LỜI CẢM ƠN

Trước tiên, tôi xin được gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Thầy giáo, PGS. TS. Nguyễn Trí Thành đã tận tình chỉ bảo, hướng dẫn, động viên và giúp đỡ tôi trong suốt quá trình thực hiện luận văn tốt nghiệp.

Tôi xin gửi lời cảm ơn tới các thầy cô trường Đại Học Công Nghệ - Đại Học Quốc Gia Hà Nội – những người đã tận tình giúp đỡ, cổ vũ, và góp ý cho tôi trong suốt thời gian tôi học tập và nghiên cứu tại trường.

Tôi xin gửi lời cảm ơn tới các anh chị, các bạn học viên cùng học tập nghiên cứu tại Trường Đại học Công nghệ đã hỗ trợ tôi rất nhiều trong quá trình học tập cũng như thực hiện luận văn.

Cuối cùng, tôi muốn gửi lời cảm ơn tới gia đình và bạn bè, những người thân yêu luôn bên cạnh, quan tâm, động viên tôi trong suốt quá trình học tập và thực hiện luận văn tốt nghiệp này.

Tôi xin chân thành cảm ơn!

Hà Nội, tháng 05 năm 2016

Học viên

Cấn Mạnh Cường

LỜI CAM ĐOAN

Tôi xin cam đoan giải pháp ***Xử lý trùng lặp, phân loại, xác định từ khóa quan trọng và sinh tóm tắt cho văn bản trong một hệ thống thu thập tin tức tự động*** được trình bày trong luận văn này do tôi thực hiện dưới sự hướng dẫn của PGS. TS. Nguyễn Trí Thành.

Tôi đã trích dẫn đầy đủ các tài liệu tham khảo, công trình nghiên cứu liên quan ở trong nước và quốc tế. Tất cả những tham khảo từ các nghiên cứu liên quan đều được nêu nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong luận văn.

Hà Nội, tháng 5 năm 2016

Tác giả luận văn

Cấn Mạnh Cường

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

Số thứ tự	*Ký hiệu, viết tắt*	*Chú giải*
1	Crawler	Trình thu thập nội dung trang web
2	WebBrowser	Trình duyệt web
3	HTTP	Giao thức truyền tải siêu văn bản
4	URL	Địa chỉ liên kết của trang web
5	Seed URL	Tập hợp các URL hạt nhân xuất phát của Crawler
6	Frontier	Kho chứa các URL chưa được thăm
7	Finger print	Dấu vân, đại diện cho tài liệu độc lập
8	Front End	Phần xử lý giao diện tương tác với người dùng
9	ID	Định danh của 1 tài liệu
10	IP	Giao thức kết nối Internet
11	Hashing	Băm tài liệu
12	Search Engine	Máy tìm kiếm
13	SEO	Tối ưu hóa trang web hỗ trợ máy tìm kiếm
14	TF	Tần số từ
15	IDF	Tần số tài liệu đảo ngược

DANH MỤC CÁC HÌNH

DANH MỤC CÁC BẢNG

MỞ ĐẦU

Báo điện tử đã không còn là khái niệm xa lạ với mỗi chúng ta, nó đang dần thay thế các hình thức phát hành báo, tạp chí truyền thống bởi các đặc điểm ưu việt như: tính thời sự - khả năng cập nhật trực tiếp, khả năng truyền tải đa phương tiện, khả năng lưu trữ và tìm kiếm thông tin, khả năng tương tác với người dùng cao, báo điện tử đã khắc phục những hạn chế của các loại hình báo chí truyền thống để trở thành loại hình báo chí ưu việt trong thời điểm hiện nay.

Tính đến ngày 25/12/2014, cả nước có 838 cơ quan báo chí in với 1.111 ấn phẩm báo chí (trong đó các cơ quan Trung ương có 86 báo in và 507 tạp chí; địa phương có 113 báo in và 132 tạp chí); 90 báo và tạp chí điện tử, 215 trang tin điện tử tổng hợp của các cơ quan báo chí. Số báo và tạp chí điện tử đã tăng gấp gần 1.5 lần so với con số 62 báo điện tử vào năm 2012 [1]. Cũng theo thống kê của một trang tổng hợp thông tin điện tử lớn là Baomoi.com[1] trong 3 tháng từ tháng 12/2015 đến tháng 2/2016, về số lượng tin bài trên báo, tạp chí điện tử, trang thông tin điện tử thì:

Bảng 0.1 Thống kê số lượng tin tức báo mới 3 tháng đầu 2016

Tổng số tin	583827
Tổng số tin đăng lại	137823
Tổng số tin gốc bị đăng lại	123805
Tổng số tin gốc không bị đăng lại	446004

Với lượng thông tin khổng lồ từ hơn 300 trang báo và tin điện tử như hiện nay thì việc tổng hợp chọn lọc một cách thủ công để mang lại nguồn thông tin hữu ích dường như là một điều không thể, việc thu thập thông tin tự động để xây dựng một hệ thống đọc tin tự động thông minh bằng máy tính không còn là chủ đề mới, xong việc cải tiến, ứng dụng các công nghệ mới vào hệ thống để hệ thống vận hành tốt trong bối cảnh dữ liệu lớn dần là cả một bài toán không hề đơn giản.

Để xây dựng được một hệ thống như vậy ta có nhiều bước cần phải sử dụng các giải thuật xử lý văn bản được nghiên cứu nhiều trong khai phá dữ liệu văn bản, dữ liệu web như: Thu thập nội dung tin tức, xử lý trùng lặp tin tức, phân loại bản tin theo danh mục, xác định từ khóa quan trọng của nội dung tin tức và sinh tóm tắt cho bản tin, kiểm lỗi chính tả tin tức, phát hiện chủ đề nóng, chủ đề nhạy cảm, xu hướng đọc tin trong thời

[1] http://www.baomoi.com/Statistics/Report.aspx

gian gần, …

Đó cũng chính là lý do mà tác giả chọn và nghiên cứu đề tài: "*Xử lý trùng lặp, phân loại, xác định từ khóa quan trọng và sinh tóm tắt cho văn bản trong một hệ thống thu thập tin tức tự động*". Luận văn được chia thành 4 phần như sau:

Chương 1. Giới thiệu đề tài

Chương này trình tổng quan về hệ thống thu thập tin tức tự động đồng thời giới thiệu một số bài toán khai phá dữ liệu trong hệ thu thập tin tức tự động, và giới thiệu cơ bản về các bài toán trong khuôn khổ đề tài.

Chương 2. Một số phương pháp tiếp cận

Chương này tập trung trình bày các phương pháp tiếp cận cho các bài toán xử lý trùng lặp, bài toán phân loại tin tức, bài toán xác định từ khóa quan trọng và chọn câu tóm tắt cho tin tức, trong mỗi phương pháp đều có nhận xét hữu ích.

Chương 3. Đề xuất mô hình giải quyết

Từ những kết quả nghiên cứu từ chương 2, chương này của luận văn sẽ chỉ ra phương pháp phù hợp cho bài toán thực tế được chọn lựa để đưa vào thực nghiệm. Tiếp đến trình bày, mô tả mô hình chi tiết và cách giải quyết cho từng bài toán.

Chương 4. Thực nghiệm và đánh giá

Chương cuối của luận văn sẽ dựa trên những phương hướng thực nghiệm cải tiến đã trình bày ở chương 3, để tiến hành các bước thực nghiệm với ba bài toán: Phát hiện tin tức trùng lặp, phân loại tin tức, xác định từ khóa quan trọng và chọn câu tóm tắt cho bản tin. Với mỗi bài toán, luận văn đưa ra những phương pháp đánh giá, những phép so sánh phù hợp và trình bày kết quả đạt được tương ứng.

Phần tổng kết: Phần tổng kết sẽ nêu lên những kết quả đạt được, những khó khăn hạn chế gặp phải trong quá trình giải quyết các bài toán và cuối cùng là định hướng phát triển trong tương lai.

Chương 1. GIỚI THIỆU ĐỀ TÀI

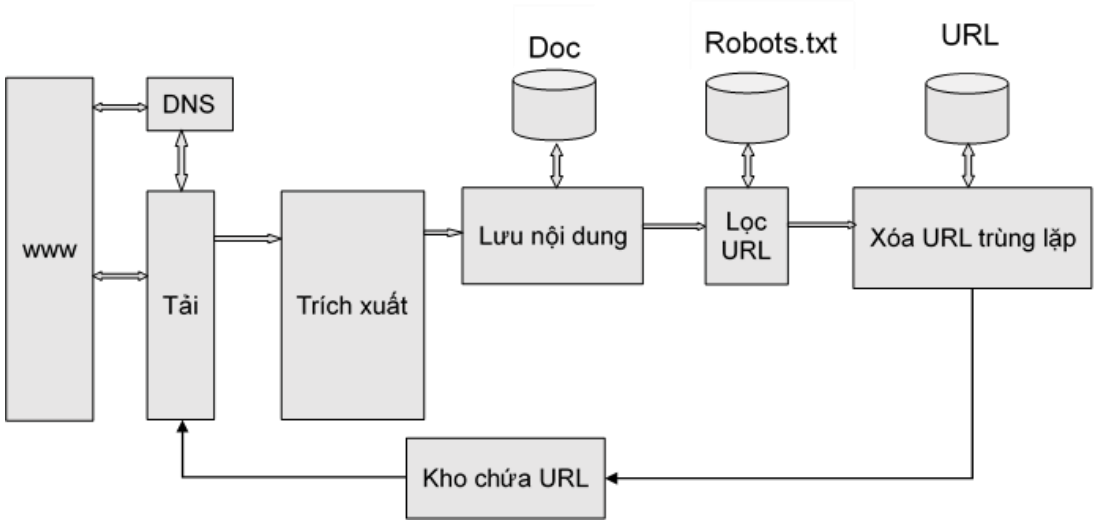
Trong chương này, luận văn tập trung giải quyết các vấn đề sau: giới thiệu tổng quan về hệ thống thu thập tin tức tự động, các bài toán trong khuôn khổ đề tài, ý nghĩa khoa học và ý nghĩa thực tiễn của bài toán đó.

1.1. Tổng quan về hệ thống thu thập tin tức tự động

1.1.1. Tổng quan về Crawler

Hệ thu thập tin tức tự động có thành phần cốt lõi là trình thu thập nội dung trang tin tức từ Internet (gọi là NewsCrawler), mô hình kiến trúc các thành phần của News Crawler giống với các trình thu thập nội dung Web (Web Crawler) thông thường khác, chỉ khác là khi áp dụng mới hệ thu thập tin tức tự động thì thành phần URL nhân (hay còn gọi là Seed) sẽ là tập các trang tin tức. Phần này sẽ giới thiệu mô hình tổng quan của Crawler và vấn đề áp dụng vào bài toán thu thập tin tức tự động.

Web Crawler (một số với tên gọi khác là WebRobot hoặc Web Spider) là một chương trình máy tính có thể "duyệt web" một cách tự động theo một phương thức, hành vi nào đó được xác định trước. Vì là một chương trình máy tính nên quá trình "duyệt web" của các Web Crawler có thể không hoàn toàn giống với quá trình duyệt web của con người (Web Crawler có thể sử dụng các phương thức dựa trên HTTP trực tiếp chứ không thông qua WebBrowser như con người). Kiến trúc cơ bản của một Crawler bao gồm các thành phần như sau:



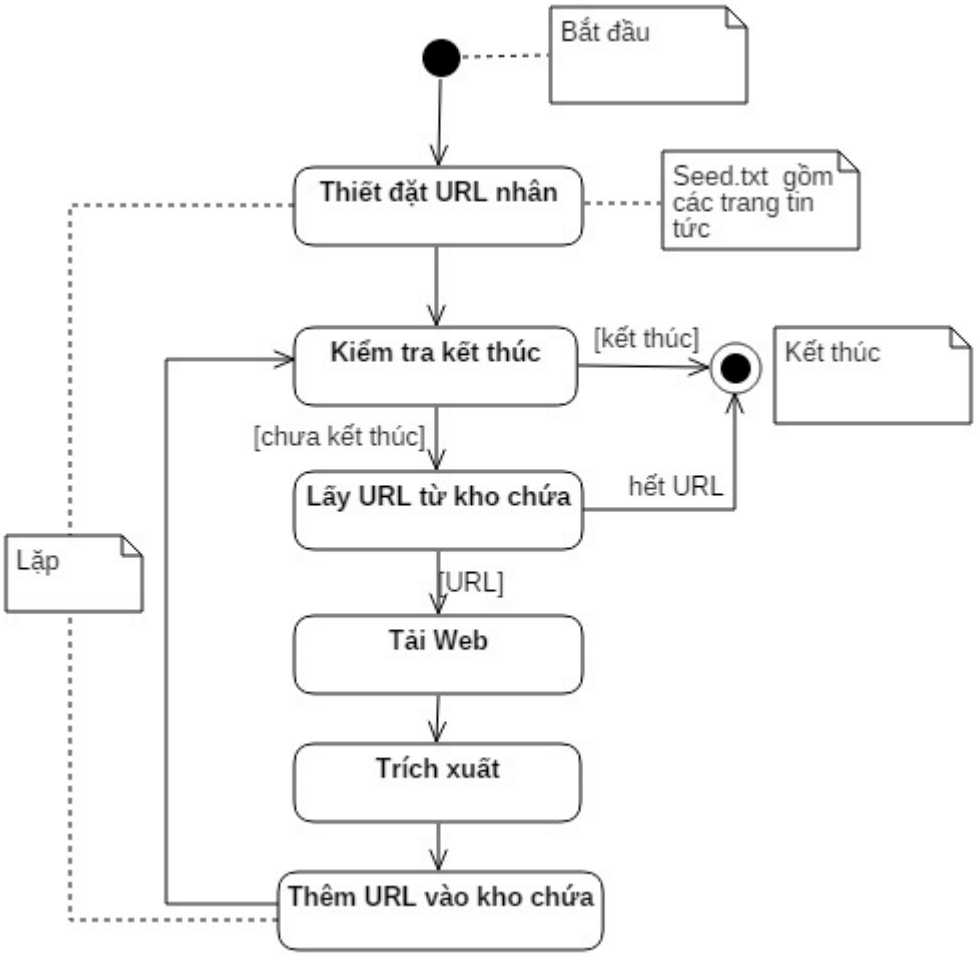
Hình 1.1. Kiến trúc các thành phần cơ bản của Web Crawler

Giải thích các thành phần trong hình 1.1:

- WWW là thành phần đại diện cho các trang Web trên internet.

- DNS viết tắt của Domain Name Service, dịch vụ phân rã tên miền phục vụ cho việc tìm kiếm địa chỉ IP thực của trang Web.
- Tải dữ liệu (Fetch) là quá trình tải trang Web, thường sử dụng giao thức HTTP để tải về nội dung các trang Web.
- Trích xuất (Parse) là quá trình trích xuất nội dung trang Web, trích xuất dữ liệu văn bản, dữ liệu đa phương tiện (hình ảnh, video, âm thanh,…) , liên kết Web,…
- Lưu nội dung (Store content) là việc lưu trữ nội dung trong pha trích xuất vào cơ sở dữ liệu dưới dạng tài liệu (Document).
- Lọc URL (URL filter) thường gồm các quá trình:
    - Kiểm tra tập tin robots.txt để xem URL nào được phép truy cập tuân theo luật của trang WEB mà Web Crawler đang thăm.
    - Chuẩn hóa các URL chẳng hạn như vấn đề mã hóa văn bản (encoding) hay vấn đề tuyệt đối hóa các đường dẫn tương đối.
- Xóa URL trùng lặp (Dup URL Remove) là quá trình loại bỏ các URL trùng lặp trong quá trình đi thăm trang Web.
- URL Frontier là nơi chứa các đường dẫn Web(URL) chưa được Crawler duyệt đến, ban đầu URL Frontier sẽ chứa các URL nhân hay gọi là Seed URL.

Chi tiết về quá trình hoạt động của Web Crawler được mô tả bởi biểu đồ trạng thái sau:

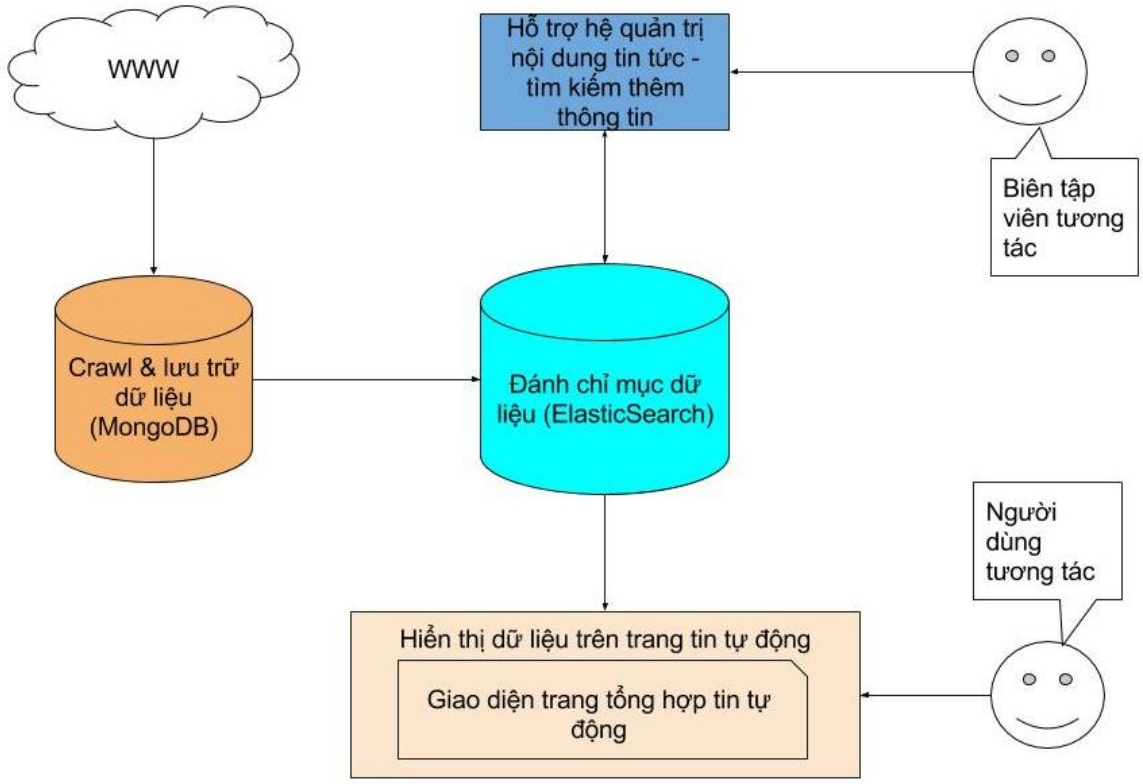


Hình 1.2. Biểu đồ trạng thái của Web Crawler

Crawler chứa một danh sách các liên kết chưa được thăm thường được thiết kế dưới dạng hang đợi (queue) gọi là kho chứa URL (frontier). Danh sách này được tạo ra bởi các URL hạt nhân (Seed URL) trong hệ thống thu thập tin tức Seed URL là tập các URL của các trang tin tức. Mỗi vòng lặp thu thập dữ liệu sẽ gồm các bước sau: chọn URL tiếp theo từ kho chứa URL(frontier), đi thăm URL đó (thường dùng giao thức HTTP), bóc tách nội dung trang web vừa tải về để lấy ra nội dung, các thông tin cần và các URL để đi thăm tiếp, kết thúc vòng lặp bằng việc thêm các URL này vào kho chứa. Quá trình crawling có thể kết thúc khi một số lượng nhất định các trang web đã được tải tùy chọn của người quản lý Crawler hoặc cho tới khi không còn đường dẫn đi thăm tiếp theo. Chương trình crawler sẽ không có trang web mới để tải và dừng lại.

1.1.2. Hệ thống thu thập tin tức tự động

Hệ thống thu thập tin tức tự động với kì vọng dữ liệu tin tức lấy được từ Crawler sẽ được đánh chỉ mục và phục vụ các mục đích khác nhau thể hiện bởi hình 1.3 dưới đây:



Hình 1.3. Mô hình tổng quan hệ tổng hợp tin tự động cơ bản

Tin tức sau khi thu thập bởi trình thu thập được đánh chỉ mục lên máy tìm kiếm để hỗ trợ việc tra cứu tìm kiếm thông tin cho biên tập viên - những người tương tác, tra cứu tìm hiểu, tham khảo thông tin. Hơn thế, dữ liệu tin tức sau khi thu thập còn được dùng với mục đích là xuất bản nội dung tin ra một trang tổng hợp tin tức động phục vụ người đọc tương tác tra cứu tìm kiếm thông tin.

Với hệ thống hiện tại như hình 1.3 dữ liệu tin tức lấy về được đánh chỉ mục thẳng lên máy tìm kiếm và kết nối trực tiếp đến hệ quản trị nội dung cũng như trang tổng hợp thông tin tự động nảy sinh các vấn đề bất cập sau:

- Số lượng tin tức bị trùng lặp do các trang tin dẫn nguồn đăng lại khá nhiều
- Các tin tức không được phân loại dẫn đến khó khăn trong việc tra cứu theo lĩnh vực, chủ đề.
- Nhiều tin không có phần tóm tắt, không có từ khóa quan trọng nêu bật chủ đề, gây khó khăn trong việc tra cứu, tìm hiểu nội dung chính của tin một cách nhanh chóng

Với Crawler thông thường chỉ giải quyết được nhu cầu cơ bản nhất đó là việc thu thập dữ liệu. Hệ thống thu thập tin tức tự động trong thực tế cần nhiều hơn thế. Để đáp ứng được nhu cầu tổng hợp tin tức không trùng lặp, phân loại, xác định các từ khóa quan trọng và câu quan trọng, của nội dung tin tức, các phần tiếp theo của luận văn sẽ thực hiện việc xây dựng các mô-đun xử lý dữ liệu tin tức mở rộng hệ thống. Chi tiết các bài

toán và cách giải quyết vấn đề từng bài toán trong thực tế sẽ được giới thiệu trong các chương tiếp của luận văn.

1.2. Các bài toán trong khuôn khổ đề tài

1.2.1. Bài toán xử lý trùng lặp tin tức

Với crawler phân tán việc thực hiện đi thăm chỉ đơn thuần chống lại việc trùng lặp ở mức URL, tuy nhiên như thế là chưa đủ, vấn đề đặt ra của chúng ta là những trang tổng hợp tin hoặc các tin đăng lại chiếm một lượng khá lớn gần như 100% các tin tức mới đăng được đăng lại ở ít nhất một nơi khác (Theo số liệu của Baomoi.com tháng 2/2016). Việc mở rộng các site hạt nhân (seed) do chính tác giả kiểm nghiệm bao gồm 120 trang báo chí và thông tin điện tử càng làm cho vấn đề trùng lặp trở nên phức tạp, lượng tin tức đổ về ngày một nhiều và bình quân ước lượng với iNews một tin có gần tới 3,5 lần đăng lại và sao chép với nội dung tương tự. Rõ ràng việc kiểm tra trùng lặp đơn thuần bằng URL không còn hiệu quả nữa bởi thực tế có cả trường hợp một báo dẫn hai URL cùng nói về một nội dung. Đến đây việc xử lý trùng lặp nội dung trở nên cấp thiết.

Không chỉ đơn giản là lấy dấu vân (finger print) của nội dung một cách chính xác vì chỉ một chỉnh sửa rất nhỏ cũng có thể làm thay đổi dấu vân, biện pháp kiểm tra trùng lặp cũng trở nên rất khó khăn khi lượng dữ liệu lớn, trải rộng qua các luồng, các node của crawler phân tán.

Tất cả tạo nên một bài toán khó và sẽ được thảo luận tìm hướng giải quyết trong chương tiếp theo.

Phát biểu bài toán:

Input:

- Tập các tin tức được thu thập trên web.
- Tin tức mới được thu thập, cần kiểm tra sự trùng lặp với tập cũ.

Output:

Tin tức mới thu thập có bị trùng lặp hay không? Trong đề tài này luận văn lấy ngưỡng(threshold) là giống lớn hơn hoặc bằng 70% nội dung được coi là trùng lặp, lưu lại ID của bài gốc và tỉ lệ phần trăm trùng lặp.

1.2.2. Bài toán phân loại tin tức

Một vấn đề khác là khi tin tức đổ về lượng lớn, Crawler rất khó có thể cung cấp cho bộ phân tích tin đó thuộc chủ đề nào, rõ ràng chúng ta phải có biện pháp phân loại lại danh mục của tin để dễ dàng sử dụng với các mục đích sau này chẳng hạn như để người dùng tra cứu với phần Front-end trang iNews. Không phủ nhận rằng đã có rất

nhiều thuật toán phân loại văn bản được giới thiệu, việc áp dụng thuật toán nào, cải tiến đóng góp ra sao để phục vụ được mục đích riêng và cho ra kết quả "chấp nhận được" cũng là một trong những bài toán cần cân nhắc để giải quyết hợp lý.

Phát biểu bài toán:

Input:

- Tập các tin tức được thu thập trên web đã được chọn dữ liệu mẫu phân đúng theo các danh mục.

- Tin tức mới được thu thập, cần kiểm tra xem thuộc danh mục nào.

Output:

Danh mục của bản tin mới được thu thập.

1.2.3. Bài toán xác định từ khóa quan trọng và chọn tóm tắt.

Việc xác định từ khóa quan trọng, nêu lên trọng tâm của bản tin đóng góp cực kì quan trọng đến việc hình thành xu hướng tin phục vụ bạn đọc, và nó có ý nghĩa lớn trong việc chọn một vài câu tóm tắt trong nội dung tin cũng có thể giúp người đọc hiểu được ý chính của bản tin, các từ khóa cũng hỗ trợ việc hình thành một chủ đề con (tag, hashtag) của tin tức phục vụ truy vấn dữ liệu theo luồng thông tin. Vậy làm sao để phát hiện từ khóa quan trọng và xu hướng của tin trong bản tin? Đây cũng là một bài toán sẽ được làm rõ trong nội dung của đề tài.

Phát biểu bài toán chọn từ khóa quan trọng:

Input:

- Tập dữ liệu các tin tức.

- Nội dung tin tức.

Output:

Các từ khóa quan trọng phản ánh nội dung của bản tin.

Phát biểu bài toán chọn các câu có thể là câu tóm tắt của bản tin:

Input:

- Tập dữ liệu các tin tức.

- Nội dung tin tức.

Output:

Các câu có thể chọn và sửa hỗ trợ biên tập viên làm câu tóm tắt (mô tả bản tin) nằm trong bản tin.

1.3. Ý nghĩa của các bài toán được giải quyết trong đề tài

1.3.1. Ý nghĩa khoa học

Để xây dựng được các mô đun giải quyết các bài toán trên cần tìm hiểu và áp dụng khá nhiều bài toán học thuật liên quan đến khai phá dữ liệu lớp, thống kê dữ liệu phổ biến, và khai phá từ khóa xu hướng và bài toán xử lý trùng lặp nội dung cơ sở dữ liệu lớn phân tán. Các nội dung khoa học đã được tham khảo áp dụng và cải tiến trong đề tài hi vọng mang lại một phần ý nghĩa đóng góp vào việc giải quyết các vấn khoa học, định hướng mở rộng sau này.

1.3.2. Ý nghĩa thực tiễn

Các mô đun trong khuôn khổ đề tài cũng góp phần vô cùng quan trọng cho một hệ tổng hợp nội dung tự động cung cấp dưới dạng trang tổng hợp và hệ hỗ trợ biên tập tổng hợp nội dung phục vụ các tác vụ phân tích hay các trang tin chuyên biệt. Việc tổng hợp tin tức, cập nhật liên tục, phát hiện được xu hướng mới trong tin, tóm lược từ khóa chứa nội dung chính trong tin giúp người đọc tiếp cận nhanh nhất đến nguồn tin tức khổng lồ đó là một trong những ý nghĩa thực tiễn quan trọng của đề tài.

Ngoài ra việc cung cấp các API cũng cho phép bên thứ ba tiếp cận nguồn tin để phục vụ các mục đích riêng của mình như thống kê, phân tích, khai phá dữ liệu khác cũng là ý nghĩa thực tiễn không nhỏ.

1.4. Kết luận

Trong chương này, luận văn trình tổng quan về hệ thống thu thập tin tức tự động đồng thời giới thiệu mộn số bài toàn khai phá dữ liệu trong hệ thu thập tin tức tự động, và giới thiệu cơ bản về các bài toán trong khuôn khổ đề tài, đồng thời nói lên ý nghĩa khoa học và ý nghĩa thực tiễn, một số khó khăn và các vấn đề cần giải quyết với mỗi bài toán.

Chương 2. MỘT SỐ PHƯƠNG PHÁP TIẾP CẬN BÀI TOÁN

Trong chương này luận văn sẽ đề cập đến cơ sở lý thuyết các thuật toán cũng như một số phương pháp tiếp cận các bài toán đã nêu ở chương 1, phân tích những ưu điểm nhược điểm của từng phương pháp tạo tiền đề để phục vụ việc lựa chọn, đề xuất giải pháp trong chương tiếp theo. Các bài toán kèm theo phương pháp tiếp cận được trình bày trong chương này bao gồm: Bài toán xử lý trùng lặp tin tức, bài toán phân loại tin tức, bài toán xác định từ khóa quan trọng của tin tức.

2.1. Các phương pháp tiếp cận bài toán trùng lặp tin tức

Về cơ bản tin tức sau khi thu thập dữ liệu và tiền xử lý loại bỏ các phần thừa, cũng như chuẩn hóa dữ liệu tin đầu vào thì bài toán phát hiện trùng lặp tin tức có thể quy về bài toán phát hiện trùng lặp nội dung văn bản text. Có rất nhiều phương pháp khác nhau để phát hiện trùng lặp văn bản - Gọi là các phương pháp NDD (Near Duplicate Detection)[3]. Luận văn sẽ giới thiệu một số phương pháp cơ bản bao gồm:

- Bag of Words – So sánh các từ và tần số của những từ đó trên một bản tin với những bản tin khác.
- Shingling – Phương pháp này cải tiến trên "Bag of Words" phương pháp tiếp cận bằng cách so sánh các cụm từ ngắn, cung cấp một số ngữ cho các từ.
- Hashing – Phương pháp này sẽ cải thiện được quá trình kiểm tra trùng lặp bằng cách loại bỏ sự cần thiết để lưu trữ các bản sao của tất cả các nội dung. Các cụm từ được băm vào con số, mà sau đó có thể được so sánh để xác định sự trùng lặp.
- MinHash – Hàm băm giúp lưu trữ phản ánh một phần nội dung trùng lặp theo ngữ cảnh dựa trên sự tương đồng các vec-tơ nhị phân.
- SimHash – Hàm băm giúp lưu trữ phản ánh một phần nội dung trùng lặp theo ngữ cảnh dựa vào dữ liệu thực thông qua độ đo cosine.

Phần tiếp theo, luận văn sẽ đi vào phân tích chi tiết từng phương pháp tiếp cận trên để làm rõ hơn bài toán, cũng như phân tích những thuận lợi khó khăn khi áp dụng các phương pháp này vào thực tế.

2.1.1. Bag of Words

Bag of Words là một trong những kĩ thuật cơ bản nhất trong việc thực hiện kiểm tra phát hiện trùng lặp nội dung văn bản. Giả định rằng chúng ta có một tập hợp các tài liệu độc lập, và muốn tìm thấy một bản sao trùng lặp của nó. Với mỗi tài liệu chúng ta sẽ so khớp nội dung trùng với các tài liệu khác. Nội dung trùng là các từ trùng lặp trong một túi từ (bag of word) bao gồm các từ (được tách độc lập) từ nội dung bản tin.

Chẳng hạn một đoạn tài liệu: A = “khám phá vẻ đẹp tiềm ẩn của Sơn Đoòng”

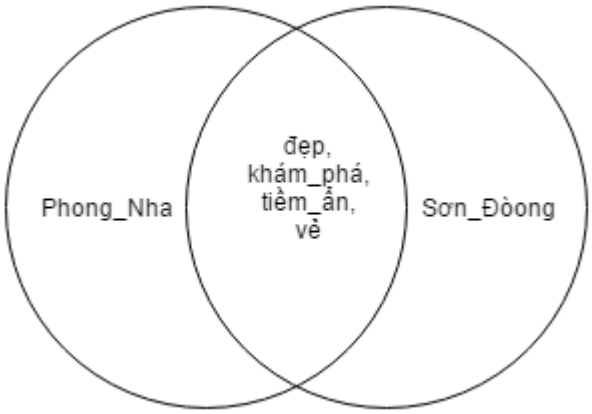
sẽ được chuyển về một tập hợp các từ bao gồm:

$$BagA = \{của, đẹp, khám\_phá, tiềm\_ẩn, Sơn\_Đoòng, vẻ\}$$

Để so sánh hai tài liệu chúng ta tìm ra các từ chungcủa hai tài liệu so với tập hợp từ của cả hai tài liệu độ đo này được gọi là hệ số Jaccard.

Chẳng hạn để so sánh câu $B = "khám phá vẻ đẹp tiềm ẩn của Phong Nha"$ ta làm như sau:

$$BagB = \{của, đẹp, khám\_phá, Phong\_Nha, tiềm\_ẩn, vẻ\}$$



Hình 2.1. Mô phỏng BagofWords

Hệ số Jaccard trong trường hợp này:

$$J(A,B) = \frac{J(A \cap B)}{J(A \cup B)} = \frac{4}{6} \sim 0.67$$

Giải pháp này đơn giản, và thuận lợi khi hai đoạn văn bản nội dung khác nhau với các từ trong túi từ khác nhau nhiều. Tuy nhiên nó cũng gây ra sự nhầm lẫn vì có những trường hợp hai câu có lượng lớn các từ giống nhau nhưng nghĩa có thể khác xa nhau. Hay nói cách khác, cách làm này không giữ lại được ngữ cảnh và sẽ xảy ra trường hợp sai sót. Chẳng hạn như câu: "***tôi thích bạn***" và câu: "***bạn thích tôi***".

Rõ ràng ngữ cảnh nói chung hay trật tự sắp đặt các từ trong câu là quan trọng trong việc kiểm tra nội dung, để khắc phục nhược điểm này người ta đề xuất cải tiến thêm một phương pháp tiếp cận mà chúng ta sẽ nghiên cứu trong mục tiếp theo đó là Shingling.

2.1.2. Shingling

Shingling được trình bày vào năm 1997 bởi Broder và cộng sự. Thuật toán Shingling dựa trên tập hợp các bộ từ (token) chồng lên nhau (giả sử là k token). Trong shingling, tất cả các chuỗi con từ của các từ liền kề sẽ được trích xuất. Qua đó, mỗi tài liệu D lấy được một tập SD. Đó là việc chuyển đổi một tài liệu thành một tập hợp của

các shingle (có thể là các *k-gram*) độc nhất (tức là các chuỗi con kề nhau của k tokens). Sự giống nhau giữa hai tài liệu được đo bằng cách sử dụng hệ số Jaccard giữa các vectơ shingle. Các tài liệu có độ tương đồng cao được coi là gần như trùng lặp. Xem xét trình tự của các từ trong một tài liệu. Tập hợp các shingle cấu thành tập các đặc trưng của một tài liệu.

Việc lấy giá trị k rất nhạy cảm, và ảnh hưởng trực tiếp tới kích thước của shingle và qua đó ảnh hưởng đến tốc độ xử lý cũng như độ chính xác của việc phát hiện trùng lặp.

- Kích thước shingle dài: Những thay đổi ngẫu nhiên nhỏ của tài liệu gây ảnh hưởng lớn.
- Kích thước shingle ngắn: Các tài liệu không liên quan có thể có quá nhiều sự tương đồng.

Trở lại ví dụ ở trên hai mệnh đề: $\mathrm{d1} = \text{"tôi thích bạn"}$ và $\mathrm{d2} = \text{"bạn thích tôi"}$

Nếu theo cách tiếp cận Bagofword thì hai mệnh đề này giống nhau 100%. Theo cách tiếp cận này giả sử chọn k=2.

$$Sd1(2) = \{tôi\ thích, thích\ bạn\}$$

$$Sd2(2) = \{bạn\ thích, thích\ tôi\}$$

$$J2(d1, d2) = \frac{Sd1(2) \cap Sd2(2)}{Sd1(2) \cup Sd2(2)} = 0$$

=> Hai mệnh đề tương đối khác nhau

Lại một lần nữa quay lại với ví dụ ở phần trước

A = "khám phá vẻ đẹp tiềm ẩn của Sơn Đoòng"

B= "khám phá vẻ đẹp tiềm ẩn của Phong Nha"

Với *k=2*

SA(2)={khám_phá vẻ, vẻ đẹp, đẹp tiềm_ẩn, tiềm_ẩn của, của Sơn_Đoòng}

SB(2)={khám_phá vẻ, vẻ đẹp, đẹp tiềm_ẩn, tiềm_ẩn của, của Phong_Nha}

$$J(A, B) = \frac{J(A \cap B)}{J(A \cup B)} = \frac{4}{6} \sim 0.67$$

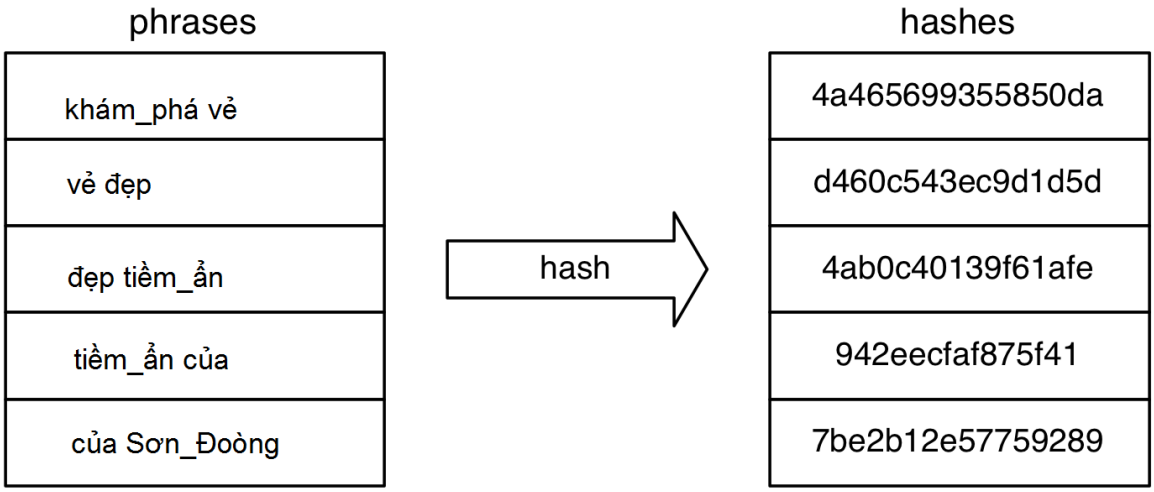
Vẫn có sự tương đồng giữa hai mệnh đề.

Kết luận: Shingling có thể kiểm tra trùng lặp giữ lại một phần ngữ cảnh của tài liệu. Tuy nhiên có một vấn đề xảy ra là việc lưu trữ tập shingle lớn, việc kiểm tra trùng lặp trở nên khó khăn và không khả thi trong thực tế.

2.1.3. Hashing

Như đã đề cập ở mục trước, vấn đề lớn của phương pháp trên là việc lưu trữ và lưu trữ trùng lặp các đoạn *k-gram* từ diễn ra thường xuyên, và có k từ trong một cụm từ thì độ phức tạp lưu trữ sẽ rơi vào khoảng *O(nk)*, Để giảm thiểu điều này chúng ta chuyển mỗi cụm từ qua một hàm băm nhất định để tạo đại diện, và thay vì lưu trữ cả một túi các từ ta sẽ lưu trữ đại diện tạo ra từ hàm băm, việc này sẽ thuận lợi hơn và giảm thiểu được không gian lưu trữ.

Ví dụ như trên khi lưu trữ các cụm từ với *k-2* sẽ có các đoạn hash sau:



Hình 2.2 Ví dụ về hashing

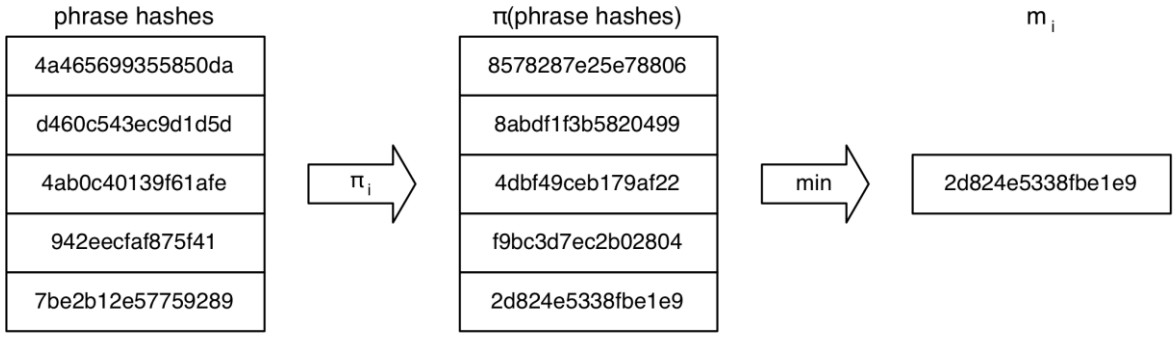
Việc giảm được không gian lưu trữ là một bước tiến đáng kể tuy nhiên trong môi trường thực tế việc lưu trữ đầy đủ các hash của các cụm từ để so sánh hai tài liệu vẫn là một việc làm vô cùng khó khăn. Rất nhiều tài liệu có độ dài lớn, khi so sánh hai tài liệu với mô hình K-gram với các cụm từ (phrases) trùng lặp việc lưu trữ và tính toán vẫn là rất lớn. Đã có một vài nghiên cứu phát triển thêm để giảm bớt thời gian tính toán trùng lặp. Trong luận văn này sẽ đề cập đến hai hàm băm đặc biệt đó là MinHash và SimHash, chi tiết sẽ được giới thiệu trong mục tiếp.

2.1.4. MinHash

MinHash là một cách tiếp cận mới với khả năng sử dụng bộ nhớ không phụ thuộc vào độ dài của tài liệu đồng thời cung cấp phương thức tốt hơn để tính toán độ tương đồng. Cách tiếp cận này dựa trên việc băm mỗi tài liệu ra một tập cố định các hash như một dạng chữ kí thô của tài liệu đó.

Việc băm đặc biệt này được thực hiện bằng cách sử dụng một tập hợp k hàm băm ngẫu nhiên. Với mỗi hàm băm ngẫu nhiên kí hiệu là πi,, chúng ta truyền tải nội dung

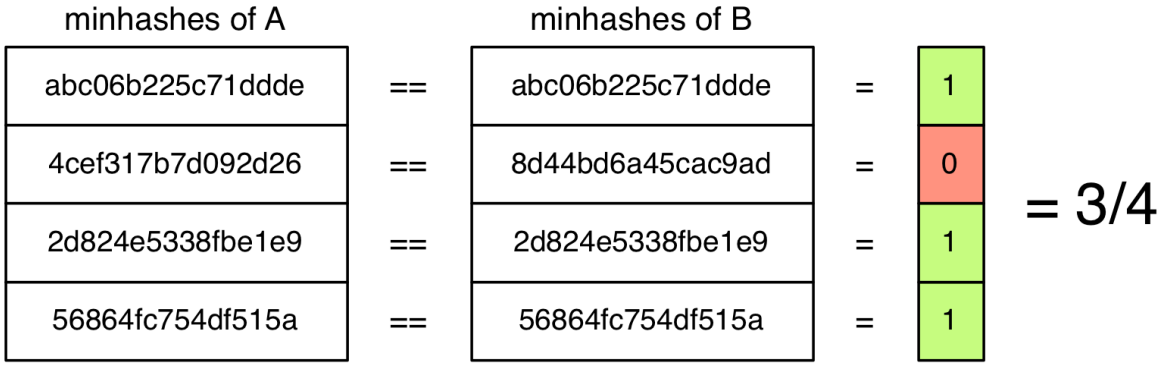
của các cụm từ trong tài liệu thông qua hàm băm để tạo một dãy băm nhỏ nhất (minimum) kí hiệu là m_i.



Hình 2.3. Mô phỏng minhash

Chữ kí của tài liệu giờ sẽ là danh sách thứ tự các hàm băm tối thiểu m_0. Tiếp đó một cách gần đúng ta có thể đo tương tự bằng hệ số Jaccard thông qua việc so sánh từng cặp mã băm của tập hàm băm tối thiểu của tài liệu, và đưa ra kết quả sự giống nhau của tài liệu.

Ví dụ:



Hình 2.3. Ví dụ về minhash

Việc làm này có 2 lợi điểm lớn: Về lưu trữ mỗi tài liệu chỉ yêu cầu không gian lưu trữ O(1) về mặt độ phức tạp tính toán trùng lặp cặp tài liệu đem ra so sánh cũng chỉ là O(1).

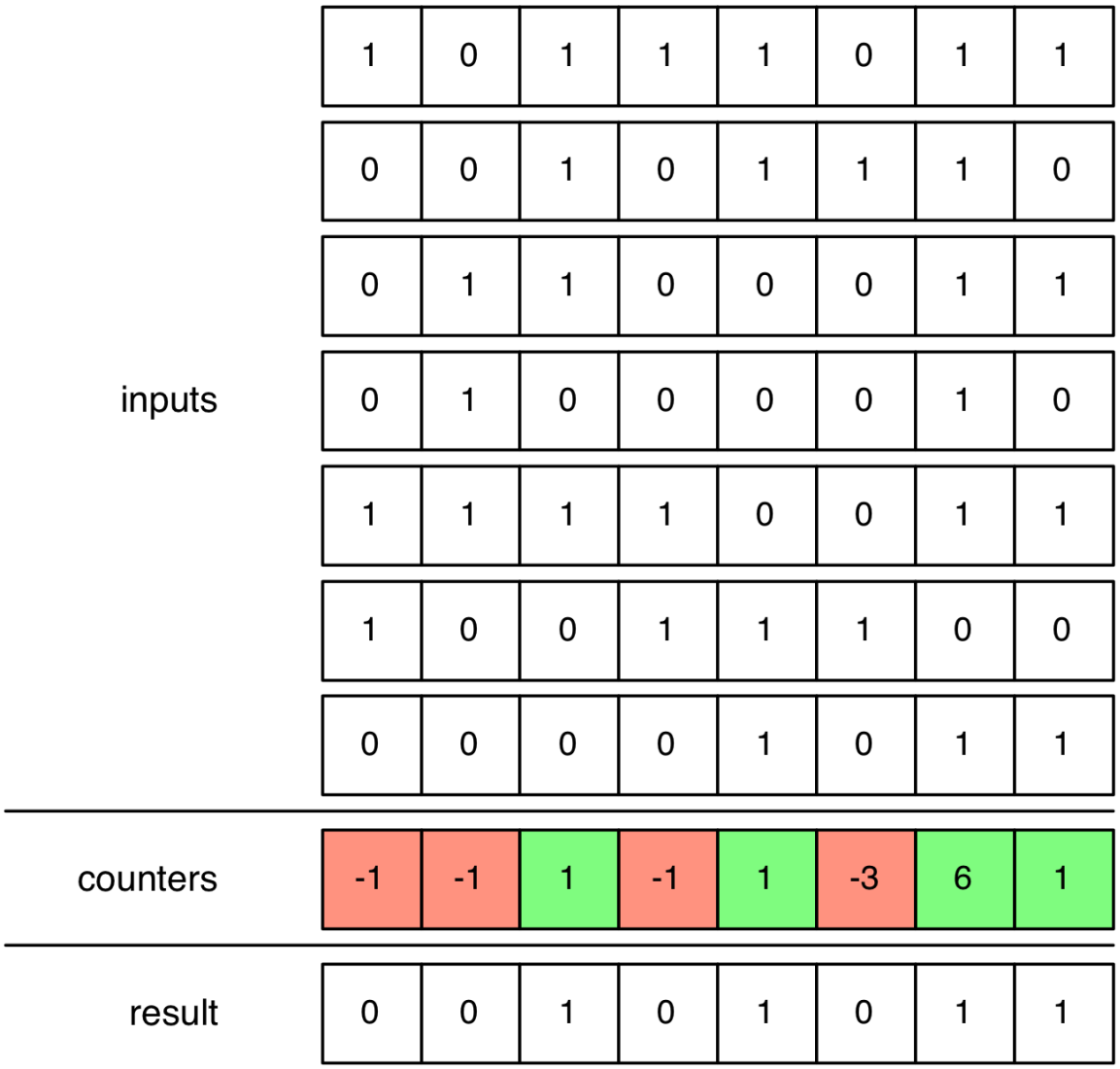
Sử dụng Minhash đã cải thiện rất lớn việc tính toán trùng lặp giữa cặp tài liệu bất kì. Nhưng trong thực tế chúng ta phải đối mặt với vấn đề truy vấn việc trùng lặp một tài liệu mới với một tập các tài liệu có sẵn, áp dụng phương pháp này thì độ phức tạp thời gian tính toán đã trở nên tuyến tính O(n). Trong Crawler, chúng ta phải thu thập tất cả dữ liệu từ các bài tin và xác định tất cả sự trùng lặp của các trang tin, số lượng tin tức

phải xử lý trùng lặp lên đến hàng triệu trang, ở điểm này dường như Minhash có thể trở nên hạn chế hơn về tốc độ.

2.1.5. SimHash

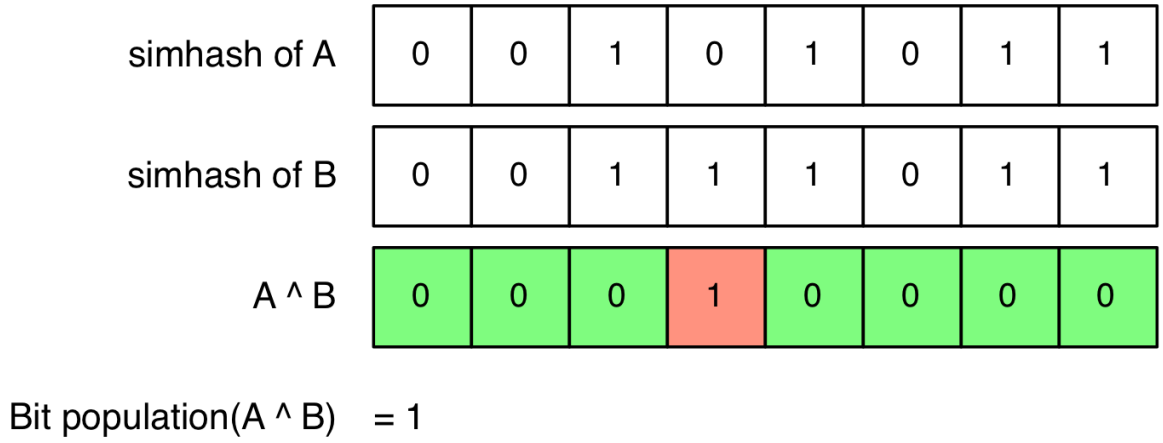
Simhashing là kĩ thuật có thể giúp chúng ta khắc phục vấn đề này. Đầu vào của chúng ta là tập các hash, simhash sẽ tạo ra một mã hash duy nhất với một đặc tính rất đặc biệt - hai tập hashed đầu vào sẽ cho ra một kết quả hashes tương tự. Hầu hết các loại hàm băm khác thường có đặc tính đầu vào dù khác nhau rất ít nhưng kết quả băm rất khác nhau ở phía đầu ra.

Với mỗi vị trí bit, chúng ta đếm số hash đầu vào với tập bit được set và trừ đi số input hash với bit không đc set. Sau khi thực hiện trừ mỗi vị trí với giá trị âm sẽ được set là 0, các vị trí khác sẽ set là 1:



Hình 2.4. Mô phỏng việc lấy simhash

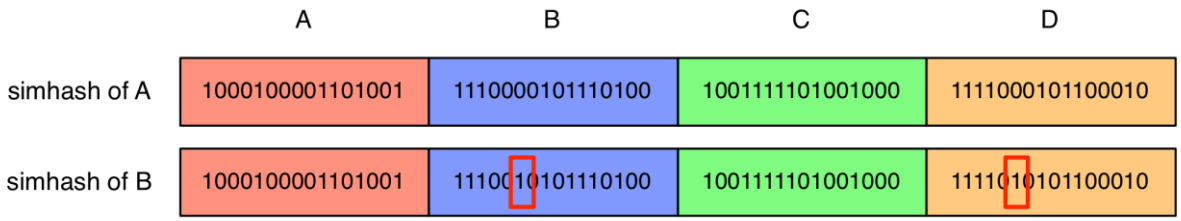
Để tính toán sự giống nhau giữa hai đoạn simhash, chúng ta đếm số bit khác nhau giữa hai dãy bit chính là sự khác nhau giữa hai tài liệu. Ngược lại, số bit giống nhau được coi như sự thể hiện giống nhau của hai tài liệu.



Hình 2.5. Mô phỏng việc tính trùng lặp bằng simhash

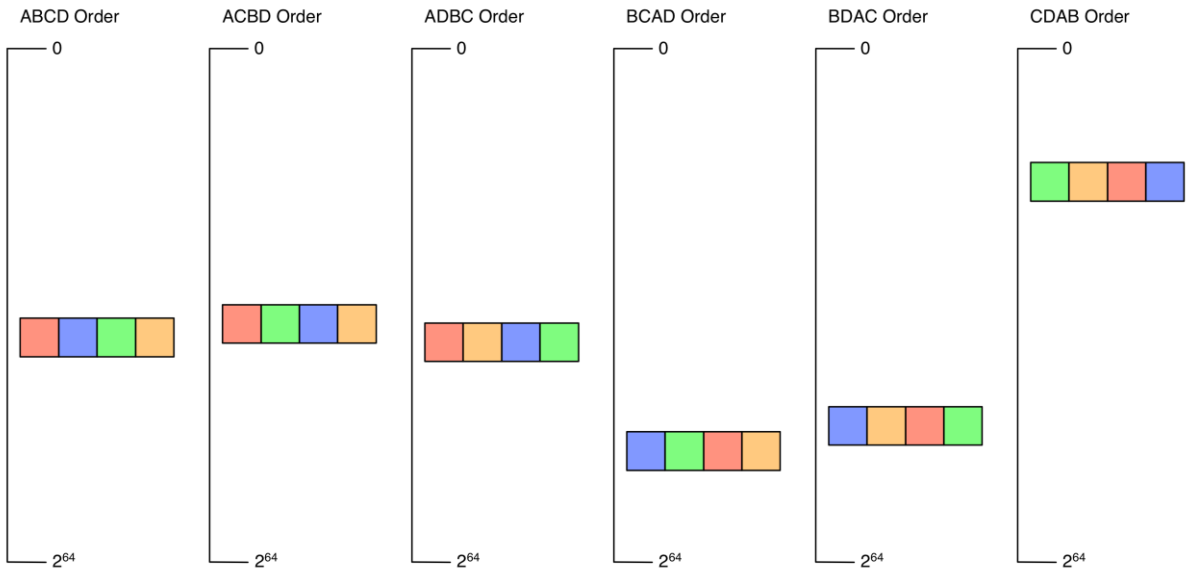
Rõ ràng việc tính toán này thuận lợi hơn nhiều so với việc lưu trữ những dãy hash dài cho mỗi tài liệu, với phương pháp này ta chỉ cần lưu lại một dãy bit hữu hạn như một dấu vân. Việc tính toán trùng lặp cũng trở nên dễ dàng hơn, tuy nhiên việc tính toán trùng lặp sẽ tốt hơn khi dãy bit lớn hơn.

Ví dụ, khi xác định hai dãy AB không trùng lặp ở dải 64 bit chia làm bốn khối (bucket) như hình, thì việc sắp xếp các dãy hash có phần đầu tương tự nhau gần với nhau, sẽ giúp cho việc tính toán simhash mới có thể được thực hiện trong thời gian lograrit.



Hình 2.6. Mô phỏng việc chia simhash theo bucket(khối)

Nhưng cũng ở hình trên, chúng ta có thể cải tiến việc lưu trữ simhash theo từng phân đoạn để cải thiện hiệu năng tính toán hơn. Giả sử dãy simhash được lưu trữ dưới dạng đã sắp xếp, sẽ thật thuật lợi nếu trong trường hợp trên A nằm cạnh C vì AC là tiền tố giống hệt nhau. Vậy nên có một phương pháp tối ưu hơn để cải tiến việc tính toán trùng lặp đó là thay vì lưu trữ một tập đã sắp xếp ta lưu trữ nhiều tập đã sắp xếp với các hoán vị như sáu hoán vị sau: ABCD, ACDB, ADBC, BCAD, BDAC và CDAB.



Hình 2.7. Ví dụ hoán vị các khối với simhash

Với mỗi truy vấn bất kì, ta kiểm tra một tập cố định danh sách các simhash đã được sắp xếp. Tìm kiếm khoảng $O(d * ln(n))$ và một vài so sánh nhỏ chúng ta sẽ tìm ra được kết quả truy vấn trùng lặp. Trong môi trường phân tán ta có thể truy vấn song song d truy vấn. Cách tiếp cận này hoàn toàn phù hợp với việc xử lý crawler lượng lớn dữ liệu trùng lặp.

2.2. Các phương pháp tiếp cận bài toán phân loại tin tức

Bài toán phân loại tin tức có thể quy về bài toán phân lớp văn bản thuần túy, với cách phát biểu bài toán như sau:

Cho x là một văn bản. Biết x thuộc một trong các loại $y \in \{1,2,\ldots,K\}$. Hãy tìm loại văn bản phù hợp nhất với x.

Ví dụ:

- Giả sử x là một tin tức được thu thập về từ internet, cần quyết định xem x thuộc thể loại nào là thích hợp nhất: "chính trị – xã hội", "quốc tế ", "thể thao". . .
- Giả sử x là một người đi vay ngân hàng với hồ sơ lý lịch biết trước, từ đó ngân hàng cần phân tích xem khoản vay x đề xuất thuộc một giá trị trong tập: {nợ tốt, nợ xấu} để cân nhắc ra quyết định cho vay hay không và cho vay bao nhiêu.

Gọi y = hθ(x) là hàm phân loại của x trong đó θ là tham số của hàm. Ta cần tìm hθ (·) có khả năng phân loại tốt. Để tìm hθ, ta sử dụng phương pháp học có hướng dẫn từ dữ liệu mẫu:

Dữ liệu học gồm N mẫu: $(x1, y1), (x2, y2), \ldots, (xN, yN)$.
Hàm hθ được xây dựng sao cho nó khớp nhất với dữ liệu huấn luyện này.

Mỗi văn bản x là một đối tượng cần phân loại, thông thường x được chuyển thành một biểu diễn véc-tơ thực D chiều:

$$x = (x1, x2, \ldots, xD), xj \in R$$

Các thành phần xj, j = 1,2, . . ., D được gọi là các đặc trưng hay thuộc tính của x.

Có nhiều phương pháp phân loại văn bản, phần tiếp theo chúng ta sẽ tiếp cận một vài phương pháp cơ bản

2.2.1. Tiếp cận dựa trên phương pháp cây quyết định

Cây quyết định là một cây trong đó mỗi nút nhánh đại diện cho một lựa chọn giữa một số các lựa chọn khác thay thế, và mỗi nút lá đại diện cho một lớp hoặc một quyết định nào đó. Đây là phương pháp học xấp xỉ các hàm mục tiêu có giá trị rời rạc. Giải thuật này cũng có thể biến đổi thể hiện dưới dạng cây *Nếu – Thì*.

Ý tưởng

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

Cách thực hiện

Cây quyết định này được tổ chức như sau: Các nút trong được gán nhãn bởi các thuật ngữ, nhãn của các cung tương ứng với trọng số của thuật ngữ trong tài liệu mẫu, nhãn của các lá tương ứng với nhãn của các lớp. Cho một tài liệu d_j, ta sẽ thực hiện so sánh các nhãn của cung xuất phát từ một nút trong (tương ứng với một thuật ngữ nào đó) với trọng số của thuật ngữ này trong d_j, để quyết định nút trong nào sẽ được duyệt tiếp. Quá trình này được lặp từ nút gốc của cây, cho tới khi nút được duyệt là một lá của cây. Kết thúc quá trình này, nhãn của nút lá sẽ là nhãn của lớp được gán cho văn bản.

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

Các thuật toán cây quyết định ngày càng được phát triển và cải tiến, hầu hết các thuật toán này đều dựa vào cách tiếp cận từ trên xuống và chiến lược tìm kiếm tham lam trong không gian tìm kiếm của cây quyết định. Đáng kể nhất là cải tiến từ giải thuật ID3 là thuật toán C.4.4 và C.4.5 mang lại độ chính xác cao và được sử dụng rộng rãi.

2.2.2. Phân loại dữ liệu Naïve Bayes

Naive Bayes (NB) là một trong những thuật toán cơ bản trong phân lớp xác suất dựa trên việc áp dụng lý thuyết của Bayes một cách "ngây thơ" bằng việc giả định xác suất độc lập giữa các đặc trưng với lớp cần so sánh.

Thuật toán Naïve Bayes được nghiên cứu từ những năm 1950, và được giới thiệu trong công cộng đồng truy hồi thông tin vào đầu những năm 1960, hiện tại vẫn là một trong những phương pháp phổ biến trong phân loại dữ liệu văn bản.

Thuật toán Naïve Bayes dựa trên định lý Bayes được phát biểu như sau:

$$P(Y|X) = \frac{P(XY)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}$$

Áp dụng trong bài toán phân loại, các dữ kiện gồm có:

- D: tập dữ liệu huấn luyện đã được vector hóa dưới dạng $\vec{x} = (x_1, x_2, \dots, x_n)$
- C_i: phân lớp i, với i = {1,2,…,m}.
- Các thuộc tính độc lập điều kiện đôi một với nhau.

Theo định lý Bayes:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Theo tính chất độc lập điều kiện:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

Trong đó:

- $P(C_i|X)$ là xác suất thuộc phân lớp i khi biết trước mẫu X.
- $P(C_i)$ xác suất là phân lớp i.
- $P(x_k|C_i)$ xác suất thuộc tính thứ k mang giá trị x_k khi đã biết X thuộc phân lớp i.

Các bước thực hiện thuật toán Naïve Bayes:

Bước 1: Huấn luyện Naïve Bayes (dựa vào tập dữ liệu), tính $P(C_i)$ và $P(x_k|C_i)$

Bước 2: Phân lớp $X^{new} = (x_1, x_2, \dots, x_n)$, ta cần tính xác suất thuộc từng phân lớp khi đã biết trước X^{new}. X^{new} được gán vào lớp có xác suất lớn nhất theo công thức

$$\max_{C_i \in C} \left(P(C_i) \prod_{k=1}^{n} P(x_k|C_i) \right)$$

Ứng dụng trong phân loại văn bản

Ý tưởng: Việc đánh giá một tài liệu có thuộc một lớp này hay thuộc những lớp khác hay không được đánh giá thông qua việc xác định các từ (thường dùng tần số từ) hay gọi là đặc trưng trong tài liệu đó có xác suất có điều kiện với loại của một văn bản cần phân loại thông qua công thức Bayes, với giả định như đã nói: xác suất độc lập giữa các đặc trưng với lớp cần so sánh. Kết quả dự đoán bị ảnh hưởng bởi kích thước tập dữ liệu, chất lượng của không gian đặc trưng…

Ví dụ thực tế:

Mô tả vector đặc trưng của văn bản: Là vector có số chiều là số đặc trưng trong toàn tập dữ liệu, các đặc trưng này đôi một khác nhau. Nếu văn bản có chứa đặc trưng đó sẽ có giá trị 1, ngược lại là 0.

Thuật toán gồm hai giai đoạn huấn luyện và phân lớp:

Huấn luyện: tính $P(C_i)$ và $P(x_k|C_i)$

Đầu vào:

- Các vector đặc trưng của văn bản trong tập huấn luyện (Ma trận MxN, với M là số vector đặc trưng trong tập huấn luyện, N là số đặc trưng của vector).
- Tập nhãn/lớp cho từng vector đặc trưng của tập huấn luyện.

Đầu ra:

- Các giá trị xác suất $P(C_i)$ và $P(x_k|C_i)$.

  Công thức tính $P(C_i)$ đã làm trơn Laplace

$$P(C_i) = \frac{|docs_i| + 1}{|total\ docs| + m}$$

Trong đó:

- $|docs_i|$: số văn bản của tập huấn luyện thuộc phân lớp i.
- |total docs|: số văn bản trong tập huấn luyện.
- m số phân lớp

  Cài đặt:

- Khởi tạo mảng A, B có kích thước m.

- Duyệt qua các văn bản trong tập dữ liệu, đếm số văn bản trong mỗi phân lớp lưu vào A.
- Tính xác suất cho từng phân lớp theo công thức trên và lưu vào mảng B.

Công thức tính $P(x_k|C_i)$ đã làm trơn Laplace:

$$P(x_k|C_i) = \frac{|docs_{x_k i}| + 1}{|docs_i| + d_k}$$

Trong đó:

- $|docs_{x_k i}|$: Số văn bản trong trong phân lớp i có đặc trưng thứ k mang giá trị x_k. (hay số văn bản trong lớp i, có xuất hiện/không xuất hiện đặc trưng k)
- $|docs_i|$: Số văn bản của tập huấn luyện thuộc phân lớp i.
- d_k: Số giá trị có thể có của đặc trưng thứ k

Cài đặt:

- Với vector đặc trưng như mô tả bên trên, d_k ở đây mang giá trị là 2, tương ứng với xuất hiện và không xuất hiện. Do chỉ có 2 giá trị, ta có thể tính nhanh xác suất không xuất hiện theo công thức $P(\bar{x}) = 1 - P(x)$
- Khởi tạo mảng ba chiều C, chiều 1 có kích thước là m (số phân lớp), chiều 2 có kích thước là N (số đặc trưng), chiều 3 có kích là 2 (d_k) để lưu các giá trị $P(x_k|C_i)$.
- Duyệt qua các văn bản trong tập dữ liệu, tiến hành thống kê các chỉ số cần thiết để tính xác suất $P(x_k|C_i)$ theo công thức trên và lưu vào mảng C.

Phân lớp:

Đầu vào:

- Vector đặc trưng của văn bản cần phân lớp.
- Các giá trị xác suất $P(C_i)$ và $P(x_k|C_i)$.

Đầu ra:

- Nhãn/lớp của văn bản cần phân loại.

Công thức tính xác suất thuộc phân lớp i khi biết trước mẫu X

$$P(C_i|X) = P(C_i)\prod_{k=1}^{n} P(x_k|C_i)$$

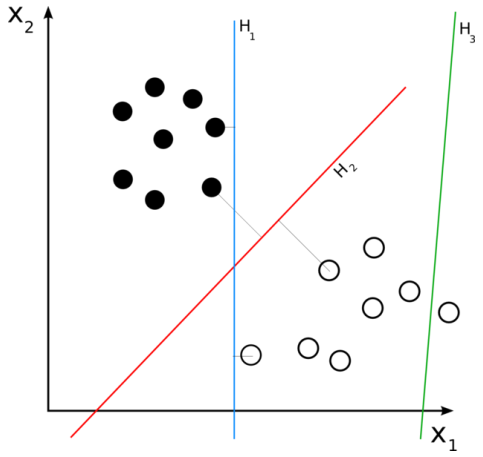
Dựa vào vector đặc trưng của văn bản cần phân lớp, áp dụng công thức trên tính xác suất thuộc từng phân lớp cho văn bản, và chọn ra lớp có xác suất cao nhất.

2.2.3. Tiếp cận theo phương pháp SVM

SVM là một phương pháp phân lớp xuất phát từ lý thuyết học thống kê. Giảm thiểu tối đa việc phát sinh lỗi trong phân loại chủ đề là ý tưởng xuyên suốt thuật toán này. 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[4].

Giả định rằng, người ta lấy một tập hợp dữ liệu đặc trưng là $F = \{f1, f2, \dots, fn\}$, gọi x_i là vector thể hiện của văn bản. Ta có: $x_i=(we_1, we_2, \dots, we_n)$, trong đó $we_n \in R$ là trọng số của đặc trưng f_n. Với tập dữ liệu huấn luyện $T_r=\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, $(x_i \in R^n)$, $y_i \in \{+1, -1\}$, cặp (x_i, y_i) được hiểu là vector x_i được gán nhãn là y_i.

Coi x_i là một điểm trên không gian n chiều, SVM cố gắng tìm một siêu phẳng tối ưu trong không gian đó để tách các phần dữ liệu dương và âm nằm về hai phía của siêu phẳng đó, bởi với mỗi một điểm bất kì với một siêu phẳng ta luôn xác định được trạng thái nó nằm trên phần nào của siêu phẳng hay thuộc siêu phẳng đó.



Hình 2.10. H_2 là mặt phẳng tốt nhất.

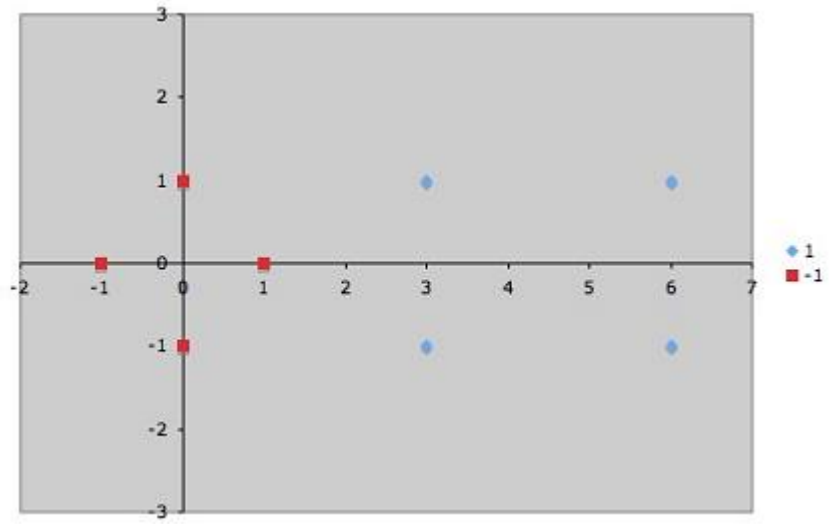
Sử dụng công thức Lagrange trong bài toán tối ưu toàn cục để biến đổi tìm ra siêu phẳng là khá hóc búa. Hiện nay đã có những bộ thư viện đã hỗ trợ cho việc tính toán trên như : SVM^{light}, LIBSVM, jSVM, …

Ví dụ: Giả sử ta có một tập các điểm được gán nhãn dương (+1):

{(3,1), (3, -1), (6, 1), (6, -1)}

Và tập các điểm được gán nhãn âm (-1) trong mặt phẳng R^+:

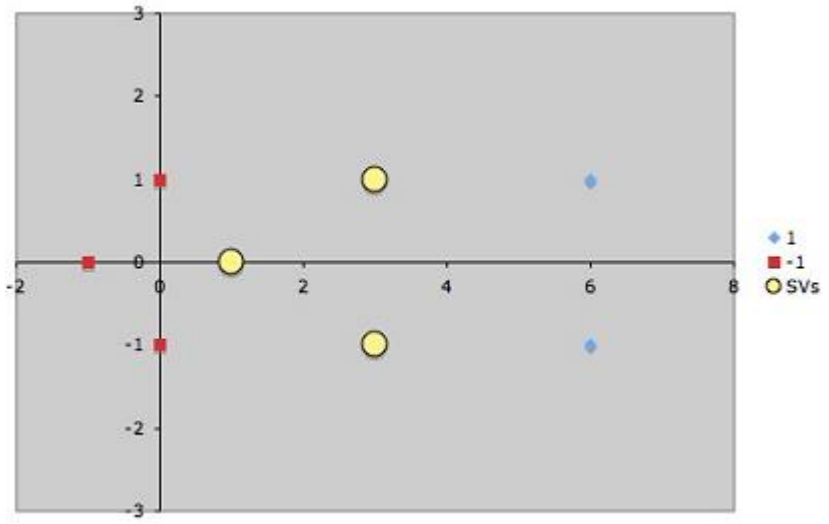
{(1, 0), (0, 1), (0, -1), (-1, 0)}



Hình 2.11. Các điểm dữ liệu được biểu diễn trên R^+.

Chúng ta sẽ dùng SVM để phân biệt hai lớp (+1 và -1). Bởi vì dữ liệu được chia tách một cách tuyến tính, rõ ràng, nên chúng ta sử dụng linear SVM (SVM tuyến tính) để thực hiện. Theo quan sát hình 2, chúng ta chọn ra ba vector hỗ trợ để thực thi các phép toán nhằm tìm ra mặt phẳng phân tách tối ưu nhất:

$\{s_1 = (1,0), s_2 = (3,1), s_3 = (3, -1)\}$



Hình 2.12. Các vector hỗ trợ (support vector) được chọn.

Các vector hỗ trợ được tăng cường (augmented) bằng cách thêm 1. Tức là $s_1 = (1, 0)$, thì nó sẽ được chuyển đổi thành $\tilde{s} = (1, 0, 1)$. Theo kiến trúc SVM, công việc của chúng ta là tìm ra những giá trị α_i.

$$\alpha_1\Phi(s_1).\Phi(s_1)+\alpha_2\Phi(s_2).\Phi(s_1)+\alpha_3\Phi(s_3).\Phi(s_1)=-1$$
$$\alpha_1\Phi(s_1).\Phi(s_2)+\alpha_2\Phi(s_2).\Phi(s_2)+\alpha_3\Phi(s_3).\Phi(s_2)=+1$$
$$\alpha_1\Phi(s_1).\Phi(s_1)+\alpha_2\Phi(s_2).\Phi(s_3)+\alpha_3\Phi(s_3).\Phi(s_3)=+1$$

Bởi vì chúng ta sử dụng SVM tuyến tính nên hàm $\Phi()$ - dùng để chuyển đổi vector từ không gia dữ liệu đầu vào sang không gian đặc trưng – sẽ bằng $\Phi()=I$. Biểu thức trên được viết lại như sau:

$$\alpha_1\tilde{s}_1.\tilde{s}_1+\alpha_2\tilde{s}_2.\tilde{s}_1+\alpha_3\tilde{s}_3.\tilde{s}_1=-1$$
$$\alpha_1\tilde{s}_1.\tilde{s}_2+\alpha_2\tilde{s}_2.\tilde{s}_2+\alpha_3\tilde{s}_3.\tilde{s}_2=+1$$
$$\alpha_1\tilde{s}_1.\tilde{s}_3+\alpha_2\tilde{s}_2.\tilde{s}_3+\alpha_3\tilde{s}_3.\tilde{s}_3=+1$$

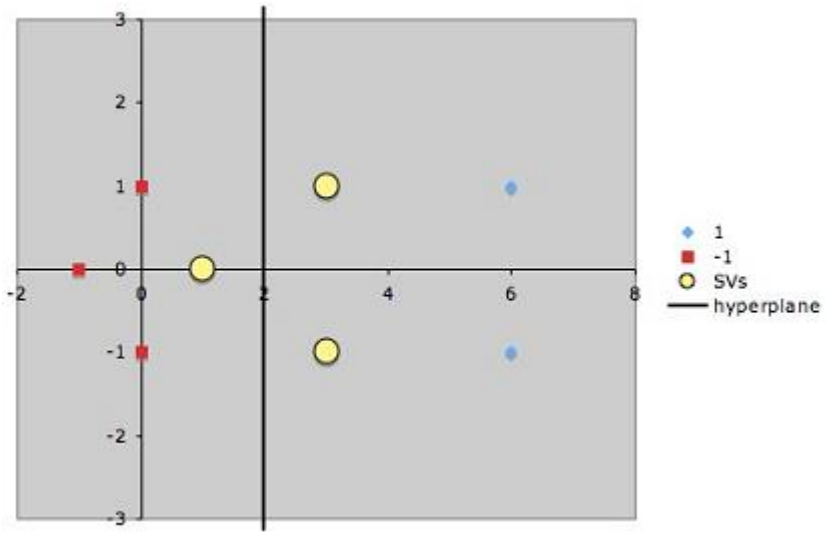
Ta rút gọn biểu thức trên thông qua việc tính toán tích vô hướng giữa các vector.

$$2\alpha_1+4\alpha_2+4\alpha_3=-1$$
$$4\alpha_1+11\alpha_2+9\alpha_3=+1$$
$$4\alpha_1+9\alpha_2+11\alpha_3=+1$$

Giải hệ phương trình ba ẩn trên ta có: $\alpha_1 = -3.5$, $\alpha_2 = 0.75$, $\alpha_3 = 0.75$. Tiếp đến ta tính trọng số $\tilde{\omega}$ thông qua công thức:

$$\tilde{\omega}=\sum_i \alpha_i\tilde{s}_i=-3.5\begin{pmatrix}1\\0\\1\end{pmatrix}+0.75\begin{pmatrix}3\\1\\1\end{pmatrix}+0.75\begin{pmatrix}3\\-1\\1\end{pmatrix}=\begin{pmatrix}1\\0\\-2\end{pmatrix}$$

Siêu phẳng phân chia hai lớp đó là: y = wx + b với w = (1, 0) và b = -2



Hình 2.13: Siêu phẳng được biểu diễn trên R^+.

Ưu điểm của SVM

Một cách công bằng có thể nói, mọi phương pháp phân loại đều có những ưu nhược điểm riêng, điều này là nhiều hay ít quan trọng phụ thuộc vào dữ liệu nào mà ta đang phân tích, do vậy có một sự liên quan tương đối giữa đặc điểm của dữ liệu phân tích và ưu nhược điểm của phương pháp phân loại, sau đây là một số ưu điểm của phân lớp bằng SVM:

Việc sử dụng các hạt nhân tính toán (kernel), SVM đạt được sự linh hoạt trong việc chia các ngưỡng, việc lựa chọn kernel phù hợp một cách dễ dàng cũng là một thuận lợi lớn. Hơn thế nữa không chỉ đơn thuần việc sử dụng hạt nhân tính toán (kernel) thuật toán SVM cải tiến năm 1993[5] đã cho thấy khả năng sử dụng hạt nhân linh hoạt (Kernel trick). Kernel trick là các hàm tối ưu để tìm ra siêu phẳng mà không cần thực hiện việc chiếu các điểm lên không gian nhiều chiều hơn. Điều này có lợi gì? Việc sử dụng kernel trick giúp hạn chế việc tính toán nhiều vì khi ánh xạ dữ liệu lên không gian nhiều chiều hơn lượng xử lý tính toán sẽ rất lớn.

Việc sử dụng các quy tắc tham số trong SVM cũng hạn chế việc quá vừa dữ liệu (over-fitting). SVM được định nghĩa bởi một vấn đề tối ưu hóa lồi (không có cực tiểu địa phương) có những phương pháp hiệu quả để giải quyết, có thể dễ dàng tùy biến áp dụng phương pháp tối ưu hơn vào phân lớp. Cơ chế cực đại hóa biên cũng giúp giảm thiểu tỉ lệ lỗi đáng kể.

Nhiều nghiên cứu từ trước đến giờ đã cho thấy SVM có độ chính xác cao hơn so với các thuật toán phân loại phổ biến khác, cụ thể:

Nghiên cứu của Jin Huang, Jingjing Lu, Charles X. Ling (2003) cho thấy trong phân lớp tập dữ liệu xã hội SVM có độ chính xác cao hơn các thuật toán Bayes, Cải tiến cây quyết định C4.4 và C4.5 [6]

Theo nghiên cứu của Sarini, Sarini, McGree, James, White, Nicole, Mengersen, Kerrie, & Kerr, Graham (2015), về phân loại dịch bệnh dựa trên văn bản cũng cho thấy SVM có kết quả cao hơn khá nhiều so với thuật toán cây quyết định với độ nhạy chính xác lớn hơn 92% so với 88% của thuật toán cây quyết định [7].

Theo nghiên cứu của A. Sopharak và B. Uyyanonvara, S. Barman(2014) việc so

sánh giữa SVM và thuật toán Naïve Bayes cũng cho thấy độ chính xác, độ hồi tưởng của SVM cao hơn.[8]

Ranjeeta Rana, Mrs. Vaishali Kolhe (2015)[9], trong việc khai phá dữ liệu text trên mạng xã hội Twitter chỉ ra rằng độ chính xác ở các lần thực nghiệm đều cho thấy SVM vượt trội hơn so với Naïve Bayes.

Các nghiên cứu cũng cho thấy SVM hoàn toàn phù hợp và thực tế chứng minh đã và đang được dùng phổ biến trong phân lớp văn bản vì những ưu điểm và độ chính xác thực tế được kiểm chứng của thuật toán.

2.3. Tiếp cận bài toán xác định từ khóa quan trọng và chọn câu tóm tắt

2.3.1. Phương pháp TF-IDF

Hans Peter Luhn (1958) được coi là "cha đẻ của lĩnh vực Information Retrieval" và là tác giả của bài báo "The Automatic Creation of Literature Abstracts - 1958" [10]. Phương pháp của Luhn xuất phát từ một ý tưởng tóm tắt các tài liệu văn học chuyên ngành. Phương pháp này dựa trên ý tưởng với giả định rằng: tần số xuất hiện của từ mang lại một ý nghĩa nào đó trong việc thể hiện độ quan trọng của từ đó trong văn bản.

Luhn sử dụng tần số từ cho tóm tắt bởi các từ quan trọng thường được lặp đi lặp lại nhiều lần trong văn bản. Thêm vào đó, thuật toán lại đơn giản, tốn ít thời gian xử lý nên chí phí rẻ. Phương pháp này không phân biệt số ít hay số nhiêu, từ loại dạng thức từ. Tuy nhiên nếu chỉ xét tần số từ trong văn bản thì những từ phổ biến sẽ xuất hiện nhiều nên độ quan trọng của từ đó cũng sẽ tăng chẳng hạn những từ phổ biến như Hà Nội,Việt Nam,….Giải pháp được đưa ra là việc loại bỏ những từ tần số quá thấp hoặc quá cao gây nhiễu ảnh hưởng đến độ quan trọng của từ trong câu, bằng việc đặt ra ngưỡng (threshold). Phương pháp này cũng cho phép loại bỏ từ dừng. (như "rằng", "thì", "mà", "là" ...).

Để lấy số lần xuất hiện của từ nổi bật, Luhn đã tính phân phối của từng từ trong tài liệu xác định (tf) và phân phối của từ ở trong tập văn phạm (idf - inverted document frequency).

$$idf(term) = \log \frac{NumDoc}{NumDoc - term}$$

NumDoc: số tài liệu trong tập văn bản

NumDoc(term); số tài liệu mà có term xuất hiện.

Gọi $We = tf(term) \times idf(term)$ là trọng số của các từ, và được sắp xếp từ cao xuống thấp và gán trọng số với giá trị W_e sau đó các câu gồm các cụm từ sẽ được tính

trọng số câu bằng tổng trọng số các từ. Các câu với tổng trọng số cụm cao nhất được chọn. Ngoài ra việc tham chiếu với kho từ khóa (tags) của trình thu thập và tham chiếu với kho từ khóa xu hướng nổi bật cũng làm cho việc xác định từ khóa quan trọng trở nên chính xác hơn.

2.3.2. Phương pháp Edmundson

Phương pháp Edmundson phục vụ việc tóm tắt văn bản, với ý tưởng quan tâm đến các yếu tố được đánh giá là "quan trọng" của văn bản bao gồm: các từ chốt, các từ khóa của văn bản, tiêu đề của văn bản và vị trí của câu trong văn bản.

Cụm từ chốt (cue) của văn bản

Các cụm từ chốt thường theo sau nó là các câu quan trọng của văn bản, cũng có những cụm từ chốt mà theo sau nó là các câu không mang ý nghĩa quan trọng trong câu. Chẳng hạn như với các cụm từ 'Trong bài này, 'Tóm lại',... thường theo sau chúng chính là phần quan trọng trong văn bản. Hoặc như cụm từ 'chẳng hạn như' thường chỉ ra phần không quan trọng của văn bản.

Tiêu đề (title) của văn bản, đoạn văn bản

Tiêu đề thường được đặt ngắn gọn, xúc tích và nêu bật phần nội dung chính mà văn bản muốn hướng tới, thể hiện. Vì thế các từ trong tiêu đề giúp tìm ra nội dung có liên quan. Cơ sở của yếu tố này là các câu có chứa các từ,cụm từ cùng từ, cụm từ trong tiêu đề thường sẽ có nét nghĩa quan trọng nêu lên nội dung chính của câu.

Các câu tiêu đề và các câu đầu đoạn thường là các câu nêu bật chủ đề, tóm lược nội dung trong văn bản. Giả định rằng một đoạn văn bản chỉ có một tiêu đề, và cũng có thể không có tiêu đề nào thì tiêu đề thường là câu đầu đoạn. Tìm trong phần đầu của tài liệu, nếu chỉ có một câu thì câu này có thể coi là câu tiêu đề. Cách xác định này phụ thuộc định dạng của văn bản đầu vào. Các từ trong tiêu đề có ý nghĩa vô cùng quan trọng, nó giúp xác định các từ khóa quan trọng, và các câu khác càng liên quan đến những câu tiêu đề này thì càng có ý nghĩa quan trọng và thường phải có trọng số cao hơn các câu khác

Vị trí (location) của câu

Như đã nói vị trí các câu trong đoạn văn, trong văn bản có ý nghĩa vô cùng quan trọng, một thống kê đơn giản chỉ cần lấy các câu trong phần đầu văn bản, đầu đoạn đem ra tổng hợp độ chính xác đã đạt khoảng 33%.

Ngoài ra, các văn bản có xu hướng có cấu trúc phụ thuộc vào kiểu của chúng. Chẳng hạn theo quy tắc hành văn thông thường, văn bản sẽ có: phần mở đầu, phần thân và phần kết luận. Đối với bản tin thì có phần tiêu đề, phần mô tả và phần nội dụng, trong phần nội dung lại có cách bố cục rất có thể giống với cách hành văn thông thường, rất

có thể phần đầu với phần kết ngắn nhưng mang ý nghĩa tốt nhất. Trong văn bản kiểu này:

- Đầu câu đầu đoạn thường là các câu nêu bật chủ đề.

- Các câu quan trọng có xu hướng xuất hiện ở cuối của văn bản.

Từ các lập luận trước, ta có thể đề xuất phương pháp chọn ra phần quan trọng của văn bản: Tùy thuộc vào loại văn bản, bố cục văn bản sẽ ảnh hưởng đến vị trí các câu tóm tắt, thông qua một ràng buộc nhất định ta có thể tìm ra những vị trí câu này một cách tự động [19].

Tần số từ trong văn bản

Các câu quan trọng chứa nội dung các từ xuất hiện thường xuyên trong văn bản. Các từ xuất hiện nhiều ở một mức nào đó như ở phần 2.3.1 sẽ có ý nghĩa và thường nêu lên chủ đề của tin tức. Cách tính độ quan trọng của câu theo tần số từ được thực hiện giống với phương pháp TF-IDF đã nói ở trên.

Theo các lập luận trên, Edmundson đã đề xuất ra một công thức với các tham số tùy biến để kết hợp các yếu tố mà ông cho là ảnh hưởng đến độ quan trọng của câu trong văn bản để từ đó phát hiện phần có thể coi là tóm tắt của văn bản:

$$Weight(S) = \alpha.Title(S) + \beta Cue(S) + \gamma.Keyword(S) + \delta.Possition(S)$$

Các câu có trọng số cao nhất sẽ được đưa vào tóm tắt. Trong phương trình trên:

- Các tham số được điều chỉnh phù hợp bằng cách sử dụng tập huấn luyện.

- Trọng số Cue của câu: β (Trọng số Cue của mỗi từ trong câu)

- So sánh mỗi từ trong câu với từ điển Cue.

- Các từ Title được cho trọng số lớn hơn các từ Heading.

Trọng số vị trí của câu:

- Các câu của đoạn đầu tiên được đánh dấu trọng số O_1

- Các câu của đoạn cuối cùng đực đánh dấu trọng số O_2

- Câu đầu tiên trong một đoạn được đánh dấu trọng số O_3

- Câu cuối cùng của đoạn được dánh dấu trọng số O_4

Thứ tự trọng số của câu: $O1 + O2 + O3 + O4$

Trong thực tế, đây chưa phải là phương pháp tối ưu nhất trong việc ứng dụng phương pháp này và một số cải tiến sẽ được giới thiệu trong chương tiếp theo của luận văn.

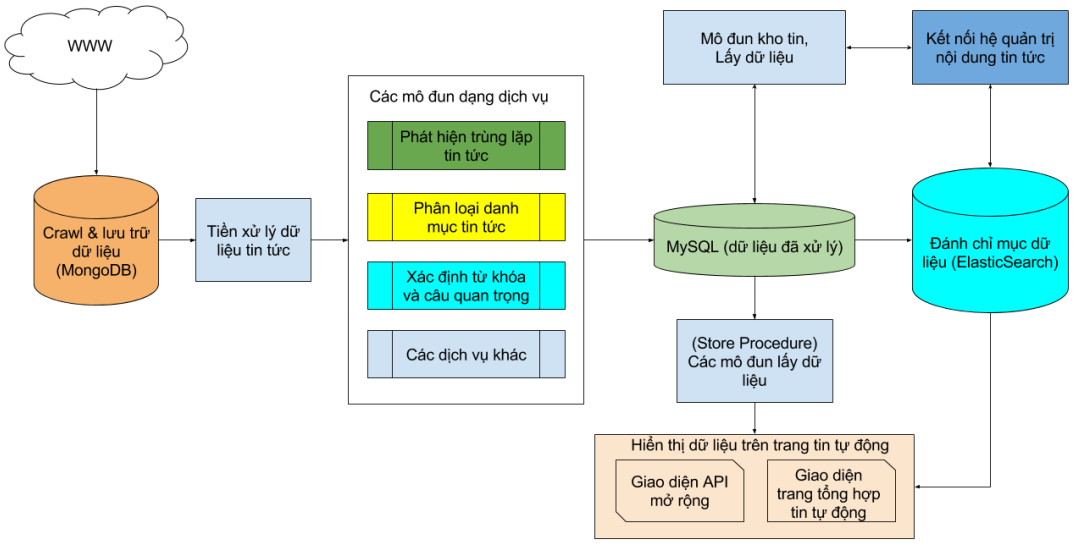
2.4. Tổng kết

Chương này tập trung trình bày các phương pháp tiếp cận cho các bài toán xử lý trùng lặp, bài toán phân loại tin tức, bài toán xác định từ khóa quan trọng và chọn câu tóm tắt cho tin tức, trong mỗi phương pháp đều có nhận xét hữu ích tạo tiền đề cho chương tiếp theo triển khai đề xuất áp dụng mô hình thực tế xử lý giải quyết các bài toán.

Chương 3. ĐỀ XUẤT GIẢI PHÁP VÀ CẢI TIẾN ÁP DỤNG GIẢI QUYẾT CÁC BÀI TOÁN TRONG THỰC TẾ

3.1. Hệ thu thập tin tức tự động mở rộng

Dựa theo cơ sở lý thuyết, những đánh giá trong quá trình tìm hiểu tài liệu, cũng như quá trình triển khai của các hệ thống, công trình nghiên cứu trước. Hệ thống thu thập tin tức mở rộng với các mô đun mới được thể hiện như hình dưới đây:



Hình 3.1. Mô hình tổng quan hệ tổng hợp tin tự động

Hệ thu thập tin tức tự động trong khuôn khổ đề tài được đề xuất như mô hình 3.1 gồm các thành phần chính:

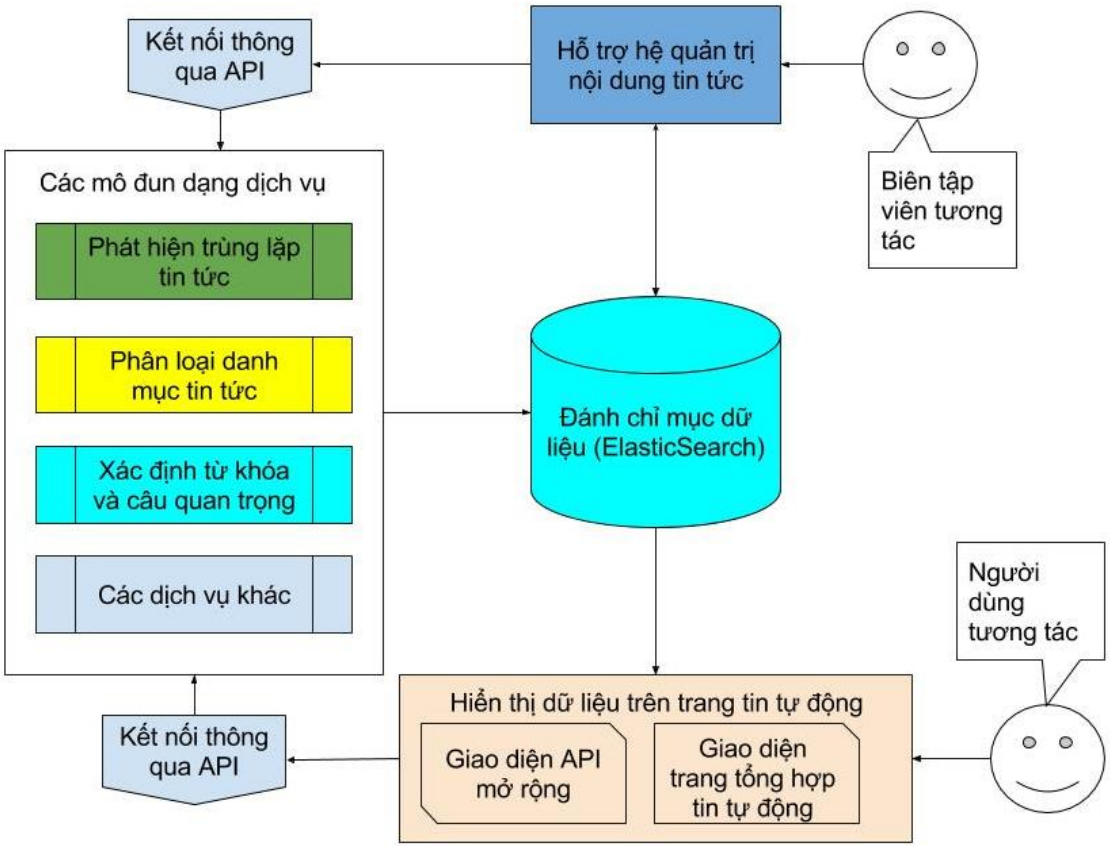
- Crawler phân tán giữ nhiệm vụ thu thập dữ liệu liên tục một cách tự động, cập nhật liên tục.
- Các giai đoạn xử lý dữ liệu bao gồm:
    - Tiền xử lý dữ liệu: chuẩn hóa phông chữ, chuẩn hóa văn bản lọc các kí tự phần thừa, xử lý tách từ, tách câu.
    - Dữ liệu được xử lý trùng lặp bằng dịch vụ xử lý trùng lặp.
    - Bộ khai phá dữ liệu làm nhiệm vụ khai phá phân tích dữ liệu nhằm phân loại, từ khóa quan trọng, tóm tắt nội dung của văn bản, ngoài ra còn các dịch vụ khác chạy kèm như phát hiện sắc thái tin tức, bộ phát hiện xu hướng tin tức,…
- Dữ liệu sau khi xử lý được lưu vào cơ sở dữ liệu cố định và đánh chỉ mục tự động lên máy tìm kiếm phục vụ việc tìm kiếm tra cứu nhanh.

- Các mô đun kho tin, các mô đun thao tác dữ liệu phục vụ việc thao tác với dữ liệu xử lý được, các mô đun ở phục vụ lấy dữ liệu được viết bởi các thủ tục (Stored Procedure) là một tập hợp các câu lệnh truy vấn có cấu trúc dùng để thực thi một nhiệm vụ lấy dữ liệu nhất định.

Các luồng đi được miêu tả theo hướng mũi tên như hình 3.1:

Dữ liệu tin tức sau khi được thu thập bởi trình thu thập dữ liệu (crawler) phân tán được lưu vào cơ sở dữ liệu dưới dạng thô, sau đó được tiền xử lý bởi dịch vụ tiền xử lý và thực hiện việc phát hiện trùng lặp, phân danh mục tự động cho tin tức và xác định từ khóa quan trọng cũng như đề xuất sẵn câu có thể chọn làm câu tóm tắt nếu crawler tin tức không lấy được phần tóm tắt (hay phần mô tả). Sau đó dữ liệu được lưu trữ phục vụ các bên khai thác dữ liệu đồng thời đánh chỉ mục (index) lên elasticsearch (một opensource khá mạnh về máy tìm kiếm) phục vụ việc tra cứu dữ liệu nhanh. Dữ liệu này được chia sẻ xuống trang tin tức tổng hợp tự động, cũng như được chia sẻ đến hệ quản trị nội dung tin tức giúp phục vụ phóng viên biên tập tin tức tổng hợp tin tác nghiệp.

Ngoài ra bộ xử lý dữ liệu cũng cung cấp API liên lạc trực tiếp với hệ quản trị nội dung tin tức phục vụ biên tập viên, phóng viên có thể kiểm tra trùng lặp bài tự viết để tham khảo nguồn bài tương tự, tự động chọn từ khóa quan trọng phù hợp làm tags (từ khóa chính của bài viết). Chi tiết được mô tả ở hình 3.2 dưới đây.



Hình 3.2. Mô hình dịch vụ xử lý phục vụ người dùng thông qua API

Hơn thế nữa bộ xử lý dữ liệu cũng cung cấp cho phía trang tin tự động hàng loạt API phục vụ bên thứ ba, người sử dụng API có thể kiểm tra trùng lặp trực tiếp trên đối sánh với dữ liệu đã có của hệ thống, cũng như đề xuất từ khóa quan trọng và câu tóm tắt của một bản tin ngẫu nhiên gửi từ phía người sử dụng API. Mô hình giải quyết thực tế chi tiết các mô-đun sẽ được giới thiệu trong các mục tiếp theo của luận văn.

3.2. Giải quyết bài toán trùng lặp tin tức

3.2.1. Yêu cầu thực tế bài toán xử lý trùng lặp tin tức

Hình 3.3. Minh họa thực tế ứng dụng bài toán xử lý trùng lặp

Trong thực tế việc xử lý trùng lặp được nghiên cứu trong đề tài nhằm đáp ứng ba yêu cầu chính sau đây:

- Crawler đánh dấu tin trùng lặp trong kho.
- Biên tập viên tham khảo bài liên quan.
- Cảnh báo việc BTV đạo văn.

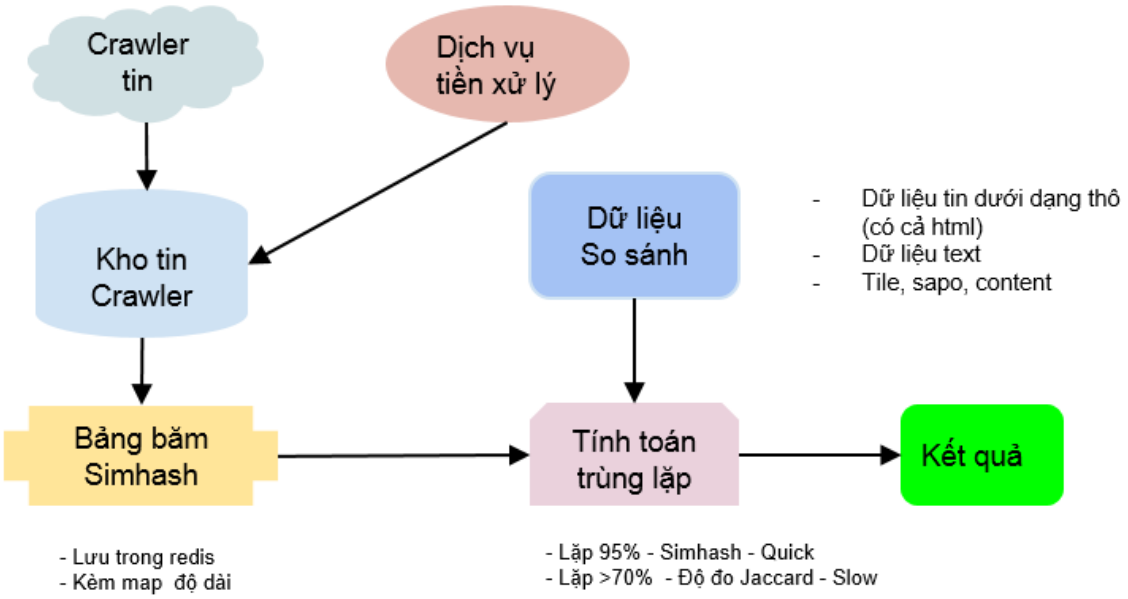
Crawler khi thu thập lượng lớn tin tức sẽ đánh dấu các tin trùng lặp phục vụ mục đích loại bỏ trùng lặp nội dung tin tức hiển thị, và hiển thị thống kê việc trùng lặp bằng phương pháp lưu vết (tức là lưu id của bản tin gốc) – với quy ước bản tin vào trước được coi là bản tin gốc.

Biên tập viên viết tin tức cũng có thể khi truy vấn một bài tìm các bài liên quan có độ giống nhau ở tỉ lệ nhất định để tham khảo phục vụ việc tổng hợp tin tức, vì việc truy vấn trùng lặp theo tỉ lệ % là khó khăn đòi hỏi tính toán lớn nên ở đây giới hạn 10 bản ghi giống gần nhất và chỉ xét tỉ lệ trùng lớn hơn 65%.

Một chức năng khác hỗ trợ hệ thống CMS viết báo là cảnh báo việc Biên tập viên, phóng viên copy bài của người khác, với mức trùng bài 70% sẽ được cảnh báo.

3.2.2. Mô hình giải pháp thực tế

Như đã phân tích ở chương 2, phần 2.1.5 Simhash ở đây là biện pháp tối ưu phục vụ cho crawler với nhiệm vụ kiểm tra trùng lặp hàng triệu dữ liệu, thời gian thực. Mô hình triển khai sau đây được áp dụng thực tế.

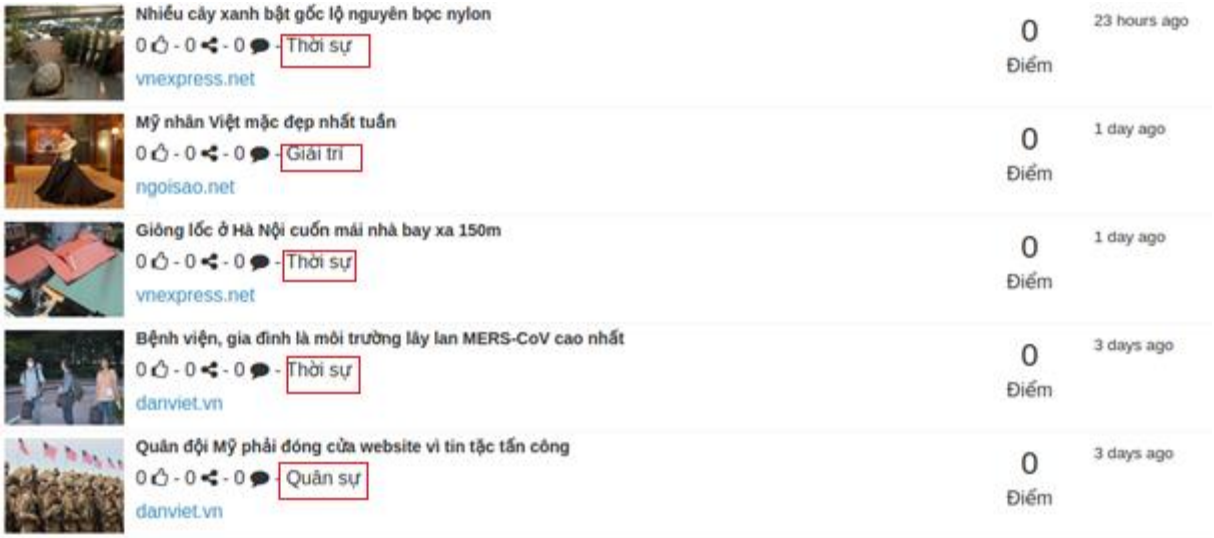


Hình 3.4. Minh họa thực tế triển khai bài toán xử lý trùng lặp

Dữ liệu tin tức sau khi thu thập sẽ được tiền xử lý và lấy Simhash tiêu đề và Simhash phần nội dung, Simhash tiêu đề được dùng dãy bit 32 bit do tiêu đề thường ngắn, Simhash nội dung dùng dãy bit Simhash 64 bit và được lưu thành các hoán vị mô tả như trong chương 2 mục 2.1.5 trong, và được lưu trên bộ nhớ memory – Redis Cluster. Khi bản ghi mới thu thập hệ thống sẽ tính toán song song và trả về kết quả có trùng lặp không trong thời gian chấp nhận được. Mô hình sẽ được đánh giá về mặt hiệu năng tốc độ so với một số thuật toán khác ở chương tiếp theo.

3.3. Giải quyết bài toán phân loại tin tức

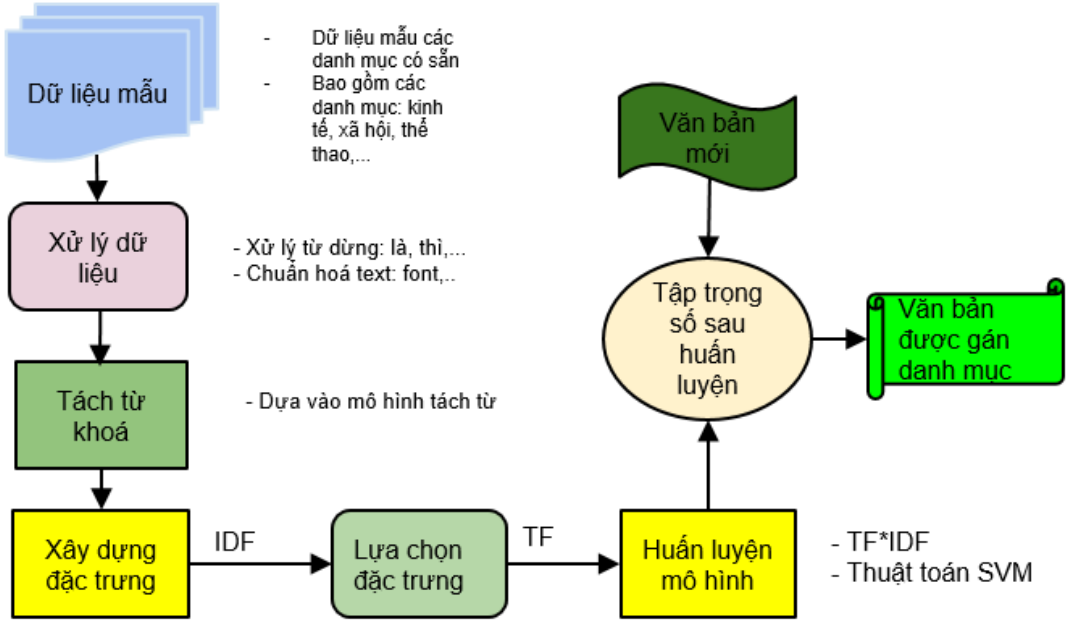
3.3.1. Yêu cầu bài toán thực tế



Hình 3.5. Minh họa thực tế ứng dụng bài toán phân loại tin tức

Bài toán thực tế phân loại tin tức như đã nói rõ ở chương một có thể quy về bài toán phân lớp văn bản thuần túy nhằm mục đích chính là để tổ chức sắp xếp tin đúng theo danh mục, phục vụ biên tập viên tra cứu theo danh mục đặc thù riêng của biên tập viên báo. Việc phân loại cũng có ý nghĩa quan trọng nhằm đáp ứng nhu cầu phân danh mục tin tức cho trang tin tổng hợp tự động.

3.3.2. Mô hình giải pháp thực tế



Hình 3.6. Mô hình triển khai thực tế triển khai bài toán phân loại tin tức

Dữ liệu mẫu sau khi được tiền xử lý sẽ được tách từ khóa và xây dựng đặc trưng, đặc trưng ở đây đây được thử nghiệm bằng TF-IDF trọng số từ trong nội dung tin và đưa vào triển khai huấn luyện mô hình với thuật toán SVM để tạo ra mô hình (model) sau huấn luyện.

Một bản tin mới chưa được phân danh mục được xử lý và biểu diễn dưới dạng Vector với trọng số cũng là TF-IDF sẽ được tham chiếu với mô hình sau huấn luyện để kết luận văn bản đó thuộc danh mục nào.

Một số yếu tố đóng góp quyết định đến chất lượng của bộ phân lớp:

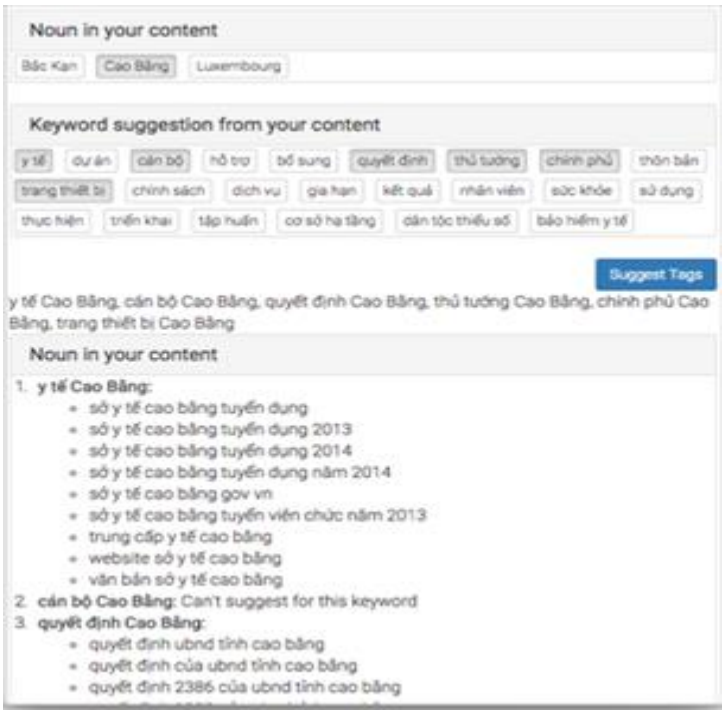
- Dữ liệu mẫu được cần lựa chọn rất kĩ để đảm bảo đặc trưng của từng lớp văn bản bộ dữ liệu mẫu trong luận văn được sự hỗ trợ của nhóm biên tập viên chọn lọc từ các danh mục của báo điện tử VNExpress. Tiêu chuẩn của dữ liệu mẫu cũng được xem xét, một tin mẫu được xác định là chuẩn với độ dài là lớn hơn 300 và nhỏ hơn 4000 kí tự - một bản tin không quá ngắn và cũng không quá dài.

- Các chủ đề được phân loại kép chia các chủ đề ra các danh mục nhỏ hơn, ví dụ tin tức được chia thành 2 danh mục lớn là tin trong nước và tin nước ngoài, trong danh mục tin trong nước sẽ có những danh mục con khác, và danh mục tin nước ngoài cũng vậy.
- Việc lựa chọn đặc trưng cũng được xem xét chỉ nên lấy phần tiêu đề và mô tả, và các câu quan trọng trong bài, hay cả nội dung bài để xây dựng nên vector bản tin.
- Với bộ phân lớp sử dụng SVM cần thực hiện tùy chỉnh các tham số để kiểm nghiệm nhằm đạt được kết quả phân loại tốt nhất.

3.4. Giải quyết bài toán xác định từ khóa quan trọng và chọn câu tóm tắt

3.4.1. Yêu cầu bài toán thực tế

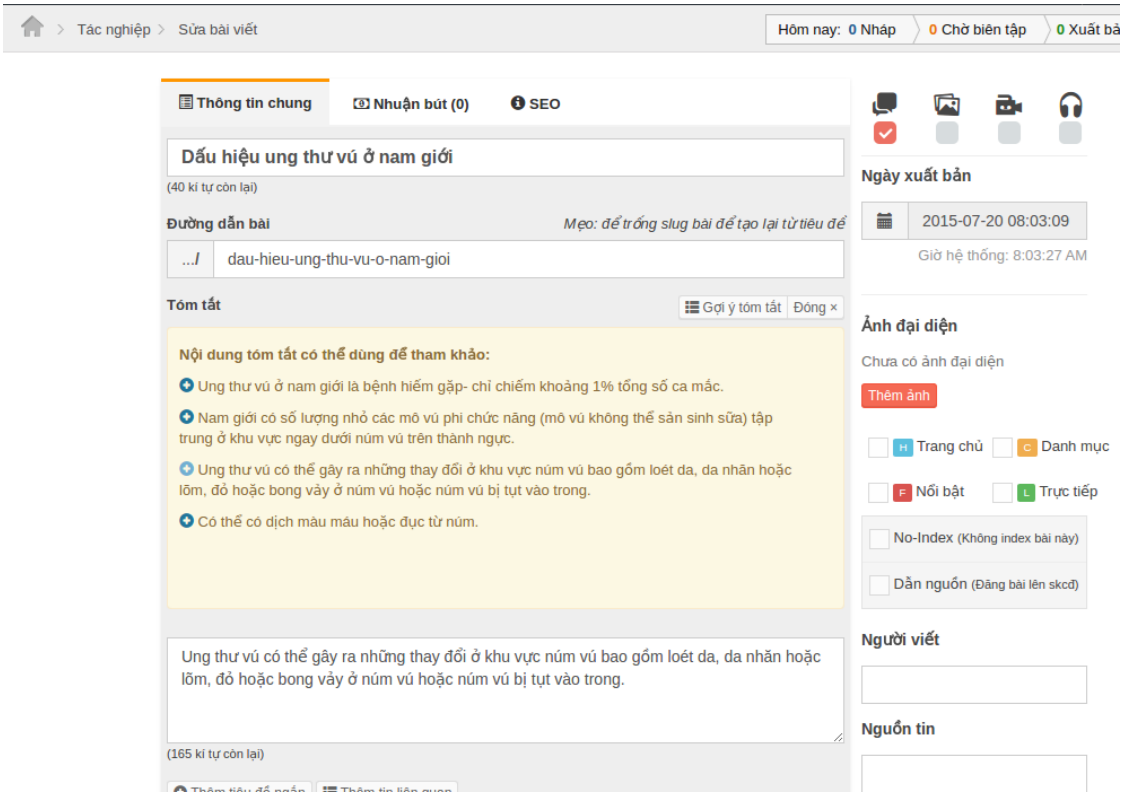
Bài toán xác định từ khóa quan trọng



Hình 3.7. Minh họa thực tế ứng dụng xác định từ khóa quan trọng

Mục đích thực tế của bài toán xác định từ khóa quan trọng là hỗ trợ việc tóm tắt đại ý của nội dung tin và phục vụ việc tạo ra các chủ đề con liên kết sự liên quan giữa các bài báo, hỗ trợ tối ưu máy tìm kiếm.

Bài toán chọn câu tóm tắt

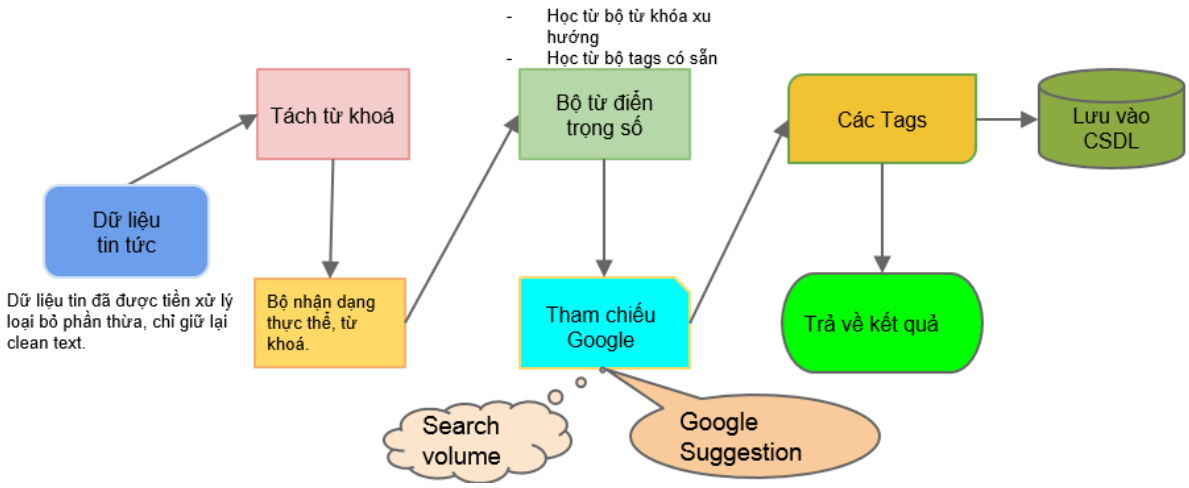


Hình 3.8. Minh họa thực tế ứng dụng chọn câu tóm tắt

Đối với một số nội dung không lấy được đoạn trích dẫn tóm tắt nội dung, hệ thống có thể tự tóm tắt một đoạn trích dẫn nội dung tóm tắt cho bài viết. Hoặc hỗ trợ biên tập viên, phóng viên đề xuất câu dùng làm câu tóm tắt mô tả của bản tin.

3.4.2. Mô hình giải pháp thực tế

Bài toán xác định từ khóa quan trọng

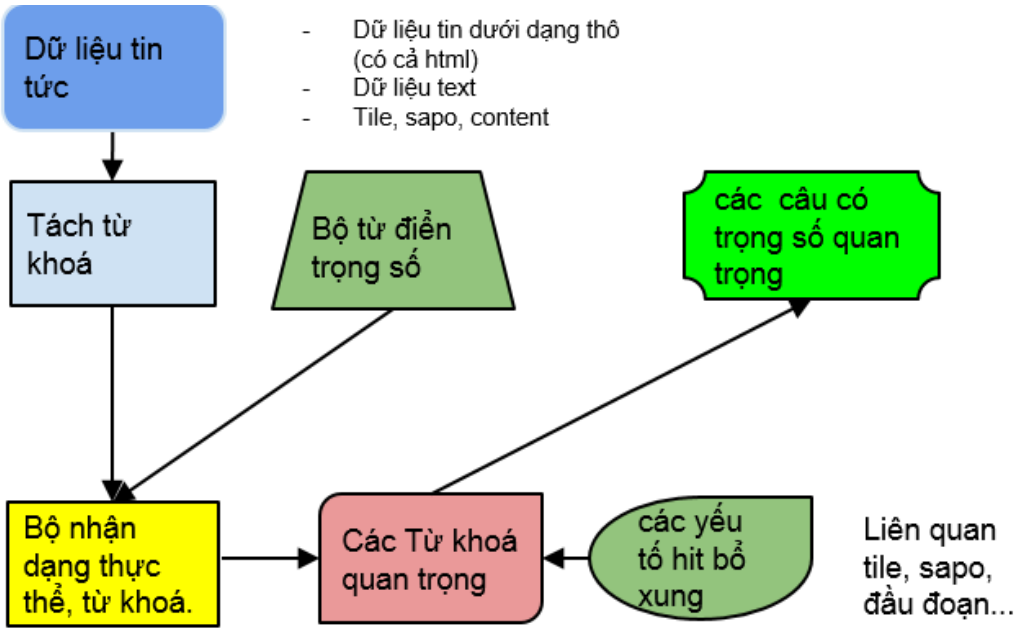


Hình 3.9. Mô hình thực tế bài toán xác định từ khóa quan trọng

Các đóng góp quan trọng trong bộ xác định từ khóa quan trọng:

- Tham chiếu vị trí trong câu, vị trí trong tiêu đề, phần mô tả và nội dung, sử dụng thêm trọng số Tf-idf.
- Tham chiếu từ bộ từ khóa(Tags) có sẵn khi thu thập dữ liệu từ internet, và bộ các từ khóa từ việc phân tích xu hướng thông tin.
- Tham chiếu kết quả Google Suggestion và Search Volumne để lấy lượng tìm kiếm, lượng tìm kiếm càng cao có nghĩa là từ khóa có mức độ quan trọng càng cao.

Bài toán chọn câu tóm tắt



Hình 3.10. Mô hình thực tế bài toán xác định câu tóm tắt

Bài toán chọn câu tóm tắt trong đề tài sử dụng kết hợp 2 phương pháp Tf-idf và Edmundson, vừa có điểm trọng số cho từ khóa, câu có nhiều từ khóa quan trọng, vừa xác định độ tương quan giữa vị trí của câu, nằm trong tiêu đề, phần mô tả, nội dung, cuối đoạn đầu đoạn được tính toán hợp lý để đề xuất ra danh sách câu quan trọng trong bài tin. Việc chọn tỉ lệ câu đề xuất trên tổng số câu trong bản tin cũng là vấn đề quyết định đến độ chính xác của bản tin. Với hệ thống hiện tại sau các kết quả kiểm nghiệm thực tế 5 câu sẽ lấy đại diện một câu quan trọng phù hợp với dữ liệu tin tức.

3.5. Tổng kết

Từ những kết quả nghiên cứu từ chương 2, luận văn chỉ ra phương pháp phù hợp cho bài toán thực tế được chọn lựa để đưa vào thực nghiệm. Sau đó, phát biểu, mô tả mô hình chi tiết và cách giải quyết cho các bài toán, cũng như một số đóng góp quan trọng cải thiện độ chính xác kết quả. Phần tiếp theo của luận văn sẽ tiến hành đánh giá

các kết quả thực nghiệm đạt được sau khi áp dụng các mô hình.

Chương 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

Ở chương này, luận văn sẽ tiến hành quá trình thực nghiệm và đánh giá kết quả đề xuất dựa trên các bài toán. Với đặc điểm riêng của mỗi bài toán sẽ có những cách đánh giá, so sánh riêng phù hợp với yêu cầu thực tế, đồng thời đảm bảo ý nghĩa khoa học của bài toán.

4.1. Môi trường thực nghiệm và các công cụ sử dụng trong thực nghiệm

Cấu hình phần cứng, phần mềm các gói đi kèm thực nghiệm được sử dụng trong luận văn được mô tả trong hai bảng sau đây:

Công cụ phần cứng được sử dụng:

Bảng 4.1 Cấu hình phần cứng thực nghiệm

Stt	Thành phần	Chỉ số
1	CPU	Intel Core i5 4460 3.4GHZ
2	RAM	8GB
3	Hệ điều hành	Ubuntu 14.04
4	Bộ nhớ ngoài	500GB

Bảng 4.2 Các công cụ phần mềm được sử dụng

STT	Tên phần mềm	Chức năng	Nguồn
1	Apache Nutch 1.11	Tải dữ liệu từ các website	http://nutch.apache.org/
2	Elasticsearch	Index, lưu trữ dữ liệu	https://github.com/elastic/elasticsearch
3	Eclipse Java EE Luna	Tạo môi trường để viết chương trình	https://eclipse.org/downloads/

4	Readability	Trích xuất nội dung	https://github.com/mozilla/readability
5	vnSentDetector 2.0.0	PhươngLH – Trích xuất câu trong đoạn văn bản.	http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnSentDetector
6	vn.hus.nlp.tokenizer-4.1.1	PhươngLH - Tách từ trong văn bản	http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTokenizer
7	LibSVM 3.21	Chih-Chung Chang and Chih-Jen Lin – Phục vụ phân loại văn	https://www.csie.ntu.edu.tw/~cjlin/libsvm/
8	Redis	Cache Simhash vào memory, share giữa các cụm	http://redis.io/

4.2. Quá trình thu thập dữ liệu tin tức và tiền xử lý

4.2.1. Thu thập dữ liệu tin tức

Dữ liệu được thu thập với phần mềm mã nguồn mở Apache Nutch 1.11 cấu hình chạy phân tán, ở Nutch được tùy biến thêm 2 plugin kế thừa việc trích xuất dữ liệu và việc đánh chỉ mục dữ liệu lên Elasticsearch (một dạng máy tìm kiếm linh động với mức độ tùy biến tìm kiếm cao).

- Plugin trích xuất dữ liệu được implement từ Readability code, tùy biến để trích xuất được các hạng mục chính của tin là: tiêu đề, phần mô tả (tóm tắt), nội dung tin, tên tác giả và ngày đăng tin.
- Plugin index tùy biến giúp index thêm các trường cần thiết mới lên ElasticSearch.

Dữ liệu được thu thập cũng được chuẩn hóa lại font chữ, lọc các tin nội dung ảnh, video, đảm bảo dữ liệu text đã được chuẩn hóa (normalize–filter) phục vụ cho việc xử lý dữ liệu.

4.2.2. Tiền xử lý dữ liệu

Với dữ liệu được lấy về sẽ được các dịch vụ tự động tiến hành xử lý tách từ, tách

câu bằng hai công cụ mã nguồn mở là vnSentDetector 2.0.0 và vnTokenizer 4.1.1, tiếp đó bản tin sẽ được lấy dấu đại diện simhash – simhash được lưu trữ riêng dưới dạng đặc biệt để phục vụ việc phát hiện trùng lặp, ngoài ra bản tin còn được xử lý lấy từ khóa quan trọng(tags) và chọn một vài câu đề xuất tóm tắt nếu bản tin lấy về không có câu tóm tắt. Với từ khóa đã được tách, và URL gốc bản tin cũng được phân loại một cách tự động. Mô hình giải quyết chi tiết cho mỗi bài toán trong luận văn đã được nêu chi tiết trong chương 3, phần tiếp theo sẽ nêu lên phương pháp đánh giá và kết quả đánh giá của từng bài toán.

4.3. Đánh giá phát hiện trùng lặp tin tức

4.3.1. Phương pháp đánh giá.

Trong thực tế có những thuật giải kiểm tra trùng lặp cho kết quả tốt hơn việc sử dụng hàm băm Simhash để tạo đại diện. Tuy nhiên trong khuôn khổ luận văn tác giả đánh giá việc sử dụng Simhash trên phương diện phục vụ cho Crawler kiểm tra trùng lặp nên tốc độ kiểm tra trùng lặp là yếu tố được ưu tiên hàng đầu.

4.3.2. Kết quả đánh giá.

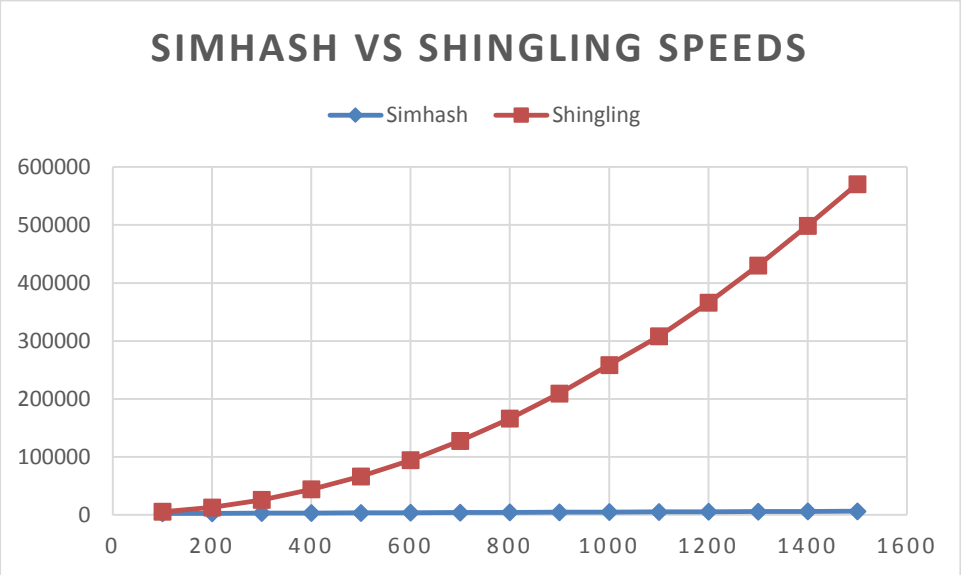
Trong thí nghiệm đánh giá, chúng ta sẽ so sánh tốc độ của hai thuật toán Simhash và Shingling trên tập dữ liệu với số lượng dữ liệu tăng dần từ 100 bản ghi lên đến 1500 bản ghi, Simhash ở đây được lấy dưới dạng Simhash 32bit và Shingling lấy dạng token sau khi đã tách từ, kết quả thu được cho dưới bảng 4.3:

Bảng 4.3 Thống kê thời gian chạy với simhash và shingling

NumRecords	Simhash(ms)	Shingling(ms)
100	2466	5389
200	2692	12851
300	3052	25841
400	3253	43918
500	3437	66225
600	3664	94262
700	3869	127710
800	4140	166124
900	4419	209418
1000	4697	258469
1100	4985	307823
1200	5261	366019
1300	5575	429911

1400	5935	498562
1500	6240	570506

Mô hình hóa dưới dạng biểu đồ:



Hình 4.1. So sánh tốc độ simhash và shingling

Thuật toán Shingling thể hiện rõ độ phức tạp tính toán theo thời gian là $O(n^2)$ trong khi áp dụng Simhash cho thấy kết quả tốt rõ rệt đúng với lý thuyết thời gian chạy logarit. Hoàn toàn phù hợp với việc áp dụng vào thực tế.

4.4. Đánh giá bộ phân loại tin tức

4.4.1. Phương pháp đánh giá.

Trước tiên cần nói thêm về quá trình thu thập dữ liệu của crawler, các danh mục thuộc diện tin văn bản được lấy và được đánh dấu riêng nằm trong 12 danh mục bao gồm:{"cong-nghe","giai-tri","giao-duc","kham-pha","kinh-te","phap-luat","quan-su","suc-khoe","tam-su","the-gioi","the-thao","xe-360"}

Việc đánh giá thuật toán phân loại sẽ sử dụng độ đo precision/recall và F1 để đánh giá bộ học dữ liệu sẽ bao gồm 56400 văn bản được chọn sẵn danh mục để học dựa trên nguồn VNExpress, 54000 văn bản thuộc 12 chủ đề (tương đương với 4500 bản tin/1 chủ đề) sẽ được dùng để huấn luyện(train), và 2400 văn bản sẽ được dùng để kiểm định (test), trong khuôn khổ luận văn thực hiện đánh giá trên phương diện việc sử dụng SVM thuần túy với nội dung bản tin và việc cải tiến cho kết quả thực tế ra sao, chi tiết sẽ được nêu tại phần kết quả.

Sau đây là một số độ đo được sử dụng trong đánh giá:

Ma trận nhầm lẫn (Confusion Matrix)

TP_i **:** Số lượng các bản tin thuộc lớp c_i được phân loại chính xác vào lớp c_i

FP_i: Số lượng các bản tin không thuộc lớp c_i bị phân loại nhầm vào lớp c_i

TN_i: Số lượng các bản tin không thuộc lớp c_i được phân loại (chính xác)

FN_i: Số lượng các bản tin thuộc lớp c_i bị phân loại nhầm (vào các lớp khác c_i)

Độ đo Precision và recall

Hay còn gọi là **Độ chính xác** và **Độ bao phủ, Precision** là việc thể hiện trong tập tìm được thì bao nhiêu cái (phân loại) đúng. **Recall** là việc thể hiện trong số các tồn tại, tìm ra được bao nhiêu cái (phân loại). Đây là hai độ đo phổ biến, rất hay được sử dụng để đánh giá các hệ thống phân loại văn bản.

- ***Precision*** đối với lớp c_i là một lớp trong tập các lớp C ={c_1, c_2, …, c_n}

$$Precision = \frac{tp}{tp + fp}$$

Tổng số các bản tin thuộc lớp c_i được phân loại chính xác chia cho tổng số các bản tin được phân loại vào lớp c_i

- ***Recall*** đối với lớp c_i

$$Recall = \frac{tp}{tp + fn}$$

Tổng số các bản tin thuộc lớp c_i được phân loại chính xác chia cho tổng số các ví dụ thuộc lớp c_i

Recall cũng được gọi là True Positive Rate hay Sensitivity (độ nhạy), và ***precision*** cũng được gọi là Positive predictive value (PPV); ngoài ra, ta có các độ đo khác như True Negative Rate và Accuracy. True Negative Rate cũng được gọi là Specificity.

Độ đo *F_1*

Tiêu chí đánh giá F_1 là sự kết hợp của hai tiêu chí đánh giá *Precision* và *Recall*

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

F_1 là một **trung bình điều hòa (harmonic mean)** của các tiêu chí ***Precision*** và **Recall.**

F_1 có xu hướng lấy giá trị gần với giá trị nào nhỏ hơn giữa hai giá trị Precision và Recall, F_1 có giá trị lớn nếu cả hai giá trị Precision và Recall đều lớn.

4.4.2. Kết quả đánh giá.

Kết quả sau khi tiến hành phân loại sử dụng SVM kernel linear với dữ liệu văn bản bao gồm nội dung văn bản thuần túy chưa có cải tiến.

Bảng 4.4 Kết quả phân loại khi chưa được cải tiến

CatNo	Category	Precison	Recall	F1
1	*cong-nghe*	75.5	82.51	78.85
2	*giai-tri*	66	75.43	70.4
3	*giao-duc*	79	92.4	85.18
4	*kham-pha*	65	71.43	68.06
5	*kinh-te*	66.5	73.08	69.63
6	*phap-luat*	76.5	83.61	79.9
7	*quan-su*	46.5	93.94	62.21
8	*suc-khoe*	67.5	87.66	76.27
9	*tam-su*	89	84.36	86.62
10	*the-gioi*	88.5	40.69	55.75
11	*the-thao*	83	91.71	87.14
12	*xe-360*	81.5	66.8	73.42
	Avg	**73.71**	**78.64**	**74.45**

Áp dụng các cải tiến vào phân loại xác định chủ đề văn bản, bằng các biện pháp đã được nêu trong chương 3, kết quả đạt được được cho trong bảng 4.5:

Bảng 4.5 Kết quả phân loại khi được cải tiến

CatNo	Category	Precison	Recall	F1
1	*cong-nghe*	80.9	90.58	85.47
2	*giai-tri*	81.7	83.29	82.49
3	*giao-duc*	82.1	93.26	87.32
4	*kham-pha*	73.5	81.4	77.25
5	*kinh-te*	76.9	77.25	77.07
6	*phap-luat*	77.6	88.92	82.88
7	*quan-su*	73.2	95.97	83.05

8	*suc-khoe*	84.9	94.04	89.24
9	*tam-su*	91.2	93.58	92.37
10	*the-gioi*	88.7	93.41	90.99
11	*the-thao*	92.6	92.62	92.61
12	*xe-360*	73.9	88.24	80.44
	Avg	**81.43**	**89.38**	**85.1**

Kết quả ở bảng trên cho thấy, toàn bộ kết quả phân loại đã được cải thiện cả về độ chính xác và độ hồi tưởng, độ chính xác **Precision** trung bình từ *73.71%* lên đến *81.43%*, độ hồi tưởng **Recall** cũng tăng từ *78.64%* lên tới *89.38%*, kéo theo đó độ đo $\mathbf{F_1}$ cũng tăng khá rõ rệt.

4.5. Đánh giá kết quả xác định từ khóa quan trọng và chọn câu tóm tắt

4.5.1. Phương pháp đánh giá.

Việc đánh giá bài toán này được thực hiện một cách thủ công một phần dựa trên ý kiến chuyên gia (expert judgment) bởi đặc điểm đặc biệt của bài toán: Để đánh giá bài toán xác định từ khóa quan trọng (tags) để phục vụ vấn đề nêu bật chủ đề của bản tin cũng như hỗ trợ việc phục vụ tối ưu máy tìm kiếm (SEO) và chọn câu tóm tắt cho chủ đề bản tin nếu bản tin thiếu phần tóm tắt khá phức tạp đòi hỏi người đánh giá vừa có kinh nghiệm về SEO và vừa có kinh nghiệm trong biên tập bản tin. Luận văn sử dụng việc tổng hợp kết quả đánh giá từ ba người trong ban biên tập viên đã được đào tạo kĩ năng SEO để thực hiện đánh giá với mỗi bạn 100 bản tin. Tổng số bản tin được lấy từ khóa quan trọng, và chọn câu tóm tắt là 300 bản tin, tỉ lệ chọn (nén câu tóm tắt là 5:1)[2]. Chi tiết kết quả thu được có trong phần kết quả đánh giá.

4.5.2. Kết quả đánh giá.

Kết quả đánh giá thủ công ba lần do ba biên tập viên có kinh nghiệm SEO được đào tạo bài bản cả về mảng biên tập lẫn kinh nghiệm về đánh giá nội dung được cho ở bảng 4.6.

Bảng 4.6 Thống kê tỉ lệ tag và tóm tắt đạt yêu cầu

	Tỉ lệ tags đạt	**Tỉ lệ tóm tắt đạt**
Lần 1 (100 tin)	73%	71%
Lần 2 (100 tin)	76%	69%
Lần 3 (100 tin)	78%	64%
Bình Quân	76%	68%

Giải thích:

Tỉ lệ Tags đạt 76% tức là trong 100 bản tin được lấy Tags tự động thì có 76 bản tin đạt yêu cầu theo ý kiến của người đánh giá, có nghĩa là phần tags chứa các từ khóa này có thể thay thế người sử dụng phần tag nội dung tự động không cần người biên tập phải can thiệp, dùng làm tags phản ánh nội dung chính của bản tin.

Tỉ lệ tóm tắt đạt 68% tức là trong 100 bản tin lấy tổ hợp câu tóm tắt tự động thì có 68% tổ hợp câu có chứa một câu có thể chọn đại diện hỗ trợ biên tập viên đặt làm câu tóm tắt của bản tin.

Qua đánh giá lấy ý kiến, sau ba lần với kết quả bình quân cho việc chọn tags tự động là 76% và việc đề xuất câu tóm tắt tự động là 68% được đánh giá cao và có khả năng triển khai thực tế, ứng dụng vào hệ thống CMS tin tức trong tương lai.

4.6. Tổng kết

Chương này tác giả đã trình bày các kết quả thực nghiệm chứng minh phương pháp đề xuất trong chương 3. Kết quả thực nghiệm tập trung vào ba bài toán chính đó là kiểm tra trùng lặp, phân loại tin tức và sinh các từ khóa nội dung chính, sinh câu đề xuất tóm tắt của văn bản. Kết quả thực nghiệm cho thấy phương pháp đề xuất phù hợp ở mức chấp nhận được và đã có những phần kết quả khả quan hơn sau thi được đóng góp cải tiến.

TỔNG KẾT

Kết quả đạt được

Luận văn đã trình bày các kiến thức cơ bản về phát hiện trùng lặp, phân loại tin tức, xác định từ khóa quan trọng và đề xuất câu tóm tắt cho tin tức trên miền dữ liệu tiếng Việt. Bên cạnh đó, luận văn đã trình bày chi tiết các phương pháp tiếp cận bài toán, cũng như hướng giải quyết và kết quả thực tế. Với bài toán phát hiện trùng lặp tin tức từ phía Crawler luận văn đã đề cập phân tích ưu nhược điểm của một số phương pháp phổ biến để phát hiện trùng lặp và sau đó đề xuất mô hình giải quyết bài toán với giải thuật SimHash từ đó đánh giá và so sánh với thuật toán phát hiện trùng lặp phổ biến là shingling. Với bài toán phân loại luận văn cũng đưa ra một vài bài toán phân loại cũng như lý do sử dụng học máy bán giám sát với SVM, Cuối cùng là bài toán xác định từ khóa quan trọng, và đề xuất câu đại diện chọn tóm tắt cho tin tức được giải quyết bằng việc tổng hợp các biện pháp Edmundson và TF-IDF.

Các kết quả cho thấy phương pháp sử dụng Simhash để kiểm tra trùng lặp có tốc độ tính toán tăng theo hàm loragit cải thiện hơn rất nhiều so với $O(n^2)$ của phương pháp shingling, cụ thể khi tập dữ liệu chỉ lên tới 1500 bản tin tốc độ của SimHash đã nhanh hơn tốc độ của Shingling tới 91,4 lần. Phương pháp SVM tích hợp vào mô đun phân loại cũng cho kết quả tốt sau khi đóng góp một số cải tiến so với sử dụng SVM thuần túy trên tập dữ liệu, với kết quả tốt. Sử dụng độ đo chính xác (precision), độ đo hồi tưởng (recall), và độ đo F-1 (F-1 measured) để đo lường kết quả cho thấy: độ đo chính xác (89.38%), độ đo hồi tưởng (89.3%), và độ đo F-1 (85.1%). Với bài toán tự động đề xuất tags bao gồm các từ khóa quan trọng và đề xuất một trong những câu có thể chọn làm tóm tắt cũng cho một kết quả tích cực sau khi áp dụng các biện pháp cải tiến ở chương 3, tỉ lệ chấp nhận được ở góc độ đánh giá của người được đào tạo (expert) trong lĩnh vực biên tập và SEO cho thấy tỉ lệ tags đạt 76% và tỉ lệ chọn câu tóm tắt chấp nhận được đạt 68%.

Hạn chế

Mặc dù kết quả đạt được khả quan tuy nhiên các giải pháp trong luận văn cũng không tránh khỏi một số hạn chế và nhược điểm cần khắc phục chẳng hạn như:

Việc lấy hàm đại diện Simhash là việc ánh xạ từ tập vô hạn sang tập hữu hạn vậy nên vẫn xuất hiện tỉ lệ trùng Simhash với hai văn bản khác nhau, điều này khiến bộ kiểm tra trùng lặp mất thêm thời gian để kiểm định thêm các trường hợp kể trên do đó tốc độ kiểm tra trùng lặp bị giảm xuống một phần.

Việc phân loại hiện tại phải thiết đặt luật cho Crawler để giới hạn tập danh mục

cụ thể của bản tin phục vụ việc phân danh mục có độ chính xác cao, các tin vắn, tin có chất lượng thấp vẫn chưa được hỗ trợ.

Việc chọn từ khóa tóm tắt(tags) và chọn câu tóm tắt vẫn còn phụ thuộc nhiều vào việc tham chiếu kho từ cũ, kho từ xu hướng có sẵn để tăng cao độ chính xác, mà chưa tự chủ được từ việc dựa vào bản thân của văn bản.

Hướng phát triển

Trong thời điểm tương lai gần, hướng phát triển trước mắt của luận văn là khắc phục những hạn chế khuyết điểm của các mô đun hiện tại và nâng cao khả năng chính xác của các thuật toán, cụ thể là: cải thiện tốc độ hơn nữa việc áp dụng Simhash để ứng phó với môi trường dữ liệu lớn hơn, cải thiện độ chính xác phân loại với nguồn tin tức đa dạng hơn đồng thời nâng cao độ chính xác việc sinh từ khóa, và đề xuất câu tóm tắt.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Bộ Thông tin và Truyền thông (2015), *Tình hình phát triển lĩnh vực báo chí năm 2015*, Hà Nội.

2. Trần Mai Vũ (2009), *Tóm Tắt Đa Văn Bản Dựa Vào Trích Xuất Câu*, Đại Học Quốc Gia Hà Nội, Trường Đại Học Công Nghệ, 2009, tr.4.

Tiếng Anh

3. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze (2009), *Introduction to Information Retrieval*, Cambridge University Press. 2009.

4. Martin Law (2011), *A Simple Introduction to Support Vector Machines*, Michigan State University, Lecture for CSE 802

5. T. Joachims (1999). *Transductive Inference for Text Classification using Support Vector Machines*. International Conference on Machine Learning (ICML), 1999.

6. Jin Huang, Jingjing Lu, Charles X. Ling (2003). *Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy*. The Third IEEE International Conference on Data Mining (ICML2003).

7. Sarini, Sarini, McGree, James, White, Nicole, Mengersen, Kerrie, & Kerr, Graham (2015), *Comparison of decision tree, support vector machines, and Bayesian network approaches for classification of falls in Parkinson's disease*. International Journal of Applied Mathematics and Statistics, 53(6), pp. 145-151.

8. A. Sopharak, B. Uyyanonvara, S. Barman, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:8, No:5, 2014

9. Ranjeeta Rana, Vaishali Kolhe (2015). *Analysis of Students Emotion for Twitter Data using Naïve Bayes and Non Linear Support Vector Machine Approachs*. International Journal on Recent and Innovation Trends in Computing and Communication. ISSN: 2321-8169

10. HP Luhn (1958), *The Automatic Creation of Literature Abstracts*, IBM JOURNAL, pp. 159-161.

PHỤ LỤC

CHỨNG NHẬN PHÁT TRIỂN VÀ TRIỂN KHAI THỰC TẾ

CÔNG TY CỔ PHẦN INEWS

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

GIẤY CHỨNG NHẬN PHÁT TRIỂN VÀ TRIỂN KHAI SẢN PHẨM

xác nhận:

Nhân viên: **Cấn Mạnh Cường**

Ngày sinh: 28/09/1991

Nơi sinh: Kim Quan, Thạch Thất, Hà Nội

Phòng ban: Kĩ thuật

Từ ngày: ...../01../2015 đến ngày ...../.12../2015

Đã tham gia phát triển, ứng dụng và triển khai các bài toán sau trên miền dữ liệu tiếng Việt:

- ***Bài toán xử lý trùng lặp tài liệu tin tức tự động.***
- ***Bài toán phân loại tài liệu tin tức theo chủ đề tự động.***
- ***Bài toán xác định từ khóa quan trọng và sinh tóm tắt cho tin tức tự động***

Và đạt kết quả khả quan và tính ứng dụng cao trong thực tiễn vào sản phẩm.

....Hà Nội..., ngày 30 tháng 05 năm 2016

ĐẠI DIỆN ĐƠN VỊ

TP. Kĩ thuật

Nguyễn Đỗ Bình