

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH
BỘ MÔN KHOA HỌC MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP

**Phát triển động cơ tìm kiếm để hỗ trợ tiếng Việt và
làm việc trên hệ thống dữ liệu lớn**

HỘI ĐỒNG 1 – CÔNG NGHỆ PHẦN MỀM

GVHD: PGS.TS. Quản Thành Thơ

Th.S. Lê Đình Thuận

GVPB: Th.S. Nguyễn Cao Trí

---o0o---

SVTH : Võ Xuân Thịnh (51003226)

TP. Hồ Chí Minh – 12/2014

LỜI CẢM ƠN

Để hoàn thành được đề tài luận văn tốt nghiệp này, em xin gửi lời cảm ơn chân thành nhất tới PGS. TS Quản Thành Thơ, Thầy đã hướng dẫn tận tình, định hướng, giải đáp khúc mắc và cùng em đi qua hết những khó khăn trong quá trình làm đề tài. Bên cạnh đó, em cũng gửi lời cảm ơn sâu sắc tới Thầy Lê Đình Thuận đã theo sát em hàng tuần, lắng nghe và đưa ra những lời khuyên bổ ích để em vững bước hơn trong lúc thực hiện.

Em chân thành biết ơn sự tận tình dạy dỗ, giúp đỡ của các thầy cô trong khoa Khoa học và Kỹ thuật Máy tính đã truyền đạt những kinh nghiệm, kiến thức, những bài giảng vô cùng sâu sắc và hữu ích để em có thể hoàn thành tốt được đề tài thực tập này.

Cuối cùng, em gửi lời cảm ơn đến gia đình, bạn bè, những người đã quan tâm, động viên, giúp đỡ cả về thể chất lẫn tinh thần để em có đủ nghị lực, sức khỏe hoàn thành tốt thực tập tốt nghiệp này.

Với lòng biết ơn chân thành, em xin gửi lời chúc sức khỏe, lời biết ơn và những lời chúc tốt đẹp nhất tới các thầy cô ở khoa Khoa học và Kỹ thuật Máy tính trường Đại Học Bách Khoa thành phố Hồ Chí Minh.

Trân trọng!

Tp. Hồ Chí Minh, tháng 12 năm 2014

Sinh viên thực hiện

Võ Xuân Thịnh

LỜI CAM ĐOAN

Tôi cam đoan rằng ngoại trừ các kết quả tham khảo từ các nguồn khác có ghi rõ trong phụ lục thì các công việc trình bày trong báo cáo luận văn tốt nghiệp này đều cho chính tôi thực hiện và chưa có phần nội dung nào của báo cáo luận văn tốt nghiệp được nộp ở trường này hoặc trường khác. Nếu có bất kỳ sai phạm nào, tôi xin chịu hoàn toàn trách nhiệm trước Ban Chủ Nhiệm Khoa và Ban Giám Hiệu Nhà Trường.

Tp. Hồ Chí Minh, tháng 12 năm 2014

Sinh viên thực hiện

Võ Xuân Thịnh

TÓM TẮT LUẬN VĂN

Thương mại điện tử đã trở thành một khái niệm khá quen thuộc với Việt Nam. Với thời gian phát triển cũng khá dài, số lượng người dùng hiện nay sử dụng thương mại điện tử đã tăng lên đáng kể. Các dạng thương mại điện tử phát triển khá đa dạng từ các trang trung gian cho phép người dùng đăng các sản phẩm của mình lên bán, đến các trang mua bán deal bán hàng giảm giá, trong hai năm gần đây, mô hình bán hàng truyền thống lại nở rộ với các công ty tầm cỡ lớn. Giá sản phẩm trong thương mại điện tử luôn là vấn đề được nhắc đến. Được biết đến với một cách mạng về giá khi không phải trả các chi phí về cửa hàng, chi phí nhân công, mọi thông tin về sản phẩm và giá sẽ được cập nhật liên tục trên các trang web. Một sản phẩm sẽ được bán trên khá nhiều trang web. Việc cạnh tranh giá là rất khốc liệt đối với thị trường buôn bán này. Với số lượng website khá lớn, người dùng đang gặp rất nhiều khó khăn khi lựa chọn một sản phẩm trong muôn vàn người bán.

Xuất phát từ khó khăn trong thực tế đó, luận văn tốt nghiệp này hướng tới việc hỗ trợ dịch vụ để người dùng có thể lựa chọn sản phẩm ở cửa hàng với giá tốt nhất và trong thời điểm thích hợp nhất. Dữ liệu giá là mục tiêu hướng đến của luận văn.

Nội dung của luận văn bao gồm các phần chính như sau:

Mở đầu:

Giới thiệu chung bô cục của luận văn, trình bày các lý do chọn luận văn.

Phần 1: Tổng quan đề tài.

Nội dung chính của chương này là giới thiệu về thị trường thương mại điện tử của Việt Nam, các mô hình thương mại điện tử; dữ liệu được sinh ra từ thương mại điện tử; mục tiêu và phạm vi của luận văn; các công cụ sử dụng để hiện thực.

Phần 2: Phân tích vấn đề

Nội dung chương này bao gồm: Các ví dụ minh họa, phân tích yêu cầu, đề nghị các giải pháp cần hiện thực, và các thảo luận xung quanh vấn đề. Ngoài ra, phần này còn có các đánh giá các hiện thực có cùng nội dung đang được chạy trên thị trường.

Phần 3: Kiến thức nền; các công nghệ sử dụng.

Để thực hiện được các yêu cầu và chức năng đã nêu ở trên, cần một số kiến thức áp dụng các nền tảng đã được xây dựng sẵn để tiết kiệm và tận dụng các công cụ đã được xây dựng sẵn. Ngoài ra, việc sử dụng công cụ hợp lý giúp tiết kiệm thời gian sản phẩm còn tăng sự ổn định và phát triển sau này. Chương này nội dung chính giới thiệu hai nền tảng sử dụng chính trong luận văn.

Phần 4: Giải thuật so trùng đối tượng sách.

Trong chương này, luận văn sẽ trình bày hai thuật toán chính được sử dụng để so sánh các đối tượng sách với nhau. Hai thuật toán bao gồm thuật toán Levenshtein Distance và thuật

toán tính Cosine Similarity dựa trên các thông tin thu thập được là tên và mô tả cuốn sách. Các giải thuật này đã được áp dụng thêm phần Tokenizer để tăng độ chính xác với tiếng Việt.

Phần 5: Thiết kế và hiện thực.

Nội dung chính của chương này là các công việc đã thực hiện trong luận văn xây dựng chương trình thu thập dữ liệu, xử lý dữ liệu trùng lặp, đánh chỉ mục dữ liệu và viết trang giao diện cho người dùng tương tác; và triển khai trên hệ thống các máy chạy thực tế.

Phần 6: Thí nghiệm, đánh giá và kết luận.

Nội dung chính của phần này trình bày các kết quả đã đạt được, kiểm tra thí nghiệm chủ quan và kết luận của luận văn. Đây là phần cuối cùng tổng kết lại các phần đã và chưa làm được cũng như đánh giá kết quả hiện thực.

ABSTRACT

E Commerce is a familiar term in Viet Nam. As a long time being applied, number of consumer using E-commerce for shopping and trading has been incredible increasing. Then many category of e-commerce including markets online or forums help sellers list their items for sales then many coupon/deal site and in recent two years, giant online stores launched to carve out the market share. Pricing is always one of the most important things in E Commerce. Since cutting down value chain by do not have to rent nice place for their shop, lower number of labors, although everything has updated to website. The fact that, one product could be sell in many sites, so it would be competitive racing to reach consumer's wallets. On the other hand, consumers see problems to decide where to buy what they want.

Thesis mainly towards helping people with a new service to choice good product's price in huge number of websites. Pricing data is one target of this thesis.

Mục lục

LỜI CẢM ƠN

LỜI CAM ĐOAN

TÓM TẮT LUẬN VĂN

ABSTRACT

PHẦN 1. GIỚI THIỆU	1
1.1. Tổng quan đề tài	1
1.1.1. Tổng quan về thương mại điện tử	1
1.1.2. Tổng quan về dữ liệu lớn trong thương mại điện tử	3
1.2. Mục tiêu của đề tài.....	4
1.3. Phạm vi đề tài.....	4
1.4. Ý nghĩa thực tiễn	4
PHẦN 2. PHÂN TÍCH VĂN ĐỀ	5
2.1. Ví dụ minh họa	5
2.1.1. Ví dụ 1.....	5
2.1.2. Ví dụ 2.....	5
2.1.3. Nhận xét	6
2.2. Phân tích	6
2.2.1. Xây dựng website tương tác với người dùng:.....	6
2.2.2. Xây dựng hệ thống backend:	7
2.3. Giải pháp đề nghị	8
2.3.1. Xây dựng website	8
2.3.2. Hệ thống thu thập dữ liệu.....	8
2.3.3. Hệ thống lưu trữ, đánh chỉ mục và tìm kiếm	9
2.4. Thảo luận	9
2.4.1. Tại sao lựa chọn sản phẩm là sách.....	9
2.4.2. So sánh các trang website so sánh giá khác đang hoạt động tại Việt Nam.....	9
PHẦN 3. KIẾN THỨC NỀN.....	13
3.1. Công cụ thu thập dữ liệu Scrapy	13
3.1.1. Kiến trúc công cụ	14
3.1.2. Hoạt động của Scrapy	15
3.1.3. Sử dụng Xpath Selector để trích xuất dữ liệu	16
3.1.4. Sử dụng Scrapy Framework.....	17
3.2. Elasticsearch	17
3.2.1. Giới thiệu về Elasticsearch	17
3.2.2. Kỹ thuật tokenizer bằng Apache Lucene và Inverted index	18
3.2.3. Định nghĩa cấu trúc trong Elasticsearch	20
3.2.4. Sử dụng Elasticsearch trong luận văn	20

3.3. Cấu trúc của CMS PhalconEye	20
3.3.1. Cấu trúc thư mục của PhalconEye:	21
3.3.2. Thiết kế và mô hình hoạt động của PhalconEye	22
3.3.3. Sử dụng và phát triển thêm PhalconEye trong luận văn:	24
PHẦN 4. GIẢI THUẬT SO TRÙNG ĐỐI TƯỢNG SÁCH	25
 4.1. Thuật toán sử dụng.....	25
4.1.1. Công thức Levenshtein distance và giải thuật tính Edit Distance	25
4.1.2. Công thức tính Levenshtein Distance hay Minimun Edit Distance (MED)	25
4.1.3. Ví dụ công thức Minimun Edit Distance	26
4.1.4. Giải thuật tính Minimun Edit Distance	26
4.1.5. Áp dụng trong luận văn.....	27
 4.2. Độ tương tự của hai chuỗi dựa vào Minimun Edit Distance	29
 4.3. So trùng dựa trên TF-IDF và Cosine Similarity	31
4.3.1. Xác định trọng số mỗi từ dựa trên thuật toán TF-IDF	31
4.3.2. Thuật toán Cosine Similarity	33
4.3.3. Áp dụng thuật toán Cosine Similarity để so trùng	33
4.3.4. Áp dụng trong luận văn.....	35
PHẦN 5. THIẾT KẾ VÀ HIỆN THỰC	38
 5.1. Thiết kế bộ thu thập dữ liệu.....	38
5.1.1. Mô hình tổng quát thu thập dữ liệu.....	38
5.1.2. Các hàm hỗ trợ để phục vụ cho quá trình làm sạch và ánh xạ kiểu dữ liệu.....	41
5.1.3. Khó khăn gặp phải khi thu thập dữ liệu	43
 5.2. Xây dựng bộ lưu trữ thông tin theo thời gian	44
 5.3. Triển khai trên hệ thống máy thật	45
 5.4. Màn hình giao diện của trang người dùng	46
PHẦN 6. THÍ NGHIỆM, ĐÁNH GIÁ VÀ KẾT LUẬN.....	50
 6.1. Thí nghiệm và đánh giá kết quả	50
6.1.1. Thí nghiệm hai giải thuật Minumin Edit Distance và TF-IDF:	50
6.1.2. Thí nghiệm sản phẩm	51
 6.2. Kết luận.....	54
PHẦN 8. Phụ lục	57
 8.1. Tài liệu tham khảo	57

Mục lục bảng

Bảng 1.1 Phân loại các mô hình thương mại điện tử	2
Bảng 3.1 Biểu thức và giải thích cú pháp của Xpath Selector.....	16
Bảng 4.1 Cấu trúc dữ liệu của đối tượng sách	28
Bảng 4.2 Kết quả thực nghiệm với các tựa sách khi tính MED.....	29
Bảng 4.3 Thủ tính với hai công thức với chuỗi	30
Bảng 4.4 Thống kê các từ xuất hiện nhiều nhất.....	35
Bảng 4.5 Danh sách đại diện các từ xuất hiện 1 lần trong văn bản	36
Bảng 4.6 Kết quả TF-IDF của một số từ trong văn bản	37
Bảng 5.1 Định nghĩa Xpath của một trang web cụ thể (tiki.vn)	41
Bảng 5.2 Kết quả thống kê sau một lần chạy thu thập.....	43
Bảng 5.3 Hệ thống máy chủ đang hoạt động	46
Bảng 5.4 Thông tin các dịch vụ đang chạy trên máy Phalcon1	46
Bảng 5.5 Thông tin các dịch vụ đang chạy trên máy Python 1	46
Bảng 6.1 Mức ngưỡng ứng với số đối tượng trùng nhau của giải thuật	50
Bảng 6.2 Báo cáo kết quả thực hiện	55

Mục lục hình ảnh

Hình 2.1 Biểu đồ giá theo thời gian	5
Hình 2.2 Mô hình tổng quan của hệ thống.....	7
Hình 2.3 Kết quả trả về từ việc tìm kiếm một máy giặt cũ thê.....	9
Hình 2.4 Kết quả tìm kiếm trả về từ trang websosanh.vn.....	10
Hình 2.5 Chức năng thông báo khi giá giảm	11
Hình 2.6 Giao diện trang chủ giadaure.com	11
Hình 2.7 Các sản phẩm được thống kê	12
Hình 2.8 Kết quả thống kê sản phẩm.....	12
Hình 3.1: Cấu trúc Scrapy Framework	14
Hình 3.2 Mô hình Data Flow của Framework Scrapy	15
Hình 3.3 Tổng quan Lucene.....	19
Hình 3.4 Cấu trúc cây thư mục của PhalconEye Framework	21
Hình 3.5 Cấu trúc cây thư mục app trong PhalconEye CMS	21
Hình 3.6 Cấu trúc cây thư mục Public của PhalconEye CMS	22
Hình 3.7 Mô hình MVC	23
Hình 3.8 Lượt đồ hoạt động của Mô hình MVC	23
Hình 4.1 Giải thuật Minimun Edit Distance	26
Hình 4.2 Ma trận kết quả khi chạy thuật toán Minumun Edit Distance	27
Hình 4.3: Giải thuật Custom Minimun Edit Distance.....	30
Hình 4.4 Kết quả chạy với tựa sách "Kinh Tế Học Hài Hước (Tái Bản 2014)"	31
Hình 4.5 Kết quả chạy với tựa sách "Siêu Kinh Tế Học Hài Hước (Tái Bản 2014)"	31
Hình 4.6 Kết quả chạy với tựa sách "Harry Potter Và Phòng Chứa Bí Mật"	31
Hình 4.7 Kết quả chạy với tựa sách "Harry Potter Và Phòng Chứa Bí Mật"	31
Hình 5.1 Ví dụ về một Trang tổng thể	38
Hình 5.2 Ví dụ về một Trang chi tiết	39
Hình 5.3 Flow chart quá trình thu thập các trang chi tiết	40
Hình 5.4 Quá trình thu thập dữ liệu	41

Hình 5.5 Hàm ánh xạ số từ chuỗi	42
Hình 5.6 Hàm làm sạch chuỗi.....	42
Hình 5.7 Hàm ánh xạ danh sách chuỗi thành từ điển	42
Hình 5.8 Mô hình Inverse Index	45
Hình 5.9 Màn hình giao diện chính	47
Hình 5.10 Màn hình chi tiết một cuốn sách	47
Hình 5.11 Thông tin của một cuốn sách	47
Hình 5.12 Kết quả tìm kiếm theo từ khoá.....	48
Hình 5.13 Biểu đồ giá theo thời gian của 2 tựa cuốn sách	48
Hình 5.14 Trang đăng kí	
Hình 5.15 Trang đăng nhập	49
Hình 6.2 Bản đồ giá của cuốn “Đắc Nhân Tâm” [1]	52
Hình 6.3 Bản đồ giá của cuốn “Đắc Nhân Tâm” [2]	52
Hình 6.4 Bản đồ giá của cuốn sách “7 Bí Quyết Giúp hôn Nhân Hạnh Phúc”	53
Hình 6.5 Biểu đồ giá sách Toeic Analyst Second Edition.....	53
Hình 6.6 Biểu đồ giá sách Barron's Toeic Test (4th Edition)	54

Mục lục công thức

Công thức 4.1 Navie similar	29
Công thức 4.2: Custom similar	30
Công thức 4.3 Tính Term Frequency.....	32
Công thức 4.4 Tính Term Frequency (2)	32
Công thức 4.5 Tính Inverse Document Frequency	32
Công thức 4.6 Tính TF-IDF	33
Công thức 4.7 Công thức tính Cosine Similarity giữa 2 Vector.....	33

PHẦN 1. GIỚI THIỆU

Nội dung chính của chương này là giới thiệu về thị trường thương mại điện tử của Việt Nam, các mô hình thương mại điện tử; dữ liệu được sinh ra từ thương mại điện tử; mục tiêu và phạm vi của luận văn; các công cụ sử dụng để hiện thực.

1.1. Tổng quan về tài

1.1.1. Tổng quan về thương mại điện tử

a) Định nghĩa và đặc điểm của thương mại điện tử

Định nghĩa Thương mại điện tử: Kinh doanh điện tử (E-business) là quá trình mở rộng bất cứ quá trình kinh doanh nào trong một công ty hoặc trong quá trình giao dịch với đối tác bằng cách sử dụng các công nghệ của Web nhằm tự động hóa các quá trình kinh doanh. Thương mại điện tử (E-commerce) là một trường hợp đặc biệt bao gồm các hoạt động tiếp thị, buôn bán, hỗ trợ khách hàng, và giao dịch với đối tác bằng cách sử dụng kinh doanh điện tử. Đây là định nghĩa khá phổ quát của IBM được đưa ra năm 1999 bởi David Liederbach, Giám đốc bộ phận thương mại điện tử lúc bấy giờ định nghĩa. Với sự phát triển nhanh và mạnh của Internet, mọi thiết bị từ điện thoại, ti-vi, và các thiết bị nghe nhìn khác có khả năng kết nối đều truy cập được các thông tin từ Internet. Số lượng người sử dụng thương mại điện tử tăng gấp đôi trong hai năm gần đây, và số lượng đó còn tiếp tục tăng trưởng khi số thiết bị không phải PC (non-PC) như máy tính bảng, thiết bị đeo trên người như vòng tay, kính, và đồng hồ ngày càng nhanh chóng tiếp cận người dùng.

Những đặc trưng tiêu biểu của thương mại điện tử bao gồm:

- Các bên tiến hành tìm hiểu, giao dịch không cần đòi hỏi với nhau từ trước. Uy tín của trang được xây dựng bằng số lượng giao dịch và số lượng truy cập của bên bán. Các hình thức dịch vụ quảng cáo điện tử cùng ngày càng được áp dụng nhiều hơn và ngày càng phức tạp hơn.
- Các thông tin về sản phẩm luôn sẵn sàng trước khi người dùng bắt đầu tìm kiếm về sản phẩm.
- Người dùng phải thông qua kết nối để tìm kiếm sản phẩm, đây có thể là một rủi ro cho người dùng khi người dùng có thể bị tấn công và ăn cắp một số thông tin cá nhân khi thực hiện giao dịch.

Từ những đặc điểm trên dẫn đến các ưu điểm khi áp dụng kinh doanh thương mại điện tử:

- Thu thập được nhiều thông tin, thông tin được đưa cho người dùng các nhiều, các chính xác sẽ giúp cho người dùng dễ lựa chọn. Đây cũng là nhược điểm của thương mại điện tử. Mọi thông tin có thể được người dùng kiểm tra chéo giữa các trang với nhau và với trang của nhà sản xuất. Nên ngoài thông tin đầy đủ, các hình ảnh và số liệu đòi hỏi phải chính xác.

- Giảm chi phí bán hàng, tiếp thị và giao dịch bởi nhiều phần đã được thay thế bằng máy móc trong chuỗi giá trị sản phẩm.
- Xây dựng quan hệ với các đối tác và tạo điều kiện sớm tiếp cận kinh tế tri thức. Uy tín và thương hiệu của các công ty có thể được xây dựng và bùng nổ nhanh chóng qua các sự kiện. Người dùng sẽ có nhiều sự lựa chọn đúng đắn hơn.

b) *Phân loại thương mại điện tử*

Cách phân loại chính của thương mại điện tử dựa vào các thực thể tham vào quá trình giao dịch và kinh doanh. Có năm phân loại giao dịch điện tử thường được thấy bao gồm: doanh nghiệp với doanh nghiệp (business-to-business), doanh nghiệp với người tiêu dùng (business-to-customer), người tiêu dùng với người tiêu dùng (customer-to-customer), doanh nghiệp với nhà nước (business-to-government) và các hoạt động hỗ trợ hoạt động mua và bán.

Bảng 1.1 Phân loại các mô hình thương mại điện tử

Mô hình kinh doanh	Mô tả	Ví dụ tiêu biểu
Doanh nghiệp với người tiêu dùng (B-2-C)	Doanh nghiệp cung cấp dịch vụ và sản phẩm cho từng khách hàng.	Walmart.com bán tất cả hàng hóa của mình thông qua Website tới người tiêu dùng. Một điển hình mới ở Việt Nam là Lazada.vn
Doanh nghiệp với doanh nghiệp (B-2-B)	Doanh nghiệp đưa các giải pháp hỗ trợ các doanh nghiệp khác hoạt động. Mô hình này khác với mô hình trên ở đối tượng hướng người dùng hướng tới là doanh nghiệp thay vì khách hàng.	Atlassian.com bán các sản phẩm hướng tới việc hỗ trợ các doanh nghiệp công nghệ thông tin khác phát triển, vận hành sản phẩm và làm việc nhóm.
Người tiêu dùng với người tiêu dùng (C-2-C)	Người tiêu dùng sử dụng một dịch vụ của bên thứ ba, một trang chợ điện tử, để buôn bán các sản phẩm của mình cho người dùng khác. Đây là một dạng cụ thể của mô hình B-2-C	Điển hình nhất của mô hình này các chợ thương mại điện tử như vatgia.com, chodientu.vn và 5giay.vn
Doanh nghiệp với chính quyền	Doanh nghiệp bán hàng hóa, dịch vụ cho các chính phủ và các cơ quan chính phủ. Đây cũng tính như là một phần của thương mại điện tử B2C.	eoffice.com.vn là một sản phẩm hướng tới chính phủ điện tử mà BKAV đang xây dựng cho và hỗ trợ cho các tỉnh.
Các hoạt động hỗ trợ việc mua bán giữa các bên.	Doanh nghiệp và các tổ chức khác sử dụng các thông tin để hỗ trợ cho việc đánh giá và quyết định của khách hàng, nhà cung cấp và nhân viên. Các	Dell sử dụng một kênh thông tin được bảo mật an toàn để chuyển giá sản phẩm cũng như dự đoán giá trong tương lai cho các nhà cung cấp. Các nhà

	doanh nghiệp ngày càng chú trọng tới việc cung cấp và quản lý các thông tin được giao cho khách hàng, đối tác, nhà cung cấp.	cung cấp sẽ sử dụng thông tin để lên kế hoạch sản xuất và cung cấp thiết bị cho công ty Dell vào đúng thời điểm với số lượng hợp lý.
--	--	--

1.1.2. Tổng quan về dữ liệu lớn trong thương mại điện tử

Thương mại điện tử trong những năm gần nay gần như bùng nổ, số lượng các trang thương mại điện tử tăng gấp hai lần trong hai năm trở lại đây. Dữ liệu được sinh ra trong môi trường thương mại điện tử cũng được tăng đáng kể, rất đa dạng và và tốc độ tăng rất đáng kể.

a) Định nghĩa dữ liệu lớn

Trong những năm đầu tiên của thế kỷ 21, dữ liệu trong liệu lớn ngày càng tăng một cách đáng kể. Trong năm 1999, số lượng thông tin được sinh ra khoảng 1.5 tỷ Gigabytes và trong 2003 con số đó đã tăng gấp đôi. Những nhà phân tích đã mô tả dữ liệu bằng ba tính chất: khối lượng (volume), sự đa dạng (variety) và tốc độ tăng (velocity) là những tính chất quan trọng nhất trong thách thức quản lý dữ liệu cho các doanh nghiệp. Vài năm sau đó, ba tính chất (3Vs) đó đã được nhiều người công nhận như định nghĩa và mô tả của dữ liệu lớn.

Dữ liệu lớn là dữ liệu có khối lượng lớn, tốc độ tăng nhanh, và có tính đa dạng cao đòi hỏi phải thông qua nhiều quá trình xử lý, tinh chế để nâng cao chất lượng ra quyết định, có cái nhìn sâu sắc hơn và tối ưu hóa quá trình.

b) Ứng dụng của dữ liệu lớn

Dữ liệu lớn ngày càng thể hiện vai trò của mình trong thương mại điện tử, ngày càng nhiều hệ thống được xây dựng phục vụ cho quá trình quảng cáo, xác định giá, khuyến mãi cũng như phân tích thông tin người tiêu dùng. Bài toán cổ điển nhất của thương mại điện tử là gợi ý món hàng mà người dùng muốn mua khi lựa chọn một món hàng khác dựa vào các thông tin cho trước như thông tin của khách hàng, các sản phẩm mà người dùng đã xem trước, các sản phẩm mà người dùng khác đã xem. Bằng các phương pháp xử lý bằng các giải thuật khai phá dữ liệu, các bộ máy ngày càng thông minh hơn.

Giám sát dữ liệu (monitoring data) cũng là một nhánh đang phát triển trong lĩnh vực dữ liệu lớn. Giá luôn là vấn đề đầu tiên khi thương mại điện tử phổ biến. Giá của các sản phẩm biến đổi không ngừng qua từng ngày, khác nhau với từng trang, phụ thuộc vào nhiều yếu tố khác (lễ hội, khuyến mãi). Giám sát giá dữ liệu từ một bên thứ 3, không phải là chủ trang web đòi hỏi phải có nhiều kỹ thuật thu thập, trích xuất và tiền xử lý trước khi đưa vào phân tích dữ liệu cùng với dữ liệu của nhiều trang lân cận khác.

1.2. Mục tiêu của đề tài

Sách là sản phẩm đầu tiên được đưa vào hoạt động thương mại điện tử với tính chất dễ bảo quản, ít bị hư hại theo thời gian, dễ vận chuyển, gọn nhẹ. Đây là một mặt hàng khá dễ bán, số lượng người dùng ngày càng nhiều, việc giảm các chi phí ra, sách còn được các nhà sản xuất chiết khấu tốt. Nên sách luôn là sản phẩm có giá tốt nhất trong thị trường thương mại điện tử. Số lượng các trang bán sách cũng ở một số lượng đáng kể trên thị trường. Theo dõi giá sách và cung cấp những thông tin gợi ý cho người dùng về địa chỉ có giá tốt nhất là mục tiêu của đề tài.

1.3. Phạm vi đề tài

Trong phạm vi có hạn về thời gian nên luận văn chỉ thực hiện việc giám sát dữ liệu trên ba trang bán sách chính là tiki.vn, nobita.vn và vinabook.vn. Thực hiện các bước thu thập và trích xuất, tiền xử lý và đánh giá các dữ liệu liên quan với nhau. Luận văn đưa một cách giải quyết vấn đề đồng nhất giữa các đối tượng ở các trang khác nhau.

1.4. Ý nghĩa thực tiễn

Luận văn nhằm hướng tới một dạng trong thương mại điện tử, cung cấp thông tin để hỗ trợ cho việc đánh giá và quyết định của khách hàng hướng tới một trang trung gian cung cấp các sản phẩm ứng với các địa chỉ bán. Nâng cao chất lượng và nội dung tìm kiếm cho người dùng thay vì phải dùng các công cụ tìm kiếm toàn văn (full-text search) khác.

PHẦN 2. PHÂN TÍCH VĂN ĐỀ

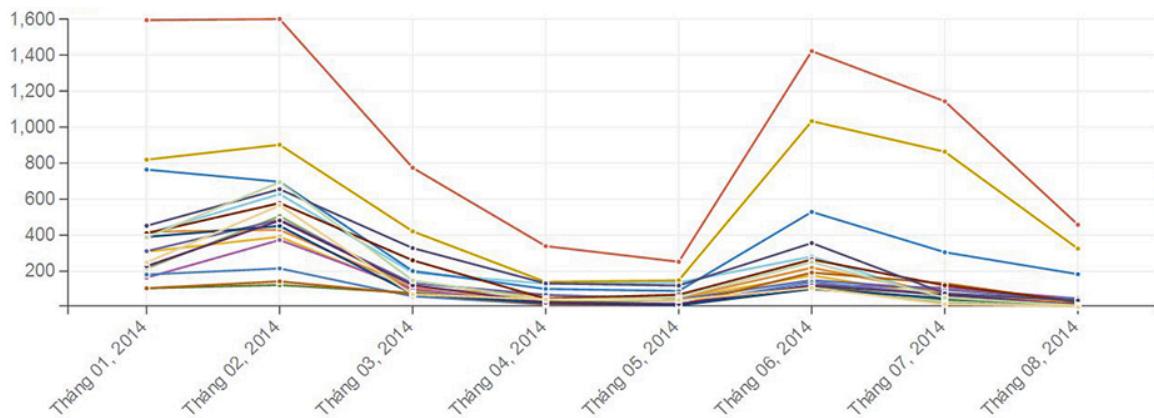
Nội dung chương này bao gồm: Các ví dụ minh họa, phân tích yêu cầu, đề nghị các giải pháp cần hiện thực, và các thảo luận xung quanh vấn đề. Ngoài ra, phần này còn có các đánh giá các hiện thực có cùng nội dung đang được chạy trên thị trường.

2.1. Ví dụ minh họa

2.1.1. Ví dụ 1

Cách truyền thống: Khi đi mua một cuốn sách, khi được một người bạn giới thiệu tên sách “Siêu kinh tế học hài hước”, người dùng sẽ sử dụng các công cụ tìm kiếm đã được xây dựng sẵn để tìm kiếm theo từ khóa với “mua sách siêu kinh tế học hài hước”: có khoảng 672.000 kết quả trả về. Trong 10 kết quả trả về đầu tiên có 3 trang bán sách với giá sản phẩm lần lượt là 48.000, 40.000, 47.000 VNĐ. Việc lựa chọn địa điểm mua hàng phụ thuộc vào quyết định của người dùng, kết quả trả về từ công cụ tìm kiếm.

Với website so sánh giá sách: Người dùng nhập tên sách vào ô tìm kiếm. Các thông tin trả về bao gồm sách, giá sách được cập nhật trong ngày, những địa điểm đang bán sách, và biểu đồ giá của sản phẩm theo thời gian. Việc còn lại của người dùng khá đơn giản nếu muốn chọn một cuốn sách với giá ưng ý, người dùng sẽ đến trang bán sách gốc để đặt cuốn sách.



Hình 2.1 Biểu đồ giá theo thời gian

2.1.2. Ví dụ 2

Người dùng muốn mua một cuốn sách giá khá là cao trong ví dụ này là từ điển “Oxford Advance Learner Dictionary”. Giá của cuốn sách này từ 300 đến 450 VNĐ. Người dùng muốn mua được giá tốt của cuốn sách này phải lướt qua các trang bán sách định kỳ xem có khuyến mãi mới cho cuốn sách này hoặc đơn giản hơn có thể đăng ký nhận email từ các trang bán sách.

Với trang so sánh giá người dùng có thể bookmark cuốn sách. Khi có giá giảm, người dùng sẽ nhận được email thông báo trong vòng 24 giờ cùng với các đường dẫn tới trang mua sách.

2.1.3. Nhận xét

Thương mại điện tử có tính chất đặc biệt hơn khi giá cả được cập nhật một cách tức thời và ưu đãi nhiều hơn cách truyền thống. Nhưng trong thời đại thương mại điện tử bùng nổ, số lượng trang bán sản phẩm thương mại điện tử khá lớn, việc cạnh tranh giá giữa các trang ngày càng gay gắt thì lựa chọn một sản phẩm đối với người dùng ngày càng khó khăn.

2.2. Phân tích

Từ nhu cầu cụ thể của người dùng, việc phân tích yêu cầu giúp cụ thể các công việc cần phải làm, thiết kế ban đầu, các công việc cần chuẩn bị cũng như các khó khăn sẽ gặp phải trong quá trình thiết kế và hiện thực.

2.2.1. Xây dựng website tương tác với người dùng:

Website giao diện là nơi tương tác chính với người dùng, thể hiện nội dung và hỗ trợ trong việc lựa chọn sách.

Các yêu cầu đặt ra với website giao diện:

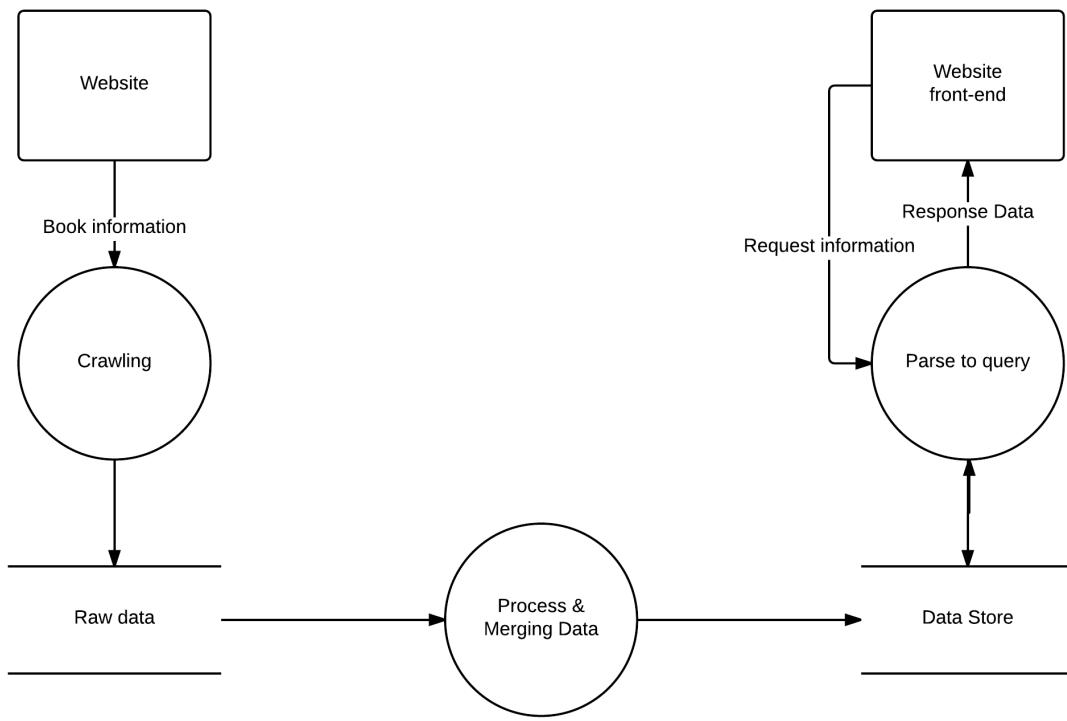
- Có chức năng tìm kiếm theo từ khoá. Người dùng có thể tìm kiếm sách theo tên sách, tên tác giả, nhà xuất bản, mô tả sách. Một số nguồn sách đã được lựa chọn sẵn để phục vụ người dùng.
- Kết quả trả về là thống kê của nhiều trang bán sách đã được sắp xếp theo mức độ phù hợp với từ khoá.
- Người dùng có thể bookmark để nhận email thông báo về sau.
- Mức độ sẵn sàng thông tin được lấy về, tiền xử lý và phân tích từ trước khi người dùng tìm kiếm.
- Khả năng đáp ứng nhanh: Thời gian phản hồi nhanh luôn là yếu tố hàng đầu để đánh giá một website.
- Giao diện thân thiện với người dùng, hỗ trợ các màn hình điện thoại.

Ứng với mỗi yêu cầu được đặt ra là một chức năng của website giao diện:

- Chức năng 1: Tìm kiếm sách theo các thông tin tên, tác giả, mô tả chung của sách. Các kết quả trả về được thống xếp theo độ chính xác ứng với mỗi từ khoá.
- Chức năng 2: Mỗi kết quả tìm kiếm bao gồm tên sách, các địa điểm bán sách, biểu đồ giá của các trang theo thời gian cũng như các đường dẫn tới trang gốc để mua sách.
- Chức năng 3: Phản hồi cập nhật mới qua email khi có sự thay đổi về giá cũng như có sản phẩm mới.

Các yêu cầu về khả năng nhanh chóng và mức độ sẵn sàng của trang web được thể hiện ở phần backend của hệ thống.

Mô hình tổng quan của hệ thống:



Hình 2.2 Mô hình tổng quan của hệ thống

- (1) Website: là các trang bán buôn bán thương mại điện tử có các sản phẩm về sách được cập nhật và bán vẫn đang hoạt động. Các trang ngày gọi là các trang nguồn. Việc lựa chọn các trang nguồn thông qua mức độ uy tín của trang web trong cộng đồng.
- (2) Website front-end: là trang giao diện cần xây dựng để người dùng có thể lựa chọn địa điểm mua sách thích hợp.
- (3) Quá trình Crawling data: là quá trình được diễn ra nhằm thu thập và cập nhật các thông tin mới từ các trang nguồn đã được lựa chọn trước. Đối với mỗi trang, việc trích xuất các thông tin cần thiết phụ thuộc vào template của trang web đó.
- (4) Quá trình Process & Merging Data: là quá trình làm sạch dữ liệu, tiền xử lý cũng như giải quyết vấn đề nhận diện các cuốn sách giống nhau ở các trang khác nhau cũng như các sản phẩm thu về là sản phẩm mới hay được cập nhật giá.
- (5) Data store: là nơi lưu trữ chính dữ liệu. Việc lưu trữ phục vụ cho việc tìm kiếm, và truy xuất nhanh từ trang giao diện.

2.2.2. Xây dựng hệ thống backend:

Để phục vụ cho website giao diện, cần xây dựng một hệ thống backend đáp ứng được hai nhu cầu được đặt ra là mức độ sẵn sàng cao, và mức độ đáp ứng nhanh. Mức độ sẵn sàng cao ngoài ý nghĩa người dùng truy cập được website mà còn muốn nói đến mức độ sẵn sàng của

dữ liệu và tính toán. Để đạt được thời gian phản hồi nhanh cần phải thiết kế cơ sở dữ liệu truy xuất dữ liệu chuỗi thời gian.

Yêu cầu và chức năng đối với hệ thống backend

- Thu thập dữ liệu từ các trang, trích xuất được các trường theo một template có sẵn trước. Tiền xử lý các dữ liệu, kiểm tra độ tương quan giữa các sản phẩm giữa các trang bán hàng thương mại điện tử. Cập nhật thông tin được lưu trữ theo chuỗi thời gian và xử lý các dữ liệu mới.
- Lưu chuỗi thông tin theo chuỗi thời gian.
- Kiểm tra độ tương đồng giữa các sản phẩm. Thiết kế giải thuật so trùng sách giữa các trang.

2.3. Giải pháp đề nghị

2.3.1. Xây dựng website

Vì thời gian thực hiện có hạn nên việc xây dựng website đã sử dụng một CMS open source là PhalconEye. Việc sử dụng CMS PhalconEye nhằm tiết kiệm thời gian vì CMS đã xây dựng khá đầy đủ phần giao diện cho admin và skeleton của ứng dụng. CMS được viết theo kiến trúc multi-module có khả năng mở rộng về sau. Ngoài ra, CMS được viết dựa trên Phalcon PHP Framework, một framework mới có các file Core được viết bằng C nhằm tăng tốc độ xử lý. Thời gian đáp ứng nhanh là đặc điểm nổi bật nhất của Framework này. Nhược điểm chính của CMS là phải build thêm một file thư viện nên khó sử dụng ở các Hosting Serve, thường thì sẽ dễ làm ở trên droplet sẽ phù hợp và dễ dàng hơn. Framework được xây dựng khá đầy đủ và không dễ cho người mới học lập trình Web.

Tận dụng được các phần đã được xây dựng sẵn bởi nhà phát triển thay vì phải thiết kế lại khung sườn sản phẩm từ đầu. Sử dụng Web Framework ngoài thời gian xây dựng lại khung sườn chương trình còn tồn các thời gian xây dựng các phần khác để bảo trì và phát triển về sau. Các phần đó trong CMS thường được định nghĩa các lớp abstract ban đầu trước.

Áp dụng theme Metronic E-commerce để tạo giao diện đẹp mắt, thu hút người dùng được thiết kế responsive cho các màn hình mobile và tablet. Ngôn ngữ được sử dụng để xây dựng website là PHP.

2.3.2. Hệ thống thu thập dữ liệu

Hệ thống thu thập dữ liệu sử dụng là Scrapy Framework. Việc lựa chọn Scrapy dựa trên các tiêu chí dễ phát triển, có kiến trúc rõ ràng được chia thành các module khác nhau. Có thể phát triển về sau. Scrapy được viết bằng ngôn ngữ Python nên tận dụng được các ưu điểm khác của Python về xử lý dữ liệu chuỗi.

2.3.3. Hệ thống lưu trữ, đánh chỉ mục và tìm kiếm

Elasticsearch được phát triển nhằm phục vụ cho quá trình đánh chỉ mục và tìm kiếm. Elasticsearch trong luận văn này còn là nơi lưu trữ thông tin chính thay thế cho database phục vụ cho việc truy xuất dữ liệu từ trang giao diện chính.

2.4. Thảo luận

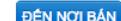
2.4.1. Tại sao lựa chọn sản phẩm là sách

Sách là sản phẩm đầu tiên được đưa vào buôn bán trong thương mại điện tử. Sách có nhiều ưu điểm thích hợp như ít hư hỏng, không mất giá trị, có giá bìa niêm yết, dễ vận chuyển và bảo quản, mặt hàng khá đa dạng và có nhu cầu cao từ thị trường. Điểm hình nhất về bán sách của thương mại điện tử chính là trang Amazon. Ở Việt Nam cũng có một số trang bán sách tiêu biểu, đó là trang Tiki.vn.

2.4.2. So sánh các trang website so sánh giá khác đang hoạt động tại Việt Nam

a) Trang sosanhgia.com

Đối tượng hàng hoá mà trang sosanhgia.com hướng tới là các mặt hàng điện tử gia dụng như máy giặt, tủ lạnh, thiết bị di động, máy tính và thiết bị kỹ thuật số. Các mặt hàng rất đa dạng và đầy đủ được phân chia thành các nhóm hàng.

Hình ảnh	Mô tả sản phẩm	Nơi bán	Giá (VND)
	MÁY GIẶT 7KG SANYO AWD-D700VT (MEDIAMART) Từ 13/12/19/12: QUÀ KHUYẾN MẠI TẶNG THÊM - Tặng thêm Bộ nồi hoặc Bếp từ trị giá đến 890.000đ - Từ 01/11-20/1/2015: Với giao dịch qua thẻ MB Bank từ 300.000đ, trúng 60 iPhone 6, 09 Smart TV 3D 55. Chi tiết tại đây.	 Media Mart	10.390.000  Cập nhật: 23/12/2014
	MÁY GIẶT SANYO AWD-D700VT(N) (NGUYỄN KIM)	 Nguyễn Kim	9.800.000  Cập nhật: 23/12/2014
	MÁY GIẶT LỒNG NGANG SANYO AWD-D700VT-N - 7KG (HOMECENTER) • Bàn là Electrolux ESI-525	 HC Home Center	9.890.000  Cập nhật: 23/12/2014
	MÁY GIẶT SANYO MODEL AWD-D700VT(VÀNG) (PHAN KHANG) Mua online rẻ hơn 808.000 VND 10.192.000	 PHAN KHANG	11.000.000  Cập nhật: 23/12/2014
	MÁY GIẶT CỬA TRƯỚC SANYO AWD-D700VT 7KG (CDISCOUNT)	 CDISCOUNT.VN CAM KẾT GIÁ RẺ	8.882.000  Cập nhật: 23/12/2014

Hình 2.3 Kết quả trả về từ việc tìm kiếm một máy giặt cũ thê

Các thông tin được hiển thị rất đầy đủ bao gồm nơi bán, giá chi tiết cũng như thời điểm cập nhật. Dữ liệu được thu thập từ các nguồn như lazada.vn, Nguyễn kim, Phankhang. Thời điểm cập nhật cũng là thời điểm trong ngày đây là ưu điểm lớn nhất của trang này.

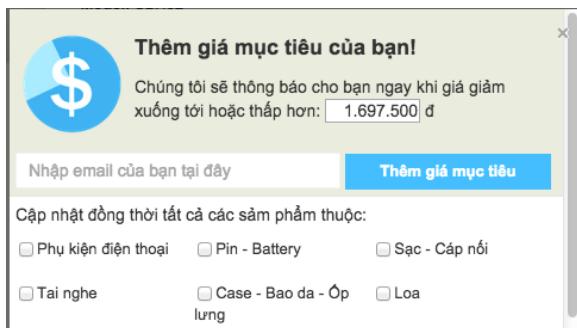
b) Trang websosanh.vn

Ngoài các thiết bị điện tử thì trang này cập nhật thêm các sản phẩm như ô tô, xe máy, thực phẩm, đồ uống, nội thất, thiết bị văn phòng, đồ thể thao. Ngoài chức năng so sánh giá, trang còn có thêm mục thông tin, cung cấp các bài viết về xu hướng công nghệ, cách lựa chọn sản phẩm. Vì vậy ngoài thu thập thông tin sản phẩm, nền tảng của trang còn phải thu thập thêm các tin tức công nghệ từ các trang báo, diễn đàn và buôn bán khác.

 phukienquaovang.com	1.750.000 đ Update 23 giờ trước Tới nơi bán	Tai nghe Bluetooth Sony SBH52 phukienquaovang.com/tai-ngh...ghe-bluetooth-sony-sbh52.html Kho hàng: Liên hệ
 tuankhanh.com.vn	1.795.000 đ Update 1 ngày trước Tới nơi bán	Tai nghe Bluetooth Sony SBH-52 tuankhanh.com.vn/san-pham/tai-nghe-bluetooth-sony-sbh52.html Kho hàng: Liên hệ
 smartphonestore.vn	1.800.000 đ Update 7 giờ trước Tới nơi bán	=> PHỤ KIỆN SONY XPERIA > TAI NGHE SONY SBH52 www.smartphonestore.vn/tai-nghe-sony-sbh52-i98.html Kho hàng: Liên hệ
 phukien24h.net	1.850.000 đ Update 21 giờ trước Tới nơi bán	Tai nghe Sony SBH52 Bluetooth chống nước phukien24h.net/san-pham/1/phu...bh52-b Tới nơi bán Tai nghe Sony SBH52 Bluetooth chống nước 1.850.000 đ Kho hàng: Liên hệ
 68mua.vn	1.850.000 đ Update 23 giờ trước Tới nơi bán	Tai nghe Bluetooth Sony SBH-52 68mua.vn/shb 52 Kho hàng: Liên hệ

Hình 2.4 Kết quả tìm kiếm trả về từ trang websosanh.vn

Người dùng có thêm các tương tác như bình chọn sản phẩm ưu thích và thông báo khi giá giảm xuống mức đặt ra trước.



Hình 2.5 Chức năng thông báo khi giá giảm

Tuy nhiên người dùng phải đặt ra một mức giá có thể chấp nhận được khi sử dụng chức năng này.

c) Trang giadaure.vn và pricexchange.vn

Giao diện không được đẹp mắt như hai trang phía trước nhưng trang này là trang cung cấp nhiều thông tin cho người dùng nhất.

The screenshot shows the homepage of giadaure.com. At the top, there is a navigation bar with links for 'Điện thoại di động', 'Máy tính bảng', 'Thương hiệu', 'Tim theo tên, thương hiệu sản phẩm...', 'Bảng giá trực tuyến', and 'Báo cáo & Thống kê'. Below the navigation bar, there is a search bar and a 'Trang chủ' link. On the right side, there is a 'Cần thông tin giá rẻ thì' button with a Facebook 'Thích' count of 132. The main content area features a large banner for 'DELACon's Advanced Call Tracking Solution' with a laptop displaying a call tracking chart. Below the banner, there are three main sections: 'Bảng Giá Trực Tuyến' (with a red background), 'Báo Cáo & Thống kê' (with a green background), and 'Biến Động Thị Trường' (with a blue background). Each section has a brief description and a 'Learn more about' button. At the bottom, there is a section titled 'Top sản phẩm giảm giá mạnh' (Top products with significant price reduction) featuring four smartphones: Samsung Galaxy Alpha, Samsung Galaxy K zoom, Sony Xperia Z3 Compact - D5833, and HTC Desire 610 - D610X, each with a red discount badge indicating a 16.7% reduction.

Top sản phẩm giảm giá mạnh



Hình 2.6 Giao diện trang chủ giadaure.com

Sản phẩm hướng tới của trang là các thiết bị điện thoại và máy tính bảng.

Sản phẩm giảm giá từ 00h ngày hôm qua đến giờ:

Sản phẩm	Tỷ lệ giảm	Theo %	Giá cũ	Giá hiện tại	Đơn vị	Ngày
[Lịch sử] Samsung Galaxy Grand Prime - G530	-40,000	-8.02 %	4,990,000	4,590,000	Tech One	23/12/2014 15:22
[Lịch sử] Samsung Galaxy S4 - I9500	-100,000	-1.26 %	7,950,000	7,850,000	Nhật Cường	23/12/2014 15:16
[Lịch sử] Samsung Galaxy Grand 2 SM-G7102	-160,000	-3.03 %	4,950,000	4,800,000	Nhật Cường	23/12/2014 15:16
[Lịch sử] Samsung Galaxy Note 4 - N910	-40,000	-0.25 %	15,990,000	15,950,000	Nhật Cường	23/12/2014 15:12
[Lịch sử] HTC Desire 310 - 2 Sim	-1,000,000	-27.1 %	3,690,000	2,690,000	Nhật Cường	23/12/2014 15:00
[Lịch sử] HTC Desire 610 - D610X	-1,200,000	-20.37 %	5,890,000	4,690,000	Nhật Cường	23/12/2014 15:00
[Lịch sử] Nokia XL	-279,000	-7.6 %	3,669,000	3,390,000	Viễn Thông A	23/12/2014 14:36
[Lịch sử] Nokia X2 - 2 sim	-200,000	-6.92 %	2,890,000	2,690,000	Viễn Thông A	23/12/2014 14:30
[Lịch sử] HTC Desire 816G	-40,000	-0.61 %	6,590,000	6,550,000	Tech One	23/12/2014 14:21

Hình 2.7 Các sản phẩm được thông kê

Điện thoại di động												
ID	Điện thoại di động:	Giá của Bạn	Hoàng Hà Mobile	Anh Vũ Mobile	Thế Giới Di Động	Viettel bán lẻ	Pico	Huyen Mobile	Nguyễn Kim	Media Mart	Mai Nguyễn	Trần Anh
	⭐ Nokia N8	0 8,590,000	Không bán	▼ 9,449,000	Không bán	9,198,000	8,600,000	9,690,000	9,098,000	9,490,000	9,099,000	
	⭐ Nokia X1-01	0 880,000	▲ 777,000	▼ 939,000	939,000	868,000	899,000	870,000	788,000	960,000	779,000	
	⭐ HTC Sensation	0 14,100,000	13,550,000	13,999,000	13,999,000	13,788,000	Không bán	14,500,000	Không bán	14,490,000	Không bán	
	⭐ Nokia C3 - FPT	0 2,340,000	2,340,000	▼ 2,479,000	2,399,000	2,198,000	2,350,000	2,290,000	2,398,000	Tham khảo	2,329,000	
	⭐ NOKIA E6	0 6,550,000	6,950,000	▼ 7,249,000	7,375,000	7,168,000	6,720,000	Không bán	6,990,000	7,490,000	7,159,000	
	⭐ Nokia E71 - Grey, Full black, Red,White	0 4,990,000	5,490,000	▼ 5,529,000	Không bán	5,338,000	5,400,000	5,850,000	5,568,000	5,590,000	Tham khảo	
	⭐ NOKIA 2730 CLASSIC	0 1,600,000	Tham khảo	▼ 1,699,000	1,899,000	1,968,000	1,880,000	Clear	1,968,000	1,990,000	Tham khảo	
	⭐ IPHONE 4 BLACK 16 GB (PHIÊN BẢN QUỐC TẾ)	0 ▲ 12,500,000	Không bán	16,500,000	Không bán	Không bán	16,000,000	15,900,000	Không bán	16,990,000	Tham khảo	
	⭐ Samsung Galaxy Y S5360	0 3,180,000	▲ 3,150,000	3,379,000	3,369,000	3,368,000	Không bán	3,390,000	Không bán	3,390,000	Không bán	
	⭐ NOKIA 701	0 7,320,000	Tham khảo	▼ 7,799,000	7,989,000	7,798,000	7,350,000	7,990,000	7,288,000	Không bán	7,799,000	
	⭐ LG Optimus One P500	0 3,990,000	4,680,000	▼ 4,499,000	4,799,000	4,698,000	Không bán	4,500,000	Không bán	Tham khảo	Không bán	
	⭐ Nokia 2700	0 1,500,000	1,570,000	▼ 1,569,000	1,629,000	1,598,000	1,520,000	1,580,000	1,638,000	Tham khảo	1,649,000	
	⭐ Nokia E63 - Red, Blue, Black, White	0 3,060,000	Không bán	▼ 3,369,000	Không bán	3,068,000	3,150,000	3,790,000	3,598,000	Tham khảo	2,999,000	
	⭐ NOKIA 5130 XPRESSMUSIC	0 1,800,000	Không bán	▼ 1,899,000	Không bán	1,938,000	1,880,000	2,060,000	1,990,000	Không bán	1,999,000	
	⭐ Q-mobile She	0 ▼ 1,950,000	Không bán	Tham khảo	Không bán	1,968,000	Không bán	Không bán	Không bán	Không bán	Không bán	
	⭐ Nokia 1280	0 440,000	Không bán	▼ 449,000	Không bán	488,000	430,000	Không bán	435,000	499,000	439,000	
	⭐ NOKIA E7	0 9,090,000	10,650,000	▼ 10,359,000	10,299,000	10,668,000	Tham khảo	Không bán	11,368,000	10,990,000	Tham khảo	

Hình 2.8 Kết quả thống kê sản phẩm

Các thông kê được thực hiện khá chi tiết theo dạng bảng được cập nhật theo thời gian. Hệ thống cung cấp biểu đồ so sánh trực quan, thống kê thứ hạng tăng giảm giá từng sản phẩm của từng công ty giúp khách hàng có đánh giá đúng đắn nhất về biến động thị trường.

Trong ba ví dụ được đưa ra, trang giadaure.vn đầu tư nhiều nhất vào lĩnh vực so sánh giá nhưng sản phẩm hướng tới chỉ có điện thoại và máy tính bảng. Chưa đa dạng và chưa có tương tác với người dùng là điểm yếu của trang này.

PHẦN 3. KIẾN THỨC NỀN

Để thực hiện được các yêu cầu và chức năng đã nêu ở trên, cần một số kiến thức áp dụng các nền tảng đã được xây dựng sẵn để tiết kiệm và tận dụng các công cụ đã được xây dựng sẵn. Ngoài ra, việc sử dụng công cụ hợp lý giúp tiết kiệm thời gian sản phẩm còn tăng sự ổn định và phát triển sau này. Chương này nội dung chính giới thiệu hai nền tảng sử dụng chính trong luận văn.

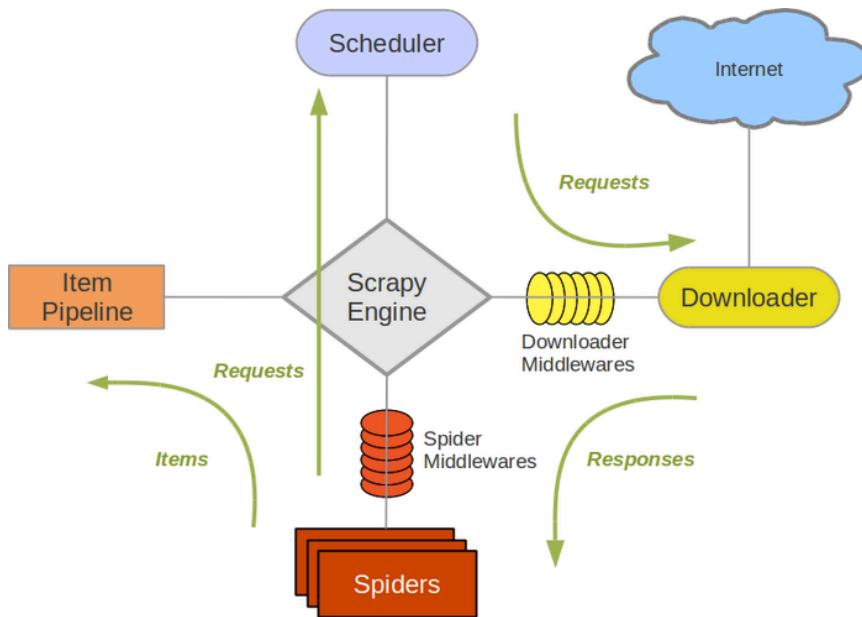
3.1. Công cụ thu thập dữ liệu Scrapy

Scrapy là một nền tảng giúp người dùng có thể xây dựng một công cụ thu thập dữ liệu tìm kiếm một cách nhanh nhất. Scrapy được viết bằng Python và bộ phận chính của nó sử dụng thư viện Twisted (<https://twistedmatrix.com/>) một thư viện xử lý các kết nối mạng bất đồng bộ khá mạnh dành cho ngôn ngữ lập trình Python. Vì viết bằng Python nên sau khi phát triển, source không cần phải build mà vẫn có thể hoạt động trên nền tảng Window, Mac, Python nếu bạn cài trình biên dịch của Python. Scrapy là một công cụ được rất nhiều sự hỗ trợ của cộng đồng, và các doanh nghiệp. Ngoài framework chính được cộng đồng xây dựng, có rất nhiều thư viện hỗ trợ xung quanh Scrapy giúp customize và hoạt động tốt hơn.

Các thống kê xung quanh Scrapy Framework:

- Hơn 2.900 câu hỏi trên trang StackOverflow.
- 6.700 Sao và 1.876 lượt Fork trên Repository chính trên Github.
(<https://github.com/scrapy/scrapy>)
- Hơn 40 công ty, sử dụng Scrapy làm công cụ thu thập dữ liệu trên trang web chính thức của Scrapy.

3.1.1. Kiến trúc công cụ



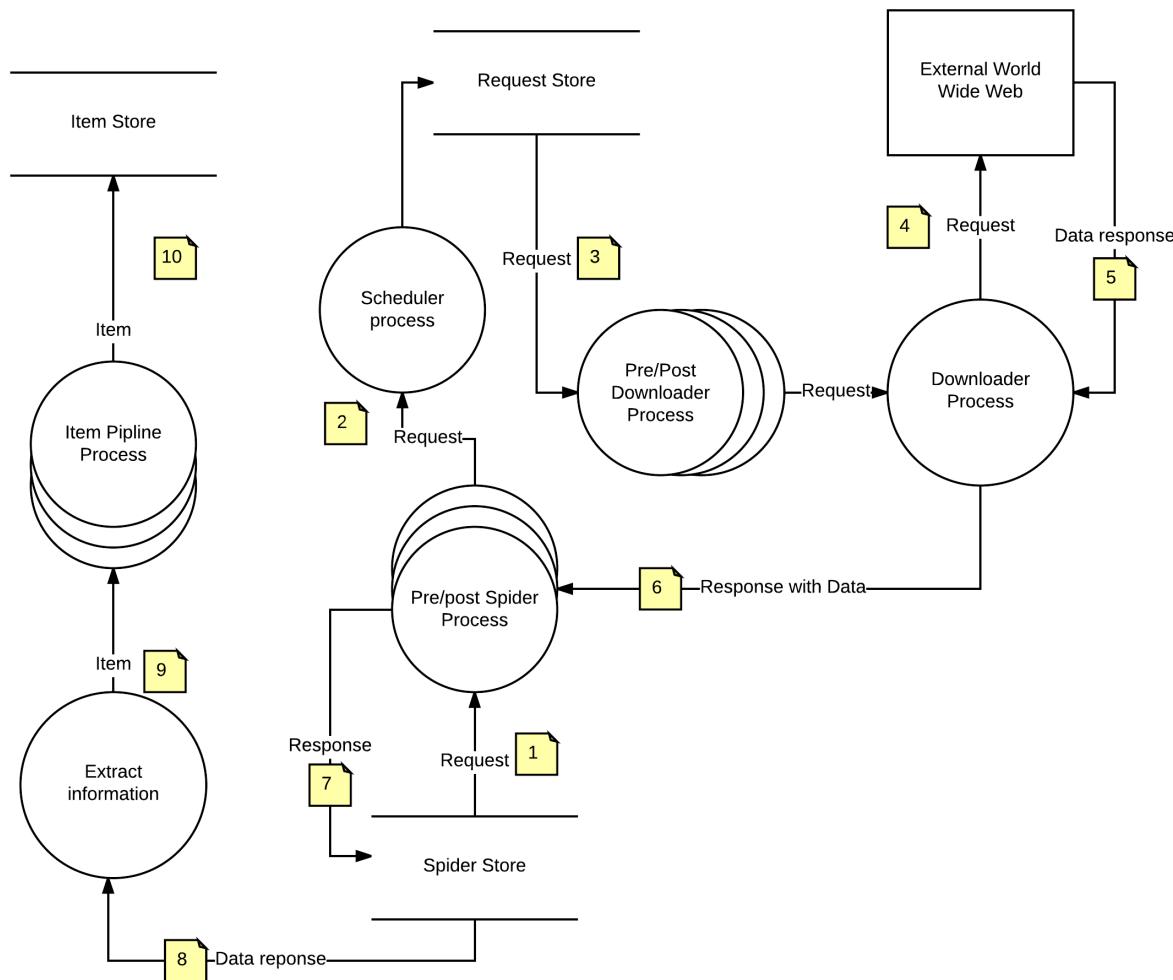
Hình 3.1: Cấu trúc Scrapy Framework

Các thành phần trong Scrapy:

- **Scrapy Engine:** đây là phần core chính của bộ thu thập thông tin. Các bộ phận khác, cũng như luồng thông tin được điều khiển bởi bộ phận này. Bản chất của Scrapy engine được hiện thực theo kiểu Event-Driven.
- **Spiders:** Mỗi Spiders sẽ ứng với mỗi trang khác nhau. Spiders sẽ cung cấp những trang bắt đầu tìm kiếm, và nhận lại phản hồi từ web service. Lấy ra các thông tin phù hợp, lấy các trang cần thu thập tiếp theo cũng là nhiệm vụ của Spiders. Nói cách khác, spider định nghĩa các chung ta thu thập dữ liệu từ một nguồn cố định.
- **Scheduler:** Các request được scrapy engine chuyển từ spider sang bộ scheduler. Bộ scheduler sẽ quản lý toàn bộ các request cũng như quyết định khi nào các request được chạy. Mình có thể sử dụng bộ Scheduler một cách hợp lý để tránh trường hợp bị trang web chặn.
- **Downloader:** Nhiệm vụ của downloader chỉ là mở các đường request tương ứng với thông tin được đưa và lấy nội dung trang và trả về lại cho bộ Scrapy Engine.
- **Downloader Middleware:** là phần trung gian giữa Scrapy Engine và Downloader. Các request trước và sau khi chạy đều được xử lý qua các tầng trung gian là các Downloader middleware. Một Downloader Middleware cần hiện thực 2 hàm chính:
 - **process_request(request, spider)** : Xử lý request trước khi download
 - **process_response(request, response, spider)** : Xử lý response sau khi download.
- **Spider middlewares:** là tầng trung gian giữa Scrapy Engine và các Spiders. Nó hỗ trợ cho người dùng có thể viết thêm các custom plugin để mở rộng. Spider Middleware cần hiện thực các hàm:

- **process_spider_input(response, spider)** : Xử lý request trước khi được spider xử lý.
 - **process_spider_output(response, result, spider)** : Xử lý request sau khi được spider xử lý. Result bao gồm các request khác hoặc là các Item được trích xuất từ văn bản.
 - **process_start_requests(start_requests, spider)** : Xử lý các request được khởi động từ spider.
- Item pipeline: các thông tin trích xuất được đóng gói vào một object gọi là Item. Các Item sẽ được đưa qua các tầng. Các Item pipeline được thiết kế nhằm làm sạch dữ liệu, xử lý dữ liệu trước khi được lưu trữ trong phạm vi luận văn này là lưu vào file JSON và ElasticSearch.

3.1.2. Hoạt động của Scrapy



Hình 3.2 Mô hình Data Flow của Framework Scrapy

Data flow của chương trình bao gồm các bước:

- Khi chương trình thu thập dữ liệu bắt đầu khởi động, các thông tin đường dẫn bắt đầu và domain cần request sẽ được lấy trong mỗi Spider (1). Các request được chuyển qua Spider Middleware (trong hình này biểu diễn bằng các process liên tục với nhau) và process cuối cùng là qua bộ định thời các request (2).
- Bộ định thời sẽ lưu trữ và quyết định các request sẽ được chạy lúc nào. Khi request được chạy, request sẽ được chuyển qua bộ Downloader. Trước khi đưa qua bộ downloader, các request sẽ được tiền xử lý như thêm header basic authenticate, thêm meta-data request (3).
- Bộ downloader sẽ tự động mở các connect ra bên ngoài (4) và lấy dữ liệu về ứng với request tương ứng (5).
- Dữ liệu trả về sẽ được tiền xử lý trong bộ Spider process (6).
- Spider sẽ xử lý các kết quả trả về. Spider sẽ lấy được các request mới từ các đường dẫn có trong request là thực hiện lại (1). Các thông tin được trích xuất sẽ dẫn sang Item pipeline (8) để có được các Item lưu vào trong cơ sở dữ liệu.

3.1.3. Sử dụng Xpath Selector để trích xuất dữ liệu

Xpath, hoặc Xpath languages, là ngôn ngữ truy vấn các node dữ liệu trong các tập tin XML. Xpath được định nghĩa bởi cộng đồng World Wide Web Consortium (W3C). Cây DOM (DOM tree) của HTML chính là một dạng cụ thể của XML. Xpath được viết dưới dạng một đường dẫn biểu thức dẫn đến một node hoặc một tập các node trong file XML. Biểu thức đường dẫn có dạng gần giống với đường dẫn truyền thống trong hệ điều hành.

Ứng với mỗi trang thương mại điện tử, sẽ có một template riêng để trình bày dạng dữ liệu vì vậy cần định nghĩa riêng Xpath Selector cho mỗi trang như link chi tiết mỗi bài nằm trong thẻ nào, title của bài viết nằm trong thẻ nào, ...

Sau đây là bảng giới thiệu cách viết Xpath selector và cách sử dụng:

Bảng 3.1 Biểu thức và giải thích cú pháp của Xpath Selector

Biểu thức	Giải thích
/	xác định tuyệt đối là thẻ con trực tiếp
//	xác định tương đối bao gồm các thẻ con trực tiếp và các thẻ con gián tiếp
@	lấy một thuộc tính của một thẻ cho trước
preceding::	Lấy các thẻ phía trước thẻ hiện tại
following::	Lấy các thẻ phía sau thẻ hiện tại
html	chọn thẻ có tên “html” ở trong vị trí đó
html/header	chọn thẻ “header” là con trực tiếp của thẻ “html”
//div	chọn tất cả các thẻ div nằm trong văn bản
html/header//meta	chọn tất cả các thẻ meta nằm trong header là con trực tiếp của thẻ html

//a[@id="next_page"]/@href	Lấy thuộc tính href của thẻ có id là “next_page”
//div[@class="book_title"]	lấy tất cả các thẻ div nằm trong văn bản có thuộc tính “class” là “book_title”
//span[@class="price"]/text()	chọn các thẻ span có “class” là price và lấy toàn bộ các chuỗi ở trong thẻ đó.
li[@class="active"]/following::a	Chọn các thẻ a theo phía sau một thẻ li có class là “active”
//div[@id="content"]/preceding::a	Chọn các thẻ a phía trước thẻ div có id là “content”

Trong phần hiện thực, toàn bộ template của một trang web được định nghĩa bằng biểu thức Xpath Selector. Ví dụ trong tiki_spider:

- xpath_list = "//a[@class='b-product-item_link']/@href"
- xpath_nextPage = "//li[contains(@class, 'b-pager_paging-arrow_next')]/a/@href"

Các title được định nghĩa là các thẻ a có class là 'b-product-item_link' sau đó mình chọn thuộc tính href của các thẻ đó.

3.1.4. Sử dụng Scrapy Framework

Nhược điểm gặp phải khi sử dụng Scrapy là framework được xây dựng khá mở và còn sơ khai. Để hiện thực Scrapy nhằm vào một mục đích cụ thể là thu thập thông tin của các trang bán sách một cách thuận tiện và tái sử dụng các đoạn code, việc thiết kế thêm các lớp abstract và các hàm hỗ trợ đã được hiện thực.

3.2. Elasticsearch

3.2.1. Giới thiệu về Elasticsearch

Elasticsearch là một open source mới được phát hành từ 02/2010 được cộng đồng sử dụng khá nhiều bởi tính dễ sử dụng và nhiều tính năng mới, là một hệ thống đánh chỉ mục (indexing) và tìm kiếm (searching) thời gian thực. Elasticsearch được sử dụng ở hạ tầng hỗ trợ cho ứng dụng thực hiện các chức năng như lưu trữ, tìm kiếm và phân tích dữ liệu khỏi lượng lớn một cách nhanh chóng và gần như realtime. Tìm kiếm luôn là nhu cầu bắt buộc khi khỏi lượng dữ liệu trở nên lớn. Tìm một tập tin trong máy tính, tìm một từ trong một bài báo cáo, tìm một món đồ trong trang thương mại điện tử. . Nhưng điều đặc biệt nhất của ElasticSearch đến từ khả năng phân tán của nó. Được thiết kế ban đầu với khả năng Scale-out khá tốt. ElasticSearch đã tránh khỏi những điều Solr gặp phải. Những trang nổi tiếng đang sử dụng ElasticSearch rất quen thuộc với chúng ta: Wikipedia, StackOverflow, FourSquare và GitHub. Ứng dụng của Elasticsearch được sử dụng khá nhiều trong thực tế:

- Wikipedia sử dụng Elasticsearch để hỗ trợ chức năng tìm kiếm full-text có kèm theo chức năng đồ độ đậm các kết quả tìm kiếm, gợi ý khi người dùng đang gõ cũng như gợi ý các kết quả tìm kiếm khác.
- GitHub sử dụng Elasticsearch để đánh chỉ mục tìm kiếm cho toàn bộ source code hơn 130 triệu dòng.

Các yêu cầu đặt ra đối với một công cụ tìm kiếm:

- Tìm kiếm bằng từ khoá
- Tìm kiếm theo ID
- Tìm kiếm theo thuộc tính
- Tô đậm các kết quả tìm kiếm
- Gợi ý tìm kiếm
- Gom nhóm và sàng lọc dữ liệu
- Thêm dữ liệu và lưu trữ

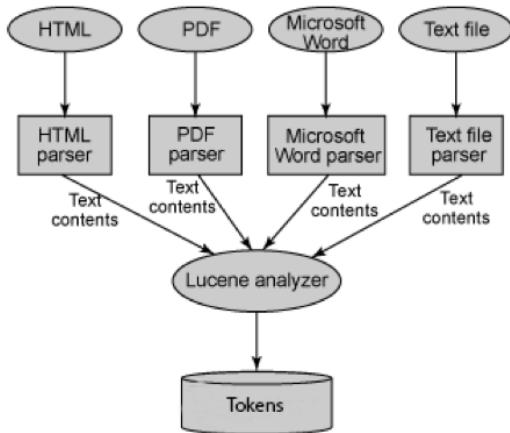
Các ưu điểm khi sử dụng Elasticsearch bao gồm:

- Cài đặt đơn giản: việc cài đặt Elasticsearch có thể thực hiện rất đơn giản và có thể sử dụng ngay. Các bước cài đặt bao gồm cài đặt Java SDK, tải Elasticsearch từ trang chủ và chạy.
- Toàn bộ kết nối, giao tiếp với Elasticsearch Server tuân thủ theo chuẩn RESTful. Mỗi method của HTTP sẽ ứng với function trên Elasticsearch Server. GET method dùng để tìm kiếm dữ liệu. POST method dùng để tạo dữ liệu mới. PUT method dùng để sửa dữ liệu đã có. DELETE method dùng để xoá các dữ liệu.
- Được xây dựng dựa trên Apache Lucene, đây là thư viện index và tìm kiếm dữ liệu được phát triển khá lâu đời.

3.2.2. Kỹ thuật tokenizer bằng Apache Lucene và Inverted index

Lucene là một thư viện nguồn mở cho việc đánh chỉ mục và tìm kiếm văn bản được phát triển bởi Dough Cutting và phát hành bởi Apache Software Foundation. Lucene được viết bằng ngôn ngữ Java, tuy nhiên đã được phát triển bởi nhiều ngôn ngữ khác. Lucene được sử dụng rất nhiều trong thực tế và nhất trong các bộ máy tìm kiếm (search engine), một số ứng dụng tiêu biểu sử dụng Lucene làm nền tảng như:

- Apache Nutch — hỗ trợ web crawling và HTML parsing
- Elasticsearch — enterprise search server
- Compass — Java Search Engine Framework
- DocFetcher — multiplatform desktop search application
- Wikipedia dùng Lucene để tìm kiếm nội dung toàn bộ văn bản.
- CNET dùng Lucene để tìm kiếm danh sách thẻ loại sản phẩm.



Hình 3.3 Tổng quan Lucene

Ví dụ về việc sử dụng apache lucene để xử lý từ tiếng Anh: Trong apache lucene. Các bộ phân tích (Analyzer) được xây dựng sẵn có các nhiệm vụ khác nhau và được dùng cho những trường hợp đa dạng trong thực tế, xét ví dụ phân tích chuỗi văn bản sau: “The quick brown fox jumped over the lazy dog,” được thực hiện với 4 bộ phân tích được xây dựng sẵn trong Apache Lucene:

Analyzing "The quick brown fox jumped over the lazy dog"

- WhitespaceAnalyzer: [The] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dog]
- SimpleAnalyzer: [the] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dog]
- StopAnalyzer: [quick] [brown] [fox] [jumped] [over] [lazy] [dog]
- StandardAnalyzer: [quick] [brown] [fox] [jumped] [over] [lazy] [dog]

Trong ví dụ trên, sau khi quá trình phân tích hoàn tất, mỗi bộ phân tích sẽ đưa ra những kết quả khác nhau. chuỗi văn bản ban đầu đã được tách thành các token và thêm vào đó là một số bộ phân tích có thể thực hiện các chức năng như loại bỏ các từ phổ biến (stop words) đưa toàn bộ về dạng lowercase,...

Sau khi tokenizer, các token sẽ được mapping ngược lại với document chứa nó, giai đoạn này gọi là Inverted Index. Ví dụ ta có 3 document bao gồm D1, D2, D3:

D1 = “hom nay troi mua”

D2 = “hom nay troi nang”

D3 = “mot ngay dep troi”

Ta sẽ có Interted Index cho 3 tài liệu trên như sau:

“hom” => {D1, D2}

“nay” => {D1, D2}

“troi” => {D1, D2, D3}

“mua” => {D1}

“nang” => {D2}

“mot” => {D3}

“ngay” => {D3}

“dep” => {D3}

Khi đó ta muốn kiểm các văn bản có từ “nắng” hoặc “đẹp” thì kết quả trả về là phép hợp của hai tập hợp $\{D2\} \cup \{D3\} = \{D2, D3\}$

3.2.3. Định nghĩa cấu trúc trong Elasticsearch

Trong Elasticsearch, một đơn vị được index được gọi là một Document. Một document bao gồm nhiều Field. Trong hệ quản trị cơ sở dữ liệu, document được ứng với một hàng (Row) còn Field được ứng với cột (Col). Nhiều document được chứa trong một Types, trong Database là Table. Nhiều table sẽ được lưu trữ trong một Database, trong Elasticsearch gọi là một Indices. Ta có một bản so sánh như sau:

Relational DB \Rightarrow Databases \Rightarrow Tables \Rightarrow Rows \Rightarrow Columns

Elasticsearch \Rightarrow Indices \Rightarrow Types \Rightarrow Documents \Rightarrow Fields

3.2.4. Sử dụng Elasticsearch trong luận văn

Trong phần hiện thực, toàn bộ dữ liệu sau khi được tiền xử lý, sẽ được lưu trữ vào trong một Elasticsearch Server riêng. Elasticsearch sẽ thay thế phần lớn chức năng của một database. Đây là điểm mới trong việc sử dụng một công cụ Index và search để lưu trữ toàn bộ dữ liệu, việc mất mát dữ liệu rất dễ xảy ra ngoài ý muốn như việc update và xoá dữ liệu có thể ảnh hưởng tới tất cả document còn lại nên cần phải backup dữ liệu cũng như test cẩn thận trước khi chạy các lệnh.

3.3. Cấu trúc của CMS PhalconEye

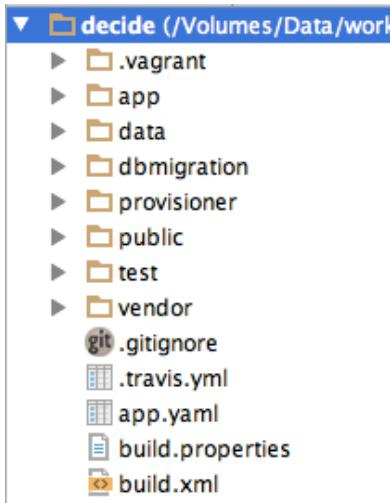
PhalconEye là một CMS mới được xây dựng dựa trên Phalcon PHP Framework. Nó chứa một số nền tảng được xây dựng sẵn nhằm nâng tốc độ phát triển sản phẩm như kiến trúc multi Module, các Widget, plugin và themes. Ban đầu PhalconEye được thiết kế để giúp các nhà lập trình web phát triển các dự án Web khác nên các phần được xây dựng nhằm cho mục đích phát triển nhiều hơn là như các CMS khác đang có mặt như Joomla hoặc Wordpress.

Phalcon là open source, framework đầy đủ tất cả các phần được viết dưới dạng C-extension để tối ưu hóa hiệu năng của hệ thống. Phalcon được xây dựng với tốc độ xử lý khá nhanh và hiệu quả nhưng cần xây dựng thêm một số chức năng phụ trợ để dễ dàng sử dụng hơn.

Người phát triển sử dụng PhalconEye dưới dạng một Foundation (nền tảng) không những tận dụng được tốc độ của Phalcon và còn tận dụng được các chức năng mà PhalconEye xây dựng.

Việc phát triển thêm sẽ ít tốn thời gian hơn phù hợp với việc sử dụng để viết trang web tiếp cận người dùng.

3.3.1. Cấu trúc thư mục của PhalconEye:

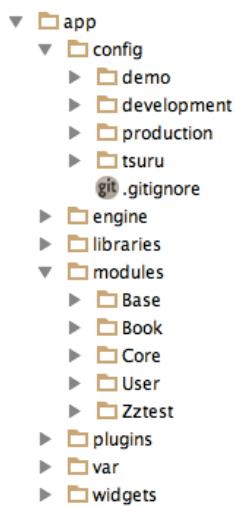


Hình 3.4 Cấu trúc cây thư mục của PhalconEye Framework

Project được chia vào hai thư mục chính là

- app: chứa toàn bộ chương trình chính được viết bởi người phát triển bằng ngôn ngữ PHP.
- public: đây là thư mục chứa các tài nguyên khác được truy cập từ HTTP Server.

a) Thư mục app là thư mục chứa các phần chính của Web bao gồm:

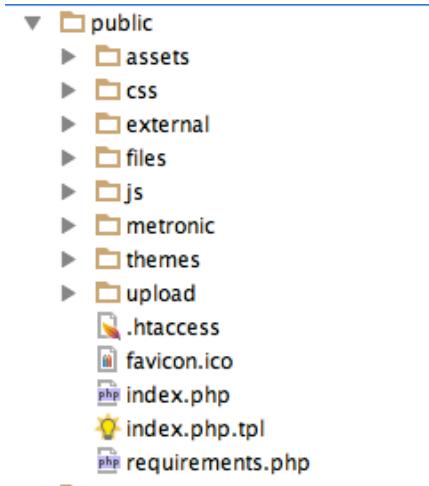


Hình 3.5 Cấu trúc cây thư mục app trong PhalconEye CMS

- Thư mục config: chứa các file cấu hình hệ thống ứng với mỗi môi trường sẽ có những cấu hình khác nhau

- Thư mục engine: đây là “trái tim” của CMS. Các phần phát triển thêm dựa trên Phalcon Framework hỗ trợ người dùng thuận tiện hơn.
- Thư mục libraries: chứa các thư viện hỗ trợ
- Thư mục modules: ứng với mỗi chức năng sẽ xây dựng một module tương ứng
- Thư mục plugins: chứa các file hiện thực plugin
- Thư mục var: chứa các file log, cache và file setting app.
- Thư mục widgets: chứa các widgets con để thêm vào view.

b) Thư mục public:

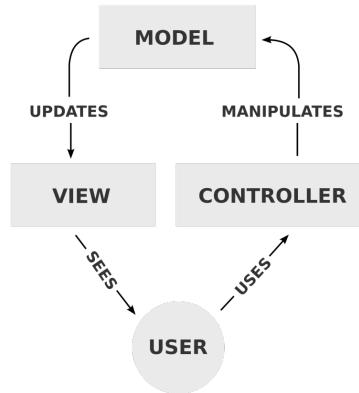


Hình 3.6 Cấu trúc cây thư mục Public của PhalconEye CMS

- Thư mục assets: chứa các file cần để chạy ứng dụng như css, js và hình.
- Thư mục external: chứa các thư viện khác hỗ trợ giao diện như jQuery và Bootstrap
- Thư mục upload: chứa các file người dùng đăng lên mạng như ảnh avatar.

3.3.2. Thiết kế và mô hình hoạt động của PhalconEye

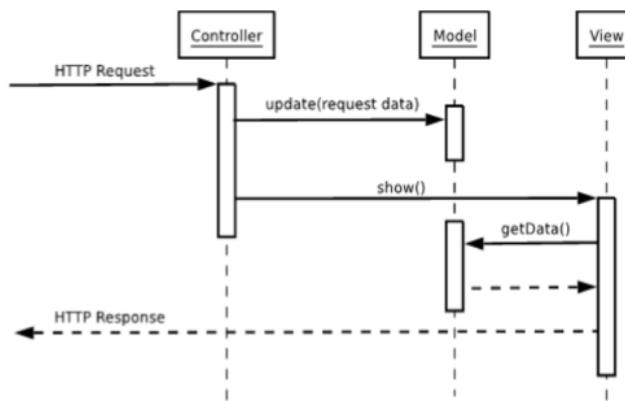
CMS hoạt động theo ba mô hình chính là Model-View-Controller (MVC), Dependency Injection (DI).



Hình 3.7 Mô hình MVC

Mô hình MVC là kiến trúc được sử dụng khá phổ biến trong phát triển phần mềm. Khi viết theo mô hình MVC người dùng sẽ tách biệt và làm rõ hơn các chức năng của các phần tránh trùng lắp, tái sử dụng code nhiều, việc viết các lớp luận lý được tách biệt với các phần xử lý request và đọc ghi database.

- Model: Giao tiếp và truy xuất cơ sở dữ liệu, đặc tả cơ sở dữ liệu bằng các class, viết các hàm kiểm tra trước khi insert/update và sau khi request từ cơ sở dữ liệu.
- Controller: Là lớp đảm nhiệm vai trò hiện thực nghiệp vụ của chương trình. Đầu vào của controller là các request từ phía web browser và trả về các thông tin sau khi xử lý. Đây là phần được viết phần test để kiểm tra. Tập hợp các Controller chính là API của hệ thống.
- View: là phần trình bày những thông tin được trả ra từ controller. Lớp View có thể chứa nhiều kiểu template và thay đổi tùy vào cấu hình. Nhiệm vụ của lớp View là thể hiện thông tin đầy đủ cho người dùng nhận biết.



Hình 3.8 Lượt đồ hoạt động của Mô hình MVC

Khi một request webservice, các bộ phận như URL (hoặc Dispatcher) sẽ quyết định request đó sẽ được controller nào xử lý. Controller sẽ xử lý thông tin và cập nhật với database thông qua lớp Model. Các thông tin sau khi được cập nhật sẽ được chuyển qua tầng View. Tầng view có

trách nhiệm lấy các thông tin cần thiết và trình diễn trên nền template. Response sau đó sẽ được trả lại cho người dùng.

Mô hình Dependency Injection hay còn có một tên là mô hình Inversion of Control được đưa vào trong Phalcon để tận dụng lại các Service được viết sử dụng chung ở nhiều lớp như Session, Request, Response, FileSystem, Flash. Mô hình này hạn chế việc liên tục tạo khi các Service được sử dụng, truy xuất các Service một cách dễ dàng. Việc sử dụng DI còn khiến cô lập các hàm và khiến các hàm có khả năng kiểm tra được. Mình có thể viết một Mock giả để kiểm tra hàm đang sử dụng như ý muốn hoặc thay thế bằng các đối tượng khác khi nâng cấp.

Ví dụ :

Trong file #app/engine/ApplicationInitialization.php:

```
$di->set(
    'es',
    function () use ($config) {
        $params['hosts'] = ['backend.thinhvoxuan.me'];
        $params['logging'] = true;
        $params['logPath'] = 'elasticSearch.log';
        $params['logPermission'] = 0664;
        $client = new \Elasticsearch\Client($params);
        return $client;
    },
    true
);
```

Ta gán giá trị 'es' là một Service là Elasticsearch\Client

Sau đó trong file #app/modules/book/Service/BookService.php

Ta có thể gọi biến đã được Inject vào trong DI và sử dụng:

```
$es = $this->di->getEs();
$params['index'] = 'book';
$params['type'] = 'tiki';
$params['body'] = [
    'query' => [
        'match_all' => []
    ],
];
$result = $es->search($params);
return $result['hits']['hits'];
```

Bản chất của mô hình DI chính là một biến có hai tính chất là singleton và global variable.

3.3.3. Sử dụng và phát triển thêm PhalconEye trong luận văn:

Sử dụng PhalconEye trong luận văn tốt nghiệp nhằm hiện thực giao diện tương tác với người dùng. Giao diện ứng dụng đã được viết lại theo giao diện Metronic E-commerce. Người dùng có thể đăng kí, đăng nhập, tìm kiếm sách và bookmark lại những cuốn sách mình thích.

PHẦN 4. GIẢI THUẬT SO TRÙNG ĐỐI TƯỢNG SÁCH

Trong chương này, luận văn sẽ trình bày hai thuật toán chính được sử dụng để so sánh các đối tượng sách với nhau. Hai thuật toán bao gồm thuật toán Levenshtein Distance và thuật toán tính Cosine Similarity dựa trên các thông tin thu thập được là tên và mô tả cuốn sách. Các giải thuật này đã được áp dụng thêm phần tokenizer để tăng độ chính xác với tiếng Việt.

4.1. Thuật toán sử dụng

4.1.1. Công thức Levenshtein distance và giải thuật tính Edit Distance

Độ giống nhau giữa hai chuỗi (Minimum Edit Distance) cho trước, còn được gọi là khoảng cách giữa hai chuỗi (String Distance), được định nghĩa khác là số phép thực hiện để biến đổi chuỗi thứ nhất sang chuỗi thứ hai. Các phép được thực hiện bao gồm thêm (insertion) một chữ, xoá (deletion) một chữ và thay (substitution) một chữ bằng một chữ khác. Định nghĩa này đã được đề xuất bởi Vladimir Levenshtein vào những năm 1965.

Ứng dụng của việc tính toán khoảng cách giữa hai chuỗi được sử dụng khá nhiều trong thực tế nhằm sửa lỗi sai khi gõ văn bản bằng cách so các từ với lại các chuỗi đã có trong cơ sở dữ liệu và đưa ra danh sách các từ có độ giống nhau cao nhất.

4.1.2. Công thức tính Levenshtein Distance hay Minimum Edit Distance (MED)

Theo toán học, khoảng cách Levenshtein giữa hai chuỗi cho trước a và b gọi là $\text{lev}_{a,b}(|a|, |b|)$

$$\text{Lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0 \\ \min \begin{cases} \text{lev}_{a,b}(i - 1, j) + 1 \\ \text{lev}_{a,b}(i, j - 1) + 1 \\ \text{lev}_{a,b}(i - 1, j - 1) + F(a_i, b_j) \end{cases} & \text{otherwise} \end{cases}$$

$$F(\text{char1}, \text{char2}) = \begin{cases} 0 & \text{if } \text{char1} = \text{char2} \\ 1 & \text{otherwise} \end{cases}$$

Giải thích công thức: Đây là công thức được viết theo kiểu đệ quy

- Nếu một trong hai chuỗi bằng rỗng, thì $\text{Lev}_{a,b}$ sẽ bằng độ dài chuỗi còn lại vì ta sẽ tốn k lần insert ta sẽ có được chuỗi từ một chuỗi rỗng.
- Ngược lại, $\text{Lev}_{a,b}$ sẽ là minimum của 3 hàm
 - Lev của chuỗi thứ nhất có xoá 1 kí tự cuối cùng và chuỗi thứ 2. Cộng 1 vì thực hiện 1 phép xoá kí tự.
 - Lev của chuỗi thứ hai và chuỗi thứ 2 có xoá một kí tự cuối cùng. Cộng 1 vì thực hiện 1 phép xoá kí tự.

- Lev của chuỗi thứ nhất có xoá 1 kí tự cuối cùng và chuỗi thứ hai có xoá 1 kí tự cuối cùng cộng với phép biến đổi kí tự cuối cùng. Nếu 2 kí tự cuối cùng giống nhau thì $F = 0$, ngược lại $F = 1$.

4.1.3. Ví dụ công thức Minimun Edit Distance

Ví dụ tính Minimun Edit Distance giữa hai từ “kitten” và “sitting”

Bắt đầu từ từ “kitten”:

- Ta thực hiện một phép đổi chữ “k” thành chữ “s”: kitten => sitten
- Ta thực hiện tiếp phép biến đổi chữ “e” thành chữ “i”: sitten => sittin
- Cuối cùng ta thực hiện một phép insert chữ “g” vào cuối: sittin => sitting

Vậy MED(“kitten”, “sitting”) = 3

4.1.4. Giải thuật tính Minimun Edit Distance

Vì định nghĩa công thức Levenshentein Distence được định nghĩa theo kiểu đệ quy, nên có nhiều giải thuật hiện thực công thức theo mô hình lập trình động nhằm tăng tốc độ tính toán và ít tốn bộ nhớ hơn.

Giải thuật nhận 2 tham số đầu vào là 2 chuỗi và trả về khoảng cách giữa chúng. Độ phức tạp của thuật toán là $O(M*N)$ với M, N lần lượt là độ dài hai chuỗi đầu vào.

Giải thuật: Minumin Edit Distance

Input: chuỗi target và chuỗi source

Output: số phép cần thực hiện N để chuyển chuỗi target sang source

```

n := LENGTH(target)
m := LENGTH(source)
Create a distance matrix distance[n+1, m+1]
distance[0,0] := 0
for each column i from 0 to n do
    for each row j from 0 to m do
        distance[i,j] := MIN(distance[i-1,j] + ins-cost(target[i]),
                               distance[i-1,j] + delete-cost(target[i]),
                               distance[i,j-1] + ins-cost(source[i]),
                               distance[i,j-1] + delete-cost(source[i]),
                               distance[i-1,j-1] + sub-cost(target[i], source[i]))
N := distance[n,m]
return N

```

Hình 4.1 Giải thuật Minimun Edit Distance

n	9	10	11	10	11	12	11	10	9	8
o	8	9	10	9	10	11	10	9	8	9
i	7	8	9	8	9	10	9	8	9	10
t	6	7	8	7	8	9	8	9	10	11
n	5	6	7	6	7	8	9	10	11	12
e	4	5	6	5	6	7	8	9	10	11
t	3	4	5	6	7	8	9	10	11	12
n	2	3	4	5	6	7	8	8	10	11
i	1	2	3	4	5	6	7	8	9	10
#	0	1	2	3	4	5	6	7	8	9
	#	e	x	e	c	u	t	i	o	n

Hình 4.2 Ma trận kết quả khi chạy thuật toán Minumun Edit Distance

Việc tính toán $distance[i,j]$ sẽ phụ thuộc vào $distance[i-1,j]$, $distance[i-1,j-1]$ và $distance[i,j-1]$ nên kết quả của giải thuật chính là $distance[m,n]$ (là góc của hình chữ nhật).

$$MED("intention", "execution") = 8.$$

Trong giải thuật này ngoài việc giải quyết bài toán tính Levenshtein Distance theo mô hình lập trình động, giải thuật còn tổng quát hoá các phép biến đổi bằng các hàm khác nhau thay vì một hằng số cố định như trong công thức gốc. Ba hàm được định nghĩa mới nhằm thể hiện 3 phép biến đổi có thể thực hiện bao gồm insert-cost (chi phí khi insert thêm một ký tự), subst-cost (chi phí khi thay thế một ký tự này bằng ký tự khác) và delete-cost(khi phí khi xoá một ký tự).

Việc tổng quát hoá này được thực hiện nhằm thay đổi Edit Distance cho phù hợp với thực tế hơn. Ví dụ trong sự lỗi sai khi gõ văn bản:

- $subst-cost('v', 'b') = 0.5$
- $subst-cost('a', 'b') = 1$

Giải thích: Xác suất gõ nhầm ký tự ‘v’ thay cho ký tự ‘b’ cao hơn xác suất gõ nhầm ký tự ‘a’ thay cho ký tự ‘b’ bởi vì lý do khách quan là phím ‘b’ và ‘v’ nằm cạnh nhau trên bàn phím máy tính nên xác xuất cao hơn. Các giá trị của các phép biến đổi càng chính xác thì giải thuật tính Edit Distance càng chính xác.

4.1.5. Áp dụng trong luận văn

Trong luận văn này, giải thuật tính Edit Distance sẽ được dùng để so trùng các quyển sách dựa trên tên của cuốn sách khác nhau. Edit Distance của tựa cuốn sách càng nhỏ nghĩa là hai cuốn sách có khả năng là chung một cuốn sách. Tuy nhiên tên sách cần được chỉnh sửa và làm sạch loại bỏ các từ còn thừa (Ví dụ: “tái bản”, “bản đẹp”, “mới”, “tặng kèm”,...) trước khi sử dụng giải thuật để nâng cao chất lượng của giải thuật đạt yêu cầu đề ra của bài toán.

- Cấu trúc dữ liệu của một cuốn sách được thu thập về

Bảng 4.1 Cấu trúc dữ liệu của đối tượng sách

Tên - Kiểu dữ liệu	Mô tả	Ví dụ
Title – Chuỗi	Tên của cuốn sách	“Harry Potter Và Phòng Chứa Bí Mật - Tập 2”
spider_name – Chuỗi	Tên crawler thu thập	nobita_spider
Description – Chuỗi	Giới thiệu sách	“Giới thiệu sách Harry Potter Và Phòng Chứa Bí Mật - Tập 2 Harry Potter và phòng chứa bí mật , không như những bộ truyện nhiều tập khác...”
sale off – Số	Giá đang giảm	102000
reg_price – Số	Giá thông thường	102000
old_price – Số	Giá sách cũ	120000
Source – Chuỗi	Nguồn thu thập sách	nobita.vn
link – Chuỗi	Đường dẫn tới đối tượng	http://nobita.vn/454/harry-potter-va-phong-chua-bi-mat-tap-2.html
Author – Chuỗi	Tác giả cuốn sách	J. K. Rowling
Information – Từ điển dạng {key1: value1, key2: value2, ...}	Thông tin chi tiết của cuốn sách. Được định dạng kiểu dictionary	{"Kích thước": "14 x 20 cm", "Ngày phát hành": "19/10/2014", "Số trang": "408", "Danh mục": "Huyền bí - Giả tưởng", "Trọng lượng": "520", "NXB": "NXB Trẻ", "Phát hành": "NXB Trẻ", "Tác giả": "J. K. Rowling", "Lượt xem": "26"}
parent_link – Chuỗi	Đường link dẫn tới link của đối tượng sách	http://nobita.vn/danh-muc/2/van-hoc-nuoc-ngoai.html?page=5
Update date	Thời gian cập nhật	2014-12-27

b) Áp dụng giải thuật Minimun Edit Distance cho các tựa sách

Thực hiện một số thực nghiệm so sánh chuỗi tự sách “Harry Potter Và Phòng Chứa Bí Mật - Tập 2” với tất cả các chuỗi sách còn lại và lấy các chuỗi có Minimun Edit Distance bé nhất.

Bảng 4.2 Kết quả thực nghiệm với các tựa sách khi tính MED

Tên sách	MED
“Harry Potter Và Hoàng Tử Lai - Tập 6”	16
“Harry Potter Và Chiếc Cốc Lửa - Tập 4”	17
“Harry Potter Và Bảo Bối Tử Thần - Tập 7”	18
“Harry Potter Và Hòn Đá Phù Thủy - Tập 1”	18
“Harry Potter Và Phòng Chứa Bí Mật - Tập 2 (Tái Bản 2013)”	18
“Nhân Chứng Đã Chết”	37
“Hoàng Tử Bé”	38
“Điều Bí Mật”	38

Nhận xét: ta thấy MED giữa hai chuỗi là một số dương tăng. Giá trị của MED càng nhỏ thì hai cuốn sách có độ tương quan càng lớn (chỉ dựa theo tên). Tuy nhiên cần phải có một phép quy đổi MED về biên độ $[0,1]$ và tính hệ số giữa MED phụ thuộc theo độ dài của hai chuỗi đầu vào. Dựa vào giá trị đó, có thể đặt một ngưỡng trên định nghĩa 2 cuốn sách là một ví dụ như 0.9.

4.2. Độ tương tự của hai chuỗi dựa vào Minimum Edit Distance

Hai chuỗi có độ dài $N1, N2$ có MED D thì D thuộc trong đoạn $[0, \text{Max}(N1, N2)]$, đơn giản ta có thể tính độ tương tự của hai chuỗi A, B bằng cách:

$$\text{similar}(A, B) = (\text{Max}(N1, N2) - D) / \text{Max}(N1, N2)$$

Công thức 4.1 Navie similar

Tuy nhiên công thức này không thể hiện được sự ảnh hưởng của độ dài của hai chuỗi đến kết quả tương đồng mà chỉ phụ thuộc vào độ dài chuỗi dài hơn. Việc này sẽ ảnh hưởng khá nhiều đến kết quả tính toán.

Ta cấu hình lại số phép biến đổi như nhau:

- Ins-cost := 1 (vì phép chỉ thêm một kí tự trên một chuỗi.)
- Del-cost := 1 (vì phép chỉ xoá một kí tự trên một chuỗi)
- Sub-cost := 2 (vì phép này thay đổi một trên mỗi chuỗi)

Giải thuật 2:

Input: Chuỗi target và chuỗi source

Output: Độ tương đồng của hai chuỗi target và source S, S thuộc [0,1]

```
n := LENGTH(target)
m := LENGTH(source)
edit := minimum_edit_distance(target, source)
S := (m + n - edit) / (m + n)
Return S
```

Hình 4.3: Giải thuật Custom Minimum Edit Distance

$$\text{similar}(A, B) = (N1 + N2 - D) / (N1 + N2)$$

Công thức 4.2: Custom similar

Thực nghiệm hai công thức:

Bảng 4.3 Thử tính với hai công thức với chuỗi

Chuỗi 1	Chuỗi 2	MED-1	CT 4.1	MED-2	CT 4.2
‘abcdkfeeee’	‘abcded’	4	0.556	5	0.667
‘abcdkfeeee’	‘abdkf’	2	0.778	2	0.875

Nhận xét:

Trong trường hợp 1: Số phép cần thực hiện là:

- Xoá kí tự ‘k’ ở cuối chuỗi thứ 1
- Xoá kí tự ‘f’ ở cuối chuỗi thứ 1
- Xoá kí tự ‘e’ ở chuỗi thứ 1
- Đổi một kí tự ‘e’ ở cuối cùng thành kí tự ‘d’

Ta sẽ có được chuỗi thứ 2.

MED được tính theo công thức 1 là 4. Độ tương đồng của hai chuỗi là 0.556

MED được tính theo công thức 2 là 5 (Vì phép hoán đổi được tính 2 lần). Độ tương đồng của hai chuỗi là 0.667.

Trong trường hợp 2: Ta thực hiện hai phép xoá kí tự cuối cùng ở chuỗi thứ 1 sẽ có được chuỗi thứ 2 như mong muốn. MED của 2 công thức đều bằng nhau. Nhưng ta lại thu được giá trị của công thức 2 lớn hơn giá trị của công thức 1.

Tiếp sau đây, luận văn sẽ sử dụng công thức 2 để tính toán độ tương đồng của hai chuỗi.

a. Xử lý các chuỗi trong tên đối tượng sách

Các kết quả ban đầu chạy hiện thực

```
-- Kinh Tế Học Hài Hước (Tái Bản 2014) --
Kinh Tế Học Hài Hước (Tái Bản 2014) 1.0
Siêu Kinh Tế Học Hài Hước (Tái Bản 2013) 0.906666666667
Khuyến Học (Tái Bản 2014) 0.7
Tự Học Móc Len Sợi (Tái Bản 2014) 0.647058823529
```

Hình 4.4 Kết quả chạy với tựa sách "Kinh Tế Học Hài Hước (Tái Bản 2014)"

```
-- Siêu Kinh Tế Học Hài Hước (Tái Bản 2014) --
Siêu Kinh Tế Học Hài Hước (Tái Bản 2013) 0.975
Kinh Tế Học Hài Hước (Tái Bản 2014) 0.933333333333
Khuyến Học (Tái Bản 2014) 0.646153846154
Tự Học Móc Len Sợi (Tái Bản 2014) 0.547945205479
```

Hình 4.5 Kết quả chạy với tựa sách "Siêu Kinh Tế Học Hài Hước (Tái Bản 2014)"

```
-- Harry Potter Và Phòng Chứa Bí Mật --
Harry Potter Và Phòng Chứa Bí Mật - Tập 2 0.891891891892
Harry Potter Và Hòn Đá Phù Thủy - Tập 1 0.611111111111
Harry Potter Và Hoàng Tử Lai - Tập 6 0.63768115942
Harry Potter Và Chiếc Cốc Lửa - Tập 4 0.657142857143
```

Hình 4.6 Kết quả chạy với tựa sách "Harry Potter Và Phòng Chứa Bí Mật"

Với chuỗi được xử lý bằng cách bỏ các phần trong giấu “(” và “)”

```
-- Harry Potter Và Phòng Chứa Bí Mật --
Harry Potter Và Phòng Chứa Bí Mật - Tập 2 (Tái Bản 2013) 0.891891891892
Harry Potter Và Hòn Đá Phù Thủy - Tập 1 (Tái Bản 2013) 0.611111111111
Harry Potter Và Hoàng Tử Lai - Tập 6 (Tái Bản 2013) 0.63768115942
```

Hình 4.7 Kết quả chạy với tựa sách "Harry Potter Và Phòng Chứa Bí Mật"

4.3. So trùng dựa trên TF-IDF và Cosine Similarity

Ngoài sử dụng giải thuật Edit-Distance, luận văn còn sử dụng công thức TF-IDF và Cosine Similarity để tính độ tương đồng giữa 2 cuốn sách.

4.3.1. Xác định trọng số mỗi từ dựa trên thuật toán TF-IDF

TF-IDF là từ viết tắt của Term Frequency-Inverse Document Frequency. Trọng số TF-IDF là một trọng số được sử dụng nhiều trong truy vấn thông tin và khai thác văn bản. Trọng số này được sử dụng để ước tính độ quan trọng của một từ trong một văn bản nằm trong một tập hợp nhiều văn bản. Trọng số TF-IDF tỉ lệ thuận với tần số xuất hiện của từ đó trong văn bản và tỉ lệ nghịch với độ thường xuyên của từ đó trong tập hợp các văn bản.

Thông thường, trọng số TF-IDF được xác định dựa trên 2 yếu tố:

- Term Frequency (TF) thể hiện tần số xuất hiện của từ trong một văn bản.

Vì mỗi tài liệu khác nhau về số từ ngữ dẫn tới một từ ngữ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Do đó, term frequency thường được tính dựa trên số lần xuất hiện của từ trong văn bản chia cho số lượng từ văn bản đó có.

$$tf(t) = f(td) / |d|$$

Công thức 4.3 Tính Term Frequency

Trong đó:

- $f(t,d)$ là tần số xuất hiện của từ t trong văn bản d
- $|d|$ là tổng số từ có trong văn bản d

Hoặc một biến thể khác của công thức TF, đó là dựa trên tần số xuất hiện của một từ trong văn bản chia cho số tần số xuất hiện của từ xuất hiện nhiều nhất trong văn bản.

$$tf = f(t,d) / \max\{f(w,d) : w \in d\}$$

Công thức 4.4 Tính Term Frequency (2)

Trong đó:

- $f(t,d)$ là tần số xuất hiện của từ t trong văn bản d
- $\max\{f(w,d) : w \in d\}$ là tần số của từ xuất hiện nhiều nhất trong văn bản d

- Inverse Document Frequency (IDF) thể hiện một từ quan trọng thế nào. Khi chỉ tính toán TF, ta chưa giải quyết được yêu cầu xuất hiện ở phương án trước. Các từ ngữ phổ biến vẫn giữ giá trị TF cao. Vì vậy chúng ta cần giảm trọng số của chúng xuống và tăng trọng số của các từ khác lên.

$$idf(t,D) = \log(|D| / |\{d \in D : t \in d\}|)$$

Công thức 4.5 Tính Inverse Document Frequency

Trong đó:

- $idf(t,D)$ là tần số nghịch của một từ trong tập văn bản
- $|D|$ là số văn bản trong tập văn bản

- $|\{d \in D : t \in d\}|$ là số văn bản có chứa t

Cuối cùng ta có thể tính toán giá trị TF-IDF bằng cách cân bằng giữa giá trị TF và IDF

$$tf-idf = tf(t) \times idf(t, D)$$

Công thức 4.6 Tính TF-IDF

Ví dụ: Giả sử một văn bản có 100 từ trong đó từ android xuất hiện 5 lần. Tập văn bản chúng ta xét gồm 10 triệu văn bản, trong đó từ android xuất hiện trong 1000 văn bản.

Ta có:

- $tf(android) = 5 / 100 = 0.05$
- $idf(android, D) = \log(10,000,000 / 1,000) = 4$
- $tf-idf = 0.05 \times 4 = 0.2$

- **Ưu điểm:** phương pháp này đánh giá tốt hơn trọng số của từ ngữ, phân loại được những từ phổ biến và những từ đặc trưng cho nội dung văn bản.

- **Nhược điểm:** nếu quá ít văn bản, giá trị trọng số tính toán bằng phương pháp này không được cải thiện nhiều. Do phương pháp này xác định từ phổ biến dựa trên tần số xuất hiện của từ đó trong tập các văn bản.

4.3.2. Thuật toán Cosine Similarity

Ta có, Cosine giữa 2 vector chỉ mức độ tương đồng giữa 2 vector. Trong đó, công thức tính Cosine giữa 2 vector là

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} = \frac{\sum_{i=1}^n a_i \cdot b_i}{\sqrt{\sum_{i=1}^n (a_i)^2} \cdot \sqrt{\sum_{i=1}^n (b_i)^2}}$$

Công thức 4.7 Công thức tính Cosine Similarity giữa 2 Vector

Giá trị Cosine này nằm trong khoảng [-1, 1]. Giá trị Cosine càng cao, mức độ tương đồng càng cao.

4.3.3. Áp dụng thuật toán Cosine Similarity để so trùng

Thuật toán Cosine Similarity dùng để so sánh mức độ tương đồng giữa 2 vector có cùng n chiều. Do đó, để so sánh 2 cuốn sách, ta cần đưa 2 phần giới thiệu chung của cuốn sách đó về dạng vector và có cùng n chiều.

Úng với mỗi đoạn giới thiệu, ta xem mỗi từ ngữ là một chiều và giá trị TF-IDF của từ đó chính là giá trị của vector tại chiều đó.

Ví dụ 1: Ta có hồ sơ sau

{ [android, 0.5], [mobile, 0.2], [fast, 0.03], [thread, 0.1] }

Ta chuyển về dạng vector

Vector A (0.5, 0.2, 0.03, 0.1)

Trong đó các chiều lần lượt của vector này là android, mobile, fast, thread.

Để so trùng 2 hồ sơ cá nhân, ta cần thực hiện các bước sau:

- Tìm không gian chung cho hai phần giới thiệu, đó là không gian chứa tất cả các chiều (ở đây các chiều là các từ của phần giới thiệu) của cả hai cuốn sách. Nói cách khác, là những từ xuất hiện ở hai phần mô tả.
- Chuyển về dạng vector trong không gian đó. Nếu chiều của từ nào xuất hiện thì nó sẽ mang giá trị TF-IDF của chiều đó, nếu không có thì sẽ mang giá trị là 0.
- Tính toán giá trị cosine similarity giữa 2 vector.

Ví dụ 2: Ta có 2 tài liệu sau:

{ [android, 0.5], [mobile, 0.2], [fast, 0.03], [thread, 0.1] }

{ [android, 0.3], [mobile, 0.1], [fast, 0.1], [thread, 0.2] }

Từ 2 tài liệu trên, ta xác định được, không gian chung của 2 tài liệu là {android, mobile, fast, thread}.

Bước thứ hai, ta chuyển hai tài liệu trên về dạng vector trong không gian mà ta đã xác định.

Vector a (0.5, 0.2, 0.03, 0.1)

Vector b (0.3, 0.1, 0.1, 0.2)

Bước thứ 3, dựa vào hai vector ở bước thứ hai, ta tính toán giá trị cosine similarity của hai vector đó.

$$\begin{aligned} \cos\theta &= \frac{0.5 \times 0.3 + 0.2 \times 0.1 + 0.03 \times 0.1 + 0.1 \times 0.2}{\sqrt{0.5^2 + 0.2^2 + 0.03^2 + 0.1^2} \times \sqrt{0.3^2 + 0.1^2 + 0.1^2 + 0.2^2}} \\ &= \frac{0.193}{0.5485 \times 0.3873} = 0.9 \end{aligned}$$

Vậy mức độ tương đồng giữa 2 vector trên là 90%, ta kết luận 2 tài liệu này có độ tương đồng là 90%.

Ví dụ 3: Ta có 2 tài liệu sau

{ [android, 0.2], [car, 0.3], [fast, 0.1], [beauty, 0.05], [thread, 0.2] }

{ [android, 0.2], [iOS, 0.3], [window, 0.1], [phone, 0.1], [mobile, 0.3] }

Từ 2 tài liệu trên, ta xác định không gian chung cho vector là {android, car, fast, beauty, thread, iOS, window, phone, mobile}

Chuẩn hoá 2 tài liệu về dạng vector

Vector a (0.2, 0.3, 0.1, 0.05, 0.2, 0, 0, 0, 0)

Vector b (0.2, 0, 0, 0, 0, 0.3, 0.1, 0.1, 0.3)

Vector a có các chiều iOS, window, phone, mobile bằng 0, do trong hồ sơ không có các từ đó.

Tương tự vector b có các chiều car, fast, beauty, thread cũng bằng 0 do trong hồ sơ không có các từ này.

Cuối cùng, ta sử dụng thuật toán Cosine Similarity để xác định độ tương đồng giữa 2 tài liệu.

$$\cos\theta = \frac{0.04}{0.4272 \times 0.4899} = 0.19$$

Vậy mức độ tương đồng giữa 2 tài liệu này là 19%.

4.3.4. Áp dụng trong luận văn

Trong luận văn tiếng việt sẽ được tokenizer dựa trên bộ VNTokenizer của anh Lê Hồng Phương. Bộ tách từ tiếng Việt sử dụng kết hợp từ điển và ngram, trong đó mô hình ngram được huấn luyện sử dụng TreeBank tiếng Việt (70.000 câu đã được tách từ). Độ chính xác hơn 97%. Sau đây là kết quả tokenizer thông tin mô tả của các cuốn sách được thực hiện trên 7370 cuốn sách.

Bảng 4.4 Thống kê các từ xuất hiện nhiều nhất

STT	Từ	Số lần xuất hiện trong tài liệu	Số tài liệu “từ” được xuất hiện
01	và	26612	5342
02	của	25608	5149
03	là	18267	4734
04	những	20787	4671
05	trong	13670	4457
06	một	17011	4295
07	được	12579	4147
08	với	9283	3842
09	cho	9915	3818
10	có	10508	3798

11	sách	8796	3786
12	người	11398	3732
13	các	11352	3713
14	đã	9664	3516
15	về	7955	3405
16	để	7354	3276
17	không	8517	3128
18	này	5957	2995
19	sẽ	6195	2941
20	đến	5800	2940

Bảng 4.5 Danh sách đại diện các từ xuất hiện 1 lần trong văn bản

STT	Từ
01	hung cường
02	TIỀN LUƠNG
03	Nói Sao Để Khích Lệ Và Giúp Con Trưởng Thành
04	Quá Trình Hình Thành Một Nhà Tư Bản Mỹ
05	Tuna Nguyễn
06	Tháng Năm Của Kẹo
07	Thế Giới Phẳng
08	vệ binh
09	Bêncạnh
10	sử ký
11	Vụ việc
12	ravages
13	Việt Nam Doanh Nhân
14	chùa chiền
15	nao nao
16	cầu hòa
17	Chương Trình Luyện Thi
18	hoà vốn
19	cả nể
20	phục thiện

Nhận xét: trong các từ tiêu biểu được lựa chọn các cụm từ phần bao gồm các từ có hai âm như “cầu hòa” và “chùa chiền”, tên sách hoặc tên tác giả như “Việt Nam Doanh Nhân” và “Thế giới Phẳng” ngoài ra các từ viết sai chính tả như “Bêncạnh” (thiếu giấu cách ở giữa) cũng được tính vào danh sách này.

Một đoạn văn bảng mẫu được áp dụng:

“Giới thiệu sách. Harry Potter Và Phòng Chứa Bí Mật - Tập 2. Harry Potter và phòng chứa bí mật , không như những bộ truyện nhiều tập khác, vẫn tuyệt hay như người anh em trước... Hogwarts là sáng tạo của một thiên tài. - Times Literary Supplement Harry khổ sở mong ngóng cho kì nghỉ hè kinh khủng với gia đình Dursley kết thúc. Nhưng một con gia tinh bé nhỏ tội nghiệp đã cảnh báo cho Harry biết về mối nguy hiểm chết người đang chờ cậu ở” (Trích Harry Porter và Phòng Chứa Bí Mật)

Sau khi áp dụng tokenizer ta được đoạn văn (các chữ trong một từ được nối với nhau bằng dấu gạch chân để phân biệt):

“Giới_thiệu_sách_Harry_Potter_Và_Phòng_Chứa_Bí_Mật_Tập_2_Harry_Potter_và_phòng_chứa_bí_mật , không_như_những_bộ_truyện_nhiều_tập_khác , vẫn_tuyệt_hay_như_người_anh_em_trước ... Hogwarts_là_sáng_tạo_của_một_thiên_tài Times_Literary_Supplement_Harry_khổ_sở_mong_ngóng_cho_kì_nghi_hè_kinh_khung_với_gia_dình_Dursley_kết_thúc Nhưng_một_con_gia_tinh_bé_nhỏ_tội_nghiệp_dã_cảnh_báo_cho_Harry_biết_về_mối_nguy_hiểm_chết_người_dang_chờ_cậu_ở ... ”

Việc tính TF-IDF cho từng từ sẽ dựa vào công thức ở trên. Vì TF-IDF đại diện cho số lần một từ xuất hiện trong tất cả các văn bản (IDF) và số lần từ đó xuất hiện trong văn bản hiện tại. Nói cách khác, số từ đó xuất hiện trong tập văn bản càng nhiều thì vai trò của từ càng quan trọng. Nhưng từ xuất hiện càng nhiều trong các tập văn bản sẽ giảm độ quan trọng của từ xuống.

Bảng 4.6 Kết quả TF-IDF của một số từ trong văn bản

Từ	Số lần xuất hiện trong văn bản	TF (với tổng số từ là 71)	Số văn bản từ xuất hiện	IDF (với tổng số văn bản là 7370)	TF-IDF
Harry Potter	1	0.014	35	2.323	0.0327
Đang	1	0.014	1435	0.71	0.01
anh em	1	0.014	84	1.94	0.027
tội nghiệp	1	0.014	20	2.566	0.036
Như	2	0.028	2215	0.522	0.014
.....					

Việc tạo hai vector được thực hiện như mô tả ở trên, cuối cùng là tính độ tương quan giữa hai vector. Ta sẽ được kết quả như mong muốn.

PHẦN 5. THIẾT KẾ VÀ HIỆN THỰC

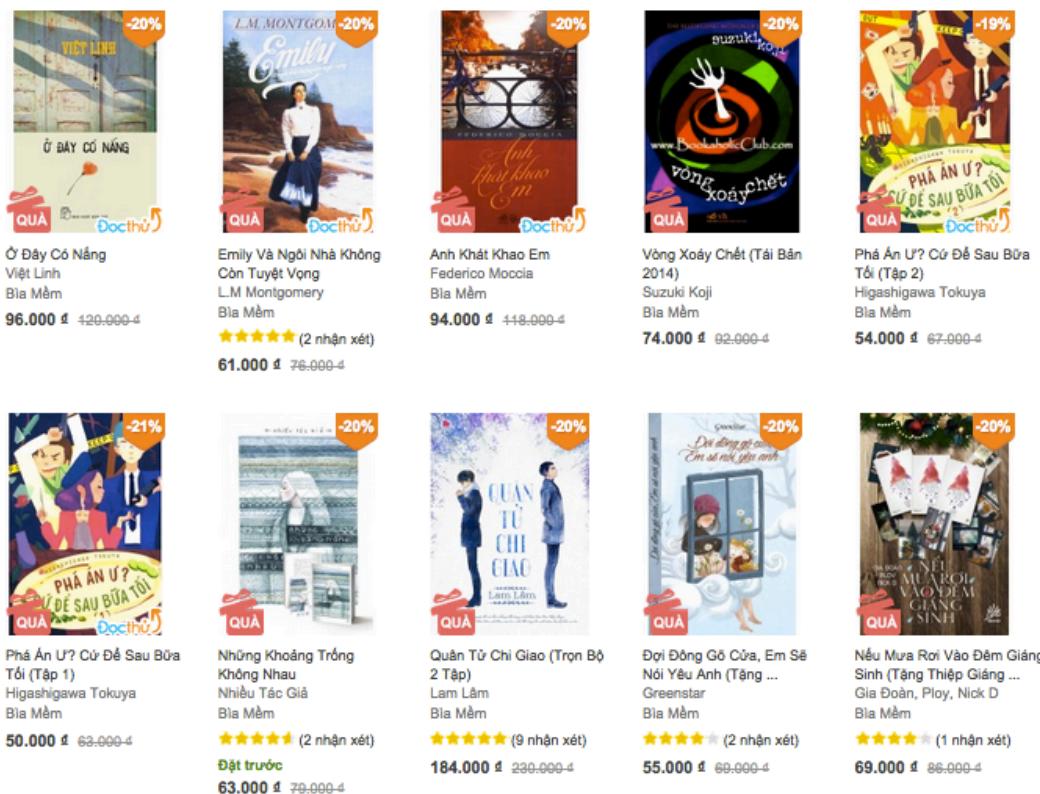
Nội dung chính của chương này là các công việc đã thực hiện trong luận văn xây dựng chương trình thu thập dữ liệu, xử lý dữ liệu trùng lặp, đánh chỉ mục dữ liệu và viết trang giao diện cho người dùng tương tác; và triển khai trên hệ thống các máy chạy thực tế.

5.1. Thiết kế bộ thu thập dữ liệu

5.1.1. Mô hình tổng quát thu thập dữ liệu

Trong luận văn tốt nghiệp này, sản phẩm hướng tới là đối tượng sách. Mô hình thu thập dữ liệu đã nêu ở Phần 2 cần được tổng quát để phù hợp và ứng dụng cho nhiều trang khác nhau. Mỗi trang bán sản phẩm sẽ ứng với mỗi loại khung nền khác nhau để trình duyệt Web có thể trình bày nội dung đẹp mắt nhưng nó vẫn chứa những đặc điểm mà chúng ta có thể lấy được thông tin và các bộ thông tin đều là chung nhất.

Ứng với mỗi trang web đều có hai loại trang: Trang tổng thể (general page) và Trang chi tiết (detail page).



1 2 3 4 5 Trang Sau >

Hình 5.1 Ví dụ về một Trang tổng thể

Ở Đây Có Nắng

Tác giả : Việt Linh
Bìa Mềm

Giá bìa: 120.000đ
Tại Tiki: 96.000đ (Đã có VAT)
Tiết kiệm: 24.000đ (20%)

Viết nhận xét để nhận tiki xu

Đổi trả trong vòng 15 ngày

Thông tin & Khuyến mãi

- Với mỗi 100.000đ trong đơn hàng, quý khách được tặng 300 Tiki Xu. > [Chi tiết](#)
- Tặng bookcard Tiki 2015 cho ĐH Sách từ 200k > [Chi tiết](#)
- Đăng ký dịch vụ BookCare để được bọc plastic đến 99% sách tại Tiki.vn
- Nhận hàng tại Hồ Chí Minh từ 2 - 3 ngày, không kể Thứ 7 & CN.

Số lượng: 1 Thêm Vào Giỏ Hàng Thêm Vào Yêu Thích

8+1 | 0 | [Email](#)

[Giới Thiệu Sách](#) | [Thông Tin Chi Tiết](#) | [Hỏi đáp](#) | [Khách Hàng Nhận Xét](#)

Ở Đây Có Nắng

Với *Ở đây có nắng*, đạo diễn điện ảnh Việt Linh muốn kể chuyện "theo cách thức montage điện ảnh", làm "một bộ phim truyền hình nhiều tập trênhững phân cảnh như những câu chuyện nhỏ, được gắn kết lại thành một câu chuyện dài, kể cho bạn nghe nhiều uẩn khúc cuộc đời. Bắt đầu bởisinh khi anh được biết mình là con nuôi trong một gia đình Việt kiều Pháp. Mở dần ra theo chuyến đi nhiều bí mật bấy lâu bị che giấu bên cạn:sống: Hội ngộ rồi chia ly tình yêu và hận thù, tài năng cùng người hâm mộ, rồi nỗi buồn, niềm vui, hạnh phúc...

Thông Tin Chi Tiết

Công ty phát hành	NXB Trẻ
Nhà xuất bản	NXB Trẻ

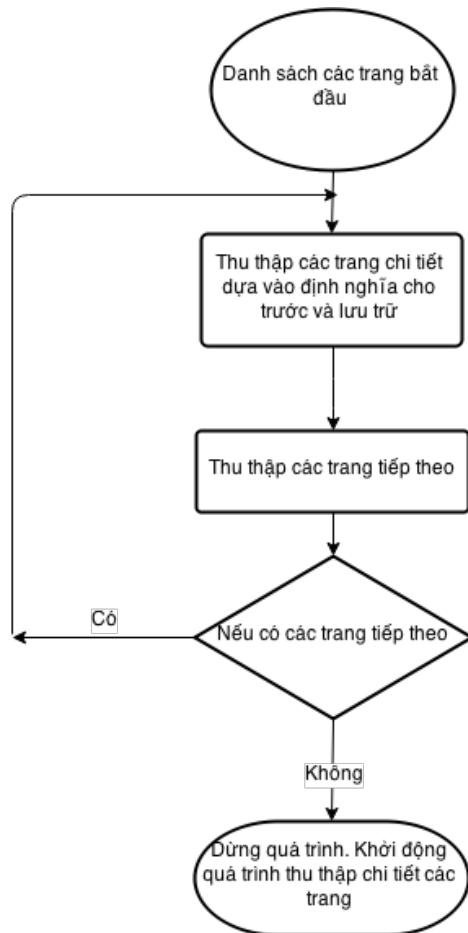
Hình 5.2 Ví dụ về một Trang chi tiết

Mục tiêu của quá trình thu thập dữ liệu là việc lấy các đường dẫn tới các trang chi tiết để lấy được nội dung văn bản cần thiết phục vụ cho các nhu cầu sau này.

Để đạt được mục tiêu này cần một số công đoạn thủ công định nghĩa một khung (template) của một trang bao gồm các chi tiết sau:

- Trang bắt đầu: Thường là một tập trang được cập nhật mới nhất các cuốn sách của cùng một địa chỉ.
- Vị trí của đường dẫn trang tiếp theo: Trong hình 5.1, trang bắt đầu có một số đường dẫn sang các trang 2, 3 và tiếp theo. Vị trí của đường dẫn sẽ được viết theo dạng Xpath Selector.
- Vị trí của đường dẫn trang chi tiết: Trong hình 5.1, trang bắt đầu sẽ bao gồm đường dẫn tới các trang chi tiết của các đối tượng sách. Vị trí đường dẫn tới trang chi tiết cũng được định nghĩa bằng Xpath Selector.

Quá trình sẽ bao gồm:



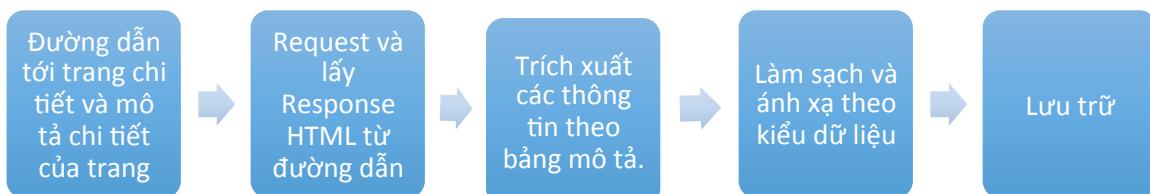
Hình 5.3 Flow chart quá trình thu thập các trang chi tiết

Quá trình tiếp theo sẽ dựa vào bảng định nghĩa Xpath Selector để lấy các thông tin:

Bảng 5.1 Định nghĩa Xpath của một trang web cụ thể (tiki.vn)

Tên	Xpath Selector
Title	'//*[@id="product_addtocart_form"]/div/div[3]/h1/text()'
Description	'//div[@class="product-description"]'
sale_off	'//p[@class="special-price2"]/span/span[@class="price"]/text()'
reg_price	'//span[@class="regular-price"]/span/text()'
old_price	'//p[@class="old-price"]/span[@class="price"]/text()'
link	Lấy trong request
author	'//div[contains(@class,"brand-box")]/ul/b/text()'
information	"//div[@id='chi-tiet']//tr/td"
parent_link	Lấy trong request

Sau quá trình thu thập được các đường dẫn tới trang chi tiết, hàng loạt request sẽ được gửi đi để lấy thông tin của một trang web cụ thể về trước đó cần thực hiện công việc lọc các đường dẫn bị trùng lắp. Có thể nhập hai quá trình lấy đường dẫn chi tiết ở phần trên và lấy thông tin cụ thể từ trang chạy song song với nhau để tiết kiệm thêm thời gian.



Hình 5.4 Quá trình thu thập dữ liệu

Úng với hai quá trình trên là hai đối tượng nền tảng Scrapy là Crawler và Spider. Để 2 đối tượng có thể hoạt động như ý muốn phải viết thêm một tầng abstract ở dưới để tái sử dụng các module được nhiều lần. Việc còn lại chỉ cần khai báo các đối tượng và định nghĩa.

5.1.2. Các hàm hỗ trợ phục vụ cho quá trình làm sạch và ánh xạ kiểu dữ liệu

Có ba kiểu dữ liệu chính là kiểu số, chuỗi và kiểu từ điển. Sau khi quá trình trích (extract) dữ liệu sẽ gặp phải vấn đề với dữ liệu chưa chuẩn, cần phải làm sạch và ánh xạ về các kiểu dữ liệu chuẩn để phục vụ vấn đề phân tích sau này. Phần này được sử dụng trong pipeline của nền tảng Scrapy.

a) Hàm ánh xạ số từ chuỗi

Các tính chất của đối tượng sách bao gồm: giá bìa, giá giảm, giá cũ sẽ được lưu ở dạng số.

```

Input: String str
Output: Int value of String
str_int := REPLACE([^\d], ' ', str)
if LENGTH(str_int) == 0:
    RETURN 0
else
    RETURN INT(str)

```

Hình 5.5 Hàm ánh xạ số từ chuỗi

b) *Hàm ánh xạ và làm sạch chuỗi*

Các tính chất như tác giả, mô tả cuốn sách, tên sách sẽ được lưu dưới dạng chuỗi. Các chuỗi sẽ được nối với nhau bằng dấu chấm “.” và xoá đi các thẻ tag HTML không cần thiết, chỉ giữ lại nội dung bên trong. Hàm Remove_tag là một hàm đệ quy. Không thay đổi và đổi thành viết thường các chữ trong quá trình này.

```

Input: Raw HTML string
Output: String content
list_of_string := SPLIT_HTML(raw_string)
result_string = ""
FOR each_string IN list_of_string:
    result_string = result_string + "." + REMOVE_TAG(each_string)
RETURN result_string

```

Hình 5.6 Hàm làm sạch chuỗi

c) *Hàm ánh xạ danh sách chuỗi thành từ điển*

Thuộc tính thông tin của sách sẽ được lưu dưới dạng từ điển. Các từ khoá và từ giá trị cũng được xử lý bằng hàm làm sạch chuỗi được nêu ở trên.

```

Input: List_string with odd position is Key and even position is Value
Output: Dictionary property
Key_list := []
Value_list := []
FOR each_string, index IN list_string:
    If EVENT(index):
        Key_list.append(LEAN_STRING(each_string))
    ELSE:
        Value_list.append(LEAN_STRING(each_string))
Result_dict = {}
FOR each_key, index in Key_list:
    Result_dict[each_key] = Value_list.at(index)
RETURN Result_dict

```

Hình 5.7 Hàm ánh xạ danh sách chuỗi thành từ điển

5.1.3. Khó khăn gặp phải khi thu thập dữ liệu

Như đã nói đến ở Phần 1, dữ liệu ngày nay là sự sống còn của doanh nghiệp, thêm vào đó, thực hiện số lượng request khá lớn vào cùng một trang trong một thời gian ngắn sẽ bị chặn, hoặc chuyển toàn bộ request từ những IP request liên tục sang một địa chỉ khác (honeybot). Đối với các trang nhỏ có thể dẫn không thể đáp ứng được nhu cầu (thường ít xảy ra). Bên cạnh việc thu thập còn phải tính đến các phương án nhằm tránh bị chặn hoặc gây phá hoại với trang web khác.

Các cách đã được luận văn thực hiện:

- Thay thế thông tin User-Agent của Request gửi đi
- Gửi các request qua Proxy miễn phí
- Dẫn cách các request một khoảng thời gian random từ 1 đến 3 giây.

Trong 3 cách đã thực hiện, các thứ 3 có hiệu quả nhất, số request lỗi gần như không có mặc dù vậy việc thực hiện các request rất lâu, thường tối 4-5 giờ chạy liên tục. Cách thứ 1 không mang lại hiệu quả nhiều lắm. Cách thứ 2 tốc độ request có tăng lên tuy nhiên có khá nhiều proxy không trả được kết quả và trả về trang lỗi 505 và 403. Nếu điều kiện cho phép việc thu thập dữ liệu có thể phân tán trên nhiều máy sẽ cải thiện tốc độ rất nhiều.

Số request không trả được kết quả (có lỗi 403 và 505) và số request trùng nhau cũng khá nhiều, sử dụng nền tảng Scrapy hỗ trợ được khó khăn này khá hiệu quả. Các request được thử lại nhiều nhất 3 lần. Việc thực hiện việc trích xuất dữ liệu phải cẩn thận kiểm tra trước khi chạy. Việc chạy code song song và bỏ qua các request lỗi sẽ giúp tiết kiệm nhiều thời gian hơn.

Ngoài ra, xử lý tiếng Việt trong quá trình ghi và đọc các file cần phải được encode theo chuẩn UTF-8, nếu không sẽ gây ra lỗi.

Bảng 5.2 Kết quả thống kê sau một lần chạy thu thập

Tên	Giá trị	Giải thích
downloader/request_bytes	3263948	Khối lượng downloader đã request về
downloader/response_status_count/200	6701	Số lượng request trả về 200
downloader/response_status_count/404	1	Số lượng request trả về 404
dupefilter/filtered	544	Số request bị trùng
response_received_count	6702	Số lượng response nhận được
start_time	2014-12-06 17:36:13.006201	Thời gian bắt đầu chạy
finish_time	2014-12-06 19:34:37.375929	Thời gian kết thúc chạy

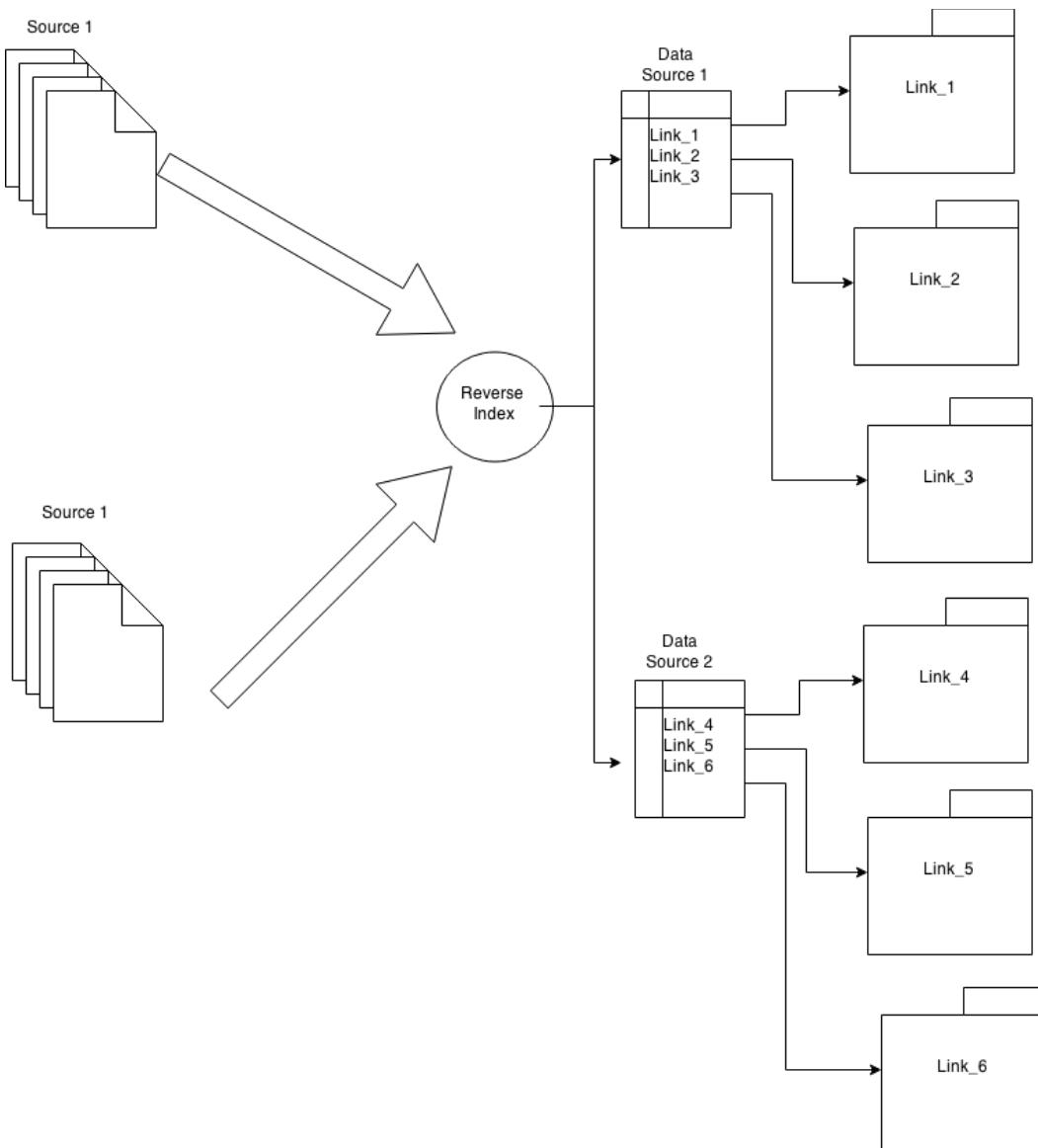
item_scraped_count	6415	Số lượng đối tượng sách thu được
--------------------	------	----------------------------------

5.2. Xây dựng bộ lưu trữ thông tin theo thời gian

Nhờ vào quá trình Reverse-Index của Lucene mà có thể sử dụng dễ dàng để lưu trữ các thông tin theo thời gian. Ý tưởng chính của việc dùng một công cụ index search để làm nơi lưu trữ dữ liệu theo thời gian dựa vào việc các trang qua chuỗi thời gian sẽ có một tính chất chung là đường dẫn tới trang URL.

Việc sử dụng các đường link làm định danh có một số rủi ro nếu trang có tình thêm một số tham số khiến cho đường dẫn khác nhau. Tuy nhiên trường hợp này rất dễ thấy và có thể khắc phục dễ dàng.

Bên cạnh lưu trữ tất cả dữ liệu và merge dữ liệu từ các trang khác nhau, chúng ta còn phải merge dữ liệu các trang từ trong cùng một trang. Xây dựng một tập chuẩn tất cả đối tượng của cùng một trang.



Hình 5.8 Mô hình Inverse Index

Mỗi đường link sẽ được tokenizer thành một token. Ứng với mỗi token sẽ trả về các tài liệu, mỗi tài liệu ứng với một lần thu thập thông tin về. Thông tin cuốn sách sẽ được thêm một trường là các đường link ứng với cùng cuốn sách ở các trang khác. Các câu truy vấn sẽ đơn giản hơn rất nhiều chỉ cần thêm một phần nhỏ là nhóm (group-facet) theo trường đường dẫn (link).

5.3. Triển khai trên hệ thống máy thật

Toàn bộ source code được đồng bộ lưu bằng Git Version. Hai repos chính được lưu tại địa chỉ:

- <https://bitbucket.org/thinhvx/decide-vn/>
- <https://bitbucket.org/thinhvx/decide-frontend>

Hiện tại hệ thống được triển khai trên hay máy chủ được thuê ở trang digitalocean.com. Và hoạt động dưới tên miền <http://demo.thinhvoxuan.me/>.

Bảng 5.3 Hệ thống máy chủ đang hoạt động

Tên	Hệ điều hành	Địa chỉ IP	Trạng thái	Bộ nhớ	Đĩa cứng	Giá
Phalcon1	Ubuntu 14.04	128.199.149.153	Đang chạy	512 MB	20GB	\$5/Tháng
Python1	Ubuntu 14.04	128.199.164.122	Đang chạy	1GB	30GB	\$10/Tháng

Máy Phalcon1 được sử dụng để chạy trang web giao diện người dùng và cơ sở dữ liệu MySQL.

Bảng 5.4 Thông tin các dịch vụ đang chạy trên máy Phalcon1

Tên	Trạng thái	Phiên bản
PHP		5.5.18
Apache2	Đang chạy	2.0
MySQL	Đang chạy	14.14
Phalcon	Đang chạy	1.3.1
Mandrill (dịch vụ email)	Đang chạy	

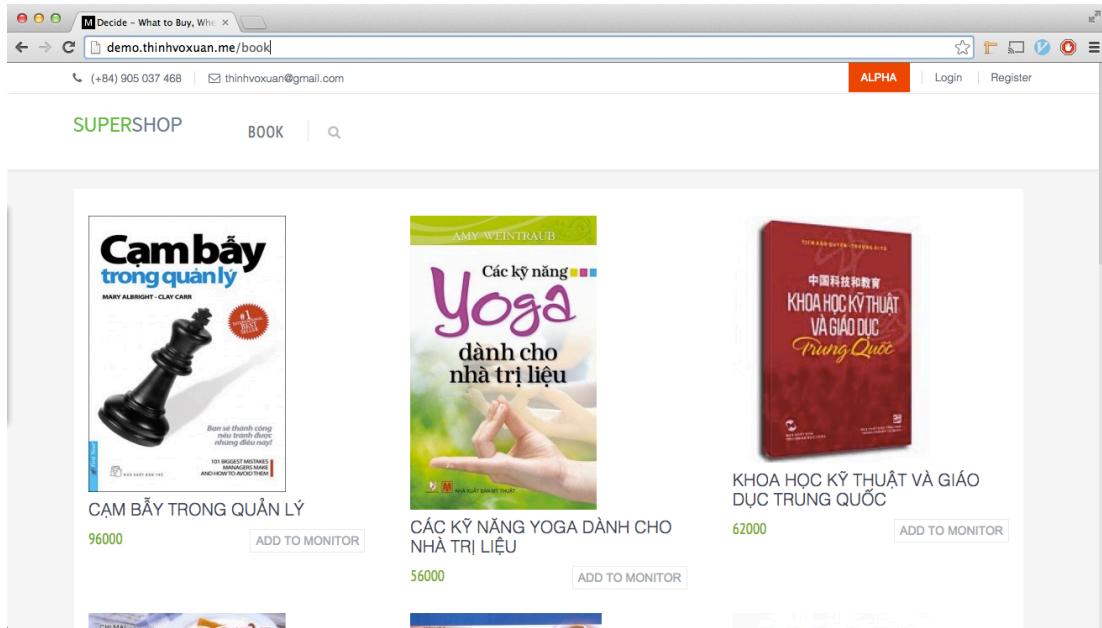
Máy Python1 được sử dụng để chạy các dịch vụ thu thập và index dữ liệu.

Bảng 5.5 Thông tin các dịch vụ đang chạy trên máy Python 1

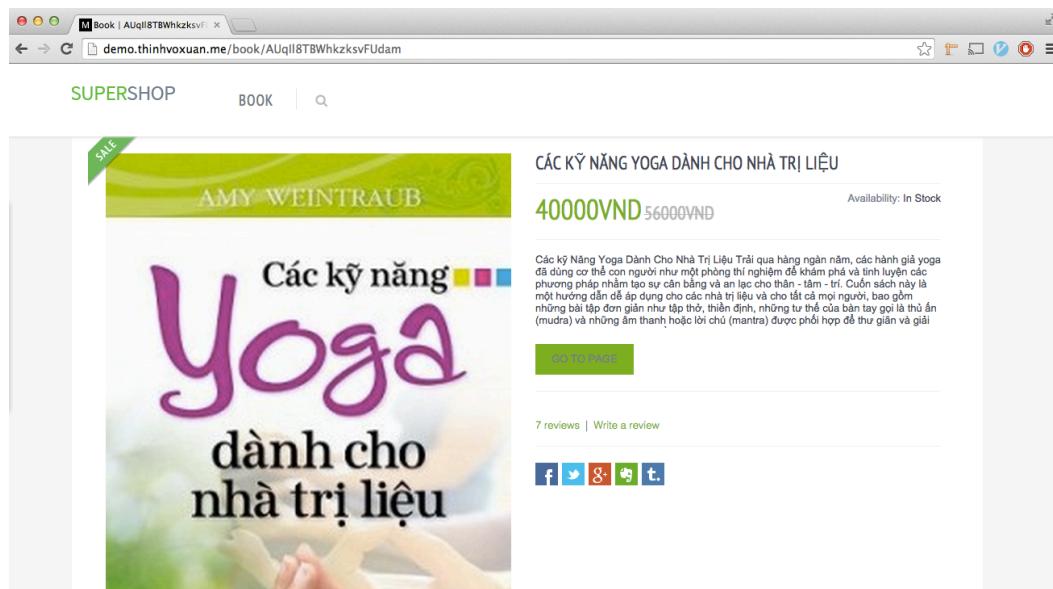
Tên	Trạng thái	Phiên bản
Python	-	2.7.6
Elasticsearch	Đang chạy	1.4.0
Lucene	Đang chạy	4.10.2
Scrapy	Đang chạy	0.24.4

5.4. Màn hình giao diện của trang người dùng

Trang người dùng được triển khai tại trang <http://demo.thinhvoxuan.me/> với môi trường được cài đặt đầy đủ.



Hình 5.9 Màn hình giao diện chính

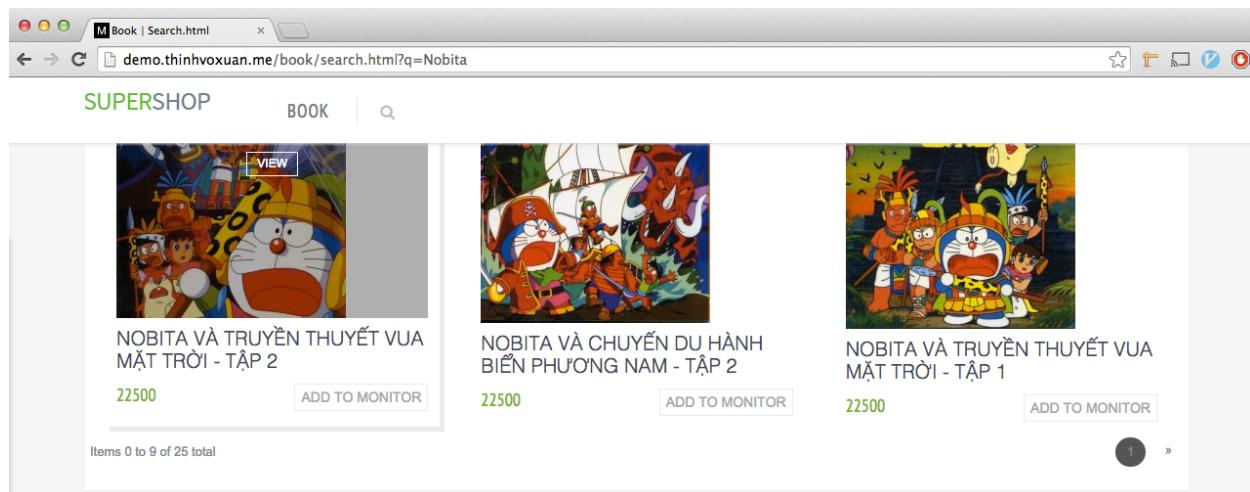


Hình 5.10 Màn hình chi tiết một cuốn sách

DESCRIPTION **INFORMATION** **REVIEWS (2)**

Các Kỹ Năng Yoga Dành Cho Nhà Trị Liệu Trải qua hàng ngàn năm, các hành giả yoga đã dùng cơ thể con người như một phòng thí nghiệm để khám phá và tinh luyện các phương pháp nhằm tạo sự cân bằng và an lạc cho thân - tâm - trí. Cuốn sách này là một hướng dẫn dễ áp dụng cho các nhà trị liệu và cho tất cả mọi người, bao gồm những bài tập đơn giản như tập thở, thiền định, những tư thế của bàn tay gọi là thủ ấn (mudra) và những âm thanh hoặc lời chú (mantra) được phối hợp để thư giãn và giải thoát. Ngày nay, có quá nhiều phương pháp tập được gọi là yoga, đây là những kỹ năng đơn giản mà sâu sắc, kết hợp những tinh hoa của yoga truyền thống và Tantric với kiến thức khoa học nhằm xử lý các vấn đề tâm trạng. Các bài tập này không bao gồm những tư thế gọi là asana, cũng không phải là những phát minh mới mẻ, nhưng chúng trở về nguồn cội của yoga, và cái đẹp thâm trầm của chúng hoàn toàn thích hợp với đời sống hiện đại.

Hình 5.11 Thông tin của một cuốn sách



Hình 5.12 Kết quả tìm kiếm theo từ khóa



Hình 5.13 Biểu đồ giá theo thời gian của 2 tựa cuốn sách

CREATE AN ACCOUNT

YOUR PERSONAL DETAILS

Username *

Email *

YOUR PASSWORD

Password * 

Repeat Password * 

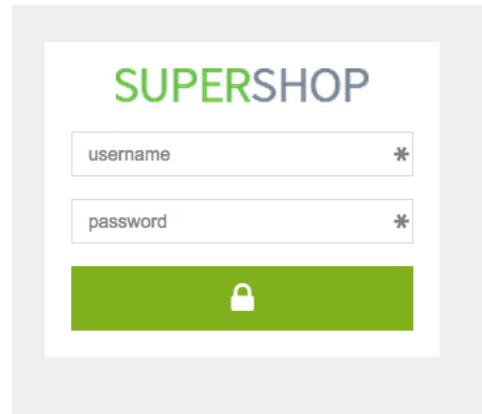
NEWSLETTER

Sign up for newsletter

CREATE AN ACCOUNT

CANCEL

Hình 5.14 Trang đăng kí



The login form is titled "SUPERSHOP" in green. It contains two input fields: "username" and "password", both marked with an asterisk (*) indicating they are required. Below the fields is a large green button featuring a white padlock icon.

Hình 5.15 Trang đăng nhập

PHẦN 6. THÍ NGHIỆM, ĐÁNH GIÁ VÀ KẾT LUẬN

Nội dung chính của phần này trình bày các kết quả đã đạt được, kiểm tra thí nghiệm chủ quan và kết luận của luận văn. Đây là phần cuối cùng tổng kết lại các phần đã và chưa làm được cũng như đánh giá kết quả hiện thực.

6.1. Thí nghiệm và đánh giá kết quả

6.1.1. Thí nghiệm hai giải thuật Minumin Edit Distance và TF-IDF:

a) Giải thuật Minimun Edit Distance

Đây là giải thuật chính được sử dụng trong phần backend của chương trình. Giải thuật Minumun Edit Distance với hàm custom_similar ở trên được sử dụng để so hào hết tên các cuốn sách với nhau. Áp dụng giải thuật với 8220 cuốn sách ở trang tiki.vn và 1049 cuốn sách ở trang nobita.vn, kết quả đạt được là 501 cuốn sách được nối với nhau từ 2 nguồn với mức ngưỡng là 0.9

Bảng 6.1 Mức ngưỡng ứng với số đối tượng trùng nhau của giải thuật

Mức ngưỡng	Số đối tượng trùng nhau
0.95	467
0.9	501
0.85	534
0.8	565
0.75	624
0.7	711

Với mức ngưỡng càng giảm số lượng đối tượng trùng nhau theo tựa ngày càng lớn và độ sai sót cũng càng lớn. Hiện nay giải thuật đang chạy với mức ngưỡng là 0.9.

b) Giải thuật TF-IDF

Giải thuật này dùng để kiểm tra độ tương đồng của hai đối tượng sách. Tuy nhiên lời tựa của cuốn sách tùy thuộc vào biên tập viên của trang nên cùng một cuốn sách có thể có nhiều lời tựa khác nhau.

VD về lời tựa của 2 cuốn sách cho chung tựa “Harry Potter Và Phòng Chứa Bí Mật - Tập 2”

Lời tựa 1:

“Giới thiệu sách Harry Potter Và Phòng Chứa Bí Mật - Tập 2 Harry Potter và phòng chứa bí mật , không như những bộ truyện nhiều tập khác, vẫn tuyệt hay như người anh em trước... Hogwarts là sáng tạo của một thiên tài. - Times Literary Supplement Harry khổ sở mong ngóng cho kì nghỉ hè kinh khủng với gia đình Dursley kết thúc. Nhưng một con gia tinh bé nhỏ tội nghiệp đã cảnh báo cho Harry biết về mối nguy hiểm chết người đang chờ cậu ở trường

Hogwarts. Trở lại trường học, Harry nghe một tin đồn đang lan truyền về phòng chứa bí mật, nơi cất giữ những bí ẩn đáng sợ dành cho giới phù thủy có nguồn gốc Muggle. Có kẻ nào đó đang phù phép làm tê liệt mọi người, khiến họ gần như đã chết, và một lời cảnh báo kinh hoàng được tìm thấy trên bức tường. Mỗi nghi ngờ hàng đầu – và luôn luôn sai lầm – là Harry. Nhưng một việc còn đen tối hơn thế đã được hé mở. Xem thêm nội dung”

(<http://nobita.vn/454/harry-potter-va-phong-chua-bi-mat-tap-2.html>)

và

“Loạt truyện Harry Potter đã đoạt giải Cuốn sách thiếu nhi hay nhất trong năm của Anh Quốc. Ba cuốn Harry Potter đầu đã in ra 35 triệu bản dịch ra 39 ngôn ngữ và xuất hiện trên 200 quốc gia (trong đó có cả Việt Nam). Bộ sách sẽ gồm 7 cuốn, mỗi cuốn cho mỗi năm học ở trường Hogwarts từ lúc Harry mười một tuổi cho đến mười bảy tuổi.”

(<http://tiki.vn/harry-potter-va-phong-chua-bi-mat-tap-2-tai-ban-2013-p59725.html>)

Giá trị trả về của TF-IDF chỉ thuộc đoạn [0, 0.3] hoặc độ tương tự thấp hơn 30% nên hiện tại chưa được áp dụng tiếp.

c) Nhận xét

Sử dụng phương pháp Minimun Edit Distance và custom_similar cho tựa sách có hiệu quả tốt hơn phương pháp sử dụng TF-IDF. Việc đặt một mức ngưỡng hợp lý cho giải thuật Minimun Edit Distance cần có nhiều thực nghiệm và nghiên cứu hơn hoặc chọn một giải thuật khác kiểm tra chéo sẽ tăng độ chính xác khi giảm mức ngưỡng của Minimun Edit Distance xuống thấp.

6.1.2. Thí nghiệm sản phẩm

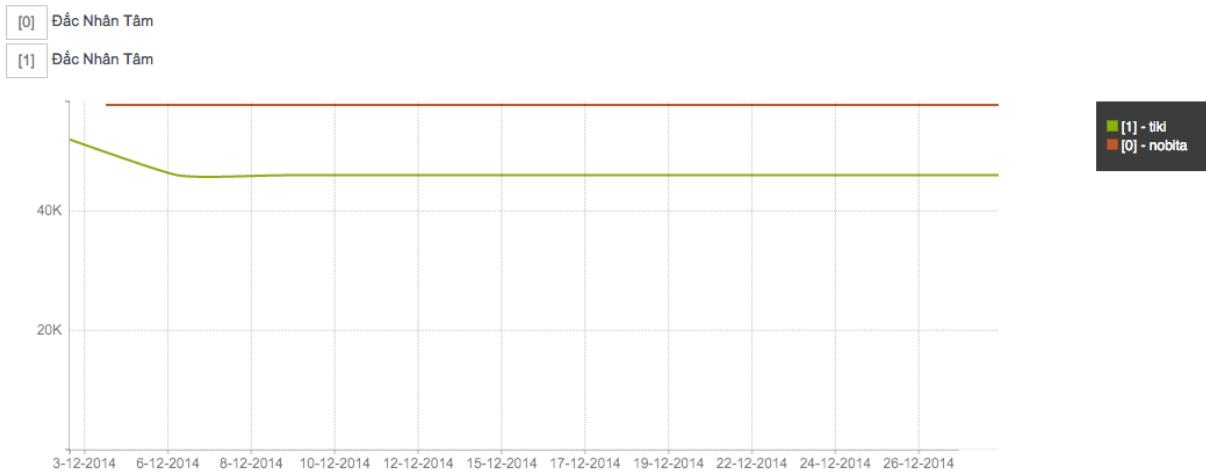
Sau đây là tổng hợp kết quả thực nghiệm tìm kiếm áp dụng cho một số lượng nhỏ người dùng sau khi sử dụng trang được thực hiện trên trang demo chính thức.

a) Trường hợp 1 với từ khoá “Đắc nhân tâm”

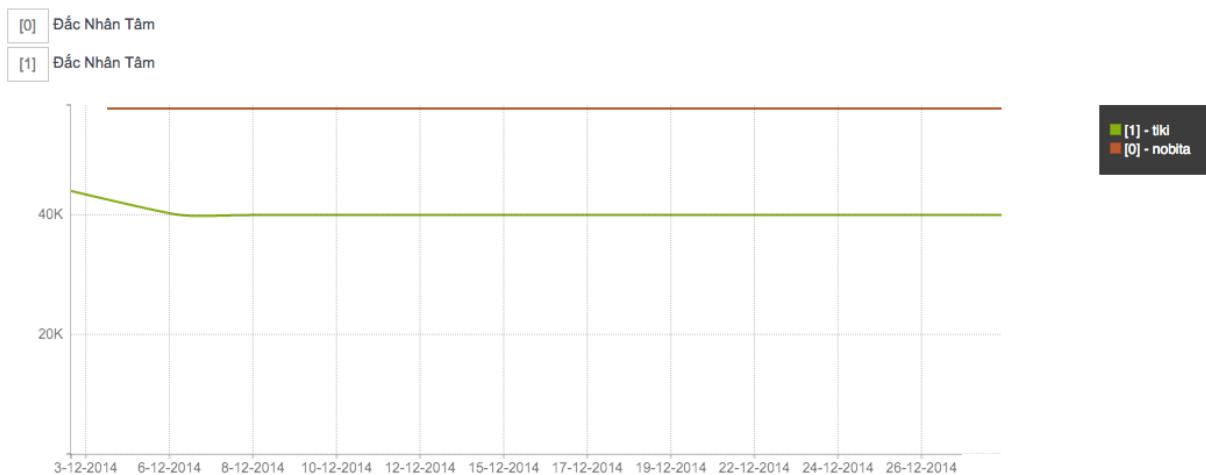
Số kết quả trả về là 6 bao gồm các cuốn sách với giá

- ĐẮC NHÂN TÂM – 46.000 [1]
- ĐẮC NHÂN TÂM – 40.000 [2]
- ĐẮC NHÂN TÂM (TÁI BẢN) – 50.000
- ĐẮC NHÂN TÂM TRONG THỜI ĐẠI SỐ – 232.000
- 25 THUẬT ĐẮC NHÂN TÂM (TÁI BẢN 2014) – 60000
- ĐẮC NHÂN TÂM THEO PHONG CÁCH PHẬT GIÁO – 40000

Bảng đồ giá của 2 cuốn đầu tiên là



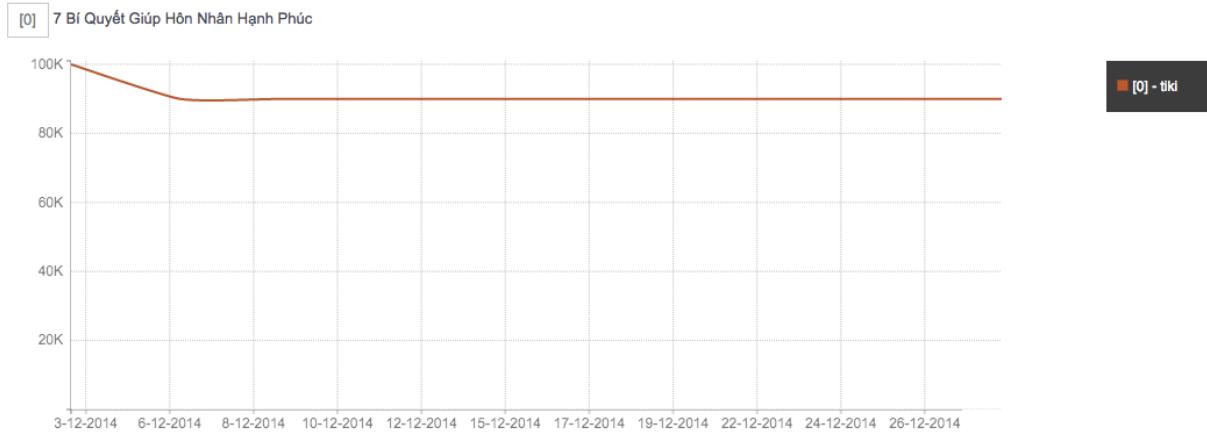
Hình 6.1 Bản đồ giá của cuốn “Đắc Nhân Tâm” [1]



Hình 6.2 Bản đồ giá của cuốn “Đắc Nhân Tâm” [2]

Hai hình vẽ này thể hiện sách trên trang tiki.vn đang có đợt giảm giá trong tháng 12 này, nên trang nobita.vn luôn có giá cao hơn.

Kiểm tra ngẫu nhiên trên vài cuốn sách khác, ta cũng nhận thấy thời gian bắt đầu giảm giá vào khoảng ngày 6 tháng 12 năm 2014.



Hình 6.3 Bán đồ giá của cuốn sách “7 Bí Quyết Giúp hôn Nhân Hạnh Phúc”

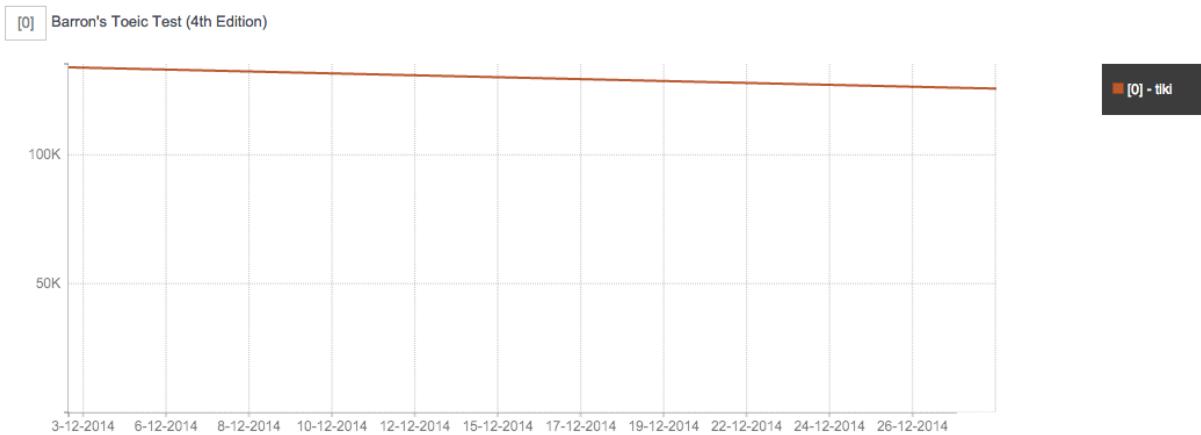
b) *Trường hợp 2 với từ khoá “toeic”*

Số lượng kết quả trả về là 80 cuốn sách.

Một số sách đang được giảm giá trong mùa Giáng Sinh và Tết Dương Lịch



Hình 6.4 Biểu đồ giá sách Toeic Analyst Second Edition



Hình 6.5 Biểu đồ giá sách Barron's Toeic Test (4th Edition)

Phát hiện này mang lại cho người dùng cảm giác rất mới và thích thú khi có thêm thông tin để lựa chọn một món quà cho người thân. Việc tìm thấy cuốn sách này hoàn toàn là tình cờ.

c) *Dánh giá của người dùng*

Sản phẩm đạt được nhiều kỳ vọng ban đầu của người dùng, giao diện thân thiện, trực quan hấp dẫn được người dùng. Người dùng muốn tăng số lượng nguồn đầu vào của trang lên. Đây là hướng sẽ phát triển trong tương lai bao gồm tăng số lượng datasource (nguồn trang đầu vào) và áp dụng mô hình cho các sản phẩm khác không chỉ là sách.

6.2. Kết luận

Em hi vọng sản phẩm của em sẽ trở thành một trào lưu mới trong thị trường thương mại điện tử. Nhóm sản phẩm cung cấp thêm các dịch vụ và quy trình trong thương mại điện tử sẽ tăng lên, không còn phụ thuộc vào nhà sản xuất và được tương tác với cộng đồng nhiều hơn.

a) *Những điều đã làm được*

Trong học kỳ này, thời gian dành cho việc hình thành ý tưởng và chuẩn bị các nền tảng ban đầu chiếm khá nhiều thời gian. Người sử dụng có nhu cầu về một trang trung gian hỗ trợ tìm kiếm sản phẩm. Em đã xây dựng hai chương trình backend và frontend. Mỗi phần đều có sự quan trọng như nhau. Backend đảm bảo sự ổn định và giá trị cốt lõi của sản phẩm còn frontend là phần tương tác với người dùng chính. Thiết kế, phát triển từng bộ phận của sản phẩm cũng như đưa sản phẩm chạy thực tế trước một số lượng nhỏ các bạn sinh viên. Bảo quản source code bằng Subversion Git và chạy môi trường trên Vagrant khiến việc khai chạy trên môi trường thật một cách dễ dàng hơn. Trong quá trình hiện thực, có viết chương trình bao đóng sử dụng thư viện xử lý tiếng Việt Java chạy trên nền Python, chương trình này sẽ được công bố lên mạng sau khi viết các testcase và document cần thiết.

Bảng 6.2 Báo cáo kết quả thực hiện

	Các vấn đề đã nghiên cứu	Kết quả
Backend	Scrapy Framework	Hiểu được cấu trúc và các thực hiện chương trình. Viết các lớp Abstract nhằm tổng quát hoá và tái sử dụng lại các lớp.
	ElasticSearch (ES) Tool	Tìm hiểu, sử dụng và triển khai chạy thực tế. Viết các kết nối từ backend vào ES để đưa dữ liệu vào Index cũng như viết kết nối từ Frontend lấy và hiển thị dữ liệu cần thiết.
	Giải thuật MED	Hiện thực và đưa thêm một hàm quy đổi về phần trăm dựa trên độ dài hai chuỗi và kết quả của hàm MED. Áp dụng vào so sánh các cuốn sách thực tế.
	Giải thuật TF-IDF	Hiện thực và áp dụng tuy nhiên giải thuật chạy không hiệu quả
Frontend	Trang chủ - PhanconEye CMS	Hoàn toàn làm chủ được CMS để phát triển thêm chức năng hiển thị được tất cả các cuốn sách đang có trong bộ lưu trữ, tìm kiếm, đưa ra thông tin chi tiết.
	Theme nền	Áp dụng được theme Metronic vào toàn trang giao diện
	Tìm kiếm	Người dùng có thể tìm kiếm bằng tiếng Việt có dấu và không dấu tên cuốn sách và tác giả cuốn sách.
	Bảng đồ giá	Thể hiện bảng đồ giá được cập nhật từ đầu tháng 12 đến nay.

b) Những điều chưa thực hiện được

Số lượng nguồn thu thập thông tin về còn hạn chế. Các giải thuật đưa ra cần có nhiều thí nghiệm chứng minh độ chính xác hơn. Ngoài ra, báo cáo luận văn còn nhiều điểm thiếu sót.

PHẦN 8. Phụ lục

8.1. Tài liệu tham khảo

- [1] Electronic Commerce - Sách - G. P. Schneider (2011), , 9th Edition, Course Technology
- [2] Natural Language Processing – Khoa học trực tuyến - Dan Jurafsky, Christopher Manning
[<https://class.coursera.org/nlp/lecture>]
- [3] Scrapy documentation [<http://scrapy.readthedocs.org/>]
- [4] Elasticsearch – The definitive guide - sách [<http://www.elasticsearch.org/blog/elasticsearch-definitive-guide/>]
- [5] Lucene in Action – Sách – Second Edition
- [6] Python documentation [<https://docs.python.org/2/>]
- [7] Fuzzy String Matching in Python [<http://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/>]
- [8] Đề tài KC01.01/06-10 "Nghiên cứu phát triển một số sản phẩm thiết yếu về xử lý tiếng nói và văn bản tiếng Việt" (VLSP) [<http://vlsp.vietlp.org:8080/demo/?page=resources>]
- [9] Git – Software ([http://en.wikipedia.org/wiki/Git_\(software\)](http://en.wikipedia.org/wiki/Git_(software)))

