

KHÓA LUẬN TỐT NGHIỆP

TÊN ĐỀ TÀI BÁO THỰC TẬP TỐT NGHIỆP:

NGHIÊN CỨU TÌM HIỂU CÔNG NGHỆ THU THẬP DỮ LIỆUVÀ ỨNG DỤNG PHÂN TÍCH QUAN ĐIỂM

Ngành đào tạo: Công nghệ thông tin

**Nhóm 7:**

* **Vũ Xuân Mừng**
* **Trần Anh Quân**
* **Nguyễn Văn Hiếu**
* **Ma Văn Nghĩa**
* **Nguyễn Mạnh Hiếu**

**G.V hướng dẫn:**

* **G.V Trần Hồng Việt**
* **G.V Đào Thuỵ Ánh**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC KINH TẾ**

**KỸ THUẬT CÔNG NGHIỆP HÀ NỘI**

**MỤC LỤC**

[DANH MỤC VIẾT TẮT 1](#_Toc62756272)

[MỞ ĐẦU 2](#_Toc62756273)

[CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU 4](#_Toc62756274)

[1.1 GIỚI THIỆU 4](#_Toc62756275)

[1.2 BÀI TOÁN THU THẬP DỮ LIỆU TỪ NGUỒN DỮ LIỆU TRỰC TUYẾN 4](#_Toc62756276)

[1.3 BÀI TOÁN PHÂN TÍCH QUAN ĐIỂM 6](#_Toc62756277)

[1.4 HƯỚNG TIẾP CẬN BÀI TOÁN 6](#_Toc62756278)

[1.5 MỤC TIÊU VÀ ĐỐI TƯỢNG NGHIÊN CỨU 9](#_Toc62756279)

[1.6 KẾT LUẬN 10](#_Toc62756280)

[CHƯƠNG 2: CÔNG NGHỆ THU THẬP DỮ LIỆU TỪ NGUỒN DỮ LIỆU TRỰC TUYẾN 11](#_Toc62756281)

[2.1 CÁC KHÁI NIỆM CƠ BẢN 11](#_Toc62756282)

[2.1.1 World Wide Web 11](#_Toc62756283)

[2.1.2 Thu thập dữ liệu 12](#_Toc62756284)

[2.1.3 Trích chọn thông tin 12](#_Toc62756285)

[2.1.4 HTML 12](#_Toc62756286)

[2.1.5 DOM 13](#_Toc62756287)

[2.1.6 JSON 14](#_Toc62756288)

[2.1.7 Web Crawler 15](#_Toc62756289)

[2.1.8 Headless Browser 18](#_Toc62756290)

[2.1.9 Puppeteer 19](#_Toc62756291)

[2.1.10 Elasticsearch 21](#_Toc62756292)

[2.1.10.1 Document 21](#_Toc62756293)

[2.1.10.2 Index 21](#_Toc62756294)

[2.1.10.3 Shard 21](#_Toc62756295)

[2.1.10.4 Node 22](#_Toc62756296)

[2.1.10.5 Cluster 22](#_Toc62756297)

[2.2 GIẢI QUYẾT BÀI TOÁN THU THẬP DỮ LIỆU TỪ NGUỒN DỮ LIỆU TRỰC TUYẾN. 22](#_Toc62756298)

[2.3 KẾT LUẬN 24](#_Toc62756299)

[CHƯƠNG 3: PHÂN TÍCH QUAN ĐIỂM NGƯỜI DÙNG 25](#_Toc62756300)

[3.1 CÁC KHÁI NIỆM CƠ BẢN 25](#_Toc62756301)

[3.1.1 Hàm Sigmoid 25](#_Toc62756302)

[3.1.2 Hàm Softmax 25](#_Toc62756303)

[3.1.3 Mạng Nơ-ron nhân tạo 26](#_Toc62756304)

[3.1.3.1 Định nghĩa 26](#_Toc62756305)

[3.1.3.2 Cấu trúc của mạng nơ-ron nhân tạo 26](#_Toc62756306)

[3.1.4 Mạng nơ-ron hồi quy 27](#_Toc62756307)

[3.1.5 Mạng bộ nhớ dài ngắn 28](#_Toc62756308)

[3.1.5.1 Khái niệm cơ bản 28](#_Toc62756309)

[3.1.5.2 Bên trong LSTM 31](#_Toc62756310)

[3.1.6 Tensorflow 33](#_Toc62756311)

[3.1.7 Word Embedding 35](#_Toc62756312)

[3.2 GIẢI QUYẾT BÀI TOÁN PHÂN TÍCH QUAN ĐIỂM NGƯỜI DÙNG 35](#_Toc62756313)

[3.2.1 Tiền xử lý dữ liệu 35](#_Toc62756314)

[3.2.2 Mô hình phân tích quan điểm người dùng 36](#_Toc62756315)

[3.2.3 Ứng dụng web cho mô hình phân tích quan điểm người dùng 37](#_Toc62756316)

[3.3 KẾT LUẬN 38](#_Toc62756317)

[CHƯƠNG 4: TRIỂN KHAI XÂY DỰNG HỆ THỐNG 39](#_Toc62756318)

[4.1 XÂY DỰNG HỆ THỐNG THU THẬP TRÍCH RÚT DỮ LIỆU. 40](#_Toc62756319)

DANH MỤC VIẾT TẮT

|  |  |
| --- | --- |
| **Từ Viết Tắt** | **Viết Đầy Đủ** |
| WWW | **W**orld **W**ide **W**eb |
| HTML | **H**yper**T**ext **M**arkup **L**anguage |
| XHTML | e**X**tensible **H**yper**T**ext **M**arkup **L**anguage |
| SVM | **S**upport **V**ector **M**achine |
| FTP | **F**ile **T**ransfer **P**rotocol |
| XML | e**X**tensible **M**arkup **L**anguage |
| PDF | **P**ortable **D**ocument **F**ormat |
| URL | **U**niform **R**esource **L**ocator |
| DOM | **D**ocument **O**bject **M**odel |
| API | **A**pplication **P**rogramming **I**nterface |
| AJAX | **A**synchronous **J**ava**S**criptand **X**ML |
| IDE | **I**ntegrated **D**evelopment **E**nvironment |
| UI | **U**ser **I**nterface |
| JSON | **J**ava**S**cript **O**bject **N**otation |
| ANN | **A**rtificial **N**eural **N**etwork |
| RNN | **R**ecurrent **N**eural **N**etwork |
| LSTM | **L**ong **S**hort-**T**erm **M**emory |
| CPU | **C**entral **P**rocessing **U**nit |
| GPU | **G**raphics **P**rocessing **U**nit |
| VLSP | **V**ietnamese **L**anguage and **S**peech **P**rocessing |
| JS | **J**ava**S**cript |

MỞ ĐẦU

Với sự phát triển không ngừng của mạng Internet, cùng với đó là khối dữ liệu dữ liệu khổng lồ trên internet ngày một gia tăng. Điều này một phần gây khó khăn với nhu cầu thu thập thông tin trên mạng internet. Để tìm kiếm và trích rút một vài thông tin trên mạng internet thì rất đơn giản, nhưng đối với công việc thu thập, trích rút số lượng lớn thông tin thì vô cùng khó khăn. Chẳng hạn, một trang web so sánh sản phẩm trên các trang thương mại điện tử cần thu thập thông tin về mặt hàng, giá cả, tính năng, khuyến mãi… của tất cả sản phẩm. Công việc này nếu làm thủ công sẽ tốt rất nhiều thời gian và công sức. Trong thực tế thu thập dữ liệu được ứng dụng với nhiều bài toán khác nhau, lĩnh vực khác nhau, có thể kể đến như: thu thập dữ liệu sản phẩm để khảo sát thị trường, tổng hợp thông tin từ các website khác, thu thập phản hồi người dùng về sản phẩm, dịch vụ… Trong đó ứng dụng thu thập phản hồi người dùng về sản phẩm, dịch vụ có số lượng lớn doanh nghiệp mong muốn phát triển. Những thông tin này không chỉ giúp doanh nghiệp đánh giá, xếp hạng sản phẩm mà còn là thông tin hữu ích nhằm cải tiến, nâng cấp chất lượng sản phẩm, dịch vụ tốt hơn.

Trong bài luận văn này, tôi xin tập trung nghiên cứu công nghệ thu thập dữ liệu và xây dựng ứng dụng phân tích quan điểm. Bố cục bài luận văn được chia thành:

**Chương 1: Tổng quan vấn đề nghiên cứu:**

Giới thiệu tổng quan về bài toán thu thập dữ liệu từ nguồn dữ liệu trực tuyến và ứng dụng phân tích quan điểm, hướng tiếp cận bài toán.

**Chương 2: Công nghệ thu thập dữ liệu từ nguồn dữ liệu trực tuyến**

Tìm hiểu, nghiên cứu về công nghệ thu thập dữ liệu từ nguồn dữ liệu trực tuyến và hướng đi giải quết bài toán.

**Chương 3: Phân tích quan điểm người dùng**

Tìm hiểu, nghiên cứu về xử lý dữ liệu, xây dựng mô hình và ứng dụng cho ứng dụng phân tích quan điểm người dùng.

**Chương 4: Triển khai xây dựng hệ thống và thực nghiệm**

Chi tiết các bước xây dựng hệ thống và đánh giá kết quả đạt được.

**Kết luận**

Kết luận cho bài luận văn và rút ra kinh nghiệm đã học hỏi được thông qua việc nghiên cứu đề tài và đưa ra hướng phát triển trong tương lai.

CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

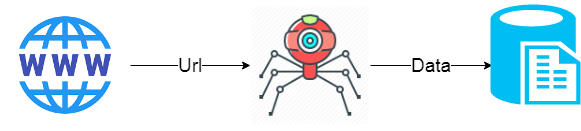
# 1.1 GIỚI THIỆU

Hiện nay với sự phát triển không ngừng của mạng internet, lượng dữ liệu thông tin qua đó cũng trở nên khổng lồ và không ngừng gia tăng. Mặt khác nhu cầu tìm kiếm sử dụng thông tin của con người ngày càng lớn. Chúng ta có thể tìm kiếm thu thập một vài thông tin rất dễ dàng qua các công cụ tìm kiếm như Google, Bring… Nhưng đối với nhu cầu thu thập một lượng lớn dữ liệu thì việc này trở nên vô cùng khó khăn và tốn kém về mặt nhân lực và thời gian. Hơn nữa thu thập dữ liệu từ nguồn dữ liệu trực tuyến còn được ứng dụng rất nhiều trong thực tế với các lĩnh vực khác nhau. Chính vì thế chúng ta cần có hệ thống thu thập dữ liệu tự động giúp tối ưu hóa chi phí và thời gian cho việc thu thập dữ liệu từ các nguồn dữ liệu trực tuyến.

# 1.2 BÀI TOÁN THU THẬP DỮ LIỆU TỪ NGUỒN DỮ LIỆU TRỰC TUYẾN

Trình thu thập dữ liệu Web [10] là một chương trình máy tính duyệt World Wide Web theo phương pháp tự động, có phương pháp hoặc theo một trật tự. Thu thập dữ liệu trên web là một phương pháp quan trọng để thu thập dữ liệu và theo kịp nguồn dữ liệu trên Internet đang ngày càng mở rộng nhanh chóng. Một số lượng lớn các trang web được thêm vào mỗi ngày và thông tin cập nhật liên tục.

World Wide Web cung cấp cho chúng ta một lượng lớn thông tin hữu ích có sẵn dưới dạng siêu văn bản. Nhóm siêu văn bản này đang thay đổi linh hoạt và không có cấu trúc về mặt ngữ nghĩa, khiến chúng ta gặp khó khăn trong việc tìm kiếm thông tin có liên quan và có giá trị. Do đó, trình thu thập dữ liệu web để tự động phát hiện thông tin có giá trị từ Web hoặc Web Mining rất quan trọng đối với chúng ta ngày nay. Trong thực tế, trình thu thập dữ liệu web này là một chương trình, tự động duyệt web bằng cách tải xuống các tài liệu và theo các liên kết từ trang này sang trang khác. Chúng chủ yếu được sử dụng bởi công cụ tìm kiếm để thu thập dữ liệu để lập chỉ mục. Các ứng dụng khác có thể bao gồm xác nhận trang, phân tích cấu trúc và trực quan hóa, thông báo cập nhật, phản chiếu và trợ lý, tác nhân web cá nhân… Trình thu thập dữ liệu web còn được gọi là spiders, robot, worm…



Hình 1.1 Tổng quan hệ thống thu thập dữ liệu

Theo một thống kê năm 2018 trên thế giới có 1,24 tỷ website cùng với nó là khối lượng thông tin khổng lồ. Bài toán đề ra làm thế nào để khai thác được những thông tin từ các website và rút trích ra thông tin có giá trị.

Ưu điểm của thu thập dữ liệu trực tuyến:

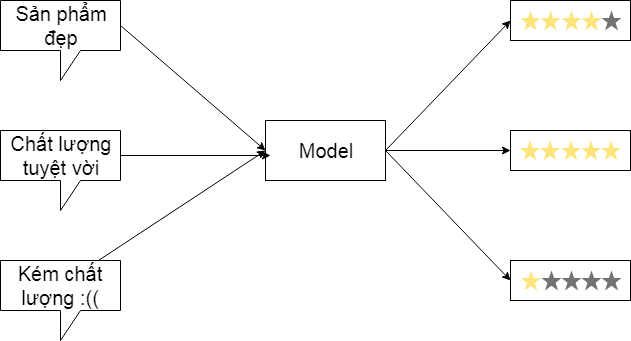
* **Chi phí thấp:** Việc thu thập dữ liệu không làm bạn phải tốn hàng ngàn đô la. Đây có lẽ là một trong những điều hấp dẫn nhất đối với các nhà nghiên cứu không có nhiều kinh phí. Chi phí tiết kiệm được có thể là do bạn sẽ không phải thuê địa điểm, thuê người phỏng vấn, hoặc tốn thời gian công sức chạy đôn chạy đáo để khảo sát.
* **Tự động hóa và truy cập thời gian thực**: Mọi thao tác thu thập, trích xuất, lưu trữ được xử lý tự động và ứng với thời gian thực giúp độ chính xác dữ liệu nâng cao, tốn ít chi phí công việc đào tạo.
* **Tốn ít thời gian**. Triển khai nhanh chóng và lặp lại là lợi thế của thu thập dữ liệu trực tuyến, điều mà các phương pháp truyền thống khó đạt được.

Khó khăn của thu thập dữ liệu trực tuyến:

* **Một số lượng lớn trang web được tạo mỗi ngày:** không ngừng phát triển đó là một đặc tính của internet và cùng với đó dữ liệu trên internet ngày khổng lồ hơn.
* **Cấu trúc các trang web thay đổi thường xuyên:** Các trang web có thể thay đổi thường xuyên về mặt cấu trúc gây khó khăn trong việc sử dụng lại trình thu thập cũ.
* **Môt số lớp người dùng chưa tiếp cận với Internet:** Lớp người dùng lớn tuổi ở Việt Nam còn khá xa lạ với Internet điều này gây khó khăn khi muốn thu thập thông tin trực tuyến liên quan đến lớp đối tượng này.

# 1.3 BÀI TOÁN PHÂN TÍCH QUAN ĐIỂM

Đây là một ứng dụng của bài toán thu thập dữ liệu từ nguồn dữ liệu trực tuyến. Hiện nay với sự bùng nổ của nguồn dữ liệu trên internet đặc biệt là sự phát triển các trang mạng xã hội (Facebook, Instagram, Twitter…) và các trang thương mại điện tử (Lazada, Shopee, Tiki…) đang dần có chỗ đứng phía người tiêu dùng. Với sự phát triển đó đã cho phép mọi người không chỉ chia sẻ thông tin trên đó mà còn thể hiện thái độ, quan điểm của mình đối với các sản phẩm, dịch vụ và các vấn đề xã hội khác. Chính vì thế có một số lượng nhu cầu rất lớn từ các doanh nghiệp muốn thu thập và đánh giá quan điểm người dùng (Sentiment Analysis) về sản phẩm của họ cũng như của các đối thủ cạnh tranh nhằm nâng cao cải thiện chất lượng sản phẩm và dịch vụ.

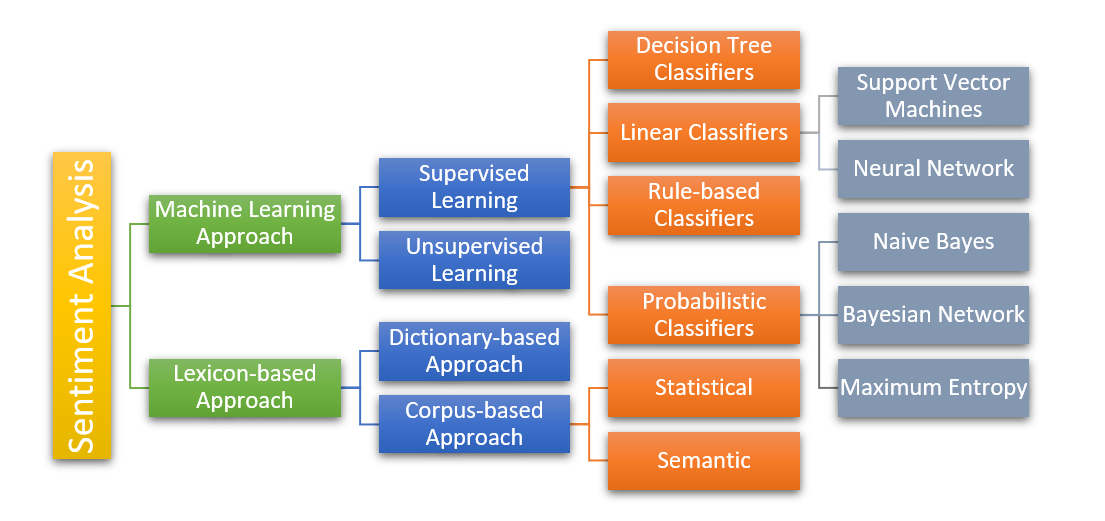


Hình 1.2 Tổng quan hệ thống phân tích quan điểm

# 1.4 HƯỚNG TIẾP CẬN BÀI TOÁN

Đối với bài toán thu thập dữ liệu từ nguồn dữ liệu trực tuyến, thu thập và trích xuất thông tin từ một trang web là lưu trữ và chọn lọc thông tin hướng mục đích trong trang web đó. Các thông tin trên một trang web nằm trong các thẻ HTML, hướng tiếp cận bài toán là phân tích mã nguồn HTML dưới dạng cây với các node là những thẻ HTML chứa thông tin khác nhau. Hướng truy xuất thông tin dựa vào đường đi từ gốc đến các node có chứa thông tin cần truy xuất. Một trang web ngoài nội dung chính có thể có nội dung khác như quảng cáo, các menu… Để tránh dư thừa về mặt thông tin không cần thiết, cần phải xác định đúng đường đi của hướng truy xuất dữ liệu.

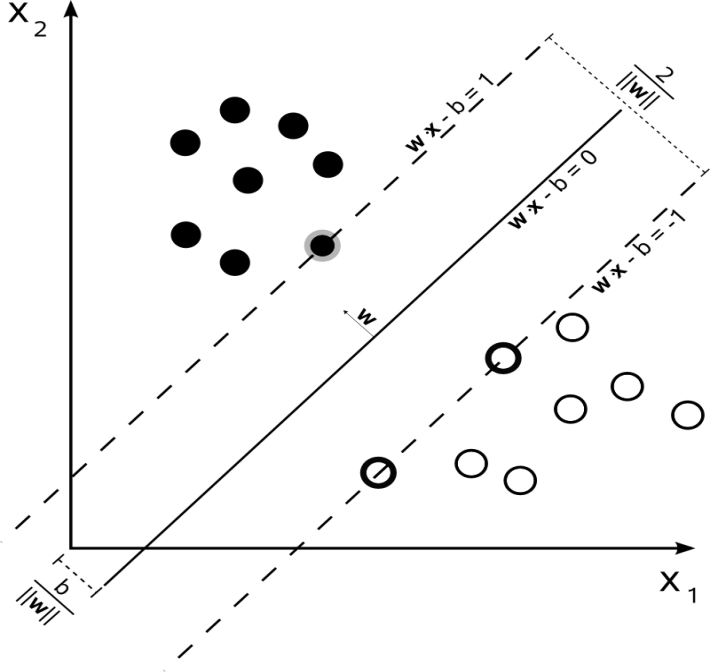
Đối với ứng dụng phân tích quan điểm, Trong những năm trở lại đây, có nhiều nghiên cứu liên quan đến phân tích quan điểm. Có thể phân loại các kỹ thuật này như sau [14]:



Hình 1.3 Các hướng tiếp cận bài toán phân tích quan điểm

Vì dữ liệu thu thập là dữ liệu phản hồi khách hàng đi kèm nhãn là số sao đánh giá tương ứng nên hướng tiếp cận bài toán phân tích quan điểm đi theo nhánh Học máy – Học có giám sát. Sơ lược các hướng tiếp cận bài toán phân tích quan điểm theo nhánh học có giám sát:

* **Support Vector Machines (SVM):** là một thuật toán thuộc nhóm [Supervised Learning](https://1upnote.me/tags/supervised-learning/) (Học có giám sát) dùng để phân chia dữ liệu (Classification) thành các nhóm riêng biệt. SVM thực hiện phân loại bằng cách tìm siêu phẳng phân biệt các lớp trong không gian n chiều.



Hình 1.4 Siêu phẳng phân tách tối ưu giữa hai lớp

* **Neural Network:** là mô hình toán học mô phỏng nơron trong hệ thống thần kinh con người. Model đó biểu hiện cho một số chức năng của nơron(neuron) thần kinh con người.



**Hình 1.5 Mô hình mạng Nơ-ron**

* **Navie Bayes:** là một trong những thuật toán rất tiêu biểu cho hướng phân loại dựa trên lý thuyết xác suất.



Hình 1.6 Kỹ thuật Navie Bayes

* **Bayesian Network:** là cách biểu diễn [đồ thị](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%93_th%E1%BB%8B_(l%C3%BD_thuy%E1%BA%BFt_%C4%91%E1%BB%93_th%E1%BB%8B)) của sự phụ thuộc thống kê trên một tập hợp các biến ngẫu nhiên, trong đó các nút đại diện cho các biến, còn các cạnh đại diện cho các phụ thuộc có điều kiện .
* **Maximum Entropy:** là một thuật toán phân loại văn bản dựa trên nguyên tắc entropy tối đa có thế mạnh là khả năng học và ghi nhớ hàng triệu tính năng từ dữ liệu mẫu.

Như ta thấy ở Hình 1.3 có hai hướng tiếp cận chính trong các kỹ thuật ứng dụng trong giải quyết bài toán phân tích quan điểm người dùng, đó là: sử dụng các thuật toán học máy hoặc tiếp cận theo hướng sử dụng các kiến thức về từ vựng và ngữ nghĩa. Trong các thuật toán học máy lại có thể được chia ra thành các thuật toán học có giám sát hay học không giám sát. Ngoài ra, trong những năm trở lại đây bắt đầu xuất hiện các ứng dụng thành công của deep learning vào trong bài toán phân tích quan điểm đạt kết quả cao. Trong bài luận văn này tôi sẽ lựa chọn mô hình mạng nơ ron hồi quy (Recurrent Neural Network) là một lớp trong Mạng nơ ron nhân tạo.

# 1.5 MỤC TIÊU VÀ ĐỐI TƯỢNG NGHIÊN CỨU

Mục tiêu xây dựng được chương trình thu thập dữ liệu tự động từ nguồn dữ liệu trực tuyến và áp dụng dữ liệu triển khai mô hình cho ứng dụng phân tích quan điểm, từ mô hình đã có xây dựng ứng dụng trực quan trên web. Đối tượng nghiên cứu đối với bài toán thu thập dữ liệu là dữ liệu từ trang thương mại điện tử Shopee.vn, đối với bài toán phân tích quan điểm sẽ sử dụng dữ liệu bình luận và nhãn là số sao tương ứng để huấn luyện mô hình.

# 1.6 KẾT LUẬN

Qua chương 1, chúng ta đã có được cái nhìn tổng quan nhất về bài thoán thu thập dữ liệu từ nguồn dữ liệu trực tuyến và ứng dụng cho bài toán phân tích quan điểm người dùng từ dữ liệu được thu thập, Tính cấp thiết của bài toán trong thực tế, ưu điểm, nược điểm của bài toán thu thập dữ liệu trực tuyến. Chúng ta cũng nắm được những hướng tiếp cận bài toán và lựa chọn hướng tiếp cận phù hợp. Trong chương 2 tôi sẽ bắt đầu tìm hiểu công nghệ và nêu hướng đi cho bài toán thu thập dữ liệu từ nguồn dữ liệu trực tuyến.

CHƯƠNG 2: CÔNG NGHỆ THU THẬP DỮ LIỆU TỪ NGUỒN DỮ LIỆU TRỰC TUYẾN

# 2.1 CÁC KHÁI NIỆM CƠ BẢN

## 2.1.1 World Wide Web

Mạng lưới toàn cầu (viết tắt WWW hay gọi tắt là Web) là một [không gian](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_gian) [thông tin](https://vi.wikipedia.org/wiki/Th%C3%B4ng_tin) toàn cầu mà mọi người có thể truy cập (đọc và viết) thông tin qua các thiết bị kết nối với mạng [Internet](https://vi.wikipedia.org/wiki/Internet); một [hệ thống](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng) [thông tin](https://vi.wikipedia.org/wiki/Th%C3%B4ng_tin) trên Internet cho phép các tài liệu được kết nối với các [tài liệu](https://vi.wikipedia.org/wiki/T%C3%A0i_li%E1%BB%87u) khác bằng các liên kết [siêu văn bản](https://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n), cho phép người dùng tìm kiếm thông tin bằng cách di chuyển từ tài liệu này sang tài liệu khác. [Thuật ngữ](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_ng%E1%BB%AF) này thường được hiểu nhầm là [từ đồng nghĩa](https://vi.wikipedia.org/wiki/T%E1%BB%AB_%C4%91%E1%BB%93ng_ngh%C4%A9a) với chính thuật ngữ Internet. Nhưng Web thực ra chỉ là một trong các [dịch vụ chạy trên Internet](https://vi.wikipedia.org/w/index.php?title=D%E1%BB%8Bch_v%E1%BB%A5_ch%E1%BA%A1y_tr%C3%AAn_Internet&action=edit&redlink=1), ngoài Web ra còn các dịch vụ khác như [thư điện tử](https://vi.wikipedia.org/wiki/Th%C6%B0_%C4%91i%E1%BB%87n_t%E1%BB%AD) hoặc [FTP](https://vi.wikipedia.org/wiki/FTP).

World Wide Web được thiết kế vào năm 1989 bởi Tim Berners-Lee tại CERN (the European Organization for Nuclear Research) tại Geneva. Đề xuất của ông đã thảo luận về các vấn đề mất thông tin trong các hệ thống phát triển phức tạp và đưa ra giải pháp dựa trên hệ thống siêu văn bản phân tán. Sự ra đời của web như một phần của internet vào năm 1991 đã giúp internet trở nên phổ biến và dễ sử dụng hơn. Web được thiết kế như một hệ thống siêu văn bản của người dùng với mục đích cho phép chia sẻ thông tin hiệu quả và dễ dàng giữa các nhóm nghiên cứu được phân tách theo địa lý. Một số người đánh đồng không chính xác World Wide Web với internet. Mặc dù web sử dụng internet làm phương tiện truyền thông tin, nhưng chúng không giống nhau. Web đơn giản là một trong những dịch vụ phổ biến nhất trên mạng internet.

Web là hệ thống thông tin của các mạng máy tính được liên kết với nhau trên toàn cầu, phục vụ các tệp được định dạng bằng HTML, XML, PDF, DOC và một số tệp khác. Một tài liệu có thể là tĩnh (được chuẩn bị và lưu trữ trước đó) hoặc được tạo động (để đáp ứng với đầu vào của người dùng). Các tệp có thể chứa văn bản, hình ảnh và các thành phần đa phương tiện, có thể bao gồm các siêu liên kết đến các tệp khác như vậy trên các máy chủ khác nhau và đặc biệt cũng có thể hoạt động như các giao diện để giúp người dùng đáp ứng nhu cầu thông tin cụ thể của họ [7][8].

## 2.1.2 Thu thập dữ liệu

Thu thập dữ liệu là cách tiếp cận có hệ thống để thu thập và đo lường thông tin từ nhiều nguồn khác nhau để có được bức tranh đầy đủ và chính xác về một lĩnh vực quan tâm. Thu thập dữ liệu cho phép một người hoặc tổ chức trả lời các câu hỏi có liên quan, đánh giá kết quả và đưa ra dự đoán về xác suất và xu hướng trong tương lai. Thu thập dữ liệu là một công việc quan trọng để phục vụ cho công việc nghiên cứu của nhiều lĩnh vực khác nhau. Thu thập dữ liệu thường được thực hiện qua những url có sẵn hoặc các engine tìm kiếm. Một số engine tìm kiếm phổ biến như: Google, Bring, DuckDuckGo… Thu thập dữ liệu web để tạo ra một hệ thống thu thập và lưu trữ nguồn dữ liệu có giá trị trên internet phục vụ nhiều cho các công việc nghiên cứu, dự đoán thị trường, tổng hợp thông tin…

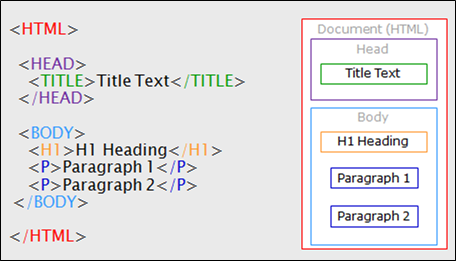
## 2.1.3 Trích chọn thông tin

Sau khi có dữ liệu một trang web việc tiếp theo là trích chọn ra những thông tin cần thiết, phù hợp với nhu cầu. Khi một trang đã được lấy, chúng ta cần phân tích nội dung của nó để trích xuất thông tin, lấy ra các URL để mở ra hướng đi tiếp theo của các trình thu thập. Phân tích nội dung có thể là quá trình khai thác hyperlink/URL đơn giản hoặc nó có thể bao gồm quá trình phức tạp hơn như lọc nội dung HTML để phân tích thành mô hình thẻ HTML dạng cây (HTML tag tree). Phân tích nội dung cũng có thể bao gồm các bước chuyển đổi URL được trích xuất thành dạng tiêu chuẩn, loại bỏ những từ ở phần đầu nội dung của trang và lấy các từ còn lại ở phần thân.

## 2.1.4 HTML

HTML [11] là chữ viết tắt của cụm từ **H**yper **T**ext **M**arkup **L**anguage được dịch là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML. Cha đẻ của HTML là **Tim Berners-Lee**, cũng là người khai sinh ra World Wide Web và chủ tịch của **World Wide Web Consortium** (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

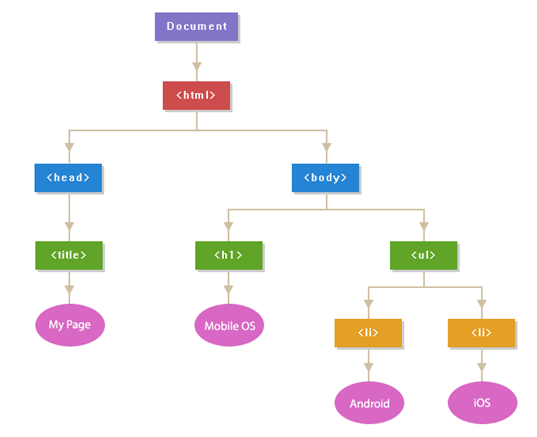
Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc ngọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví <strong> dụ </strong> và ). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ : <strong> Đây là chữ in đậm </strong>). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>). Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.



**Hình 2.1 Cấu trúc đơn giản của một file html**

## 2.1.5 DOM

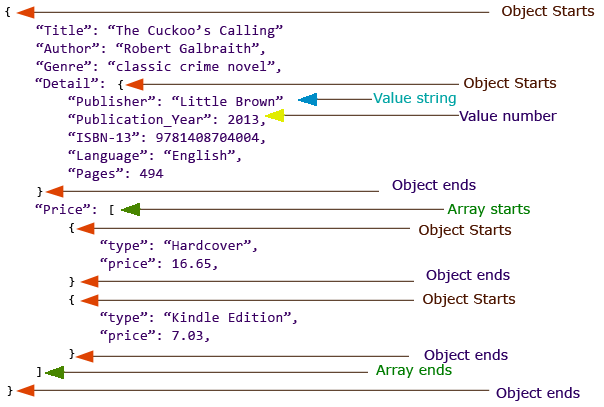
Document Object Model (DOM) hay Mô hình đối tượng tài liệu là một chuẩn được định nghĩa bởi tổ chức Web Toàn Cầu (World Wide Web Consortium – W3C). DOM cung cấp cách thức để truy cập và thay đổi nội dung của các tài liệu HTML. Phần lớn các trình duyệt web hiện này đều triển khai DOM dưới nhiều hình thức và mức độ khác nhau. Cũng như rất nhiều chuẩn khác, đặc biệt các chuẩn liên quan đến lập trình web, DOM được phát triển và cải tiến qua nhiều năm. Khi một trang web được tải, trình duyệt tạo ra một Document Object Model trang. Dưới đây là mô hình cây DOM:



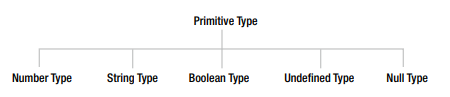
**Hình 2.2 Cấu trúc DOM**

## 2.1.6 JSON

JavaScript Object Notation (JSON) là một kiểu dữ liệu mở trong JavaScript. Kiểu dữ liệu này bao gồm chủ yếu là text, có thể đọc được theo dạng cặp "thuộc tính - giá trị". Về cấu trúc, nó mô tả một vật thể bằng cách bọc những vật thể con trong vật thể lớn hơn trong dấu ngoặc nhọn (**{ }**). JSON là một kiểu dữ liệu trung gian, chủ yếu được dùng để vận chuyển thông tin giữa các thành phần của một chương trình [1].



**Hình 2.3 Ví dụ về Json**



**Hình 2.4 Kiểu dữ liệu trong Json**

## 2.1.7 Web Crawler

Trình thu thập dữ liệu Web (Web crawler) hay còn được gọi là spider or spiderbot và thường gọi tắt là Crawler, là một bot internet duyệt hệ thống World Wide Web, khai thác cấu trúc đồ thị của web và di chuyển từ trang này qua trang khác thường dành cho mục đích lập chỉ mục web (lướt web). Mặc dù vậy cụm từ thu thập không lột tả hết tốc độ của chương trình này, vì chúng có tốc dộ làm việc đáng kinh ngạc, có thể thu thập dữ liệu lên đến hàng chục ngàn trang trong vòng một vài phút.

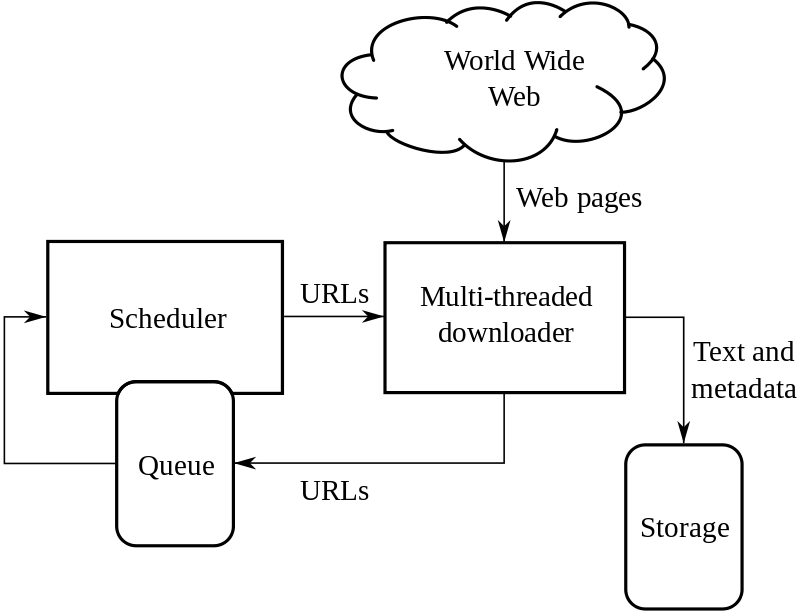
Từ thời kỳ đầu, một động lực quan trọng để thúc đẩy quá trình phát triển của việc thiết kế trình thu thập dữ liệu web là lấy được nội dung của các trang web và thêm chúng hoặc nội dung của chúng vào kho lưu trữ, một kho lưu trữ phục vụ cho các ứng dụng cụ thể trong tìm kiếm web (search engine).

Các trang web chủ yếu được viết bằng các ngôn ngữ đánh dấu như HTML, XHTML và được nhắm đến đối tượng sử dụng là con người chứ không phải máy tính. Các trang web lại chứa đựng nhiều thông tin có ích mà con người có thể muốn thu thập và lưu trữ lại, chính vì thế mà cần phải có những kỹ thuật bóc tách và trích xuất thông tin theo một cơ chế tự động. Các kỹ thuật bóc tách dữ liệu có thể ở mức đơn giản như việc bóc tách các siêu liên kết, hoặc ở mức phức tạp hơn một chút là bóc tách bất kỳ phần nội dung nào trong một trang web.

Trình thu thập thường bắt đầu bằng cách chọn một số các đường dẫn (URL) ứng với các trang web sẽ ghé thăm đầu tiên, các trang này được gọi là các trang hạt giống. Khi ghé thăm một trang hạt giống, trình thu thập sẽ đọc nội dung trang web, lọc ra tất cả các siêu liên kết (hyperlink) có trong trang web đó và đưa các URL tương ứng với chúng vào một danh sách gọi là biên giới (frontier). Dựa vào danh sách này, trình thu thập tiếp tục quá trình duyệt đệ quy để ghé thăm tất cả các URL chưa được duyệt. Quá trình này chỉ dừng lại khi trình thu thập đã thu thập đủ số trang yêu cầu hoặc frontier là rỗng, tức là không còn URL để duyệt. Tuy mô tả này có vẻ đơn giản nhưng đằng sau chúng là khá nhiều vấn đề hóc búa liên quan đến kết nối mạng, bẫy nhện, tiêu chuẩn trích xuất URL, chuẩn hóa các trang HTML, bóc tách nội dung trang HTML vv...

Sau khi đã có được một danh sách các URL dùng cho việc thu thập, ta sẽ thực hiện quá trình lấy trang. Tất cả các trang được lấy một lần và được lưu vào một kho lưu trữ giống như cơ sở dữ liệu của công cụ tìm kiếm, đến đây không cần thu thập thêm. Tuy nhiên web là một thực thể năng động với các không gian con liên tục phát triển và thay đổi nhanh một cách chóng mặt, vì thế thông tin phải liên tục được thu thập để giúp các ứng dụng luôn cập nhật, ví dụ như bổ sung các trang mới loại bỏ các trang đã bị xóa, di chuyển hoặc cập nhật các trang bị sửa đổi.

Về bản chất, quá trình thu thập web chính là quá trình duyệt đệ quy một đồ thị. Các web được xem như một đồ thị với các trang là các đỉnh (node) và các siêu liên kết là các cạnh. Quá trình lấy trang và trích xuất các liên kết bên trong nó tương tự như việc mở rộng tìm kiếm một đỉnh trong đồ thị. Việc tìm kiếm này là khác nhau trong các trình thu thập sử dụng chiến lược tìm kiếm khác nhau. Trong thực tế web crawler phải đối diện với khá nhiều khó khăn như: kích thước khổng lồ của WWW, các trang web được cập nhật thường xuyên... [10][3].



**Hình 2.5 Kiến trúc của web crawler**

Một số web crawler phổ biến hiện nay có thể kể đến như:

* **Scraper API:** là một công cụ dành cho các nhà phát triển, nó xử lý proxy, trình duyệt và CAPTCHA để nhà phát triển có thể lấy HTML thô bằng một lệnh gọi API đơn giản.
* **Scrapy:** là một công cụ nguồn mở dành cho các nhà phát triển Python đang tìm cách xây dựng các trình thu thập dữ liệu web có thể mở rộng. Nó giúp xử lý tất cả các khó khăn trong việc crawler web như xử lý hàng đợi, proxy…
* **Cheerio:** là công cụ dành cho Các nhà phát triển NodeJS muốn có một cách đơn giản để phân tích HTML.
* **Puppeteer** là một API Chrome không đầu dành cho các nhà phát triển NodeJS, những người muốn kiểm soát rất chi tiết đối với hoạt động cạo của họ.
* **Mozenda** dành cho các doanh nghiệp đang tìm kiếm một nền tảng quét web tự phục vụ dựa trên đám mây.

Trong bài luận văn này tôi xin trình bày phương pháp crawler dữ liệu sử dụng thư viện Puppeteer của Nodejs.

## 2.1.8 Headless Browser

Headless Browser là một trình duyệt web không có giao diện đồ họa người dùng. Các Headless Browser cung cấp tương tác tự động một trang web trong một môi trường giống như các trình duyệt web phổ biến khác, nhưng nó được thực hiện thông qua giao diện dòng lệnh hoặc qua một mạng truyền thông. Chúng đặc biệt hữu ích cho việc kiểm thử các trang web vì chúng có thể hiển thị và hiểu HTML giống như các trình duyệt thông thường, bao gồm các cả styling như bố cục trang, màu sắc, font chữ và thực thi Javascript và AJAX mà thường không có sẵn khi sử dụng các phương pháp kiểm thử khác.

Các trường hợp sử dụng Headless Browser:

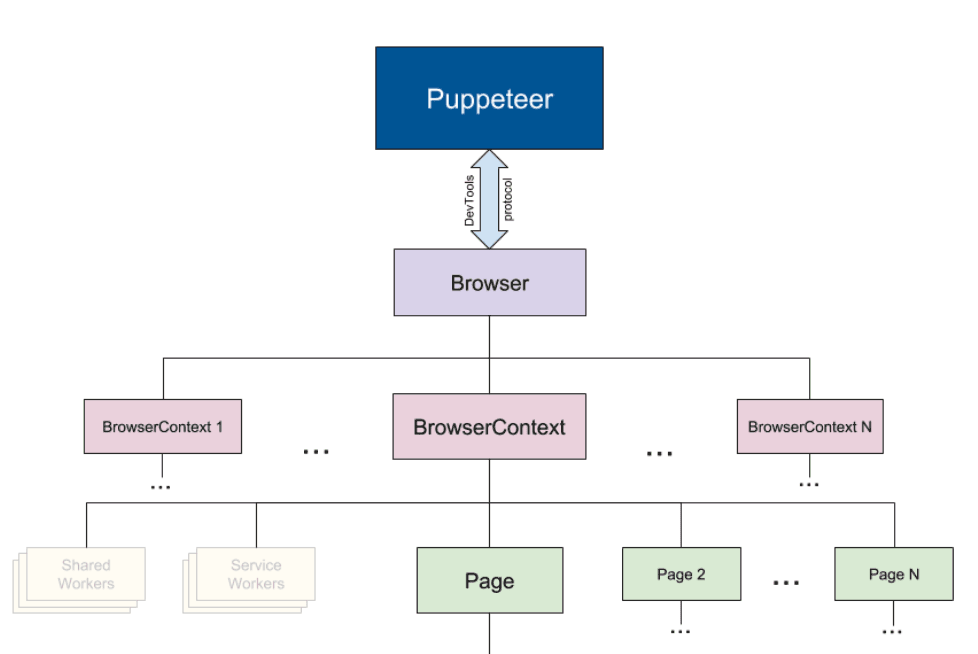
* Kiểm tra tự động hóa trong các ứng dụng web hiện đại.
* Chụp ảnh màn hình của các trang web.
* Chạy các bài kiểm tra tự động cho các thư viện JavaScript.
* Scrape các trang web cho dữ liệu.
* Tự động tương tác của các trang web.

Danh sách các Headless Browser:

* **Google Chrome**: kể từ phiên bản 59 Chrome hỗ trợ chế độ Headless ở Linux và macOS.
* **Chế độ Firefox:** headless có sẵn trên Linux kể từ phiên bản 55. Phiên bản 56 hỗ trợ thêm chế độ Headless trong Windows và macOS.
* **PhantomJS:** một trình duyệt web Headless Browser sử dụng WebKit để hiển thị các trang web và JavaScriptCore để thực hiện các bài kiểm tra theo kịch bản. PhantomJS ban đầu được phát triển bởi Ariya Hidayat vào năm 2010 và đã có được một hệ sinh thái phát triển rộng khắp.
* **HtmlUnit:** một trình duyệt Headless Browser bằng Java. HtmlUnit sử dụng công cụ Rhino để cung cấp hỗ trợ JavaScript và AJAX cũng như khả năng hiển thị một phần.
* **TrifleJS:** một trình duyệt Internet Explorer nhưng Headless bằng trình sử dụng engine Trident để hiển thị các trang và công cụ JavaScript V8 để thực hiện script. TrifleJS sử dụng ngôn ngữ API giống như PhantomJS và hoạt động bằng cách sử dụng đối tượng WebBrowser để điều khiển bất kỳ phiên bản nào của IE được cài đặt trên máy.
* **Splash:** trình duyệt web Headless Browser với API HTTP, hỗ trợ kịch bản Lua và IDE tích hợp sẵn trong IPython (Jupyter). Splash được viết bằng Python và sử dụng engine WebKit. Phát triển bắt đầu tại ScrapingHub vào năm 2013; nó được một phần tài trợ bởi DARPA.

## 2.1.9 Puppeteer

Puppeteer là một ‘Node library’ do Google phát triển cung cấp các API điều khiển Chrome hoặc Chromium thông qua [DevTools Protocol](https://chromedevtools.github.io/devtools-protocol/). Puppeteer mặc định chạy ở chế độ headless tuy nhiên ta cũng có thể cài đặt để chạy non-headless. Những gì làm được bằng giao diện người dùng trên Chrome thì ta đều có thể làm bằng Puppeteer.



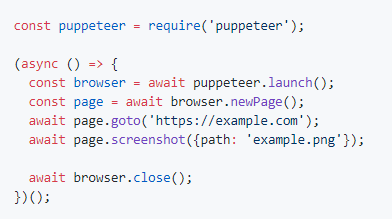
**Hình 2.6 Kiến trúc API của Puppeteer**

Một số điểm nổi bật của Puppeteer:

* Chụp màn hình và lưu lại thành file ảnh hoặc PDF.
* Lấy dữ liệu từ website cho dù site đó bắt ta phải login hay dùng AJAX để load data.
* Tự động submit form, UI testing, keyboard input…
* Tạo ra môi trường automated testing. Chạy trực tiếp test của bạn trên phiên bản Chrome mới nhất.
* Chụp lại timeline trace site của bạn để giúp cho việc chẩn đoán những vấn đề về hiện năng.

Nhược điểm của Puppeteer: chưa tích hợp với các trình duyệt như Firefox, Edge, Safari…

Ví dụ sử dụng Puppeteer để chụp ảnh màn hình trang web:



**Hình 2.7 Code ví dụ chụp ảnh màn hình trang web sử dụng Puppeteer**

Trong đó ‘https://example.com’ là địa chỉ trang web muốn truy cập. Hàm screenshot thực hiện chụp ảnh màn hình và lưu ảnh với tên ‘example.png’.

## 2.1.10 Elasticsearch

Elasticsearch là một công cụ tìm kiếm dựa trên nền tảng Apache Lucene. Nó cung cấp một bộ máy tìm kiếm dạng phân tán, có đầy đủ công cụ với một giao diện web HTTP có hỗ trợ dữ liệu JSON. Elasticsearch được phát triển bằng Java và được phát hành dạng nguồn mở theo giấy phép Apache [4].

### 2.1.10.1 Document

Document là một JSON object với một số dữ liệu. Đây là basic information unit trong Elasticsearch. Hiểu 1 cách cơ bản thì đây là đơn vị nhỏ nhất để lưu trữ dữ liệu trong Elasticsearch.

### 2.1.10.2 Index

Trong Elasticsearch, sử dụng một cấu trúc được gọi là inverted index. Nó được thiết kế để cho phép tìm kiếm full-text search. Cách thức của nó khá đơn giản, các văn bản được phân tách ra thành từng từ có nghĩa sau đó sẽ được map xem thuộc văn bản nào. Khi search tùy thuộc vào loại search sẽ đưa ra kết quả cụ thể.

### 2.1.10.3 Shard

Shard là đối tượng của Lucene, là tập con các Documents của một Index. Một Index có thể được chia thành nhiều Shard. Mỗi node bao gồm nhiều Shard . Chính vì thế Shard mà là đối tượng nhỏ nhất, hoạt động ở mức thấp nhất, đóng vai trò lưu trữ dữ liệu. Chúng ta gần như không bao giờ làm việc trực tiếp với các Shard vì Elasticsearch đã support toàn bộ việc giao tiếp cũng như tự động thay đổi các Shard khi cần thiết. Shard gồm có 2 loại là: Primary Shard và Replica Shard.

### 2.1.10.4 Node

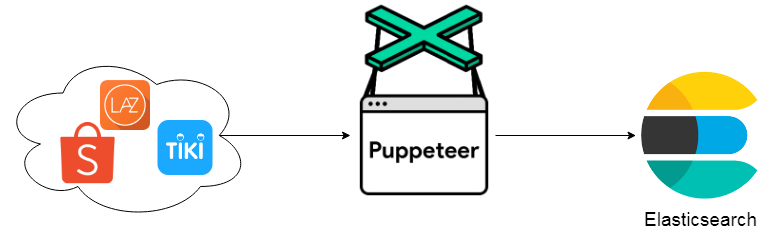
Node là trung tâm hoạt động của Elasticsearch. Là nơi lưu trữ dữ liệu, tham gia thực hiện đánh Index của Cluster cũng như thực hiện các thao tác tìm kiếm. Mỗi Node được định danh bằng một unique name.

### 2.1.10.5 Cluster

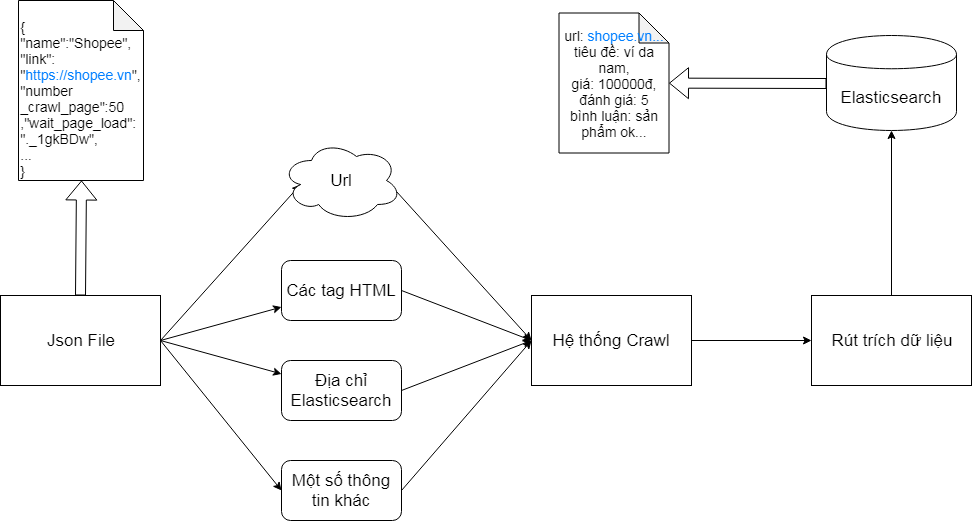
Cluster là Tập hợp các nodes hoạt động cùng với nhau, chia sẽ cùng thuộc tính cluster.name. Chính vì thế Cluster sẽ được xác định bằng một 'unique name'. Việc định danh các Cluster trùng tên sẽ gây nên lỗi cho các node vì vậy khi setup cần hết sức chú ý điểm này. Mỗi Cluster có một node chính (master), được lựa chọn một cách tự động và có thể thay thế nếu sự cố xảy ra. Một Cluster có thể gồm một hoặc nhiều Nodes. Các Nodes có thể hoạt động trên cùng một server. Tuy nhiên trong thực tế, một Cluster sẽ gồm nhiều Nodes hoạt động trên các server khác nhau để đảm bảo nếu một server gặp sự cố thì server khác (Node khác) có thể hoạt động đầy đủ chức năng so với khi có hai servers. Các Node có thể tìm thấy nhau để hoạt động trên cùng một Cluster qua giao thức unicast.

# 2.2 GIẢI QUYẾT BÀI TOÁN THU THẬP DỮ LIỆU TỪ NGUỒN DỮ LIỆU TRỰC TUYẾN.

Sơ đồ các khối chức năng của hệ thống:



**Hình 2.8 Khái quát mô hình thu thập dữ liệu**



**Hình 2.9 Khối chức năng thu thập dữ liệu**

Ý tưởng giải quyết bài toán: sử dụng Puppeteer tạo Headless Browser với đường link đã có thực hiện bóc tách dữ liệu. Với mỗi comment lưu thành một bản ghi riêng trong index của Elasticsearch. Đẩy dữ liệu thu được vào Elasticsearch. Sau khi bóc tách tất cả trang web kết thúc chương trình. Chi tiết các bước thực hiện:

**Bước 1:** Viết file config json với các thông tin cần thiết để đưa vào chương trình ví dụ: thông tin link trang web, địa chỉ server Elasticsearch, số trang tiếp theo sẽ được crawl, tên index được tạo trong Elasticsearch…

**Bước 2:** Import các thư viện cần thiết và load dữ liệu file json config. Mục đích thêm thư viện sử dụng chương trình và lấy thông tin từ file json phục vụ cho chương trình.

**Bước 3:** Kết nối với Elasticseach. Kiểm tra index trên server nếu đã có ghi tiếp dữ liệu, nếu không có tạo mới index.

**Bước 4:** Tạo Handless Browser, tạo page với link là link trong file config json.

**Bước 5:** Trên Handless Browser truy cập link trang web ban đầu. giá trị number\_crawl\_page trong file json ứng với số trang sẽ được thực hiện crawl dữ liệu. Thực hiện vòng lặp với số vòng lặp bằng number\_crawl\_page từ trang đầu tiên đi tới các trang tiếp theo để lấy url lưu vào một mảng urls.

**Bước 6:** Tạo vòng lặp với số lần lặp là độ dài của mảng urls. Với mỗi lần lặp đi tới url ứng với chỉ mục index.

**Bước 7:** Lấy ra số lượng sản phẩm trong trang. Thực hiện vòng lặp với số vòng lặp là số lượng sản phẩm. Tại mỗi vòng lặp click vào một sản phẩm với chỉ mục index sử dụng Puppeteer trích rút thông tin cần thiết qua các thẻ html. Ứng với mỗi bình luận là một bản ghi trong Elasticsearch. Sau khi trích rút dữ liệu trả về danh sách các bản ghi.

**Bước 8:** Lưu dữ liệu đã trích rút vào Elasticsearch. Và đóng Headless Browser, kết thúc chương trình.

# 2.3 KẾT LUẬN

Sau khi có được dữ liệu đã crawl trong Elasticsearch có thể lựa chọn các trường thông tin cần thiết phục vụ mục đích khác nhau để export dữ liệu ra định dạng csv làm dữ liệu đầu vào cho các bài toán phân tích quan điểm, dự đoán xu hướng thị trường, thống kê… Trong phần tiếp theo của bài luận văn tôi sẽ trình bày ứng dụng dữ liệu phản hồi khách hàng được thu thập từ các trang thương mại điện tử cho bài toán phân tích quan điểm người dùng. Qua chương 2 chúng ta đã tìm hiểu về một số khái niệm về những công nghệ liên quan đến bài toán thu thập dữ liệu từ nguồn dữ liệu trực tuyến. Vạch ra hướng đi từng bước giải quyết bài toán qua đó có cái nhìn tổng quát nhất về bài toán. Trong chương tiếp theo tôi sẽ trình bày về ứng dụng phân tích quan điểm cho bài toán thu thập dữ liệu từ nguồn dữ liệu trực tuyến.

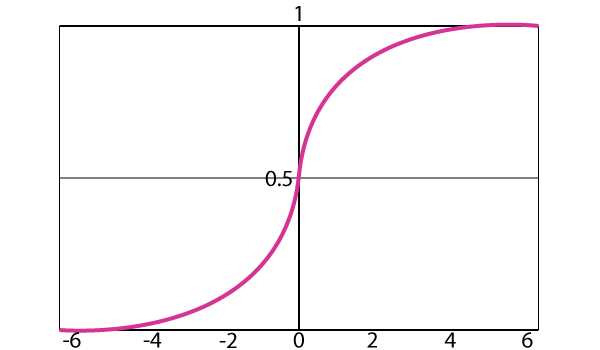
CHƯƠNG 3: PHÂN TÍCH QUAN ĐIỂM NGƯỜI DÙNG

# 3.1 CÁC KHÁI NIỆM CƠ BẢN

## 3.1.1 Hàm Sigmoid

Hàm Sigmoid (hay còn gọi là hàm số Logistic) là một hàm số toán học có đường cong dạng hình chữ "S" với công thức như sau: f(x) =

Mọi giá trị khi đi qua hàm Sigmoid sẽ nằm trong miền giá trị số thực chạy từ 0.0 đến 1.0, có nghĩa là hàm Sigmoid dùng để nén giá trị lại với đồ thị như sau:



**Hình 3.1 Đồ thị của hàm Sigmoid**

Qua hình bạn có thể thấy giá trị càng âm (càng nhỏ nhất) khi đi qua hàm Sigmoid sẽ cho kết quả càng tiến về 0 và ngược lại, giá trị càng dương (càng lớn nhất) khi đi qua hàm Sigmoid sẽ cho kết quả càng tiến về 1, tại vị trí x = 0 hàm Sigmoid sẽ cho giá trị 0.5. Nhờ vào đặc tính này, hàm Sigmoid được ứng dụng rất nhiều vào trong các bài toán trí tuệ nhân tạo trong vai trò là hàm kích hoạt. Hàm Sigmoid có một đặc điểm là một hàm số liên tục cho nên có thể dùng [đạo hàm](https://www.banhoituidap.com/p/38/) trên mọi vị trí x bất kỳ thuộc hàm số với công thức: F′(x) = F(1−F). Với F chính là bản thân hàm Sigmoid [9].

## 3.1.2 Hàm Softmax

Hàm softmax nhận đầu vào là một vec-tơ và cho đầu ra là 1 vec-tơ có cùng số chiều s(x).Hàm Softmax có công thức: Với j = 1… k [5].

Trong đó:

* 0 ≤ F(xi​) ≤ 1.
* F là hàm đồng biến. Exp (x) là hàm đồng biến mà mẫu số lại không đổi nên F cũng là đồng biến. Tức là nếu xi < xj thì F(xi​) < F(xj). Hay nói cách khác thứ tự đầu ra của các vector là không đổi.
* ∑i​F(xi​) = 1.

Hàm Softmax được áp dụng cho các bài toán phân loại với kết quả đầu ra nhiều lớp.

## 3.1.3 Mạng Nơ-ron nhân tạo

### 3.1.3.1 Định nghĩa

Mạng nơ-ron nhân tạo gọi tắt là ANN là một mô hình xử lý thông tin phỏng theo các thức xử lý thông tin của hệ thống nơ-ron sinh học. Nó được tạo lên từ một số lượng lớn các phần tử gọi là nơ-ron kết nối với nhau thông qua các liên kết gọi là trọng số liên kết. Mạng nơ-ron nhân tạo thường được mô phỏng và huấn luyện từ tập mẫu. Qua quá trình huấn luyện, các trọng số liên kết sẽ được cập nhật sao cho giá trị làm lỗi là nhỏ nhất [2].

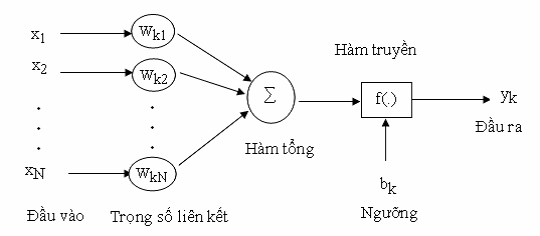
### 3.1.3.2 Cấu trúc của mạng nơ-ron nhân tạo

ANN được cấu tạo từ tầng đầu vào (input layer), các tầng ẩn (hidden layer) và tầng đầu ra (output layer). Lưu ý ANN chỉ có một tầng đầu vào và một tầng đầu ra nhưng có thể có nhiều tầng ẩn. Một ví dụ về cấu tạo ANN với hai tầng ẩn:



**Hình 3.2 Mạng Nơ-ron nhân tạo**

Trong cấu trúc ANN các tầng ẩn gồm nhiều nơ-ron, nhận dữ liệu input từ các nơ-ron ở lớp trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Quá trình xử lý thông tin của một ANN như sau:



**Hình 3.3 Bên trong một mạng Nơ-ron nhân tạo**

Trong hình 3.3 x1… xn là thuộc tính dữ liệu đầu vào là dữ liệu phục vụ cho công việc tính toán. Ví dụ trong bài toán đánh giá học sinh dữ liệu đầu vào là mã học sinh, họ tên học sinh, điểm số, hạnh kiểm … Yk là lớp đầu ra cũng là kết quả đạt được trong bài toán trên thì Yk sẽ là học sinh đó đạt loại gì học sinh trung bình, học sinh khá hay học sinh giỏi. Trọng số liên kết Wk1 … Wkn là một thành phần rất quan trọng trong mô hình ANN, nó thể hiện độ mạnh của dữ liệu đầu vào đối với quá trình xử lý thông tin. Quá trình học của ANN thực chất là quá trình thay đổi và cập nhật trọng số của đầu vào để có một kết quả mong muốn. Hàm tổng trong ANN được tính theo công thức Y = nó cho biết khả năng kích hoạt của nơ-ron đó. Transfer Function có tác động lớn đến kết quả của ANN. Hàm này xác định output của 1 nơ-ron có thể được chuyển đến layer tiếp trong mạng nơ-ron hoặc không.

## 3.1.4 Mạng nơ-ron hồi quy

Để hiểu và trích xuất phân tích quan điểm từ một văn bản ta cần lưu lại ngữ cảnh của các câu trước đó từ đó với ngữ cảnh đã lưu kết hợp phân tích câu hiện tại sẽ đưa ra kết quả có độ chính xác cao. Các mạng ANN không thể làm được điều này vì bản chất nó không mô phỏng khía cạnh thời gian. Và như vậy Mạng nơ-ron hồi quy (Recurrent Neural Network) ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại thông tin từ từ những bước tính toán xử lý trước để dựa vào nó có thể đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại [6].



**Hình 3.4 Cấu trúc liên quan tính toán thời gian của RNN**

Hình 3.4 mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự. Ví dụ ta có một câu gồm 4 chữ “Sản phẩm tuyệt vời”, thì mạng nơ-ron được triển khai sẽ gồm 4 tầng nơ-ron tương ứng với mỗi chữ một tầng. Lúc đó việc tính toán bên trong RNN được thực hiện như sau:

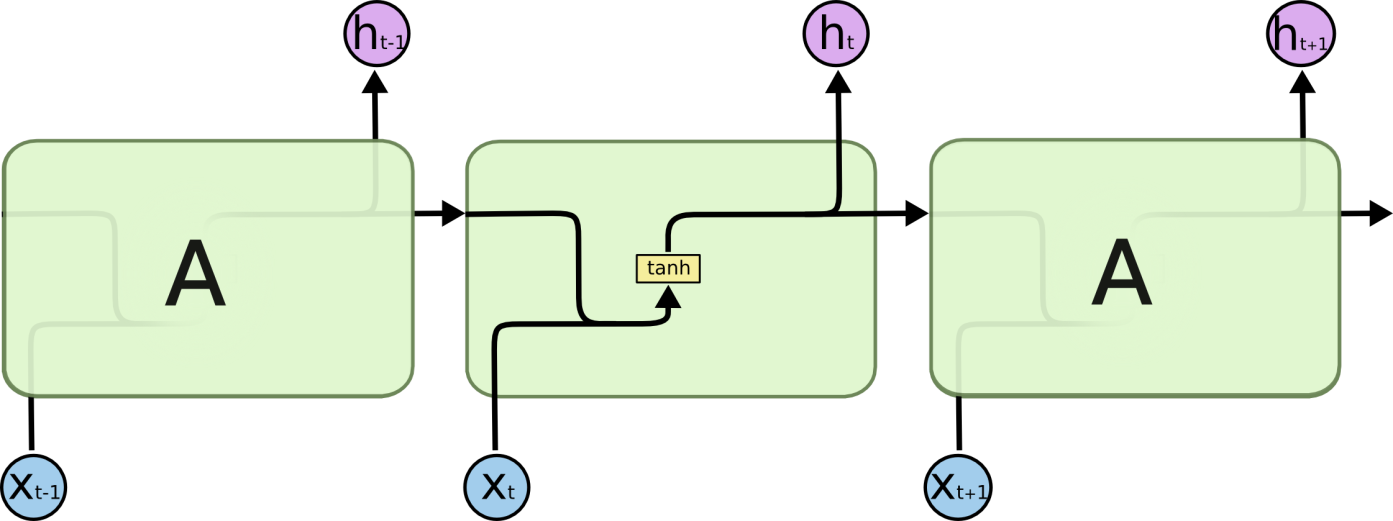
* **Xt** ​ là đầu vào tại bước t. Ví dụ,  x1​ là một vec-tơ one-hot tương ứng với từ thứ 2 của câu (phẩm).
* **St** là trạng thái ẩn tại bước t. Nó chính là bộ nhớ của mạng, St được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại bước đó: St​=f( Uxt​+Wst−1​). Hàm f thường là một hàm phi tuyến tính như [tang hyperbolic (tanh)](https://vi.wikipedia.org/wiki/H%C3%A0m_hypebolic) hay [ReLu](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)). Đề làm phép toán cho phần tử ẩn đầu tiên ta cần khởi tạo giá trị S-1, thường giá trị khởi tạo được gán bằng 0.
* **Ot** là đầu ra tại bước t. Ví dụ ta muốn dự đoán quan điểm khách hàng thì Ot chính là một vector xác xuất giá trị dự đoán tích cực hay tiêu cực Ot=Softmax (V St).

## 3.1.5 Mạng bộ nhớ dài ngắn

### 3.1.5.1 Khái niệm cơ bản

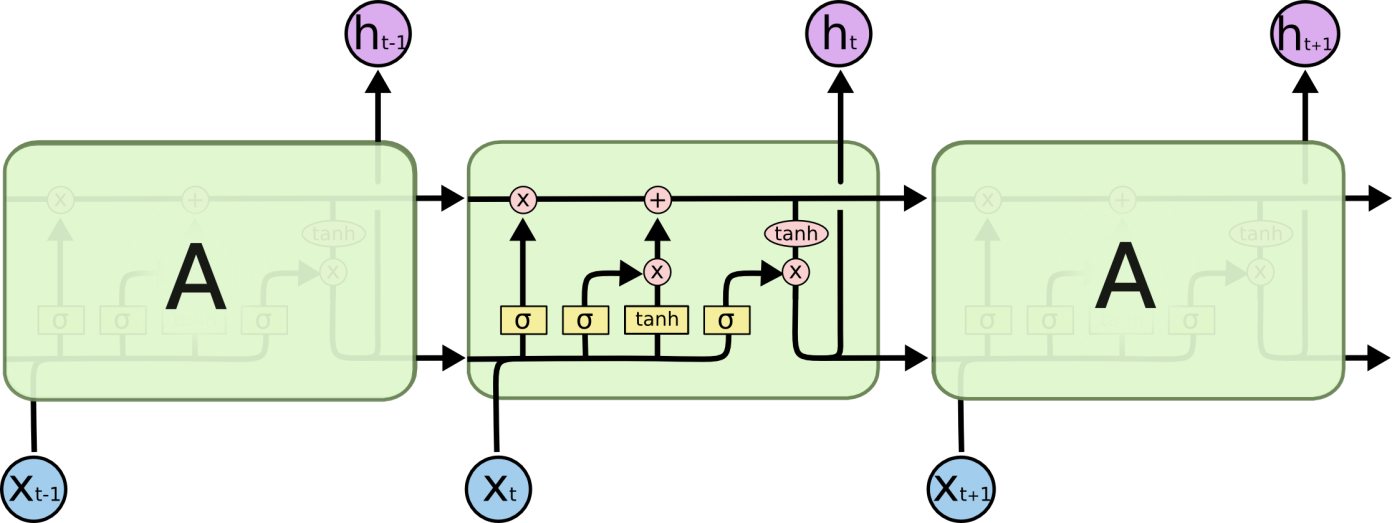
Mạng bộ nhớ dài ngắn (Long Short Term Memory Network) - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi [Hochreiter và Schmidhuber (1997)](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay. LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào [13].

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh.



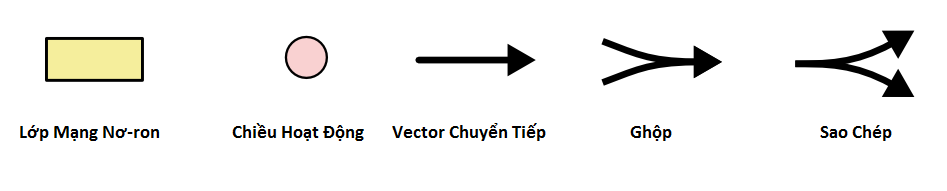
**Hình 3.5 Module lặp lại tiêu chuẩn trong LSTM**

Long Short Term Memory Networks cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



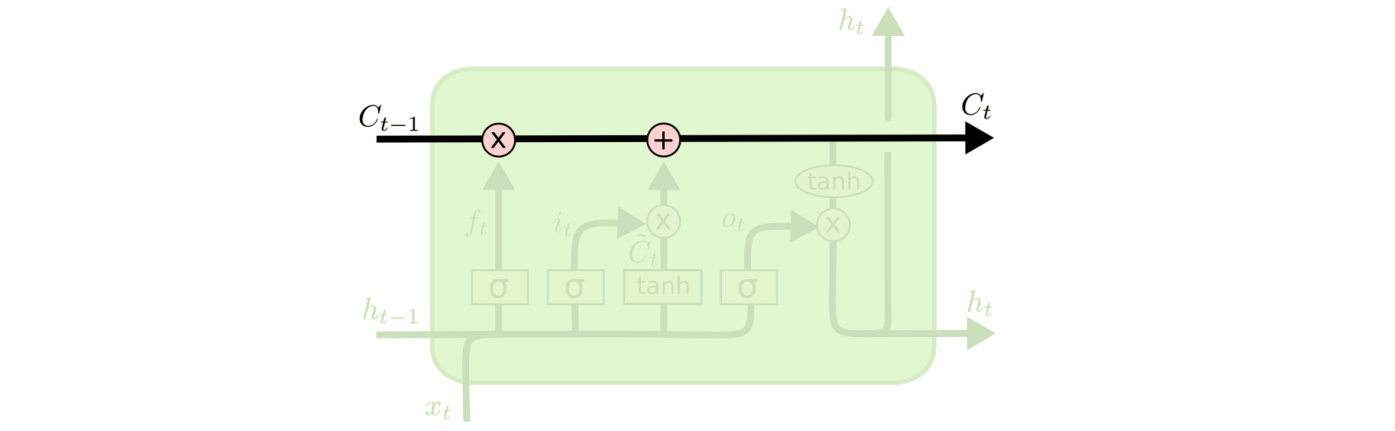
**Hình 3.6 Module lặp lại trong LSTM chứa 4 lớp tương tác**

Các kí hiệu sử dụng:



**Hình 3.7 Chú thích kí hiệu**

Ở sơ đồ Hình 3.6, mỗi một đường mang một véc-tơ từ đầu ra của một nút tới đầu vào của một nút khác. Các hình trong màu hồng biểu diễn các phép toán như phép như phép cộng véc-tơ, nhân véc-tơ …, còn các ô màu vàng được sử dụng để học trong các từng mạng nơ-ron. Các đường hợp nhau kí hiệu việc kết hợp, còn các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới các nơi khác nhau.



**Hình 3.8 Trạng thái tế bào trong LSTM**

Ý tưởng cốt lõi của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ Hình 3.8.

Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế bào, chúng được điều chỉnh cẩn thận bởi các nhóm gọi là cổng (gate).Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



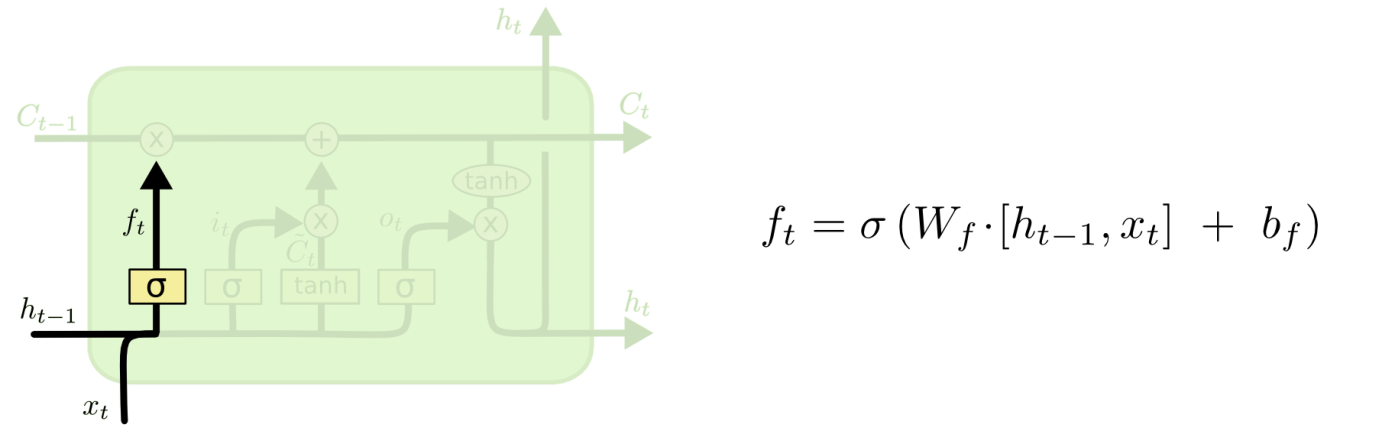
**Hình 3.9 Tầng mạng sigmoid và một phép nhân.**

Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0, 1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

### 3.1.5.2 Bên trong LSTM

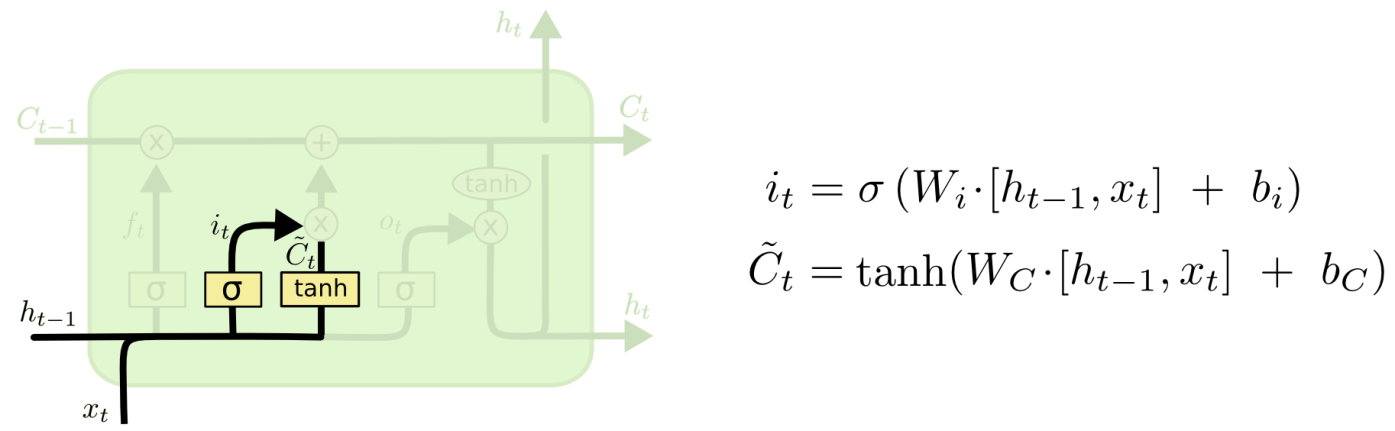
Quá trình hoạt động của LSTM được thông qua các bước cơ bản sau:

**Bước 1:** Bước đầu tiên của mô hình LSTM là quyết định xem thông tin nào cần loại bỏ khỏi cell state. Tiến trình này được thực hiện thông qua một sigmoid layer gọi là “forget gate layer” – cổng chặn. Đầu vào là h𝑡−1 và xt, đầu ra là một giá trị nằm trong khoảng [0, 1] cho cell state 𝐶𝑡−1. 1 tương đương với “giữ lại thông tin”, 0 tương đương với “loại bỏ thông tin”.



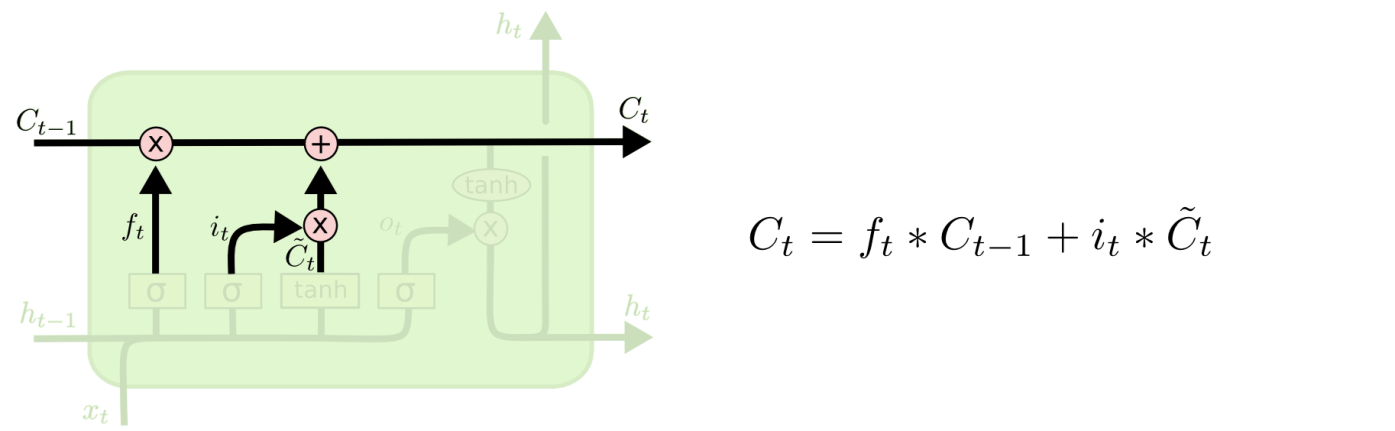
**Hình 3.10 Quyết định những thông tin loại bỏ thông qua hàm sigmoid**

**Bước 2:** Cần quyết định thông tin nào cần được lưu lại tại cell state. Ta có hai phần là single sigmoid layer được gọi là “input gate layer”- cổng vào quyết định các giá trị chúng tôi sẽ cập nhật. Tiếp theo, một tanh layer tạo ra một vector ứng viên mới C ̃t được thêm vào trong cell state.



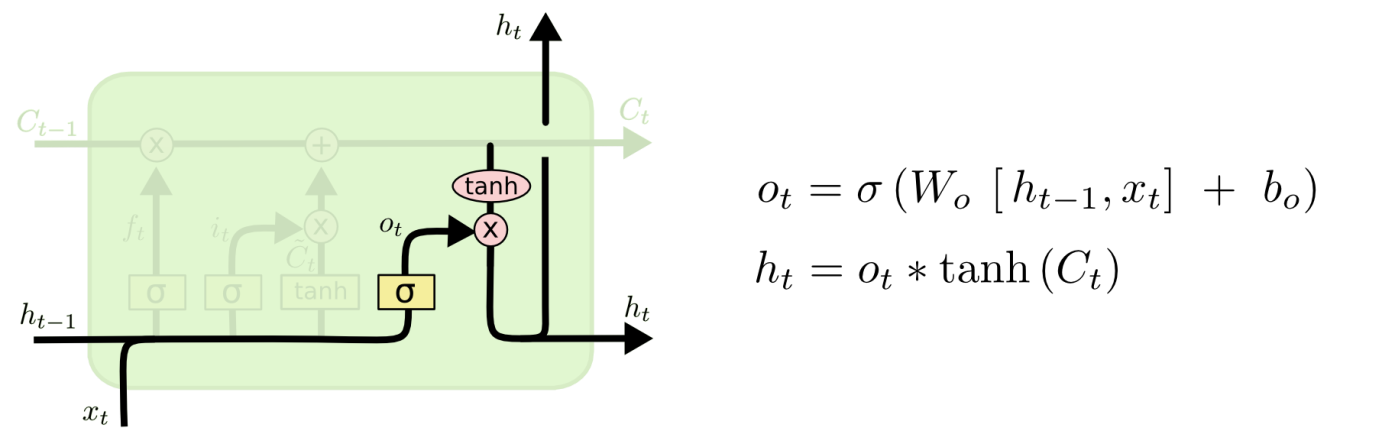
**Hình 3.11 Quyết định thông tin mới lưu vào tế bào qua hàm sigmoid và hàm tanh**

**Bước 3:** Kết hợp hai thành phần này lại để cập nhật vào cell state. Lúc cập nhật vào cell state cũ 𝐶𝑡−1 vào cell state mới Ct. Ta sẽ nhân trạng thái cũ với hàm 𝑓𝑡, để quên đi những gì trước đó. Sau đó, ta sẽ thêm 𝑖𝑡 ∗ 𝐶 ̃t. Đây là giá trị ứng viên mới, co giãn (scale) số lượng giá trị mà tôi muốn cập nhật cho mỗi state.



**Hình 3.12 Cập nhật trạng thái cell state**

**Bước 4:** Dựa vào giá trị đầu ra mong muốn, giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm tanh để co giá trị nó về khoảng [-1, 1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn. Với ví dụ về mô hình ngôn ngữ, chỉ cần xem chủ thể mà ta có thể đưa ra thông tin về một trạng từ đi sau đó. Ví dụ, nếu đầu ra của chủ thể là số ít hoặc số nhiều thì ta có thể biết được dạng của trạng từ đi theo sau nó phải như thế nào.



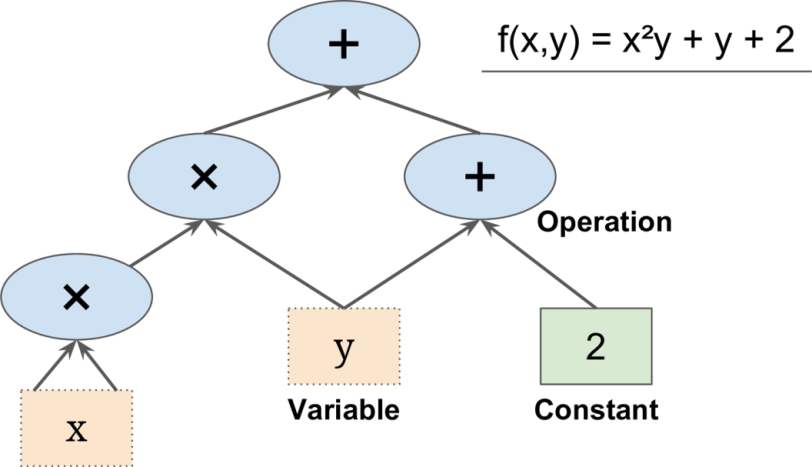
**Hình 3.13 Quyết định đầu ra trạng thái**

## 3.1.6 Tensorflow

TensorFlow [12], một nền tảng trí tuệ nhân tạo được Google phát hành theo giấy phép mã nguồn mở Apache 2.0 vào ngày 09/11/2015. Nền tảng này sử dụng kỹ thuật Machine Learning để học hỏi, nhận biết hình ảnh, giọng nói và các dấu viết dữ liệu. Hiện TensorFlow đang được sử dụng cho chức năng điều khiển giọng nói trong các ứng dụng Google, tìm hình ảnh trong Photos, và mới đây nhất là chức năng trả lời tự độnưg Smart Reply của app email Inbox. Trước TensorFlow, Google cũng từng thiết lập một hệ thống Machine Learning thế hệ thứ nhất mang tên DistBelief. Tuy nhiên, DistBelief bị trói buộc nhiều vào hạ tầng kĩ thuật của Google, lại khá nặng nề và khó mở rộng. Trong khi đó, TensorFlow thì không còn bị ràng buộc gì về mặt hạ tầng nữa, và nó có khả năng chạy trên hầu hết mọi thứ, từ các app smartphone cho đến phần mềm trên server siêu mạnh. Nếu so với hệ thống DistBelief, TensorFlow là một khái niệm xa hơn bằng cách sử dụng công nghệ máy học chuyên sâu (Deep Learning) và mạng nơ-ron hồi quy (Recurrent Neural Network) gồm nhiều lớp.

Về cơ bản, TensorFlow chứa tất cả các lớp dữ liệu, gọi là nút (Node), để tìm hiểu và xác minh một vấn đề. Lớp đầu tiên sẽ yêu cầu hệ thống tìm kiếm một cái gì đó như xác định mẫu chung của vấn đề. Hệ thống sau đó di chuyển để tập hợp các dữ liệu tiếp theo, chẳng hạn như tìm kiếm và xác minh một phần của vấn đề đó. Việc di chuyển hệ thống từ nút tới nút được thực hiện để biên dịch đầy đủ thông tin được đưa ra để xác minh và khẳng định vấn đề đó. “Dòng chảy” tensor như vậy đã tạo nên cái tên TensorFlow. Cũng chính Google đã phải nhận định: “TensorFlow nhanh hơn, thông minh hơn và linh hoạt hơn hệ thống cũ của chúng tôi (DistBelief), do đó nó có thể được điều chỉnh dễ dàng hơn với các sản phẩm mới và giúp quá trình nghiên cứu diễn ra thuận lợi hơn”.

Những đặc điểm nổi bật nhất của TensorFlow có thể kể đến như khả năng sử dụng trên các platform khác nhau từ smartphone, pc tới distributed servers. Thư viện này có thể được chạy với CPU hoặc cùng với GPU, cùng với danh sách API rất dễ sử dụng, cho phép tính toán nhanh chóng trên mạng nơ-ron. Một số 19 các mã nguồn có sẵn trong thư viện TensorFlow là nhận diện ảnh với mạng convolutional neural network, học ngữ nghĩa của từ với word2vec, mô hình hóa ngôn ngữ với recurrent neural network, máy phiên dịch và chatbot với seq2seq. Ví dụ minh họa đơn giản về tensorflow:



**Hình 3.14 Sơ đồ của một tensorflow đơn giản**

## 3.1.7 Word Embedding

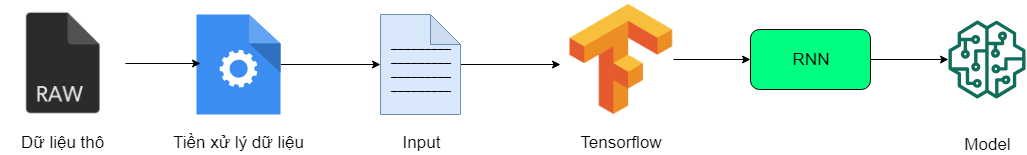
Word Embedding là một kỹ thuật cho việc học mật độ dày đặc thông tin đại diện của từ trong một không gian vector có số chiều nhỏ hơn. Mỗi một từ có thể được xem như là một điểm trong không gian này, được đại diện bởi một vector có độ dài cố định. Word Embedding thường được thực hiện trong lớp đầu tiên của mạng: Trong đó lớp embedding sẽ ánh xạ một từ (chỉ số index của từ trong từ điển từ vựng) từ từ điển sang một vector dày đặc với kích thước đã cho.

Có hai loại Word Embedding:

* **Frequency-based embedding:** dựa vào tần số xuất hiện của các từ để tạo ra các vector từ, trong đó có 3 loại phổ biến nhất là Count Vector, tf-idf Vector, Co-occurrence Matrix.
* **Prediction-based embedding:** xây dựng các vector từ dựa vào các mô hình dự đoán, tiêu biểu của mô hình này là Word2vec.

**Word Embedding**tạo ra các vector từ mà dựa vào đó ta có thể áp dụng chúng để thực hiện các thao tác về ngữ nghĩa như là tìm từ đồng nghĩa, trái nghĩa… Ngoài ra, chúng cũng là nguồn tài nguyên cho các hệ thống Machine Learning, Deep Learning nhằm thực hiện các mục đích cao hơn như là các hệ thống máy dịch, phân tích cảm xúc dựa trên ngôn từ,vv…

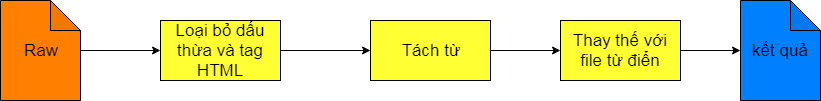
# 3.2 GIẢI QUYẾT BÀI TOÁN PHÂN TÍCH QUAN ĐIỂM NGƯỜI DÙNG



**Hình 3.15 Mô hình cơ sở bài toán phân tích quan điểm**

## 3.2.1 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu ngôn ngữ tự nhiên là một bước quan trọng trước khi đưa dữ liệu vào huấn luyện mô hình. Các văn bản vốn dĩ được liệt kê mà không có cấu trúc, để nguyên vậy để xử lý là rất khó khăn. Đặc biệt là loại văn bản trên web có lẫn các HTML tag, code JS, đó chính là nhiễu gây không ít khó khăn cho việc huấn luyện mô hình. Chính vì vậy để có mô hình có kết quả chính xác tốt công việc tiền xử lý dữ liệu là rất cần thiết



**Hình 3.16 Quy trình tiền xử lý dữ liệu**

Các bước thực hiện:

**Bước 1:** Gán nhãn dữ liệu.

**Bước 2:** Đọc file dữ liệu thô trả về kết quả là hai danh sách một là danh sách comment hai là danh sách nhãn.

**Bước 3:** Với danh sách các bình luận thực hiện đưa về chữ thường, loại bỏ dấu thừa và các html tag nhưng vẫn giữ lại các dấu có thể hiện tình cảm ví dụ: “**:)**”, ”**:(**”…

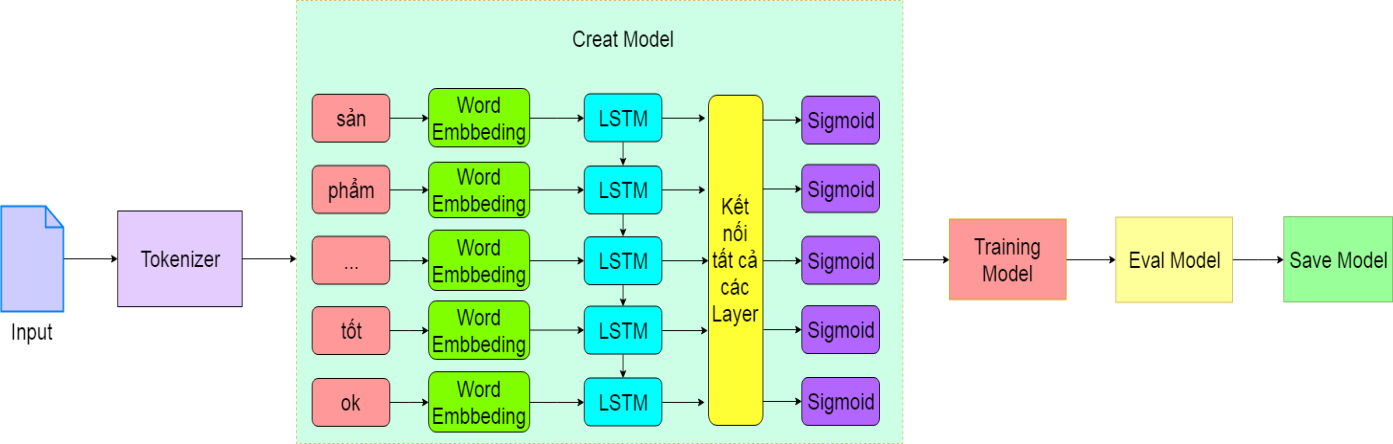
**Bước 4:** Tách từ tokenizer bằng dấu cách. Ví dụ câu “Hôm nay trời đẹp quá.” Sau khi tách từ được một danh sách các từ “Hôm”, “nay”, “trời”, “đẹp”, “quá”.

**Bước 5:** Đọc file từ điển dạng json là danh sách các từ viết tắt thường gặp. Thay thế các từ viết tắt ở mỗi câu. Ví dụ từ viết tắt “k” thay thế bằng từ “không”, từ “đc” thay thế bằng từ “được”…

**Bước 6:** Lưu lại thành file kết quả dữ liệu chuẩn.

## 3.2.2 Mô hình phân tích quan điểm người dùng

Để xây dựng mô hình trong bài luận văn này tôi sử dụng kiến trúc RNN với sự hỗ trợ của công cụ Tensorflow.



**Hình 3.17 Mô hình phân tích quan điểm người dùng**

Các bước thực hiện:

**Bước 1:** Đọc file đầu vào thành dữ liệu danh sách.

**Bước 2:** Với mỗi câu trong danh sách thực hiện tách từ và đưa vào một Hash Table. Mỗi token được nhúng thành một vector có độ rộng xác định.

**Bước 3:** Định nghĩa mô hình RNN hai chiều (Bidirectional RNN). Cấu trúc sử dụng các cell LSTM.

**Bước 4:** Sử dụng hàm kích hoạt Sigmoid để dự đoán xác suất kết quả miền xác suất chạy từ giá trị 0 đến giá trị 1.

**Bước 5:** Từ giá trị xác xuất đưa vào hàm Softmax để phân chia giá trị từ 1 đến 5 ứng với thước đo đặt ra cho bài toán.

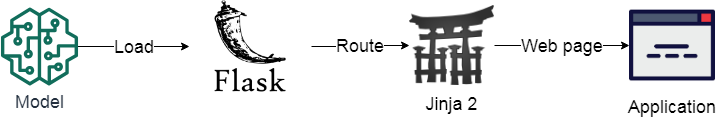
**Bước 6:** Đưa dữ liệu vào huấn luyện mô hình.

**Bước 7:** Đánh giá mô hình.

**Bước 8:** Lưu trữ mô hình.

## 3.2.3 Ứng dụng web cho mô hình phân tích quan điểm người dùng

Thiết kế trang web sử dụng mô hình đã huấn luyện vào ứng dụng thực tế. Nhằm trực quan hóa công việc đánh giá quan điểm người dùng.



**Hình 3.18 Mô hình cơ sở thiết kế ứng dụng web**

Các bước thực hiện:

**Bước 1:** Load model đã huấn luyện.

**Bước 2:** Thiết kế các trang HTML là giao diện của trương trình.

**Bước 3:** Sử dụng framework Flask định tuyến tới các trang HTML

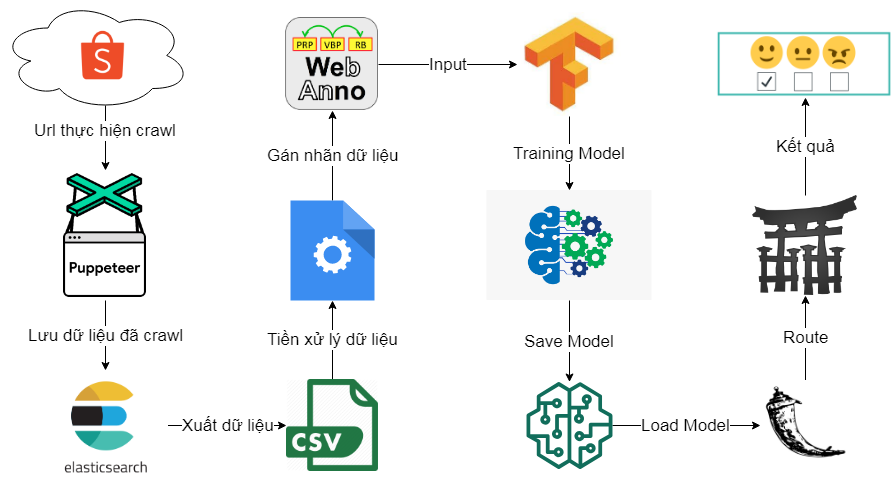
**Bước 4:** Xử lý request để trao đổi dữ liệu đến các trang HTML với Jinja.

# 3.3 KẾT LUẬN

Qua chương 3 chúng ta đã tìm hiểu cơ sở lý thuyết của những thuật toán và công nghệ áp dụng để giải quyết bài toán phân tích quan điểm người dùng. Từng bước xây dựng ứng dụng phân tích quan điểm qua đó có cái nhìn bao quát về bài toán. Trong chương 4 tôi sẽ trình bày về quá trình xây dựng ứng dụng thu thập dữ liệu từ trang thương mại điện tử Shopee và sử dụng dữ liệu để huấn luyện mô hình cho bài toán phân tích quan điểm người dùng.

CHƯƠNG 4: TRIỂN KHAI XÂY DỰNG HỆ THỐNG

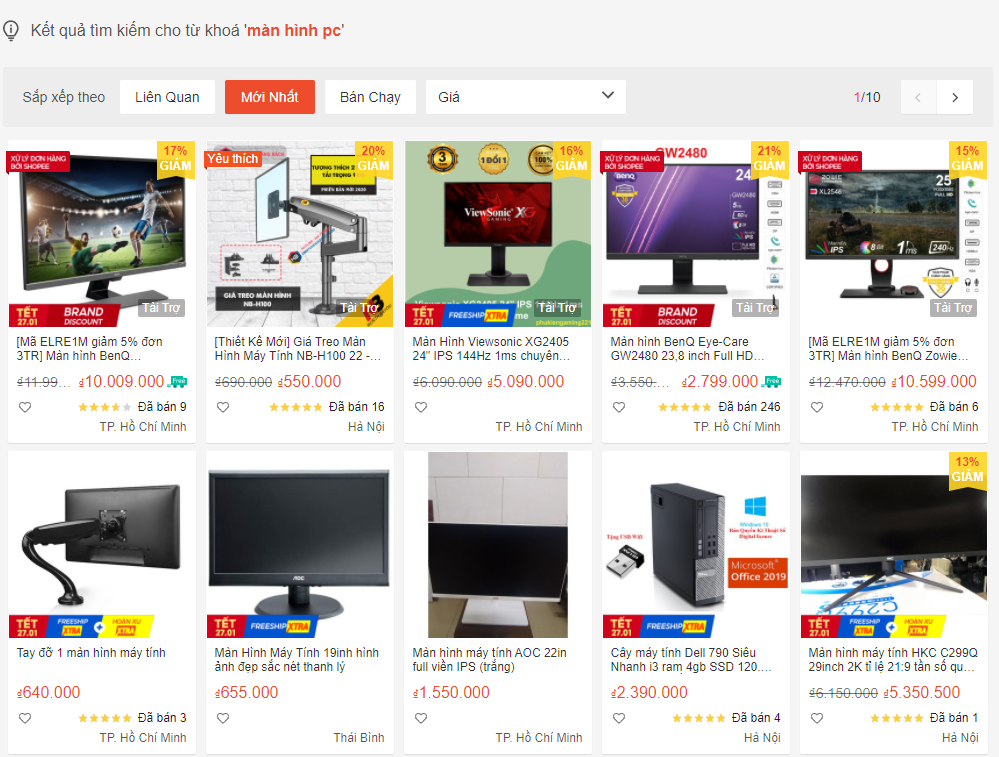
Xây dựng hệ thống với thu thập dữ liệu từ trang Shopee với ngành hàng thời trang. Với dữ liệu thu được gán nhãn dữ liệu, tiền xử lý và đưa vào huấn luyện mô hình. Xây dựng trang web ứng dụng phân tích quan điểm người dùng với 5 tiêu chí: 1 rất tệ, 2 không tốt, 3 trung lập, 4 tốt, 5 rất tốt.



**Hình 4.1 Cấu trúc toàn bộ hệ thống**

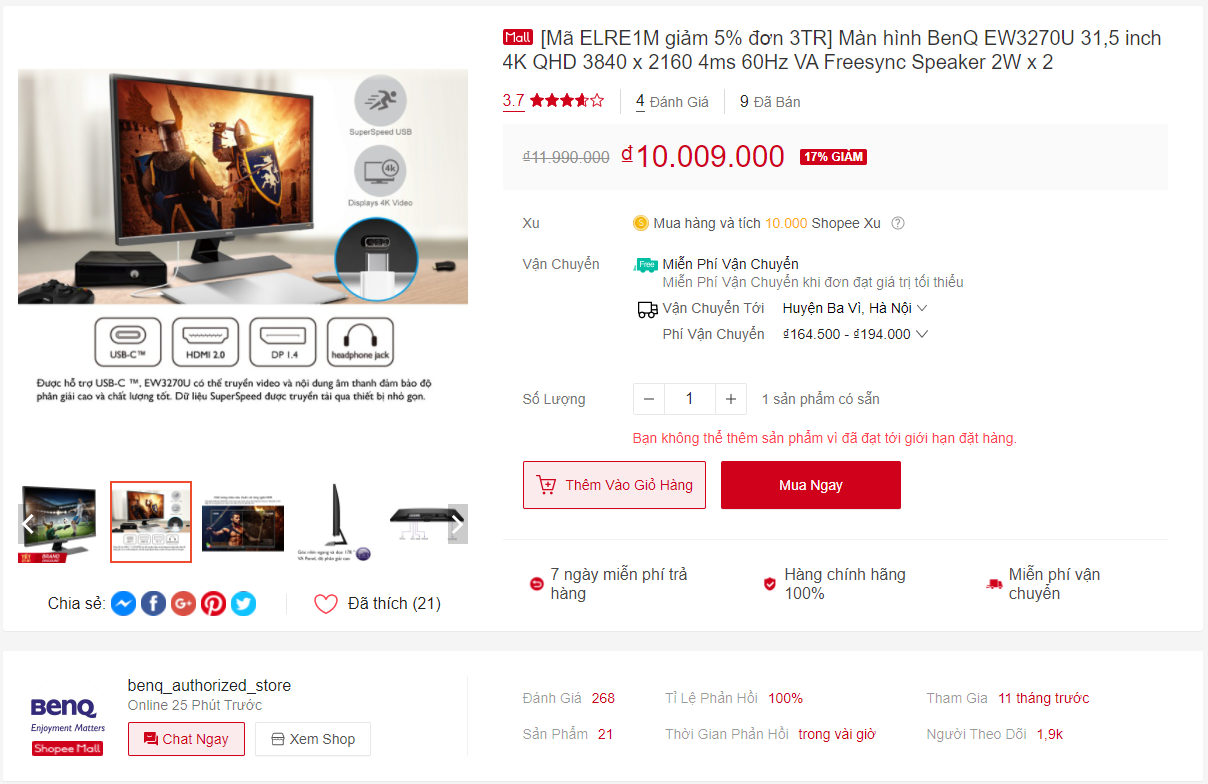
# 4.1 XÂY DỰNG HỆ THỐNG THU THẬP TRÍCH RÚT DỮ LIỆU.

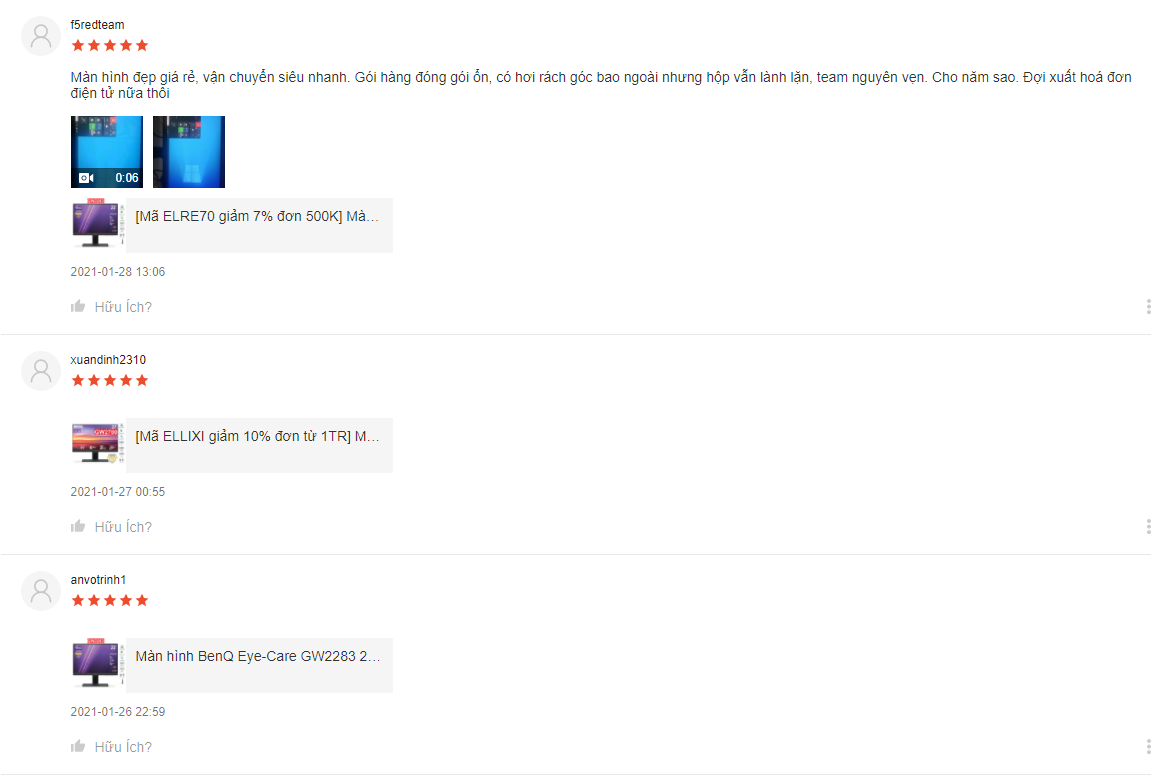
Trong hệ thống này tôi thu thập dữ liệu về thời trang trên trang Shopee. Url của trang: [https://shopee.vn/search?keyword=m%C3%A0n%20h%C3%ACnh%20pc&page=0&sortBy=ctime](https://shopee.vn/search?keyword=m%C3%A0n%20h%C3%ACnh%20pc&page=0&sortBy=ctime&fbclid=IwAR1pQXx6Yfy1_wOL97SDxzXjav1AuUtllP0zVeiOIlZo_zv1NdtLiDZdu5E)



**Hình 4.2 Trang danh sách sản phẩm**

Cần lấy dữ liệu từng sản phẩm trong trang.

**Hình 4.3 Thông tin của một sản phẩm**



**Hình 4.4 Thông tin của một sản phẩm**

Mỗi sản phẩm có nhiều thông tin như tên sản phẩm, hình ảnh, giá… Trong bài tôi sẽ trích rút một số thông tin url sản phẩm, tên sản phẩm, đánh giá chung, hình ảnh, giá, thông tin sản phẩm, bình luận và số sao. File json thông tin cho việc crawl:



***Hình 4.5 File json chứa dữ liệu phục vụ cho việc crawl***

Số trang được đưa vào crawl là 50 và mỗi trang có 50 sản phẩm ta sẽ thu thập được dữ liệu của 2500 sản phẩm. Quá trình crawl dữ liệu:



***Hình 4.6 Quá trình crawl dữ liệu***

Kết quả thu được sau quá trình crawl dữ liệu trên Elasticsearch:



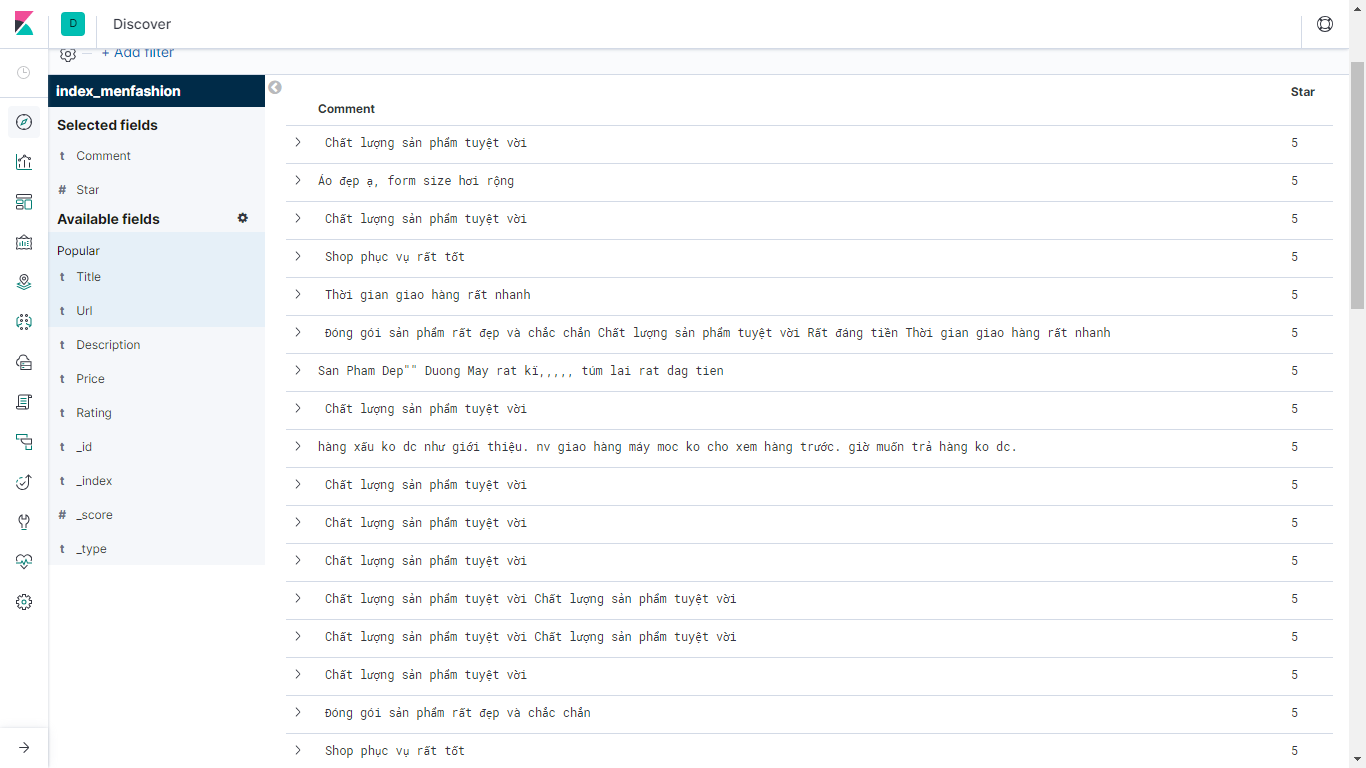
***Hình 4.7 Dữ liệu được lưu trữ***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mặt Hàng** | **Đường Dẫn** | **Số Trang Thu Thập** | **Số Lượng Sản Phẩm** | **Số Lượng Bình Luận** |
| Thời Trang Nam | <https://shopee.vn/Th%E1%BB%9Di-Trang-Nam-cat.78> | 50 | 2500 | 15032 |
| Thiết Bị Điện Tử | <https://shopee.vn/Thi%E1%BA%BFt-B%E1%BB%8B-%C4%90i%E1%BB%87n-T%E1%BB%AD-cat.2365> | 50 | 2500 | 25965 |

***Bảng 4.1Kết quả thu thập dữ liệu từ trang Shopee.vn***

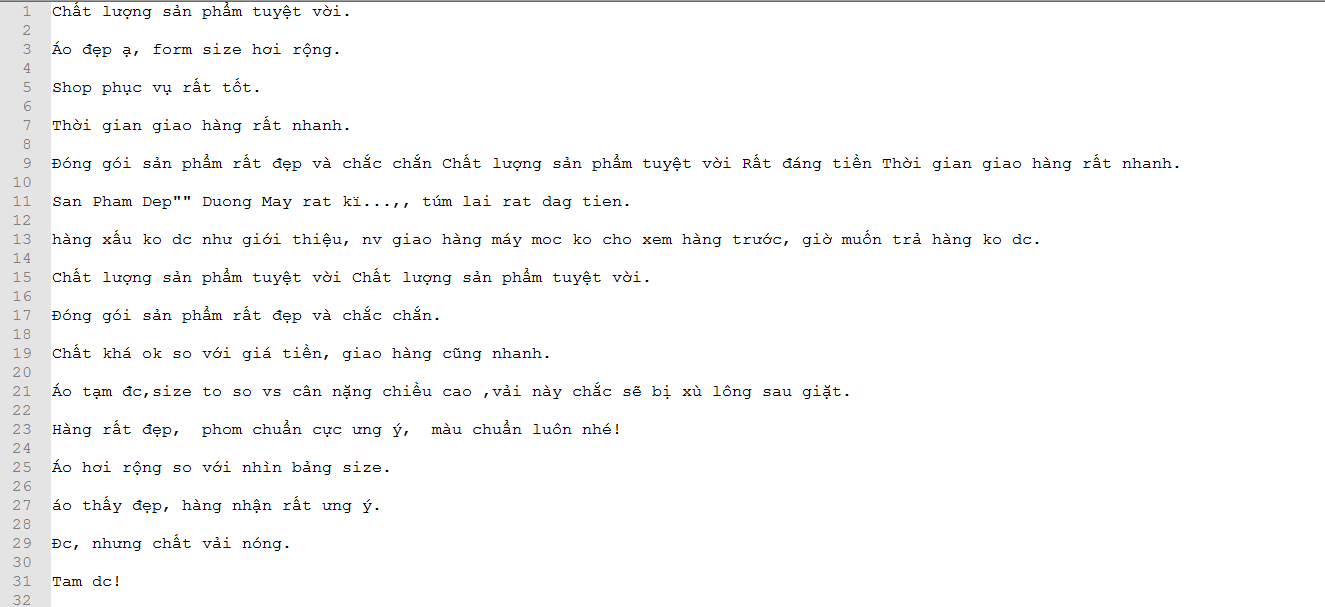
# 4.2 TIỀN XỬ LÝ DỮ LIỆU

Dữ liệu thu được trong Elasticsearch có 10588 bản ghi. Chọn các trường phù hợp với nhu cầu, trong ứng dụng này tôi sử dụng trường Comment và Star.

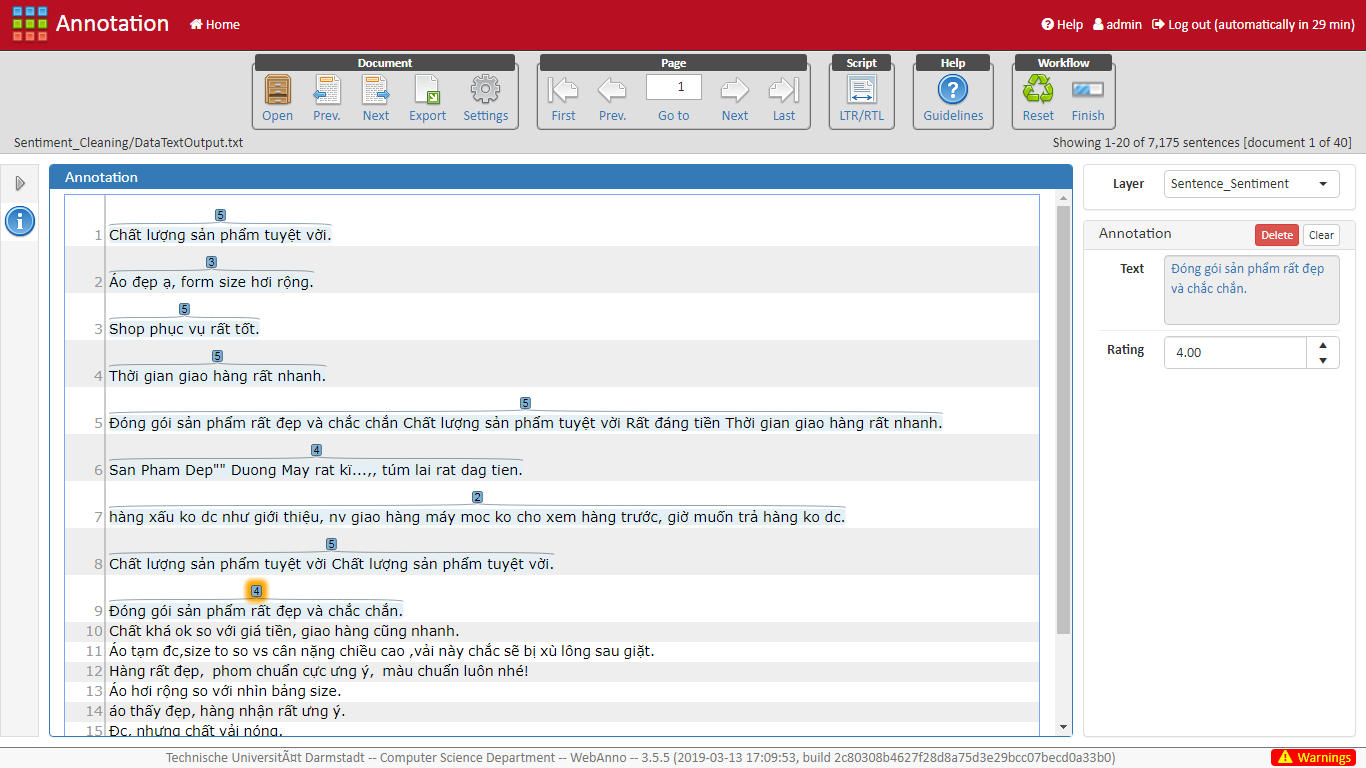


***Hình 4.8 Dữ liệu được lọc theo các trường cụ thể***

Trên Elasticsearch chọn Share, CSV Report để xuất dữ liệu ra file csv. Tại đây chúng ta có thể đưa dữ liệu vào training model với dữ liệu nhãn là “star” thu được, tuy nhiên để tăng độ chính xác cho mô hình nên gán nhãn lại dữ liệu để đảm bảo đầu vào dữ liệu có độ chính xác và khách quan nhất. Ví dụ trong trường hợp khách hàng bình luận về sản phẩm là “Rất tốt” nhưng lại để số sao là 2 gây sai lệch về dữ liệu và nhãn dẫn nhiễu trong mô hình. Để gán nhãn thực hiện đọc file csv và xử lý ghi dữ liệu bình luận vào file txt để thực hiện gán nhãn. File txt mỗi câu cách nhau bằng một dòng trắng để đưa vào Webanno. Sau khi đưa vào webanno thực hiện gán nhãn dữ liệu, Nhãn này sẽ thuộc trên thang điểm 1-5 cho mỗi câu, tương ứng với các giá trị (1) rất tệ - (2) không tốt - (3) trung lập – (4) tốt - (5) rất tốt. Các câu không được gắn nhãn sẽ thuộc trường hợp không đoán được (mức 3).

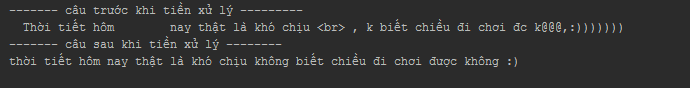


***Hình 4.9 Dữ liệu được đưua vào Webanno để gán nhãn***



***Hình 4.10 Gán nhãn dữ liệu***

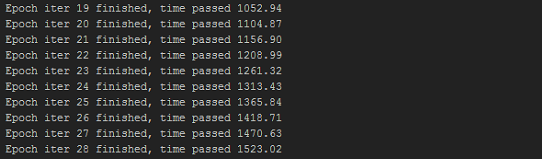
Sau khi gán nhãn dữ liệu xong đưa thực hiện xuất dữ liệu và nhãn ra file tsv. Đọc file tsv và lấy ra dữ liệu gồm bình luận và nhãn. Thực hiện tiền xử lý dữ liệu bình luận: xóa tag html, xóa dấu thừa, thay thế từ điển viết tắt. Ghi kết quả gồm dữ liệu bình luận đã tiền xử lý và nhãn vào file đầu vào json.



***Hình 4.11 Ví dụ tiền xử lý dữ liệu***

# 4.3 HUẤN LUYỆN MÔ HÌNH

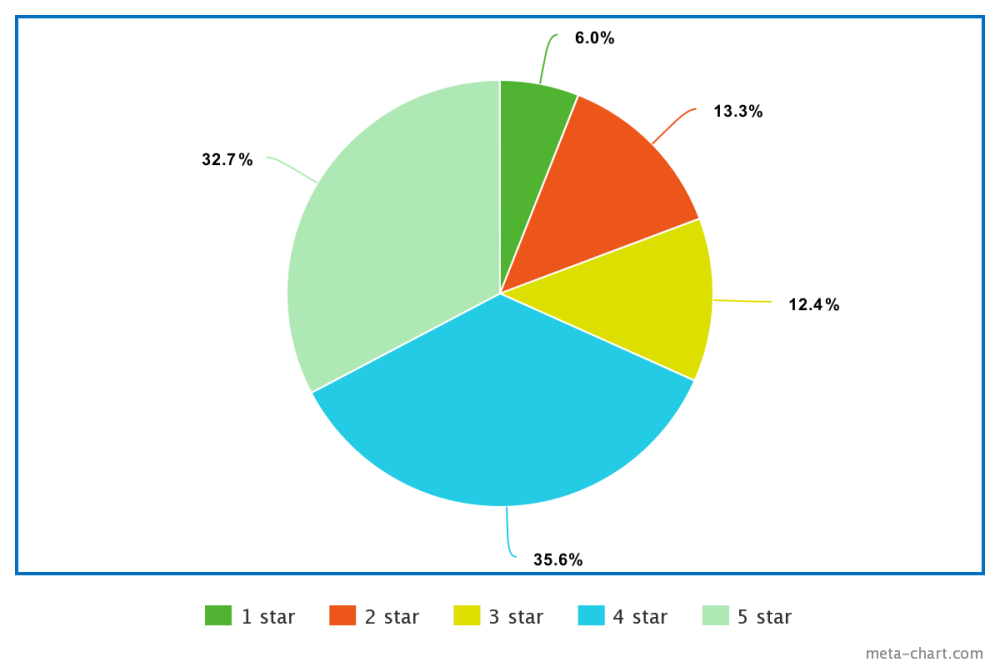
Đưa bộ dữ liệu đầu vào json huấn luyện mô hình với 100 lần lặp, sau khi huấn luyện xong đánh giá mô hình và lưu lại mô hình.



***Hình 4.12 Quá trình huấn luyện mô hình.***

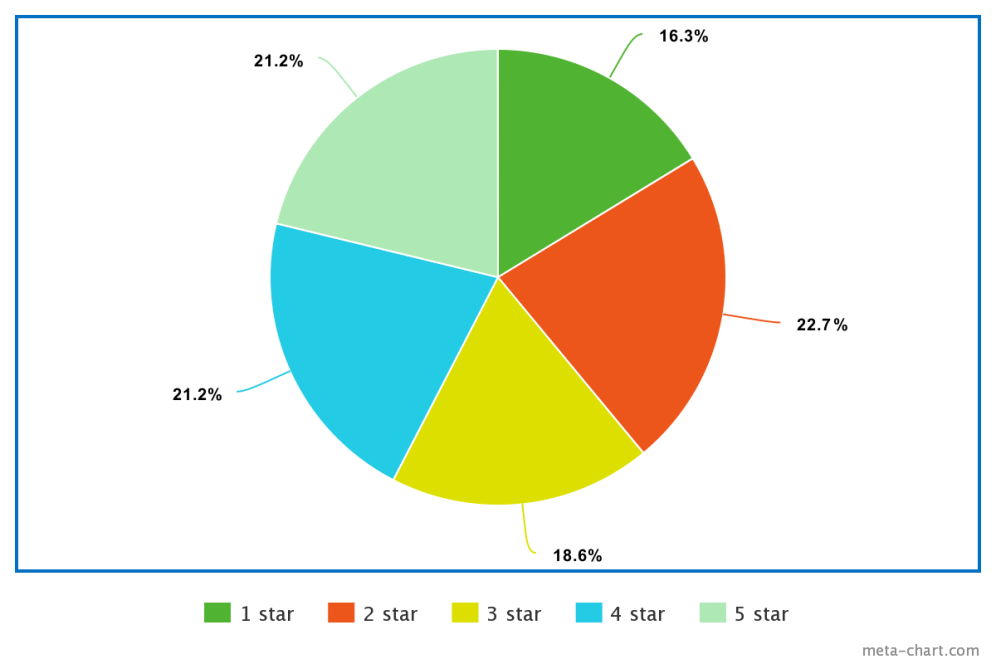
Thực nghiệm mô hình:

* **Mô hình với dữ liệu chưa được tiền xử lý:**
  + **Tiền xử lý dữ liệu:** chưa tiền xử lý dữ liệu
  + **Tổng thời gian huấn luyện mô hình:** 13627 giây
  + **Bộ dữ liệu train:** 17000 câu bình luận có nhãn
  + **Bộ dữ liệu test:** 1000 câu có nhãn
  + **Độ chính xác với đầu ra từ 1 đến 5:** 51%
  + **Độ chính xác với đầu ra negative, neutral, positive:** 70%
* **Mô hình với bộ dữ liệu chưa cân bằng phân bổ nhãn:**
  + **Phân bổ nhãn dữ liệu:** chưa phân bổ nhãn dữ liệu



***Hình 4.13 Dữ liệu chưa cân bằng phân bổ nhãn***

* + **Tiền xử lý dữ liệu:** đã tiền xử lý dữ liệu
  + **Tổng thời gian huấn luyện mô hình:** 11921 giây
  + **Bộ dữ liệu train:** 17000 câu bình luận có nhãn
  + **Bộ dữ liệu test:** 1000 câu có nhãn
  + **Độ chính xác với đầu ra từ 1 đến 5:** 64%
  + **Độ chính xác với đầu ra negative, neutral, positive:** 83%
* **Mô hình dữ liệu cân bằng phân bổ nhãn:**
  + **Phân bổ nhãn dữ liệu:** đã cân bằng phân bổ nhãn dữ liệu



***Hình 4.14 Dữ liệu cân bằng phân bổ nhãn***

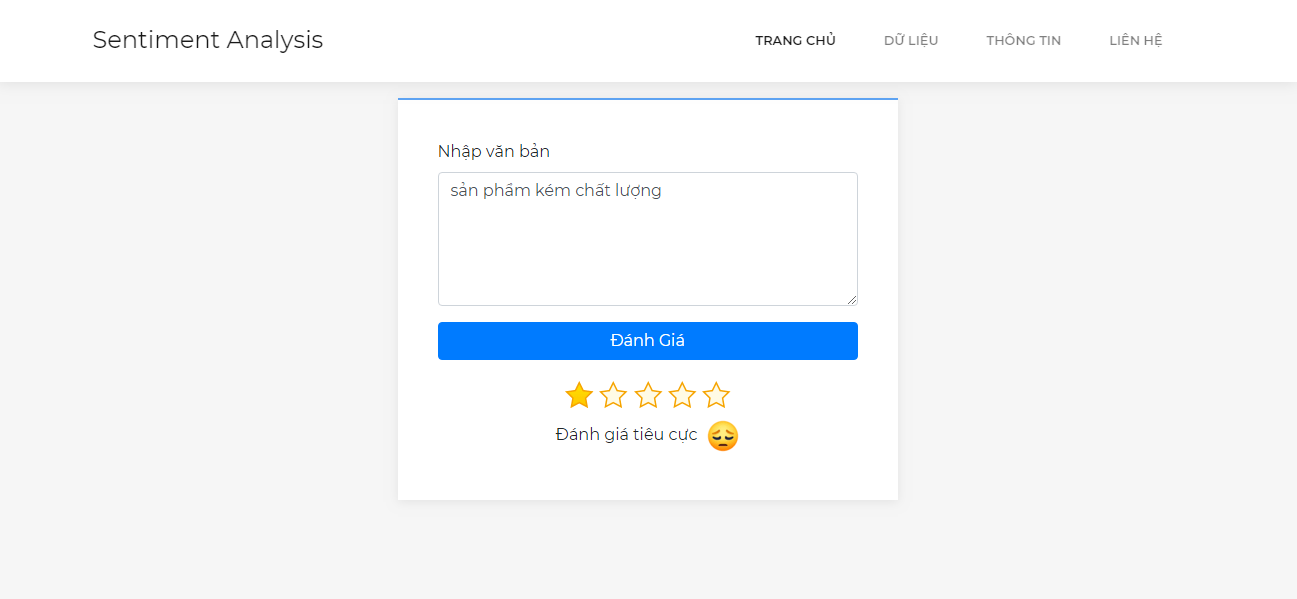
* + **Tiền xử lý dữ liệu:** đã tiền xử lý dữ liệu
  + **Tổng thời gian huấn luyện mô hình:**10370 giây
  + **Bộ dữ liệu train:** 15000 câu bình luận có nhãn
  + **Bộ dữ liệu test:** 1000 câu có nhãn
  + **Độ chính xác với đầu ra từ 1 đến 5:** 72%
  + **Độ chính xác với đầu ra negative, neutral, positive:** 86%
* **Mô hình với dữ liệu VLSP 2016:**
  + **Tiền xử lý dữ liệu:** đã tiền xử lý dữ liệu
  + **Tổng thời gian huấn luyện mô hình:** 6889 giây
  + **Bộ dữ liệu train:** bộ dữ liệu train của VLSP 5100 câu đã có nhãn
  + **Bộ dữ liệu test:** bộ dữ liệu test của VLSP 1050 câu đã có nhãn
  + **Độ chính xác với đầu ra negative, neutral, positive:** 77%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Mô hình chưa tiền xử lý dữ liệu** | **Mô hình chưa cân bằng phân bổ nhãn dữ liệu** | **Mô hình cân bằng phân bổ nhãn dữ liệu** | **Mô hình với bộ dữ liệu VLSP 2016** |
| **Độ chính xác với kết quả 3 lớp (negative, neutral, positive)** | 70% | 83% | 86% | 77% |
| **Độ chính xác với kết quả 5 lớp (1 sao đến 5 sao)** | 51% | 63% | 72% |  |

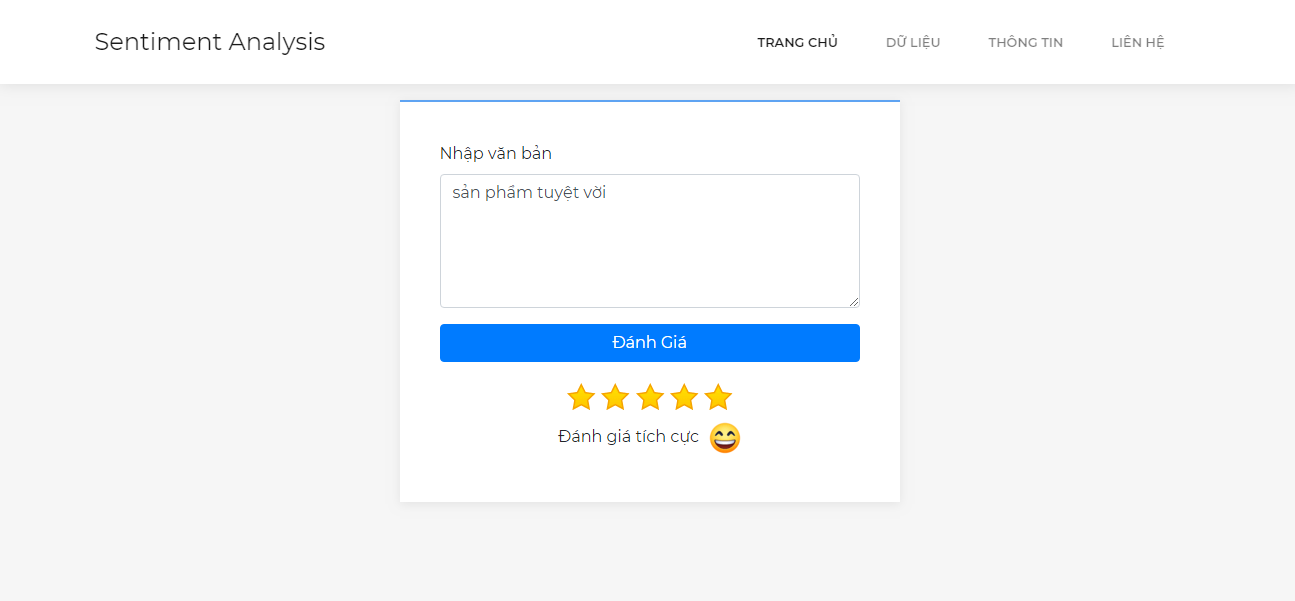
***Bảng 4.2 Kết quả đánh giá mô hình***

# 4.4 XÂY DỰNG ỨNG DỤNG TRÊN WEB

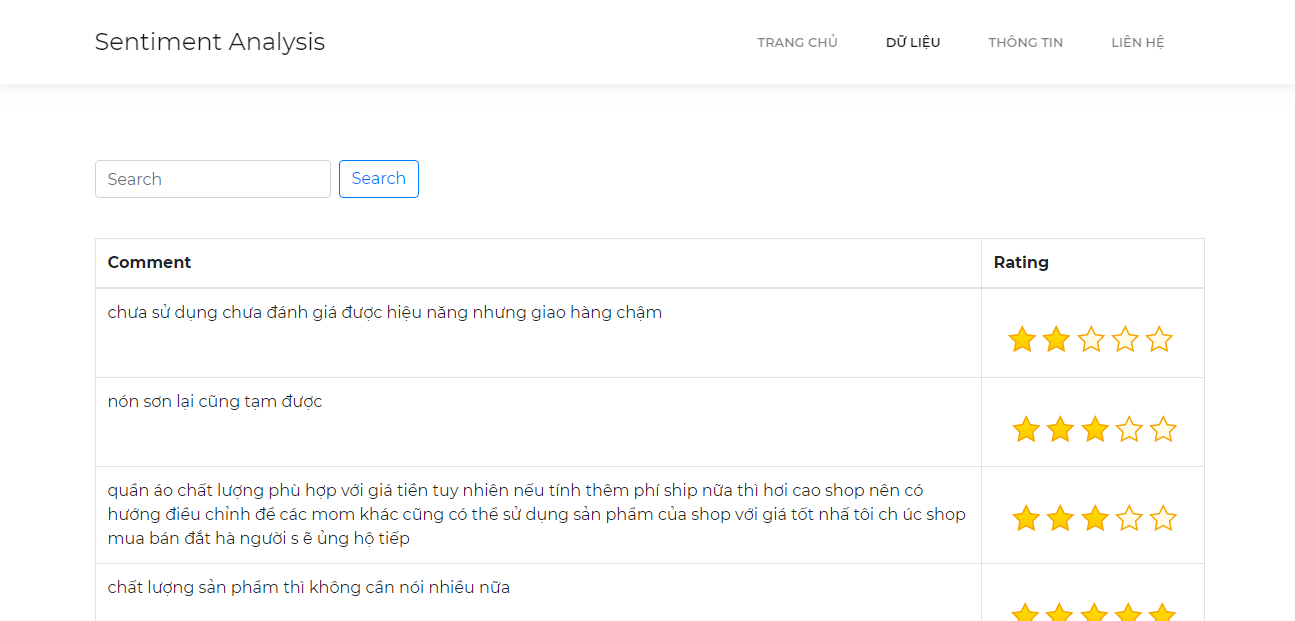
Xây dựng các trang HTML, hai trang chính: trang phân tích quan điểm và trang bộ dữ liệu.



***Hình 4.15 Đánh giá câu với quan điểm tiêu cực***



***Hình 4.16 Đánh giá câu với quan điểm tích cực***



***Hình 4.17 Liệt kê bộ dữ liệu***