phạm vi của khóa luận, chúng em đã nghiên cứu một số thuật toán phân lớp: SVM, Naive Bayes, J48.

Sau quá trình thực hiện, đề tài khóa luận thu thập được các kết quả như sau:

- Thu thập bộ dữ liệu gồm các bài đăng trên các website tin tức Việt Nam.
- Đánh giá và so sánh các thuật toán: SVM, Naive Bayes, J48.

[1] http://mic.gov.vn/Pages/TinTuc/116095/Tinh-hinh-phat-trien-linh-vuc-bao-chi-va-phat-thanh-truyen-hinh-nam-2015.html

[2] Theo thống kê từ dữ liệu thu thập bởi công ty VCCorp tính đến tháng 7/2017.

- Xây dựng được hệ thống lọc rác cho hệ thống phát hiện tin nóng.

Chương 1

MỞ ĐẦU

1.1 Dẫn nhập

Trong thời đại bùng bổ về thông tin, thông tin trên mạng internet được sinh ra với lưu lượng và khối lượng khổng lồ. Cụ thể, trong năm 2017, cứ mỗi phút lại có 149,513 email được gửi đi, 3.3 triệu bài Facebook được đăng lên, 3.8 triệu lượt tìm kiếm trên Google được sinh ra[1],... Ngày nay, để tận dụng triệt để nguồn thông tin khổng lồ đó, rất nhiều hệ thống phân tích và xử lý dữ liệu được sinh ra. Tuy nhiên ngân hàng dữ liệu càng lớn thì sự đồng bộ về chất lượng của dữ liệu càng không được đảm bảo, thêm vào đó, sự đa dạng của các nguồn dữ liệu cũng đồng nghĩa với việc không phải mẫu dữ liệu nào cũng phù hợp với nhu cầu của tất cả hệ thống phân tích và xử lý khác nhau. Việc phân tích và xử lý các mẫu dữ liệu kém chất lượng hay không phù hợp sẽ làm gia tăng chi phí, cũng như làm suy giảm hiệu suất và tính hiệu quả của hệ thống. Do đó việc sàng lọc dữ liệu trước khi đưa vào hệ thống để phân tích và xử lý là cần thiết.

Hệ thống phát hiện tin nóng hiện đang được triển khai và hoạt động tại VCCorp là một trong các hệ thống phân tích và xử lý dữ liệu tin tức, được phát triển nhằm mục đích xác định tin nóng từ các trang báo điện tử Việt Nam. Tuy nhiên, sự đa dạng và phong phú của nguồn tin dẫn đến việc bản chất của nhiều bài thu thập được không phù hợp với mục tiêu của hệ thống. Những bài đó, đối với hệ thống, được hiểu là "rác". Số lượng "rác" lớn sẽ gây nhiễu và làm cho việc phân tích và xử lý trở nên kém chính xác. Một cơ chế lọc rác hiệu quả sẽ giúp cải thiện độ chính xác cũng như giảm tải cho hệ thống từ việc phân tích và xử lý các dữ liệu không có giá trị sử dụng.

[1]https://www.smartinsights.com/internet-marketing-statistics/happens-online-60-seconds/

Bài toán đặt ra là làm thể nào để có thể lọc "rác" trong quá trình thu thập tin tức một các chính xác. Với nhu cầu và điều kiện thuận lời từ công ty VCCorp, khóa luận này hướng đến việc xây dựng một hệ thống có khả năng lọc "rác" để tăng hiệu quả cho hệ thống phát hiện tin nóng và giúp đánh giá độ đáng tin cậy của các nguồn tin.

1.2 Mục tiêu đề tài

- Tìm hiểu và đánh giá các thuật toán phân lớp cho việc lọc rác dữ liệu tin tức
- Xây dựng hệ thống lọc rác áp dụng các thuật toán phân lớp.

1.3 Nội dung thực hiện

- Tìm hiểu bài toán phân loại tin tức, tìm hiểu các phương pháp và các hướng tiếp cận
- Thử nghiệm đánh giá các phương pháp đã tìm hiểu:
  - Thu thập dữ liệu tin tức từ cơ sở dữ liệu của công ty VCCorp.
  - Tiến hành một số thống kê trên dữ liệu thu thập được.
  - Huấn luyện và so sánh các thuật toán phân lớp: Naive Bayes, SVM, J48.
- Xây dựng hệ thống:
  - Tìm hiểu về MongoDB, framework Struts 2, thư viện Apache Lucene, Weka, LibSVM.
  - Xây dựng kiến trúc hệ thống.
  - Thiết kế chức năng, giao diện hệ thống.
  - Cài đặt hệ thống.

1.4 Phạm vi đề tài

- Nguồn dữ liệu: các bài viết từ báo chính thống Việt Nam.
- Ngôn ngữ: tiếng Việt.
- Các thuật toán tìm hiểu: Naive Bayes, SVM, J48

1.5 Cấu trúc báo cáo

Luận văn được bố cục thành chương mục như sau:

- **Chương 1**: Mở đầu: Giới thiệu về đề tài.
- **Chương 2**: Bài toán lọc rác tin tức cho hệ thống phát hiện tin nóng. Trình bày cơ sở lý thuyết, các khái niệm, phương pháp tiếp cận liên quan đến bài toán lọc rác.
- **Chương 3**: Hiện thực hệ thống lọc rác: Trình bày về kiến trúc, cài đặt hệ thống phát hiện tin nóng.
- **Chương 4**: Thực nghiệm và đánh giá: Trình bày về bộ dữ liệu thu thập được, đánh giá và so sánh các thuật toán.
- **Mục Tài liệu tham khảo**
- **Phụ lục. Giới thiệu về thư viện Apache Lucene**

Chương 2

BÀI TOÁN LỌC RÁC TIN TỨC CHO HỆ THỐNG PHÁT HIỆN TIN NÓNG

2.1 Mở đầu

Chương này sẽ giới thiệu bài toán lọc rác tin tức cho hệ thống phát hiện tin nóng. Trình bày cơ sở lý thuyết và phát biểu bài toán lọc rác tin tức. Cuối cùng trình bày một số phương pháp tiếp cận bài toán lọc rác tin tức và các kiến thức liên quan.

2.2 Giới thiệu bài toán

2.2.1 Các khái niệm cơ bản

Để có thể xác định khái niệm rác, ta phải hiểu được khái nhiệm "tin nóng". Le [1] đã định nghĩa khái niệm "tin nóng" như sau:

- *Tin nóng (Hot news)*: những tin viết về một sự kiện mới xảy ra, có tính thời sự, có tầm ảnh hưởng rộng, thu hút được sự chú ý, quan tâm của cộng đồng.

Ngoài ra Gyöngyi và Garcia-Molina định nghĩa "đăng rác" như sau:

- *Đăng rác (Spamming)*: các hành động có chủ đích của con người nhằm thiên vị tính liên quan của một trang web nào đó, làm sai lệch giá trị của trang web.

Từ các khái niệm trên ta có định nghĩa "rác" cho bài toán như sau:

Định nghĩa: Rác là các tin không mang tính chất sự kiện, thời sự, và làm sai lệch kết quả phân tích và xử lý dữ liệu của hệ thống phân loại tin tức.

Từ định nghĩa trên ta có thể xác định được các loại tin sẽ được xem là rác:

- *Quảng cáo*: các bài quảng cáo, giới thiệu sản phẩm, các chương trình khuyến mãi, trúng thưởng.
- *Tuyển dụng*: các bài đăng tuyển dụng, giới thiệu việc làm của các tổ chức, doanh nghiệp.
- *Chia sẻ* : các bài chia sẻ mẹo vặt, thủ thuật, tâm sự, chuyện đời tư, giới thiệu ẩm thực, thời trang, hoặc các kiến thức chung.

2.2.2 Bài toán phân lớp văn bản

Bài toán phân lớp văn bản là bài toán có lịch sử lâu đời, một số ghi nhận cho thấy nó bắt đầu từ những năm 1800s, tuy nhiên tới nay vẫn chưa có một sự thống nhất nào về phương pháp tốt nhất cho bài toán phân lớp [2].

Theo Zechner [2], bài toán phân lớp văn bản gồm các thành phần chính cần quan tâm:

- Loại phân lớp: supervised hoặc unsupervised.
- Mục tiêu: đối tượng thành phần của văn bản mà ta muốn quan tâm.
- Kho từ vựng: kích thước và chất lượng của kho từ vựng ảnh hưởng lớn đến việc phân lớp.
- Features: đặc trưng của văn bản dùng cho việc phân lớp.
- Bộ phân lớp: thuật toán máy học dùng cho việc phân lớp.

2.2.3 Bài toán lọc rác tin tức

Bài toán lọc rác hay còn được biết đến như là bài toán phát hiện rác (Spam detection) là một dạng bài toán phân lớp văn bản nhị phân. Tức là chỉ có hai lớp: "rác" và "không rác". Đối với hệ thống phát hiện tin nóng cũng như các hệ thống phân tích và xữ lý dữ liệu khác, việc phát hiện và lọc rác phải được diễn ra trước khi quá trình phân tích và xử lý dữ liệu bắt đầu để đảm bảo quá trình phân tích và xử lý dữ liệu chính xác và đáng tin cậy.

Mặc dù nguồn dữ liệu mà chúng ta quan tâm là tin tức từ các trang báo điện tử chính thống, số lượng nguồn tin, chủ đề tin và số lượng tin rất lớn dẫn đến việc chất lượng dữ liệu tin tức không được đảm bảo cũng như tính chất dữ liệu không được phù hợp với nhu cầu của hệ thống. Do đó việc lọc rác tin tức là cần thiết cho việc khai thác và sử dụng nguồn dữ liệu tin tức một các hiệu quả.

2.2.4 Phát biểu bài toán lọc rác tin tức cho hệ thống phát hiện tin nóng

Cho trước một bài báo d_ii. Mục tiêu của bài toán là tìm ra được nhãn l_i của bài báo d_i và nhãn đó nhận một trong hai giá trị: "rác" và "không rác".

2.3 Các nghiên cứu liên quan

Vấn đề phát hiện và lọc rác là một vấn đề lâu đời và kinh điển đối với bài toán phân lớp văn bản. Do đó số lượng nghiên cứu xoay quan vấn đề này cũng rất lớn và đa dạng. Phần lớn các nghiên cứu cho đến ngày nay tập trung vào việc phát hiện rác cho email, web và mạng xã hội.

Sahami và cộng sự [3] sử dụng các tiếp cận Bayesian vào vấn đề lọc rác cho email. Thực nghiệm cho thấy các bộ phân lớp đạt được hiệu suất cao hơn khi ta cân nhắc các tính năng đặc trưng theo domain và nội dung văn bản của email. Ngày nay thì các phương pháp phát hiện và lọc rác cho email đã khá hoàn thiện, và các phương pháp phát hiện và lọc rác Bayesian được áp dụng rộng rãi trong các email client và server.

Rothwell và cộng sự[4] đã xây dựng một hệ thống phát hiện các thông điệp rác bằng cách sử dụng neural network. Khi một thông điệp được đưa vào hệ thống nó sẽ được phân rã và các thông số thống kê trong thông điệp đó sẽ được thu thập bở bộ phân tích số liệu. Một neural network được huấn luyện bằng cách sử dụng sự kết hợp tuyến tính linh hoạt đề thay đổi các trọng số cùng với bộ phân tích số liệu thống kê sẽ được dùng để quyết định xem một thông điệp có phải là rác hay không.

Idris và cộng sự[5] đã xây dựng một hệ thống phát hiện rác cho email sử dụng thuật toán negative selection và particle swarm optimization (NSA-PSO). Hệ thống áp dụng thuật toán local outlier factor (LOF) để tính toán độ mở rộng của không gian không rác để thu thập được các tính năng tốt nhất cho việc phân lớp. Mô hình NSA-SPO

được dùng để phân biệt giữa rác và không rác trong các mạng client-server.

Vấn đề phát hiện rác cho web thì khó khăn hơn vì độ lớn và tốc độ thay đổi nhanh chóng. Gyöngyi và cộng sự [6] đề xuất sử dụng thuật toán TrustRank để đo kiểm điểm tin cậy của đồ thị Web. Dựa trên điểm tin cậy, các trang web tốt sẽ được đánh giá cao hơn va những trang web xấu có thể được search engine bỏ qua.

Benczúr và cộng sự[7] tiếp cận bài toán phát hiện web rác bằng các xác định các trang web được back linked bởi nhiều trang khác, bằng việc tính toán điểm SpamRank để đánh giá lại PageRank thật sự xứng đáng của một trang web.

Phát hiện rác trên các nền tảng mạng xã hội cũng là một trong các chủ đề nóng trong lĩnh vực này. Jin và cộng sự[8] đã đề xuất một hệ thống phát tự động thu hoạch các hoạt động đăng tin rác trên ác trang mạng xã hội bằng cách sử dụng các bộ định vị xã hội đối với các cơ sở người dùng lớn. Hệ thống dùng các đặc trưng của cả hình ảnh và văn bản để xác định các nội dung rác, và sử dụng thuật toán gom cụm GAD cho dữ liệu quy mô lớn.

Wang cũng xây dựng một hệ thống phát hiện rác cho dữ liệu Twitter sử dụng thuật toán phân lớp Naive Bayes. Hệ thống thức hiện phân lớp dựa trên các đặc tính mạng xã hội của Twitter và đặc tính nội dung của dữ liệu văn bản. Theo kết quả đo kiểm đánh giá thì độ chính xác của hệ thống đạt đến 89%[9]

2.4 Giới thiệu một số phương phát trừu tượng hóa dữ liệu văn bản

Do máy tính không thể đọc hiểu và xử lý được dữ liệu thuần văn bản, ta cần có một phương pháp trừu tượng hóa dữ liệu văn bản thành các con số mà máy tính có thể hiểu được.

Term Frequency - Inverse Document Frequency (TF-IDF)

Trong lĩnh vực thu thập thông tin, TF-IDF là một dữ liệu số dùng để thể hiện độ quan trọng của một từ trong một một tập hợp hoạc một kho từ vựng [10]. Nó thường được dùng trong việc thu thập thông tin, text mining và user modeling. Giá trị của TF-IDF tăng theo tỉ lệ với số lần một từ xuất hiện trong một văn bản, nhưng thường bị giảm lại theo tần xuất của từ đó trong toàn kho từ vựng, vì một từ có thể chỉ đơn

giản là nó phổ biến hơn các từ khác. Ngày nay, TF-IDF là một trong những phương pháp tính trọng số cho các từ phổ biến nhất; 83% các hệ thống khuyến nghị dựa trên văn bản trong lĩnh vực thư viện điện tử sử dụng TF-IDF [11].

TF-IDF gồm có hai thành phần chính Term Frequency (TF) và Inverse Document Frequency (IDF).

TF của từ t trong văn bản d được thể hiện dưới dạng $tf(t, d)$. Ta có nhiều cách tính TF, trong đó cách đơn giản nhất là đếm số lần một từ xuất hiện trong một văn bản. Nếu ta gọi số lần từ t xuất hiện trong văn bản d là $f_{t,d}$ thì ta có $tf(t, d) = f_{t,d}$

IDF là thước đo lượng thông tin một từ cung cấp, có nghĩa là liệu từ đó hiếm gặp hay xuất hiện xuyên suốt trong kho từ vựng. IDF được tính bằng công thức:

$$idf(t, D) = \log \frac{N}{1 + |\{d \in D : t \in d\}|} \tag{2.1}$$

Trong đó:

- D: kho từ vựng.
- N: số lượng văn bản trong kho từ vựng.
- $|\{d \in D : t \in d\}|$: số lượng văn bản và có xuất hiện từ t.

Word Vector

Word vector là một hình thức biểu diễn các từ trong một kho từ vựng dưới dạng các vector. Các từ thường xuất hiện cùng nhau trong các nhiều ngữ cảnh khác nhau thì sẽ được biểu diễn bằng các vector gần với nhau trong không gian vector [12]. Tập hợp các kỹ thuật để nhận vào input là một kho từ vựng và cho ra một không gian vector tương ứng được gọi là word2vec.

Word vector thường được sử dụng trong các công cụ tìm kiếm và dịch thuật do nó cung cấp thông tin về sự tương qua ngữ nghĩa giữa các từ và các ngữ cảnh cũng như giữa các từ với nhau.

2.5 Các phương pháp tiếp cận phổ biến

Phân lớp là quá trình xác định xem một đối tượng sẽ thuộc về lớp nào dựa trên việc quan sát các đối tượng khác mà lớp của chúng đã được xác định từ trước. Thông

thường, các đối tượng cần phân lớp sẽ được phân tích thành các thuộc tính có thể đo đếm được. Một thuật toán dùng để phân lớp, hay còn được gọi là một bộ phân lớp, sẽ dựa trên các thuộc tính đó để ánh xạ các đối tượng input mới vào lớp của chúng. Đây là một trong những tác vụ chính của lĩnh vực khai thác dữ liệu.

Bài toán phân lớp bao gồm nhiều thuật toán khác nhau, trong phạm vi của khóa luận chúng ta sẽ chỉ quan tâm đến các thuật toán: Support vector machine, Naive Bayes, và J48 decision tree.

2.5.1 Thuật toán phân lớp Support Vector Machine (SVM)

2.5.1.1 Giới thiệu

Support vector machine hay còn gọi là support vector network là một thuật toán phân lớp dành cho các bài toán phân lớp nhịphân. SVM thực hiện phân lớp bằng việc ánh xạ các vector lên một không gian nhiều chiều. Trên không gian này sẽ sinh ra một mặt phẳng quyết định tuyến tính phân tách hai lớp dữ liệu [13].

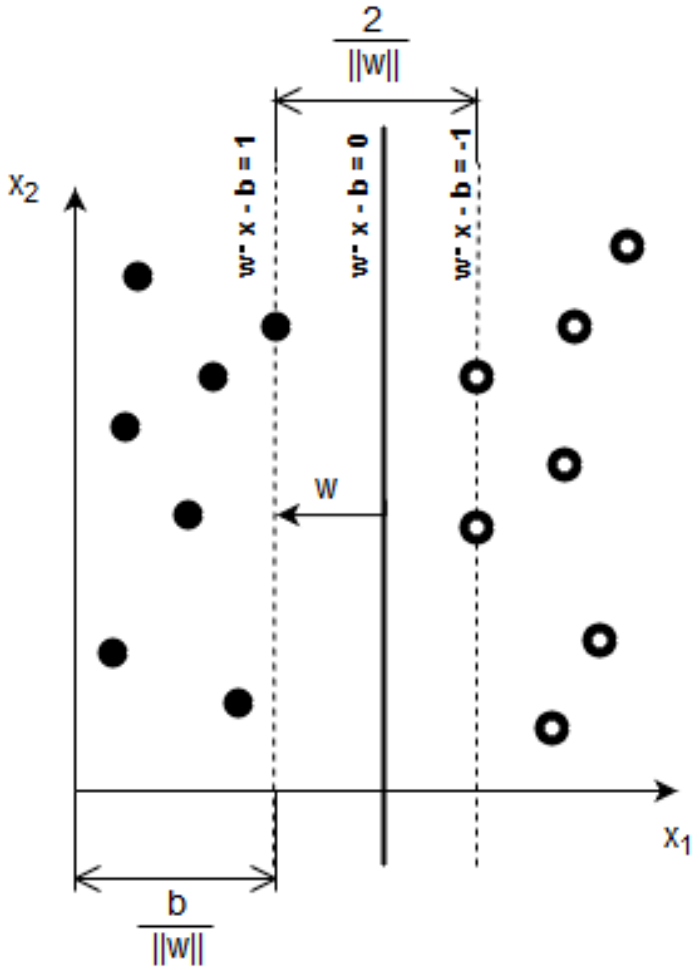
2.5.1.2 Ưu điểm, hạn chế

Ưu điểm:

- Rất hiệu quả khi tập dữ liệu có chiều không gian lớn.
- Vẫn hiệu quả trong các trường hợp số chiều lớn hơn số lượng dữ liệu.
- Chỉ sử dụng một phần dữ liệu huấn luyện để ra quyết định (các dữ liệu huấn luyện này là các support vector).
- Có nhiều kernel để sử dụng cho chức năng ra quyết định, thể hiện sự đa dạng, phân lớp dữ liệu tốt.

Nhược điểm:

- Nếu như số chiều lớn hơn nhiều so với số lượng dữ liệu thì cho hiệu năng thấp.
- Vì thuật toán SVM là thuật toán phi xác suất nên không cung cấp các ước lượng xác suất trực tiếp để kiểm tra độ chính xác mà dùng phương pháp trung gian là k-fold cross validation để kiểm tra với chi phí cao.



Hình 2.1: *Minh họa về một bộ phân lớp SVM tuyến tính. Siêu phẳng với lề cực đại và các margin của một bộ SVM được học từ các mẫu được gán nhãn thành hai lớp. Những mẫu nằm trên margin được gọi là support vector*

2.5.1.3 SVM với bài toán lọc rác tin tức

Bài toán lọc rác văn bản là một bài toán phân lớp nhị vân với các vector đặc trưng có số chiều rất lớn nên đây là một bài toán rất phù hợp cho việc sử dụng thuật toán SVM.

2.6 Thuật toán Naive Bayes

2.6.1 Giới thiệu

Thuật toán Naive Bayes là một thuật toán phân lớp xác suất đơn giản dùng để tính một tập các xác suất bằng cách đếm tần suất và kết hợp của các giá trị trong

một tập cho trước. Thuật toán sử dụng định luật Bayes sẽ giả định rằng tất cả các chiều(attributes) của dữ liệu là độc lập với nhau. Các giả định rằng các chiều là độc lập với nhau rất khó để có thể xuất hiện trong thực tế. Tuy nhiên giả thiết ngây ngô này lại mang lại những kết quả phân lớp tốt cho nhiều bài toán phân lớp [14].

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là P(A|B), và đọc là "xác suất của A nếu có B". Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó. Theo định lí Bayes, xác suất xảy ra A khi biết B sẽ phụ thuộc vào 3 yếu tố: Xác suất xảy ra A của riêng nó, không quan tâm đến B. Kí hiệu là P(A) và đọc là xác suất của A. Đây được gọi là xác suất biên duyên hay xác suất tiên nghiệm, nó là "tiên nghiệm" theo nghĩa rằng nó không quan tâm đến bất kỳ thông tin nào về B. Xác suất xảy ra B của riêng nó, không quan tâm đến A. Kí hiệu là P(B) và đọc là "xác suất của B". Đại lượng này còn gọi là hằng số chuẩn hóa (normalising constant), vì nó luôn giống nhau, không phụ thuộc vào sự kiện A đang muốn biết. Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là P(B|A) và đọc là "xác suất của B nếu có A". Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra. Chú ý không nhầm lẫn giữa khả năng xảy ra B khi biết A và xác suất xảy ra A khi biết B. Tóm lại định lý Bayes sẽ giúp ta tính ra xác suất xảy ra của một giả thuyết bằng cách thu thập các bằng chứng nhất quán hoặc không nhất quán với một giả thuyết nào đó. Khi các bằng chứng tích lũy, mức độ tin tưởng vào một giả thuyết thay đổi. Khi có đủ bằng chứng, mức độ tin tưởng này thường trở nên rất cao hoặc rất thấp, tức là xác xuất sảy ra giả thuyết sẽ thay đổi thì các bằng chứng liên quan đến nó thay đổi.

2.6.2 Ưu điểm, hạn chế

Ưu điểm:

- Đơn giản, nhanh, dễ cài đặt
- Nếu điều kiện giả định độc lập của thuật toán Naive Bayes là đúng, thì nó sẽ hội tụ nhanh hơn nhiều khi so sánh với các mô hình phân biệt như là logistic regression.

- Ngay cả khi điều kiên giả định độc lập của thuật toán Naive Bayes không được đảm bảo một cách tuyệt đối, thì thuật toán này vẫn hoạt động một cách ổn định và đáng tin cậy trong thực tế.
- Không cần nhiều dữ liệu huấn luyện.
- Khả năng mở rộng cao. Thuật toán này có thể mở rộng một cách tuyến tính theo số lượng các predictor và các điểm dữ liệu.
- Có thể dùng cho các bài toán phân lớp nhị phân lẫn các bài toán phân lớp đa lớp.
- Có thể thực hiện các dự đoán có tính xác suất.
- Xử lý được cả dữ liệu rời rạc lẫn dữ liệu liên tục.
- Không có các đặc tính thừa thải.

Nhược điểm:

- Trên thực tế điều kiện giả định độc lập là gần như không thể xảy ra, điều này dẫn đến sự hao hụt về độ chính xác.

2.6.3 Thuật toán Naive Bayes và bài toán lọc rác tin tức

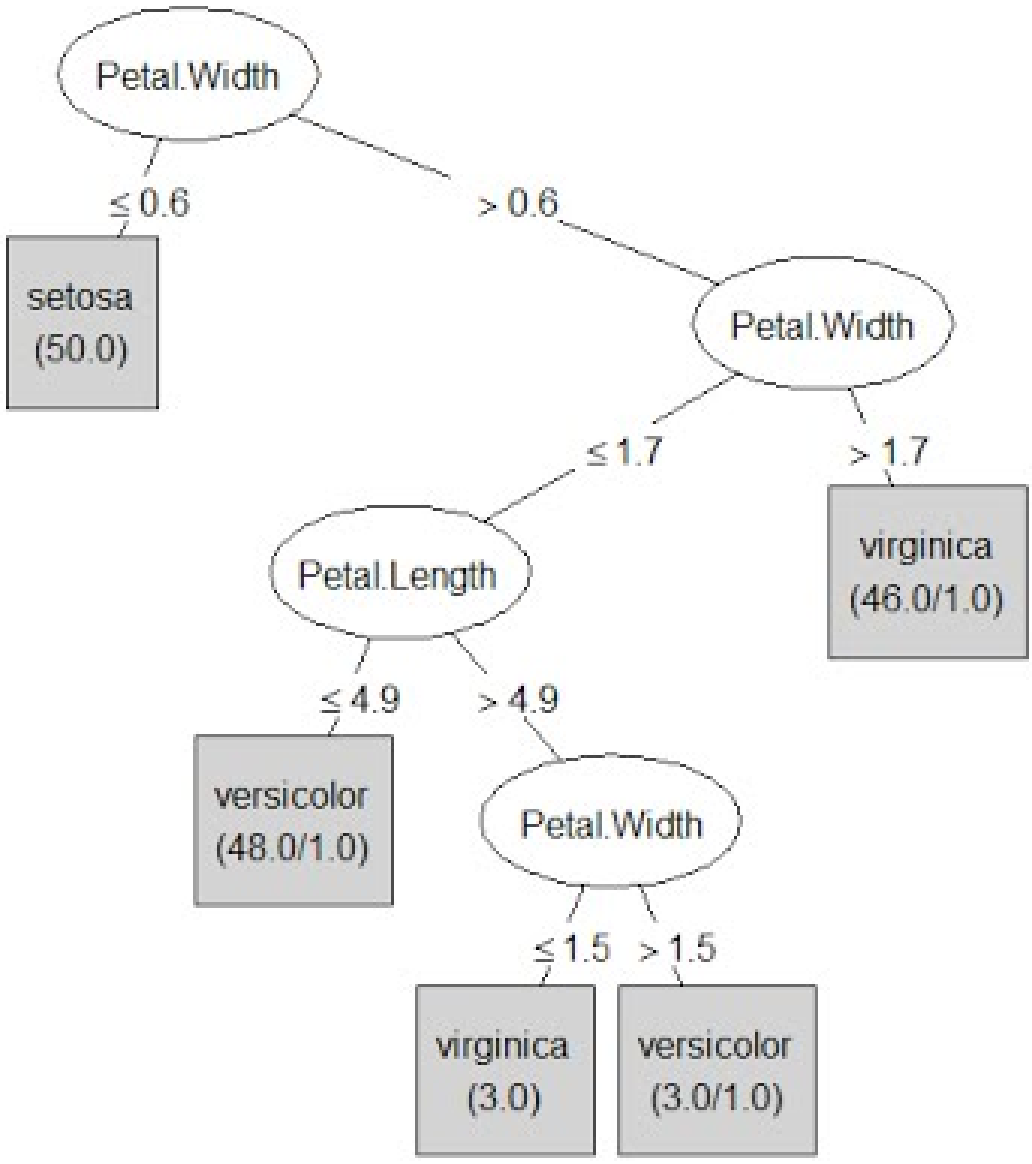
Naive Bayes là một trong các thuật toán được dùng rất phổ biến trong việc phát hiện và lọc rác cho email. Nó là một trong những kỹ thuật cơ bản và lâu đời nhất trong lĩnh vực này. Do đó, đề tài nãy cũng sẽ tìm hiểu và áp dụng Naive Bayes vào bài toán lọc rác tin tức.

2.7 Thuật toán J48

2.7.1 Giới thiệu

Thuật toán J48(C4.5) là một thuật toán sử dụng cây quyết định (decision tree) cho việc phân lớp. Thuật toán sẽ tạo ra môn cây nhị phân. Bằng cách sử dụng cây quyết định, cách tiếp cận này cũng thường đước sử dụng trong bài toán phân lớp. Cây quyết định sẽ được xây dựng để mô hình hóa quá trình phân lớp.

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật (series of rules). Các thuộc tính của đối tượng (ngoại trừ thuộc tính phân lớp – Category attribute) có thể thuộc các kiểu dữ liệu khác nhau (Binary, Nominal, ordinal, quantitative values) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal. Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các đối tượng chưa biết (unseen data).



Hình 2.2: Ví dụ về cây quyết định. Thực hiện phân lớp các loài hoa: setosa, versicolor, virginica; dựa trên độ dài và độ rộng của cánh hoa

2.7.2 Ưu điểm, hạn chế

Ưu điểm:

- Có khả năng chọn ra các đặc tính có tính phân biệt sâu sắc nhất.
- Phân loại được dữ liệu mà không cần thực hiện quá nhiều thao tác tính toán.
- Xử lý tốt dữ liệu nhiễu và không hoàn chỉnh.

Nhược điểm:

- Tỉ lệ lỗi phân lớp cao khi tập dữ liệu huấn luyện có kích thước nhỏ so với số lượng lớp.
- Tốc độ huấn luyện tăng nhanh chóng khi kích thước tập dữ liệu và số lượng các đặc tính tăng.

2.8 Các độ đo đánh giá các thuật toán phân lớp

2.8.1 Precision và Recall

Trong lĩnh vực nhận diện mẫu, thu thập thông tin và phân lớp nhị phân, precision là tỉ lệ giữa số lượng các đối tượng cần quan tâm đối với tổng các số lượng trả về, trong khi đó recall là tỉ lệ số lượng các đối tượng cần quan tâm trả về đối với tổng số các đối tượng cần quan tâm.

Đối với các tác vụ phân lớp, precision của một lớp dữ liệu là số lượng các đối tượng true positive (TP) (số đối tượng được phân lớp đúng vào lớp positive) chia cho tổng số đối tượng được phân lớp là positive (tổng của cả true positive và positive - các đối tượng phân lớp sai là positive). Trong ngữ cảnh này, recall được xác định bằng tổng số đối tượng true positive chia cho tổng số đối tượng thuộc về lớp positive. Giá trị precision tuyệt đối 1.0 của lớp C có nghĩa là tất cả các đối tượng được phân vào lớp C đúng là thuộc về lớp C (nhưng không nói gì về số lượng các đối tượng thuộc về lớp C bị phân sai vào các lớp khác), trong khi đó, giá trị recall tuyệt đối 1.0 có nghĩa là tất các đối tượng thuộc lớp C đều được phân vào lớp C (nhưng không nói gì về số lượng các dối tượng thuộc về các lớp khác cũng được phân sai vào lớp C).

Trong các tác vụ phân lớp, ta sử dụng các khái niệm *true positve (TP), false positive (FP), true negative (TN), false negative (FN)* để so sánh kết quả phân lớp. Các từ *positive* và *negative* dùng để thể hiện kết quả dự đoán của bộ phân lớp (hay còn gọi là *expectation*). Các từ *true* và *false* được dùng để chỉ kết quả đánh giá của chúng ta từ bên ngoài về kết quả phân lớp (hay còn gọi là *observation*).

Khi đó, precision và recall được xác định bằng:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

2.8.2 F-Measure

F-measure là một phương thức kết hợp giá trị của precision và recall bằng cách lấy giá trị trung bình điều hòa (harmonic mean) của chúng:

$$F = 2 \times \frac{precision \times recall}{precision \times recall}$$

Giá trị của F-measure thể hiện mức độ cân bằng của hai giá trị precision và recall, và gần bằng giá trị trung bình của chúng khi chúng xấp xỉ nhau.

2.8.3 Receiver operating characteristic (ROC) area

Trong xác suất, đường cong ROC là một dạng đồ thị thể hiện khả năng phân tích của một bộ phân lớp nhị phân khi có đa dạng ngưỡng phân biệt.

Đường cong ROC được tạo ra bằng việc bằng cách vẽ biểu đồ dựa trên tỉ lệ true positive (recall) và true negative (specifity). Và diện tich nằm dưới đường cong ROC (ROC area) bằng với xác suất một bộ phân lớp sẽ đánh giá một đối tượng positive được chọn ngẫu nhiên cao hơn một đối tượng negative được chọn ngẫu nhiên (ngầm đình là postive xếp hạng cao hơn negative)[15].

2.9 Kết chương

Chương này đã đề cập đến các nghiên cứu liên quan đến bài toán lọc rác tin tức, các thuật toán được sử dụng trong đề tài, và các phương pháp để đánh giá các thuật toán đó.

Chương 3

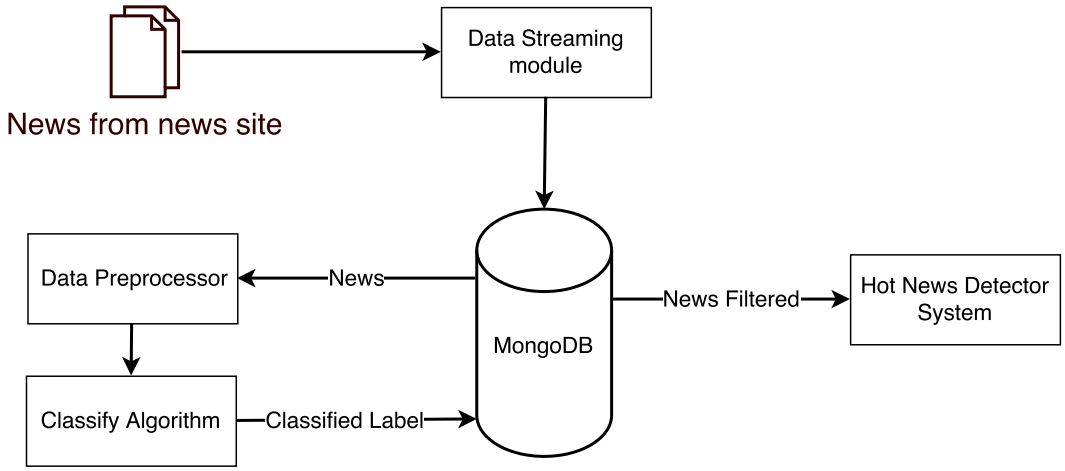
HIỆN THỰC HỆ THỐNG LỌC RÁC TIN TỨC

3.1 Mở đầu

Chương này sẽ trình bày về thiết kế, chi tiết cài đặt, các thư viện và framework được sử dụng để xây dựng hệ thống lọc rác tin tức.

3.2 Mô hình hệ thống

Dưới đây là mô hình của các thành phần chính hệ thống:



Hình 3.1: Các thành phần chính của hệ thống

Hệ thống chủ yếu phục vụ cho biên tập viên. Biên tập viên sử dụng hệ thống thông qua giao diện Spam Filtering để sử dụng chức năng phân lớp tin tức hoặc xem các số liệu thống kê về tin tức đã phân lớp lưu trong cơ sở dữ liệu.

3.3 Phân hệ thu thập dữ liệu (Data Streaming)

Module này sử dụng hệ thống Crawler để lấy các bài đăng từ nhiều nguồn tin tức lưu trữ ở cơ sở dữ liệu MySQL và đưa qua cơ sở dữ liệu MongoDB, hệ thống sẽ thực hiện các chức năng:

- Connect vào cơ sở dữ liệu mySQL để lấy các bài bài báo.
- Xóa các bài viết trùng.
- Lưu trữ xuống cơ sở dữ liệu MongoDB

3.4 Phân hệ tiền xử lý dữ liệu (Data Preprocessor)

Module tiền xử lý có nhiệm vụ chính gồm loại bỏ URL, thực hiện tách từ, loại bỏ stopwords và biểu diễn dữ liệu thành vector trọng số tf-idf. Trước khi chạy thuật toán phân loại, dữ liệu được lấy ra từ MongoDB và tiền xử lý phần nội dung của bài báo bằng các bước sau:

1. Loại bỏ URL bằng regular expression.
2. Tách từ sử dụng thư viện TPSegmenter. Bước này dùng để biểu diễn các từ ghép trong tiếng Việt bằng cách thêm gạch nối giữa các tiếng của từ.
   Ví dụ: *"Vụ tai nạn 13 người chết: Đã giám định mẫu máu tài xế xe tải"* qua bộ tách từ sẽ thành *"Vụ tai_nạn 13 người chết : Đã giám_định mẫu máu tài_xế xe_tải"*.
3. Loại bỏ các từ trong danh sách gồm 813 stopwords.
4. Tính và biểu diễn dữ liệu thành vector tf-idf và metadata.

3.5 Phân hệ phân loại tin tức

Sử dụng model đã train trước đó để phân loại tin tức. Model sử dụng thuật toán SVM đã trình bày ở chương 2 làm thuật toán chính để phân loại tin tức. Hiện tại hệ thống có 2 nhãn đã được định nghĩa trước đó:

- *K*hông rác
- *R*ác

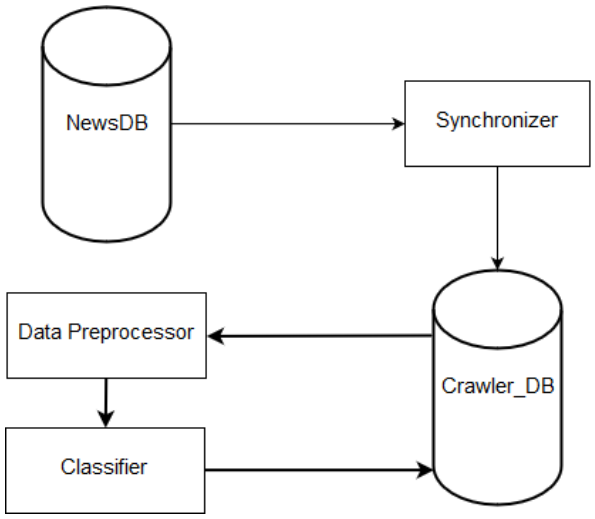
Sau khi đã phân loại, hệ thống sẽ lưu nhãn đã gán cho tin tức xuống cơ sở dữ liệu MongoDB.

3.6 Phân hệ phân loại loại rác

Hệ thống sẽ lấy những tin được gán nhãn "Rác" ở bước phân loại tin tức ở trên để phân loại loại rác cho tin đó. Hiện tại có 3 loại rác đã được định nghĩa trước:

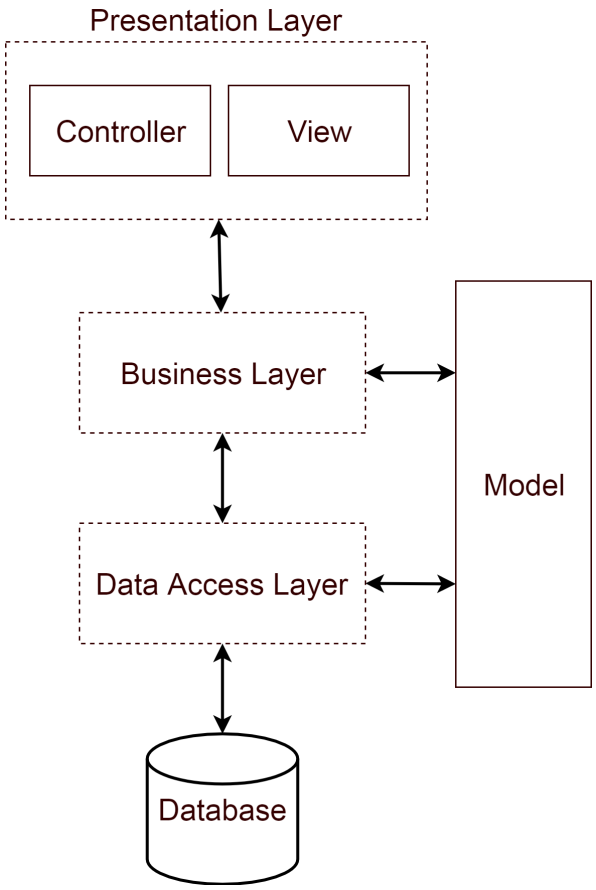
- *Q*uảng cáo
- *T*uyển dụng
- *C*hia sẻ

3.7 Thiết kế hệ thống



Hình 3.2: Kiến trúc hệ thống crawler tin tức

Hệ thống Crawler được xây dựng độc lập bao gồm phân hệ thu thập dữ liệu, phân hệ tiền xử lý và phân hệ phân lớp để chạy ngầm nhằm thu thập và phân loại thông tin



Hình 3.3: Kiến trúc hệ thống kết hợp Multilayer architecture kết hợp với mô hình MVC

Hệ thống xây dựng theo kiến trúc 3 tầng gồm: Presentation Layer, Business Logic Layer và Data Access Layer. Trong đó, Presentation Layer áp dụng mô hình Model - View - Controller (MVC). Cụ thể:

- Presentation Layer: Có trách nhiệm hiển thị thông tin, tương tác với người dùng hệ thống. Gồm 2 thành phần:
  - Controller: Điều khiển các luồng của hệ thống web, nhận các tín hiệu từ người dùng và xử lý tương ứng.
  - View: Có nhiệm vụ hiển thị các giao diện hệ thống cho người dùng, hệ thống sử dụng React để gọi api trả về dữ liệu cho giao diện người dùng.
- Model: Đối tượng chứa dữ liệu để xử lý và hiển thị.
- Business Layer: Chứa các nghiệp vụ của hệ thống. Bao gồm các bước xử lý dữ liệu, thuật toán gom cụm, các tác vụ thống kê,...

- Data Access Layer: Có nhiệm vụ giao tiếp với các hệ cơ sở dữ liệu.

3.8 Cài đặt hệ thống

Hệ thống ứng dụng được xây dựng trên nền tảng Java EE với các thành phần sau:

- Ngôn ngữ: Java, HTML, CSS, JavaScript, React.
- Hệ cơ sở dữ liệu: MongoDB và MySQL.
- Thư viện, framework: Apache Struts 2, Apache Lucene, TPSegmenter, Weka
- Server: Apache Tomcat

3.8.1 Các package

Source code chương trình được tổ chức thành các package như sau:

Hệ thống crawler:

- vn.vccorp.crawler.bo: Chứa các business object của hệ thống
- vn.vccorp.crawler.config: Các file config cho hệ thống
- vn.vccorp.crawler.constant: Các file constant của hệ thống
- vn.vccorp.crawler.dao: Data Access Object, thực hiện các tác vụ đọc, ghi database
- vn.vccorp.crawler.dbconnection: Cung cấp kết nối đến database
- vn.vccorp.crawler.dto: Data Transfer Object, các đối tượng để vận chuyển dữ liệu từ database
- vn.vccorp.crawler.main: Các lớp bao đóng để tạo thread chạy song song các tác vụ
- vn.vccorp.crawler.thread: Các lớp bao đóng để tạo thread chạy song song các tác vụ
- vn.vccorp.crawler.util: Các công cụ hỗ trợ trong hệ thống

Hệ thống HotNewsDetector:

- vn.vccorp.hotnewsdetector.action.general: lớp ảo chứa các phương thức của action.
- vn.vccorp.hotnewsdetector.action.news: chứa các action cho tin tức
- vn.vccorp.hotnewsdetector.action.twitter: chứa các action cho twitter
- vn.vccorp.hotnewsdetector.bo: Chứa các business object của hệ thống
- vn.vccorp.hotnewsdetector.config: Các file config cho hệ thống
- vn.vccorp.hotnewsdetector.constant: Các file constant của hệ thống
- vn.vccorp.hotnewsdetector.context: chứa các action context của hệ thống
- vn.vccorp.hotnewsdetector.crawler.news: Dùng để lấy dữ liệu từ trang web tin tức
- vn.vccorp.hotnewsdetector.dao: Data Access Object, thực hiện các tác vụ đọc, ghi database
- vn.vccorp.hotnewsdetector.dbconnection: Cung cấp kết nối đến database
- vn.vccorp.hotnewsdetector.dto: Data Transfer Object, các đối tượng để vận chuyển dữ liệu từ database
- vn.vccorp.hotnewsdetector.exception: Quản lý lỗi và đưa ra thông báo của hệ thống
- vn.vccorp.hotnewsdetector.thread
- vn.vccorp.hotnewsdetector.utils

3.8.2 Cơ sở dữ liệu MongoDB

Hệ thống sử dụng MongoDB để lưu trữ dữ liệu tin tức và quản lý kết quả phân lớp. Đây là một hệ cơ sở dữ liệu NoSQL, cung cấp khả năng mở rộng, sao lưu, phân mảnh dữ liệu tốt, và có thể thay đổi cấu trúc dữ liệu một cách linh hoạt.

Dưới đây là bảng so sánh một số thuật ngữ cơ bản giữa các cơ sở dữ liệu SQL truyền thống và MongoDB:

Thuật ngữ SQL	Thuật ngữ MongoDB
database	database
table	collection
row	document hoặc BSON document
column	field
index	index
table joins	$lookup, embedded documents

Bảng 3.1: So sánh các thuật ngữ giữa SQL và MongoDB

3.8.2.1 Collection News

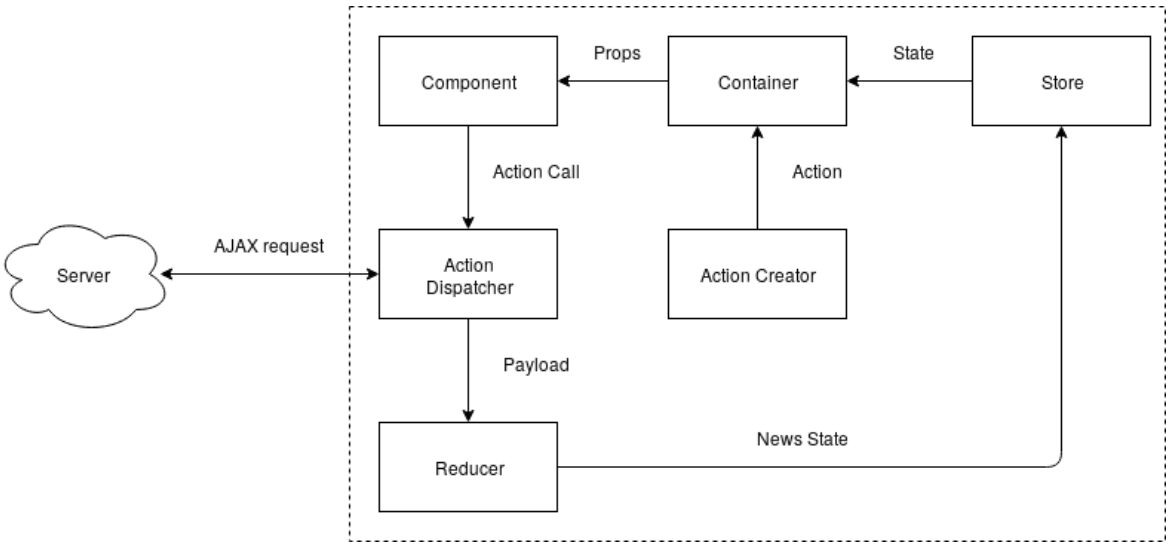
Collection này chứa thông tin về các tin lấy từ cơ sở dữ liệu MySQL, cũng là nơi lưu trữ tất cả thông tin về tweet trong hệ thống. Khi dữ liệu được stream từ MySQL về, mỗi tin chỉ có 12 trường, các trường khác sẽ được thêm vào trong quá trình hệ thống xử lý.

Thuộc tính	Loại	Ý nghĩa
_id	ObjectID	ID trong MongoDB của document
Id	Int	ID của tin tức lấy về
Title	String	Title của tin tức
Content	String	Nội dung của tin tức
Source	String	Nguồn trang báo điện tử của tin tức
CreateTime	Date	Thời gian đăng của tin
GetTime	Date	Thời gian tin tức được thu thập vào cơ sở dữ liệu MySQL
CollectDate	Date	Thời gian tweet được thu thập vào cơ sở dữ liệu MongoDB
Author	String	Người viết bài báo(nếu có)
Category	Integer	Chủ đề của bài báo
SpamLabel	String	Nhãn đã gán cho tin tức
SpamCategory	String	Nhãn đã gán cho loại tin rác
SpamLabelFeedback	Integer	Feedback của nhãn đã gán cho tin tức đó

Bảng 3.2: Các trường của collection News

3.9 Giao diện

Giao diện hệ thống được xây dựng sử dụng thư viện React, với kiến trúc Redux. Giao diện tương tác với hệ thống thông qua ajax request.



Hình 3.4: Kiến trúc giao diện hệ thống

3.9.1 Giao diện danh sách tin đã phân lớp

Tên	Loại	Mô tả
fetchSpam	Action	Truy xuất danh sách các tin đã phân lớp
sendFeedback	Action	Gửi phản hồi người dùng về label của tin đã phân lớp
spamList	State	Danh sách các tin đã phân lớp

Bảng 3.3: Bảng các thuộc tính

Tên	Loại	Mô tả
Date	Date Picker	Chọn ngày để lấy danh sách tin đã phân lớp
SpamTable	Table	Hiển thị danh sách các tin đã phân lớp
FeedbackBox	Select	Chọn và gửi phản hồi về label của tin đã phân lớp

Bảng 3.4: Bảng các đối tượng hiển thị

3.9.2 Giao diện phân lớp tin

Tên	Loại	Mô tả
fetchLabels	Action	Gửi danh sách các tin cần phân lớp và lấy về các label tương ứng
spamLabels	State	Danh sách các tin và label tương ứng

Bảng 3.5: Bảng các thuộc tính

Tên	Loại	Mô tả
InputTable	Table	Bảng chứa các tin cần hoặc đã phân lớp
DatatypeMenu	Tabs	Chuyển kiểu dữ liệu của input: text hoặc URL
ClassifyButton	Button	Thực hiện phân lớn các input
InputButton	Button	Hiển thị input form
InputForm	Form, Popup	Form để thêm input mới
ContentText	Input	Nội dung input
ExpectedLabel	Select	Label mong đợi của input
ConfirmButton	Button	Thêm input mới vào table của tab hiện tại
ImportButton	Button	Truyền dữ liệu input từ file
ClearAllButton	Button	Xóa hết dữ liệu input từ table của tab hiện tại
DeleteButton	Button	Xóa một dòng input trên bảng

Bảng 3.6: Bảng các đối tượng hiển thị

3.9.3 Giao diện thống kê dữ liệu phân lớp

Tên	Loại	Mô tả
fetchSpamStatistics	Action	Truy xuất dữ liệu thống kê phân lớp
spamStatistics	State	Dữ liệu thống kê phân lớp

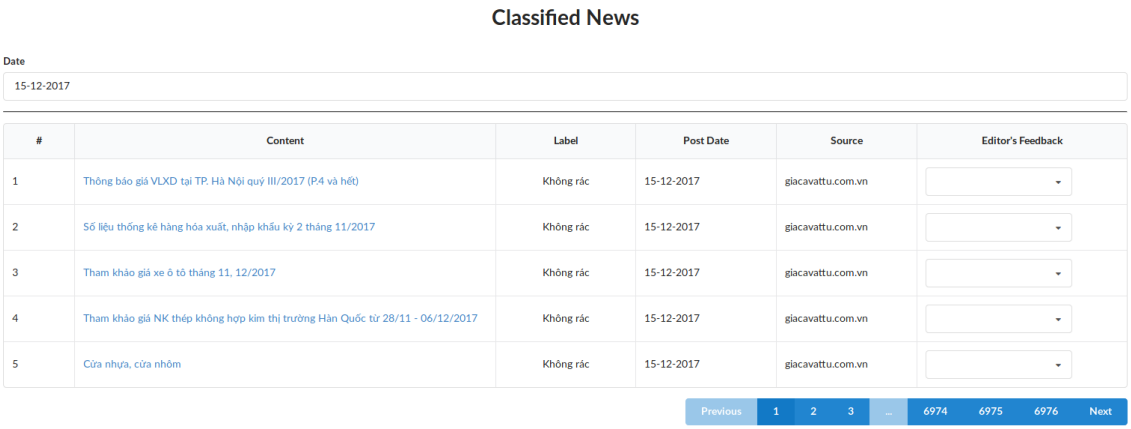
Bảng 3.7: Bảng các thuộc tính

Tên	Loại	Mô tả
DateRange	Select	Chọn khung thời gian để lấy dữ liệu thống kê
ChartsSlider	Slider	Danh sách các đồ thị hiển thị dữ liệu thống kê

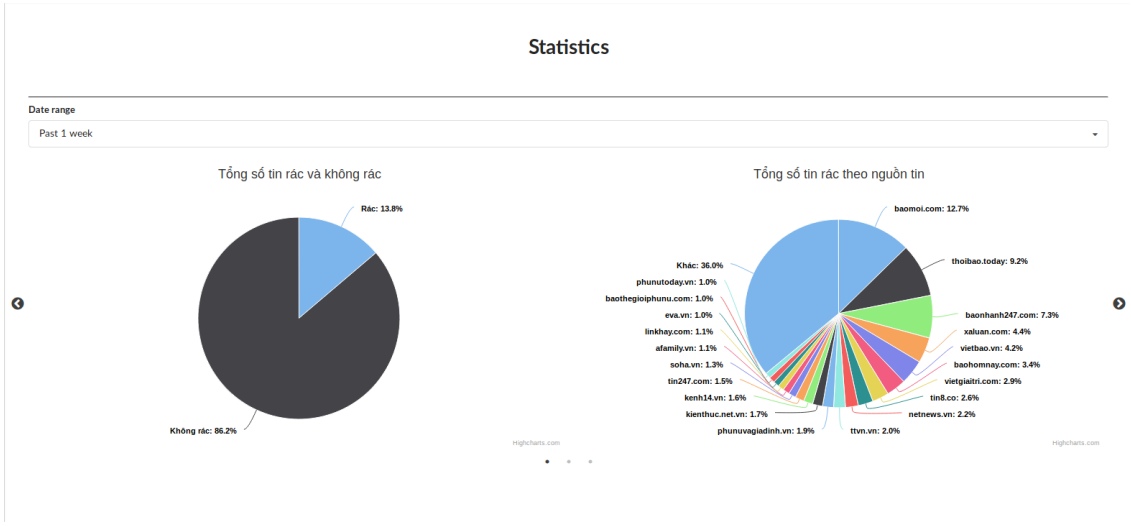
Bảng 3.8: Bảng các đối tượng hiển thị

3.10 Kết quả

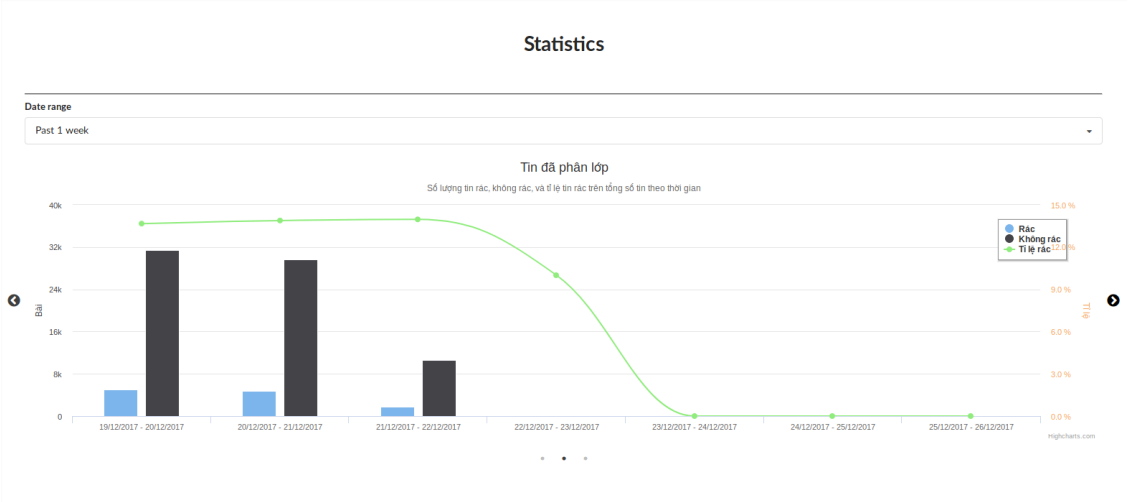
Hệ thống đã hoàn thiện các chức năng: lọc rác tin tức trên Crawler, hiển thị danh sách tin đã qua phân lớp bằng bộ lọc rác, bộ phân lớp lọc rác thủ công, thống kê dữ liệu đã phân lớp. Dưới đây là một số hình ảnh của hệ thống:



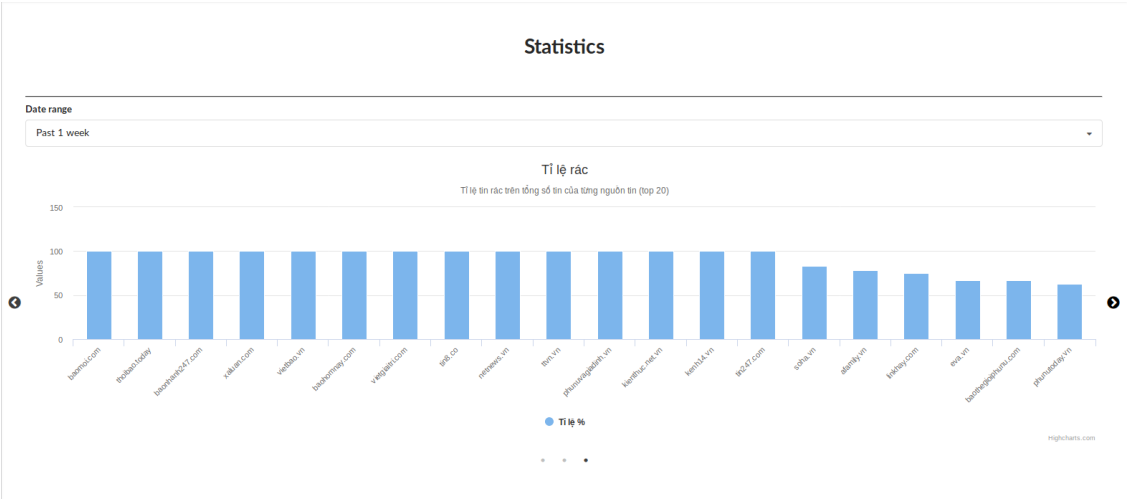
Hình 3.5: Giao diện danh sách tin đã phân lớp. Người dùng có thể chọn ngày để hiển thị tin và gửi phản hồi về nhãn của tin.



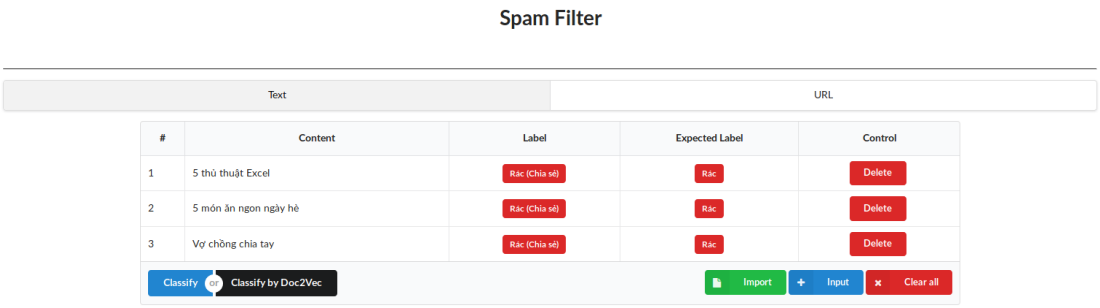
Hình 3.6: Giao diện thống kê biểu đồ với hai biểu đồ. Biểu đồ thứ nhất thể hiện tổng số tin đã phân lớp phân theo nhãn "Rác" và "Không rác". Biểu đồ thứ hai thể hiện số lượng và tỉ lệ số lượng tin rác của mỗi nguồn tin so với tổng số tin rác và số tin rác của các nguồn tin khác (biểu đồ chỉ hiển thị top 20, nguồn tin có số lượng tin rác nhiều nhất)



Hình 3.7: Giao diện thống kê. Biểu đồ thể hiện số lượng tin "Rác" và "Không rác" và tỉ lệ tin "Rác" so với tổng số tin theo thời gian. Người dùng có thể chọn hiển thị số liệu thống kê trong hôm nay, 7 ngày trước, 30 ngày trước, hoặc 12 tháng trước



Hình 3.8: Giao diện thống kê. Biểu đồ thể hiện tỉ lệ tin rác của một nguồn tin trên tổng số các tin của nguồn tin đó. Biểu đồ chỉ hiển thị top 20 nguồn tin với tỉ lệ lớn nhất và khác 0



Hình 3.9: Giao diện bộ phân lớp tin tức. Người dùng có thể thực hiện phân lớp một cách thủ công bằng cách cho cho vào input là văn bản hoặc các đường dẫn đến các bài báo cần phân lớp, kèm theo là nhãn mà người dùng gán cho tin đó. Input có thể được truyền vào bằng cách nhập từ bàn phím hoặc truyền vào từ một file. Sao khi đã hoàn thành việc phân lớp, hệ thống sẽ trả về các nhãn tương ứng kèm theo độ chính xác của việc phân lớp và confusion matrix

3.11 Kết chương

Chương này đã trình bày về các thành phần chính hệ thống, kiến trúc phân tầng, các hệ cơ sở dữ liệu được sử dụng và cách tổ chức, cùng một số kết quả cài đặt hệ thống.

Chương 4

THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Mở đầu

Mục đích của chương này là trình bày kết quả huấn luyện các thuật toán phân lớp trên bộ dữ liệu thu thập được. Qua đó đánh giá, nhận định và so sánh các thuật toán phân lớp.

4.2 Tổng quan về bộ dữ liệu

Bộ dữ liệu gồm các bài viết từ các trang báo điện tử Việt Nam. Dữ liệu được lấy về từ cơ sở dữ liệu tin tức của công ty VCCorp. Dữ liệu thu thập được được sàn lọc thông qua một tập các từ khóa đã được định nghĩa trước.

Mỗi mẫu tin bao gồm các thông tin như sau: tựa đề tin, nội dung tin, mô tả tin, ngày đăng tin, và nguồn tin.

Bộ dữ liệu huấn luyện gồm 15,612 tin tiếng Việt, được thu thập và sàn lọc trong khoảng thời gian từ.

Dữ liệu thống kê cho tập dữ liệu sử dụng để train các model:

Dữ liệu gán nhãn	Số lượng
Không rác	7331
Rác	8281
Tổng	15612

Bảng 4.1: Thống kê dữ liệu gán nhãn

Dữ liệu gán nhãn	Số lượng
Quảng cáo	4627
Chia sẻ	3279
Tuyển dụng	375
Tổng	8281

Bảng 4.2: Thống kê loại rác

Chia sẻ	thủ thuật, mẹo, tình cảm, chia tay, món ăn, ngon, tâm sự
Tuyển dụng	việc làm, tuyển dụng, cộng tác viên, lao động
Quảng cáo	giảm giá, khuyến mãi, trúng thưởng

Bảng 4.3: Danh sách từ khóa để thu thập dữ liệu

Bộ dữ liệu có thể được tải về từ địa chỉ: https://drive.google.com/open?id=1xZVBcaVtZAmQ4xU0KKPvB5ZRAUoKc2_v

4.3 Thiết lập thực nghiệm, cách đánh giá

Sử dụng 3 thuật toán phân lớp: thuật toán Naive Bayes, thuật toán J48, thuật toán Support Vector Machine.

Ta đánh giá và so sánh kết quả về thời gian xử lý và chất lượng phân lớp thông qua một số độ đo đã trình bày ở mục **??**: Precision, Recall, F-Measure, ROC Area, confusion matrix.

4.4 Kết quả thực nghiệm

4.4.1 Kết quả train model phân loại tin tức

Dưới đây là kết quả các thuật toán và kết quả của một số độ đo :

4.4.1.1 Kết quả dựa trên nội dung của tin của tập mẫu

Dữ liệu gán nhãn	Số lượng
Mẫu dữ liệu	15612
Số chiều	111363
Cross-validation:	10

Bảng 4.4: Thông số cơ bản của tập train

Time build model(seconds):		
SVM	NaiveBayes	J48
382.85	853.67	43530.21

Bảng 4.5: Thời gian train model

Class	Precision			Recall			F-Measure			ROC Area		
	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes
Rác	0.845	0.876	0.847	0.838	0.915	0.874	0.842	0.895	0.860	0.823	0.884	0.854
Tin	0.819	0.898	0.852	0.827	0.854	0.821	0.823	0.876	0.836	0.823	0.884	0.871
	0.832	0.887	0.849	0.833	0.886	0.849	0.832	0.886	0.849	0.823	0.884	0.862

Bảng 4.6: Kết quả train model dựa trên một số độ đo

4.4.1.2 Kết quả dựa trên tiêu đề của tin của tập mẫu

Dưới đây là kết quả các thuật toán và kết quả của một số độ đo :

Dữ liệu gán nhãn	Số lượng
Mẫu dữ liệu	15612
Số chiều	15133
Cross-validation:	10

Bảng 4.7: Thông số cơ bản của tập train

Time build model(seconds):		
SVM	NaiveBayes	J48
66.23	130.78	5895.41

Bảng 4.8: Thời gian train model

Class	Precision			Recall			F-Measure			ROC Area		
	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes
Rác	0.845	0.876	0.847	0.838	0.915	0.874	0.842	0.895	0.860	0.823	0.884	0.854
Tin	0.819	0.898	0.852	0.827	0.854	0.821	0.823	0.876	0.836	0.823	0.884	0.871
	0.832	0.887	0.849	0.833	0.886	0.849	0.832	0.886	0.849	0.823	0.884	0.862

Bảng 4.9: Kết quả train model dựa trên một số độ đo

4.4.2 Kết quả train model phân loại loại tin rác

Dữ liệu gán nhãn	Số lượng
Mẫu dữ liệu	8281
Số chiều	9526
Cross-validation:	10

Bảng 4.10: Thông số cơ bản của tập train

Time build model(seconds):		
SVM	NaiveBayes	J48
14.46	19.39	590.78

Bảng 4.11: Thời gian train model

Class	Precision			Recall			F-Measure			ROC Area		
	NaiveBayes	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes	J48	SVM	NaiveBayes	J48	SVM
Chia sẻ	0.97	0.952	0.96	0.730	0.985	0.986	0.833	0.968	0.973	0.973	0.978	0.980
Quảng cáo	0.931	0.990	0.99	0.945	0.968	0.975	0.938	0.979	0.982	0.962	0.980	0.981
Tuyển dụng	0.335	0.989	1	0.992	0.952	0.955	0.5	0.970	0.977	0.954	0.987	0.977
	0.919	0.975	0.979	0.862	0.974	0.979	0.877	0.974	0.979	0.966	0.980	0.981

Bảng 4.12: Kết quả train model dựa trên một số độ đo

4.5 Nhận xét

4.5.1 Nhận định về các thuật toán phân lớp cho bài toán

Dựa vào bảng 4.6, 4.9 và 4.12, ta thấy rằng với độ đo và tập dữ liệu mẫu, thuật toán Support Vector Machine cho kết quả tốt với mọi độ đo so với thuật toán Naive Bayes và J48.

Dựa vào bảng 4.5, 4.8 và 4.11, ta thấy rằng thời gian để train model cho thuật toán Support Vector Machine tốn ít thời gian hơn thuật toán Naive Bayes và J48.

4.6 Kết chương

Qua các kết quả thử nghiệm, chương này đã thể hiện được một số tính chất của các thuật toán được đánh giá như thời gian train, một số độ đo thường dùng để đánh giá một hệ phân lớp. Ta thấy thuật toán SVM có thời gian train nhanh trong khi vẫn giữ được độ chính xác tốt so với thuật toán Naive Bayes và J48. Riêng thuật toán J48 có hạn chế về thời gian train quá lớn so với hai thuật toán còn lại.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả đạt được

Mục tiêu chính của đề tài là xâu dụng một hệ thống lọc rác tin tức để cải thiện tính hiệu quả cửa hệ thống phát hiện tin nóng. Một số nội dung thực hiện được đề ra ban đầu là: tìm hiểu bài toán liên quan, thu thập dữ liệu, cài đặt và đánh giá một số thuật toán, và xây dựng hệ thống.

Sau quá trình nghiên cứu và thực hiện, khóa luận đã thu được một số kết quả sau:

- Kiến thức:
  - Tìm hiểu về bài toán phân lớp văn bản, phát hiện rác tin tức.
- Sản phẩm:
  - Thu thập bộ dữ liệu từ cơ sở dữ liệu của công ty VCCorp.
  - Khảo sát và đánh giá các các thuật toán phân lớp.
  - Xây dựng được hệ thống có khả năng phát hiện rác dữ liệu tin tức đang được triển khai tại công ty VCCorp.

Hướng phát triển

Tuy hệ thống đạt được kết quả tương đối khá tốt trong việc phát hiện rác cho dữ liệu tin tức, độ chính xác vẫn còn có thể cải thiện thêm nữa bằng việc cải thiện mô hình thông qua phản hồi người dùng và sử dụng các phương pháp, thuật toán khác.

Trong tương lai, hệ thống sẽ được phát triển thêm chức năng tự động cập nhật mô hình dựa trên phản hồi người dùng, và cho phép người dùng quản lý các mô hình đã huấn luyện. Hệ thống cũng có thể mở rộng để lọc rác cho các nguồn tin khác như mạng xã hội.

TÀI LIỆU THAM KHẢO

[1] An K. Le. Vietnamese hot news detection from twitter. chapter The Hot News Detection Problem and Approaches, pages 6–26. Ho Chi Minh, Vietnam, 2017. 6

[2] Niklas Zechner. The past, present and future of text classification. Intelligence and Security Informatics Conference (EISIC), 2013 European. IEEE, 2013. 7

[3] Mehran Sahami, Susan T. Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. 1998. 8

[4] A.C. Rothwell, L.D. Jagger, W.R. Dennis, and D.R. Clarke. Intelligent spam detection system using an updateable neural analysis engine, July 27 2004. US Patent 6,769,016. 8

[5] Ismaila Idris, Ali Selamat, Ngoc Thanh Nguyen, Sigeru Omatu, Ondrej Krejcar, Kamil Kuca, and Marek Penhaker. A combined negative selection algorithm–particle swarm optimization for an email spam detection system. *Engineering Applications of Artificial Intelligence*, 39(Supplement C):33 – 44, 2015. 8

[6] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 576–587. VLDB Endowment, 2004. 9

[7] András Benczúr, Károly Csalogány, Tamás Sarlós, and Máté Uher. Spamrank - fully automatic link spam detection work in progress. 12 2017. 9

[8] Xin Jin, Cindy Xide Lin, Jiebo Luo, and Jiawei Han. Socialspamguard: A data mining-based spam detection system for social media networks. 4:1458–1461, 08 2011. 9

[9] A. H. Wang. Don't follow me: Spam detection in twitter. In *2010 International Conference on Security and Cryptography (SECRYPT)*, pages 1–10, July 2010. 9

[10] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2011. 9

[11] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, Nov 2016. 10

[12] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 2013, 01 2013. 10

[13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. 11

[14] George Dimitoglou, James A Adams, and Carol M Jim. Comparison of the c4. 5 and a naïve bayes classifier for the prediction of lung cancer survivability. *arXiv preprint arXiv:1206.1121*, 2012. 13

[15] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006. 17

Phụ lục. Giới thiệu về thư viện React và kiến trúc Redux

Giới thiệu React

React (hay ReactJS) là một thư viện JavaScript được dùng để xây dựng giao diện người dùng. React được phát triển bởi Facebook, Instagram và cộng đồng các nhà phát triển và các doanh nghiệp. React cho phép các nhà phát triển xây dựng các ứng dụng web có thể thay đổi trạng thái mà không cần phải tải lại nội dung trang. Vì đặc tính này React thường được sử dụng để xây dựng các ứng dụng web đơn trang. Do bản thân React chỉ là một thư viện, nên nhà phát triển có thể tích hợp React vào các hệ thống kiến trúc MVC hoặc các framework giao diện người dùng khác như AngularJS.

Một đối tượng trong React được gọi là một component. Mỗi component có thể được tạo ra từ các component khác. Mỗi component có 2 thành phần chính cần quan tâm:

- *props*: là đối tượng chứa các tham số được tạo ra vào lúc khởi tạo component.
- state: là đối tượng chứa các tham số trạng thái của component.

React sử dụng virtual DOM để giúp quản lý DOM và

Lucene lưu trữ dữ liệu ở dạng chỉ mục ngược (inverted index), cho phép tìm kiếm các đối tượng văn bản dựa trên từ khóa một cách nhanh chóng. Một đối tượng văn bản trong Lucene được gọi là Document. Mỗi Document có một hoặc nhiều Field, ứng với các thuộc tính của Document đó. Một bài viết hay một trang web có thể là một Document, với các Field như: tiêu đề, nội dung, tác giả, ngày đăng,...

Một số class chính của thư viện:

- Analyzer và các class con: có nhiệm vụ phân tích dữ liệu văn bản thành những token/term trước khi ghi vào index. Một số class con như StandardAnalyzer, WhitespaceAnalyzer, SimpleAnalyzer, KeywordAnalyzer.
- IndexWriter: nhận luồng dữ liệu đã tokenize bằng Analyzer và ghi vào index.
- IndexReader, IndexSearcher: đọc và tìm kiếm trên index đã tạo từ trước, ngoài ra cung cấp các thông tin khác từ index như danh sách term, tần số của term trong một Document, trong toàn bộ dữ liệu.
- Query và các class con: dùng để xây dựng câu truy vấn và truyền vào IndexSearcher để thực hiện tìm kiếm. Một số class con như TermQuery, RangeQuery, Boolean Query.

Hệ thống chủ yếu sử dụng Lucene hỗ trợ trong bước tiền xử lý dữ liệu, nhằm tính và biểu diễn các bài viết ở dạng vector tf-idf, thông qua các thông tin về tần số term trong index.

Sử dụng Lucene trong Java

Tạo IndexWriter

Để ghi dữ liệu vào Lucene index, ta cần tạo IndexWriter như sau:

```
indexDirectory = FSDirectory.open(new File(indexDir));
analyzer = new StandardAnalyzer(Version.LUCENE_36,
   Collections.emptySet());
IndexWriterConfig config = new
   IndexWriterConfig(Version.LUCENE_36, analyzer);
config.setOpenMode(OpenMode.CREATE);
IndexWriter writer = new IndexWriter(indexDirectory, config);
```

Thêm một document vào index

Giả sử ta có một tweet với ID là "001", nội dung là "Tai nan kinh hoang khi xe tai mat phanh", dưới đây là cách tạo và thêm vào index document với 2 field tương ứng là "tweetID" và "tweetContent".

```
Field tweetID = new Field("tweetID", "001", Field.Store.YES,
    Field.Index.NO);
Field tweetContent = new Field("tweetContent", "Tai nan kinh hoang
    khi xe tai mat phanh", Field.Store.YES, Field.Index.ANALYZED,
    Field.TermVector.YES);
Document lucenceDocument = new Document();
lucenceDocument.add(tweetID);
lucenceDocument.add(tweetContent);
writer.addDocument(luceneDocument);
```

Đọc dữ liệu từ index

Sau khi tạo index, ta có thể đọc thông tin trong index thông qua IndexReader hoặc IndexSearcher.

```
IndexReader reader = IndexReader.open(indexDir);
IndexSearcher searcher = new IndexSearcher(reader);
```

Cách tính giá trị **idf** cho tất cả term trong field "tweetContent" trong bộ dữ liệu:

```
int docCount = reader.numDocs();
TermEnum listOfTerms = reader.terms();
TreeMap<String, Double> idfVector = new TreeMap<String, Double>();
while (listOfTerms.next()) {
   String currentTerm = listOfTerms.term().text();
   int docFreq = searcher.docFreq(new Term("tweetContent",
      currentTerm));
```

```
    double idf = 1 + Math.log((double) docCount / docFreq);
    idfVector.put(currentTerm, idf);
}
```

Cách tính vector tf-idf cho document thứ **i** trong index:

```
TermFreqVector tfv = reader.getTermFreqVector(i, "tweetContent");
String[] termList = tfv.getTerms(); //list of terms in this
    document
int[] termFreqList = tfv.getTermFrequencies();
int totalTermCount = 0;
LinkedHashMap<String, Double> tfidfVector = new
    LinkedHashMap<String, Double>();

// calculate (total) term count in document i
for (int temp : termFreqList) {
    totalTermCount += temp;
}
// loop through all term in doc i and calculate tfidf vector
int uniqueTermCount = termList.length;
for (int j = 0; j < uniqueTermCount; j++) {
    if (termFreqList[j] != 0 && idfVector.containsKey(termList[j])){
        double tfidf = ((double)termFreqList[j] / totalTermCount)
        * idfVector.get(termList[j]);
        tfidfVector.put(termList[j], tfidf);
    }
}
```