

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Phân tích sắc thái bình luận trên trang thương mại  
điện tử**

**LÊ ĐỨC ĐÔ**

do.ld176716@sis.hust.edu.vn

**Ngành: Công nghệ thông tin**

**Giảng viên hướng dẫn:** PGS.TS Lê Thanh Hương

Chữ kí GVHD

**Khoa:** Khoa học máy tính

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 08/2022**

# LỜI CAM KẾT

Họ và tên sinh viên: Lê Đức Đô  
Điện thoại liên lạc: 0974937387  
Email:do.ld176716@sis.hust.edu.vn  
Lớp: AS  
Hệ đào tạo: Công nghệ thông tin Việt Nhật

Tôi – *Lê Đức Đô* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS.TS Lê Thanh Hương*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

*Hà Nội, ngày tháng năm 2022*

Tác giả ĐATN

*Họ và tên sinh viên*

## **LỜI CẢM ƠN**

Đầu tiên em xin gửi lời cảm ơn chân thành nhất tới PGS.TS. Lê Thanh Hương đã luôn nhiệt tình hướng dẫn, chỉ bảo và giúp đỡ em trong suốt quá trình tham gia làm đồ án tốt nghiệp.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn giúp đỡ động viên và tạo điều kiện tốt nhất để em có thể thực hiện hoàn thành đồ án tốt nghiệp này. Trong quá trình xây dựng và hoàn thiện báo cáo cũng như đồ án tốt nghiệp, em sẽ không tránh khỏi những sai sót, vì thế em rất mong các thầy cô và các bạn đọc góp ý để em có thể hoàn thiện hơn nữa sản phẩm này.

Em xin chân thành cảm ơn !

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Ngày nay, vấn đề mua bán hàng online đang được nổi lên rất nhiều, nhất là trong đợt dịch vừa rồi chính vì thế mà các trang thương mại điện tử đang được phát triển rất nhiều như tiki, lazada, sendo,... Nhưng hiện nay trên các trang thương mại điện tử sản phẩm vẫn đang được đánh giá qua việc đánh sao cho sản phẩm kèm với những câu bình luận. Chính vì thế mà người dùng muốn có đánh giá tổng quát nhất về sản phẩm thì phải đọc hết bình luận. Nhưng đánh giá đó của người dùng chỉ mang tính cảm tính, không có độ chính xác cao.

Xuất phát từ vấn đề trên, nhiệm vụ của đồ án tốt nghiệp (ĐATN) này là phát triển mô hình phân tích sắc thái bình luận và hệ thống thu thập và quản lý các sản phẩm trên các trang thương mại điện tử. Mô hình phân tích sắc thái bình luận sẽ được tích hợp trong hệ thống giúp người dùng có thể tìm kiếm một sản phẩm có đánh giá chính xác nhất và đáng mua nhất.

Để làm được điều đó, trước tiên ĐATN tiếp cận bài toán phân tích sắc thái bình luận theo hướng End to End thành các nhãn: Positive, Negative, Neutral với độ chính xác ít nhất là 70%. Trong quá trình làm ĐATN, tác giả đã thực nghiệm các mô hình học sâu phoBERT và mô hình học máy SVM. Trong đó mô hình phoBERT đạt kết quả cao là 0.94 F1-score với bộ dữ liệu VLSP 2016 Dataset cùng với dữ liệu được thu thập từ các trang thương mại điện tử như tiki, thê giới di động.

Đồng thời, ĐATN phát triển hệ thống website thu thập và quản lý sản phẩm từ các trang thương mại điện tử và website giúp người dùng tìm kiếm được sản phẩm có đánh giá chính xác nhất sử dụng công nghệ ReactJS, NodeJS, Firebase.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Tổng quan về bài toán phân tích sắc thái bình luận .....	1
1.2 Các giải pháp hiện tại và hạn chế .....	1
1.3 Mục tiêu và định hướng giải pháp .....	2
1.3.1 Mục tiêu.....	2
1.3.2 Định hướng giải pháp .....	3
1.4 Đóng góp của đồ án .....	4
1.5 Bố cục đồ án .....	4
<b>CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT .....</b>	<b>5</b>
2.1 Ngữ cảnh của bài toán.....	5
2.2 Các kết quả nghiên cứu liên quan .....	5
2.3 Mô hình huấn luyện BERT .....	6
2.3.1 Masked Language Model (MLM) .....	8
2.3.2 Next Sentence Prediction (NSP) .....	9
2.4 Mô hình RoBERTa .....	10
2.5 Mô hình PhoBERT .....	11
2.6 Thuật toán SVM .....	12
2.6.1 Ý tưởng của phương pháp .....	12
2.6.2 Nội dung của phương pháp .....	13
<b>CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT.....</b>	<b>15</b>
3.1 Tổng quan giải pháp.....	15
3.2 Mô hình đánh giá bằng PhoBERT .....	16
3.2.1 Tinh chỉnh mô hình .....	16
3.2.2 Chiến lược tối ưu hoá .....	18

3.2.3 Chiến lược làm mịn nhãn .....	18
3.3 Mô hình đánh giá bằng SVM .....	19
3.3.1 Mô hình end-to-end.....	19
3.3.2 Hàm số Kernel Linear .....	20
3.3.3 Hàm mất mát của Multi-class Support Vector Machine.....	20
<b>CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM.....</b>	<b>22</b>
4.1 Các tham số đánh giá .....	22
4.2 Phương pháp thí nghiệm.....	23
4.2.1 Thực hiện chuẩn bị dữ liệu .....	23
4.2.2 Chuẩn bị môi trường.....	24
4.2.3 Tiên xử lý dữ liệu .....	25
4.2.4 Huấn luyện mô hình PhoBERT .....	26
4.2.5 Huấn luyện mô hình với SVM .....	26
4.3 Kết quả của mô hình PhoBERT .....	27
4.4 Kết quả huấn luyện mô hình SVM.....	28
4.5 So sánh hai mô hình.....	28
<b>CHƯƠNG 5. PHÁT TRIỂN HỆ THỐNG TRANG WEB THU THẬP VÀ ĐÁNH GIÁ SẢN PHẨM.....</b>	<b>30</b>
5.1 Phân tích yêu cầu .....	30
5.1.1 Tổng quan chức năng.....	30
5.1.2 Đặc tả chức năng.....	32
5.1.3 Yêu cầu phi chức năng.....	35
5.2 Công nghệ sử dụng .....	36
5.2.1 FrontEnd .....	36
5.2.2 BackEnd .....	36
5.3 Thiết kế kiến trúc.....	37

5.4 Thiết kế chi tiết giao diện .....	38
5.5 Thiết kế chi tiết server.....	39
5.5.1 Thiết kế kiến trúc backend .....	39
5.5.2 Thiết kế kiến trúc dịch vụ đánh giá .....	40
5.5.3 Thiết kế cơ sở dữ liệu .....	40
5.6 Thiết kế API.....	41
5.7 Xây dựng hệ thống.....	41
5.7.1 Thư viện và công cụ sử dụng .....	41
5.7.2 Kết quả phát triển.....	42
5.8 Đóng gói và triển khai.....	45
<b>CHƯƠNG 6. KẾT LUẬN .....</b>	<b>46</b>
6.1 Kết luận .....	46
6.2 Hướng phát triển trong tương lai .....	46
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>48</b>
<b>PHỤ LỤC.....</b>	<b>48</b>



## DANH MỤC HÌNH VẼ

Hình 2.1	Kết quả của Vietnamese Students' Feedback Corpus . . . . .	6
Hình 2.2	Kết quả của VLSP 2018 Shared Task . . . . .	6
Hình 2.3	Kiến trúc mô hình BERT . . . . .	7
Hình 2.4	So sánh trực quan giữa BERT, OpenAI và ELMo . . . . .	8
Hình 2.5	Sơ đồ kiến trúc BERT cho tác vụ Masked Language Model (MLM) [5]. . . . .	8
Hình 2.6	Ví dụ đầu vào của một mô hình BERT. . . . .	10
Hình 2.7	Kết quả của mô hình RoBERTa. . . . .	11
Hình 2.8	Kết quả của mô hình PhoBERT. . . . .	12
Hình 2.9	Siêu phẳng phân chia dữ liệu học thành 2 lớp + và - với khoảng cách biên lớn nhất. . . . .	13
Hình 3.1	Quy trình thực hiện đánh giá sản phẩm . . . . .	15
Hình 3.2	Minh họa kiến trúc [9] . . . . .	16
Hình 3.3	Chiến lược tinh chỉnh . . . . .	17
Hình 3.4	Tỷ lệ học tập tam giác nghiêng . . . . .	17
Hình 3.5	Mô hình chung cho các bài toán học máy. . . . .	20
Hình 4.1	Công thức Precision và Recall . . . . .	23
Hình 4.2	Code thiết lập để lấy dữ liệu bình luận trên trang thế giới di động . . . . .	24
Hình 4.3	Code cho thiết lập đầu vào của mô hình PhoBERT . . . . .	26
Hình 4.4	Code thiết lập cho quá trình huấn luyện mô hình SVM . . . . .	27
Hình 4.5	Early stoping khi training PhoBERT . . . . .	28
Hình 5.1	Biểu đồ use case tổng quan của hệ thống . . . . .	31
Hình 5.2	Biểu đồ use case thu thập sản phẩm . . . . .	31
Hình 5.3	Biểu đồ use case tìm kiếm sản phẩm . . . . .	32
Hình 5.4	Kiến trúc chung của hệ thống . . . . .	37
Hình 5.5	Thiết kế mockup giao diện . . . . .	38
Hình 5.6	Thiết kế kiến trúc backend . . . . .	39
Hình 5.7	Thiết kế kiến trúc dịch vụ đánh giá . . . . .	40
Hình 5.8	Giao diện trang tổng quan . . . . .	42
Hình 5.9	Giao diện màn cài đặt . . . . .	42
Hình 5.10	Giao diện màn quản lý sản phẩm . . . . .	43
Hình 5.11	Giao diện màn trang chủ . . . . .	43

Hình 5.12 Giao diện của modal đánh giá . . . . .	44
Hình 5.13 Giao diện của modal của sản phẩm . . . . .	44

## **DANH MỤC BẢNG BIỂU**

Bảng 4.1	Kỹ thuật sử dụng và tham số . . . . .	25
Bảng 4.2	Bảng kết quả của PhoBERT . . . . .	27
Bảng 4.3	Bảng kết quả của SVM . . . . .	28
Bảng 5.1	Danh sách use case . . . . .	32
Bảng 5.2	Đặc tả usecase "Thu thập sản phẩm tự động" . . . . .	33
Bảng 5.3	Đặc tả usecase "Đánh giá sản phẩm" . . . . .	34
Bảng 5.4	Đặc tả usecase "Tìm kiếm sản phẩm theo đánh giá" . . . . .	35
Bảng 5.5	Thiết kế chi tiết cơ sở dữ liệu của hệ thống . . . . .	40
Bảng 5.6	Danh sách API của hệ thống . . . . .	41
Bảng 5.7	Danh sách công cụ và thư viện sử dụng . . . . .	41

## **DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

<b>Thuật ngữ</b>	<b>Ý nghĩa</b>
API	Giao diện lập trình ứng dụng (Application Programming Interface)
BERT	Bidirectional Encoder Representation from Transformer
DATN	Đồ án tốt nghiệp
NLP	Xử lý ngôn ngữ tự nhiên (Natural Language Processing)
SA	Phân tích cảm xúc (Sentiments Analysis)
SVM	Support Vector Machine

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Chương này tập trung giới thiệu về những vấn đề thực tế dẫn tới việc chọn đề tài, tổng quan về hệ thống đánh giá sản phẩm trên các trang thương mại điện tử. Tiếp theo đưa ra mục tiêu và phạm vi của đồ án, định hướng giải pháp, đóng góp và bối cảnh trình bày của đồ án.

## 1.1 Tổng quan về bài toán phân tích sắc thái bình luận

Bài toán phân tích sắc thái bình luận (SA) là một bài toán được quan tâm nhiều trong xử lý ngôn ngữ tự nhiên. Bên cạnh đó bài toán cũng có một tầm ảnh hưởng tới đời sống thực tế vì bài toán có thể ứng dụng trong rất nhiều lĩnh vực trong thực tiễn. Bài toán có đầu vào là câu bình luận, đầu ra sẽ là cảm xúc của câu bình luận, có thể là cảm xúc tích cực, tiêu cực hay trung bình.

Trong thời gian gần đây, do sự phát triển của internet và nhu cầu tham khảo phản hồi của người dùng trước đó khi mua sắm trực tuyến trên các trang thương mại điện tử ngày càng tăng. Do đó, các nền tảng thương mại điện tử được phát triển cho phép người dùng có thể để lại những trải nghiệm, đánh giá, nhận xét và phản hồi về các loại dịch vụ, sản phẩm của các doanh nghiệp hay tổ chức. Khi quyết định mua sản phẩm, dịch vụ nào đó người dùng ngoài việc xem xét kĩ thông tin về sản phẩm hay dịch vụ mà còn quan tâm đến phản hồi từ những người dùng trước đó. Tuy nhiên, với lượng phản hồi và đánh giá của người dùng về sản phẩm hay dịch vụ nào đó thì người dùng khó có thể quan tâm được hết. Để giải quyết vấn đề đó người dùng cần một hệ thống có thể phân tích tự động được toàn bộ các phản hồi, đánh giá và tóm tắt lại các phản hồi để khách hàng tham khảo và đưa ra quyết định nhanh chóng.

Hiện nay, các trang mạng thường chỉ sử dụng đến thang điểm mà người dùng đánh giá về sản phẩm đó để phân tích các phản hồi của người dùng. Tuy nhiên, việc dùng thang điểm thì sẽ không khách quan mức độ hài lòng của người dùng bằng những câu văn hay những đoạn bình luận.

Hiện nay, bài toán phân tích sắc thái bình luận được quan tâm ở rất nhiều lĩnh vực khác nhau, từ giáo dục đến khảo sát ý kiến và đặc biệt nhất là lĩnh vực kinh doanh, mua sắm.

## 1.2 Các giải pháp hiện tại và hạn chế

Trong loại bài toán phân tích sắc thái bình luận được phân thành các bài toán có độ khó khác nhau như sau:

- **Đơn giản:** Phân tích cảm xúc (thái độ) trong văn bản thành 2 lớp: tích cực

(positive) và tiêu cực (negative).

- **Phức tạp hơn:** Xếp hạng cảm xúc (thái độ) trong văn bản từ 1 đến 5. Có thể là tích cực (positive), trung bình (neutral), tiêu cực (negative).
- **Khó:** Phát hiện mục tiêu, nguồn gốc của cảm xúc (thái độ) hoặc các loại cảm xúc (thái độ) phức tạp.

Hiện tại đa số giải pháp cho bài toán mới chỉ giải quyết tốt cho bài toán phân tích sắc thái ở cấp độ đơn giản, tức là phân tích sắc thái với hai phân lớp cảm xúc tích cực (positive) và tiêu cực (negative) với độ chính xác hơn 85%.

Hiện nay, bài toán phân tích sắc thái bình luận có 1 số phương pháp giải quyết như sau:

- **Phương pháp dựa trên từ điển các từ thể hiện cảm xúc:** Việc dự đoán cảm xúc dựa vào việc tìm kiếm các từ cảm xúc riêng lẻ, xác định điểm số cho các từ tích cực, xác định điểm số cho các từ tiêu cực và sau đó là tổng hợp các điểm số này lại theo một độ đo xác định để quyết định xem văn bản mau màu sắc cảm xúc gì. Phương pháp này có điểm hạn chế là thứ tự các từ bị bỏ qua và các thông tin quan trọng có thể bị mất.
- **Phương pháp kết hợp Rule-bases (dựa trên luật) và Corpus-bases (dựa trên ngữ liệu):** Phương pháp này kết hợp sử dụng mô hình Deep Learning Recursive Neural Network với hệ tri thức chuyên gia trong xử lý ngôn ngữ tự nhiên (XLNNTN) được gọi là Sentiment Treebank. Sentiment Tree là cây phân tích cú pháp của 1 câu văn, trong đó mỗi nút trong cây kèm theo bộ trọng số cảm xúc lần lượt là: rất tiêu cực (very negative), tiêu cực (negative), trung tính (neutral), tích cực (positive) và rất tích cực (very positive). Hạn chế của phương pháp này ở chỗ chỉ xử lý tốt cho dữ liệu đầu vào là một câu đơn.

Trên các trang thương mại điện tử hiện nay chỉ đưa ra đánh giá cho các sản phẩm dựa trên thang điểm sao, cùng với rất nhiều các phản hồi của người dùng dẫn tới việc khách hàng phải mất thời gian để đọc các phản hồi của người dùng trước đó, hoặc chỉ dựa trên số điểm của sản phẩm để đưa ra quyết định. Vì vậy, việc xây dựng một hệ thống đánh giá sản phẩm dựa trên phân tích sắc thái bình luận rất cần thiết.

### 1.3 Mục tiêu và định hướng giải pháp

#### 1.3.1 Mục tiêu

Từ các vấn đề đặt ra ở phần 1.2, đồ án bao gồm 2 phần:

- Nguyên cứu và thực nghiệm hai mô hình học máy và học sâu cho bài toán

phân tích sắc thái với độ khó là 3 nhãn tốt, trung bình và không tốt.

- Phát triển hệ thống đánh giá sản phẩm trên trang thương mại điện tử.

Đối với các bài toán liên quan tới xử lý ngôn ngữ tự nhiên (NLP) thì không hề xa lạ với các kiến trúc về học máy và học sâu. Trong đó, kiến trúc mạng học sâu tuy đòi hỏi lượng tài nguyên lớn trong quá trình thực nghiệm nhưng mang lại kết quả tốt cho bài toán. Ngoài ra, kiến trúc mạng học máy cũng là một lựa chọn tốt cho các bài toán liên quan tới xử lý ngôn ngữ tự nhiên. Chính vì vậy, trong đồ án này sẽ nghiên cứu, thực nghiệm của hai kiến trúc cho bài toán phân tích sắc thái bình luận.

Từ kết quả thực nghiệm của hai mô hình trên, em tiến hành so sánh kết quả đầu ra, cùng với các chỉ số của hai mô hình. Để đưa ra một mô hình tốt nhất cho bài toán phân tích sắc thái bình luận.

Từ kết quả so sánh của mô hình, em sẽ tiến hành xây dựng và phát triển một hệ thống website thu thập và đánh giá các sản phẩm điện tử từ một số trang thương mại điện tử đang được mọi người chú ý gần đây như tiki, sendo, shopee, ... Nhằm mục đích tạo ra một hệ thống giúp người dùng có góc nhìn tổng quát nhất về sản phẩm để đưa ra quyết định nhanh chóng.

### **1.3.2 Định hướng giải pháp**

Từ các mục tiêu đã nêu trong phần 1.3.1, em đề xuất định hướng giải pháp theo hướng sau: (i) Thực nghiệm mô hình PhoBERT, mô hình SVM cho bài toán phân tích sắc thái. Tiếp theo, (ii) Xây dựng hệ thống thu thập và đánh giá sản phẩm.

Trước tiên, ĐATN tập trung nghiên cứu và thực nghiệm mô hình học sâu PhoBERT - mô hình quy mô lớn đầu tiên được đào tạo cho Tiếng Việt. Cùng với tư tưởng của RoBERTa nên chỉ sử dụng tác vụ Masked Language Model để huấn luyện, bỏ đi tác vụ Next Sentence Prediction.

Tiếp theo, ĐATN cũng tập trung nghiên cứu và thực nghiệm một mô hình học máy có giám sát được sử dụng phổ biến gần đây. Chính là mô hình SVM (SVM with linear kernel)

Bên cạnh đó, ĐATN hướng tới tích hợp mô hình phân tích sắc thái vào hệ thống đánh giá sản phẩm. Chính vì vậy, ĐATN sẽ hướng tới xây dựng và phát triển một hệ thống thu thập và đánh giá sản phẩm trên các trang thương mại điện tử bằng cách sử dụng các công nghệ ReactJS [1] cho phần giao diện, NodeJS và Flask cho phần máy chủ. Việc sử dụng ngôn ngữ javascript xuyên suốt từ phía người dùng tới phía máy chủ giúp đồng nhất ngôn ngữ và dữ liệu trao đổi giữa hai phía. Bên cạnh đó, em sử dụng Firestore (một công nghệ của Google) để lưu trữ dữ liệu cho

hệ thống. Vì Firestore lưu trữ dữ liệu theo mô hình NoSQL nên tốc độ ghi và đồng bộ dữ liệu giữa các ứng dụng client một cách nhanh chóng.

## 1.4 Đóng góp của đồ án

Đồ án này có 2 đóng góp chính như sau:

- Đồ án đề xuất một phương pháp tiền xử lý dữ liệu nhằm loại bỏ nhiễu và dữ liệu ngoại lai trước khi đưa vào huấn luyện mô hình.
- Đồ án thực hiện thu thập và tổng hợp thêm các câu bình luận thực tế từ các trang thương mại điện tử.

## 1.5 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về tổng quan ngữ cảnh của bài toán, một số kết quả nghiên cứu về bài toán, cùng với những kiến thức nền tảng bao gồm mô hình BERT, mô hình RoBERTa, mô hình PhoBERT và mô hình SVM. Đây là tiền đề để hiểu được các giải pháp được trình bày trong các chương tiếp theo.

Trong Chương 3, em trình bày về tổng quan giải pháp cho bài toán phân tích cảm xúc mà ĐATN hướng tới. Từ tổng quan giải pháp đó sẽ trình bày chi tiết các thuật toán, mô hình đã mô tả trong tổng quan giải pháp.

Tiếp theo, trong Chương 4 sẽ trình bày chi tiết về hệ thống đánh giá sản phẩm bao gồm: Phần (i) Phân tích tổng quát yêu cầu của hệ thống, (ii) Trình bày những công nghệ sử dụng nhằm xây dựng hệ thống, (iii) Tổng quan về kiến trúc của hệ thống, (iv) Khái quát về giao diện của hệ thống sẽ hướng tới. Tiếp theo, phần (v) và phần (vi) lần lượt trình bày thiết kế máy chủ và thiết kế api của hệ thống. Cuối cùng, phần (vii) và phần (viii) sẽ bàn về xây dựng hệ thống và đóng gói hệ thống.

Trong Chương 5 này sẽ trình bày về các tham số đánh giá cho bài toán, phương pháp thí nghiệm mà ĐATN hướng tới. Bên cạnh đó, chương này sẽ nêu ra các kết quả mà ĐATN đạt được.

Tiếp theo, Chương 6 sẽ trình bày lại các vấn đề mà ĐATN đã giải quyết được, những vấn đề còn tồn đọng, chưa giải quyết, từ đó đưa ra hướng phát triển trong tương lai.

Cuối cùng, trong Chương 7 sẽ trình bày về toàn bộ tài liệu tham khảo trong ĐATN này.

Sau đây là chi tiết của từng chương của ĐATN này.

## CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

Chương 1 đã nêu ra các vấn đề hiện tại, cùng với mục tiêu và giải pháp cho đồ án này. Chương này đi sâu vào việc trình bày ngũ cảnh của bài toán phân tích sắc thái, các nghiên cứu tương tự. Đặc biệt, tập trung vào việc trình bày cơ sở lý thuyết, đây là cơ sở để hiểu được có giải pháp và kết quả được nêu ở Chương 3 và Chương 4. Các nội dung có trong chương này bao gồm: (i) Ngũ cảnh của bài toán, (ii) Các kết quả nghiên cứu tương tự, (iii) Mô hình huấn luyện BERT, (iv) Mô hình RoBERTa, (v) Mô hình PhoBERT, (vi) Thuật toán SVM.

### 2.1 Ngũ cảnh của bài toán

Bài toán Phân tích cảm xúc người dùng (Sentiment analysis) là một bài toán con của phân tích khía cạnh cảm xúc (ABSA-Aspect-Based Sentiment Analysis).

Trong những năm gần đây, bài toán này được đông đảo công đồng nghiên cứu xử lý ngôn ngữ tự nhiên (NLP) đặc biệt quan tâm. Bài toán xác định và phân loại văn bản thành các cảm xúc khác nhau như tích cực (positive), tiêu cực (negative), trung bình (neutral) hoặc là cảm xúc như vui, buồn, tức giận, v.v. để xác định cảm xúc con người đối với chủ đề hay thực thể cụ thể.

Bài toán phân tích cảm xúc thuộc cấp độ ngữ dụng học (Pragmatics) và ngữ nghĩa học (Semantics).

Phân tích cảm xúc cùng có ý nghĩa thiết yếu trong các ngành công nghệ - dịch vụ, nhằm nhận biết thái độ và cảm xúc của khách hàng về sản phẩm và dịch vụ họ đang dùng.

Hiện nay, bài toán này có 3 cấp độ lần lượt là cấp độ câu văn (sentence level), cấp độ văn bản (document level), cuối cùng là cấp độ khía cạnh. Đối với cấp độ câu văn, mục tiêu là phân loại một câu thành các lớp cơ bản như tích cực(positive), tiêu cực (negative), trung bình (neutral). Ở cấp độ văn bản thì có mục tiêu là xác định mức độ cảm xúc của 1 đoạn văn bản (gồm nhiều câu văn) thành các lớp cảm xúc. Cấp độ khía cạnh dùng để xác định mức độ cảm xúc cho mỗi khía cạnh của thực thể được đề cập trong một câu hay một văn bản. Trong phạm vi đồ án, em giới hạn nghiên cứu ở cấp độ câu văn.

### 2.2 Các kết quả nghiên cứu liên quan

Từ năm 2000 đến nay, bài toán phân tích cảm xúc được rất nhiều các nhân và tổ chức nghiên cứu, thực nghiệm và triển khai bài toán vào các ứng dụng thực tế, từ nước ngoài lật trong nước. Nghiên cứu đặt nền móng cho bài toán phân tích cảm xúc là nghiên cứu của Pang và công sự.

Những năm gần đây, các bài báo kèm mô hình về bài toán phân tích cảm xúc ngày càng nhiều có thể kể tới như: Vietnamese Students' Feedback Corpus (UIT-VSFC) [2], VLSP 2018 Shared Task: Aspect Based Sentiment Analysis [3].

Đầu tiên là Vietnamese Students' Feedback Corpus (UIT-VSFC), đây là đề xuất của Kiet Van Nguyen và các cộng sự, trong đề xuất này họ sử dụng mô hình Bi-LSTM kèm Word2Vec và mô hình Maximum Entropy classifier trên bộ dữ liệu 16.000 câu được chú thích bởi con người. Hai mô hình đều cho ra kết quả cao 0.84 và 0.92 F1 được biểu thị trong **Hình 2.1**.

Model	Topic (F1)	Sentiment (F1)
Bi-LSTM - Word2Vec	0.896	0.92
Maximum Entropy classifier	0.88	0.84

**Hình 2.1:** Kết quả của Vietnamese Students' Feedback Corpus

Tiếp theo là VLSP 2018 Shared Task: Aspect Based Sentiment Analysis, Ngo Xuan Bach và các cộng sự đã triển khai các mô hình SVM, CNNs trên bộ dữ liệu VLSP 2018 gồm hai lĩnh vực khách sạn và nhà hàng. **Hình 2.2** là kết quả mà nhóm của Ngo Xuan Bach đã triển khai.

Domain	Team	Phase A (Aspect)			Phase B (Aspect-Polarity)		
		Precision	Recall	F <sub>1</sub>	Precision	Recall	F <sub>1</sub>
Restaurant	SA1	0.75	0.85	<b>0.79</b>	0.63	0.71	<b>0.67</b>
	SA2						0.59
	SA3	0.78	0.65	0.71	0.71	0.59	0.64
Hotel	SA1	0.75	0.64	<b>0.69</b>	0.67	0.58	<b>0.62</b>
	SA2						0.56
	SA3	0.83	0.51	0.63	0.78	0.48	0.6

**Hình 2.2:** Kết quả của VLSP 2018 Shared Task

Ngoài hai đề xuất trên còn rất nhiều các xuất nghiên cứu và triển khai bài toán phân tích cảm xúc khác. Nhìn chung thì các đề xuất đề sử dụng những mô hình học máy để triển khai bài toán. Một số mô hình học máy vẫn chưa giải quyết triệt để một số vấn đề còn tồn đọng trong bài toán phân tích cảm xúc đối với các hệ thống mua sắm điện tử. Trong ĐATN này em sẽ hướng tới giải quyết các vấn đề còn tồn đọng trong bài toán phân tích cảm xúc trên trang thương mại điện tử.

### 2.3 Mô hình huấn luyện BERT

BERT [4] (Bidirectional Encoder Representations from Transformers) được hiểu là một mô hình học sẵn (pre-train model) học mối tương quan giữa các từ và học

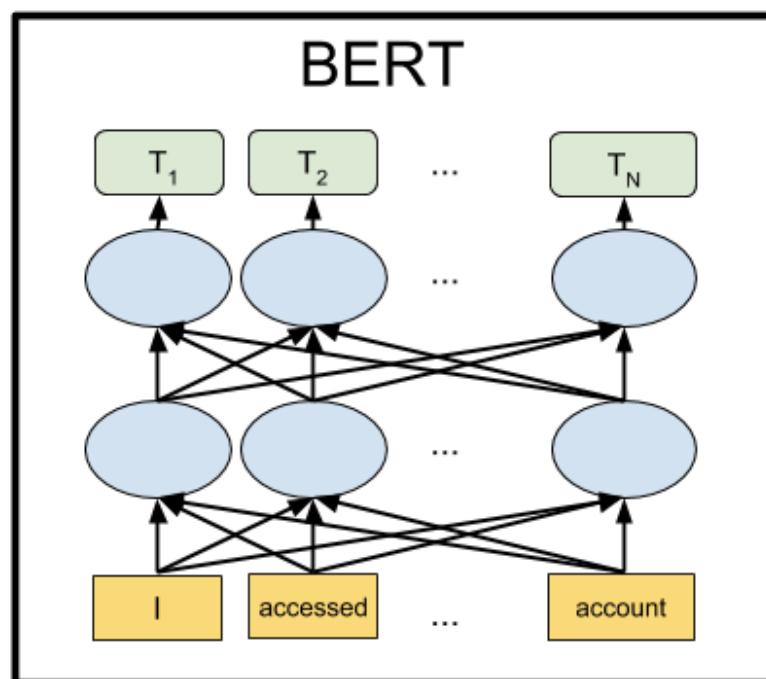
ra các vector đại diện theo ngữ cảnh 2 chiều của từ, được sử dụng để chuyển sang các bài toán khác trong lĩnh vực xử lý ngôn ngữ tự nhiên.

Mô hình BERT thông qua ngữ cảnh để tìm ra đại diện của từ trong không gian số (một không gian mà máy tính có thể hiểu được).

Mô hình BERT nhờ cách tạo ra các biểu diễn theo ngữ cảnh dựa vào từ đứng trước và đứng sau nó đã tạo ra một mô hình ngôn ngữ với ngữ nghĩa phong phú.

Transformer là một mô hình học sâu được thiết kế để phục vụ giải quyết nhiều bài toán trong xử lý ngôn ngữ và tiếng nói. Đặc biệt, Transformer không xử lý các phần tử trong một chuỗi một cách tuần tự. Transformer gồm có 2 phần chính: Encoder và Decoder, encoder thực hiện đọc dữ liệu đầu vào và decoder đưa ra dự đoán.

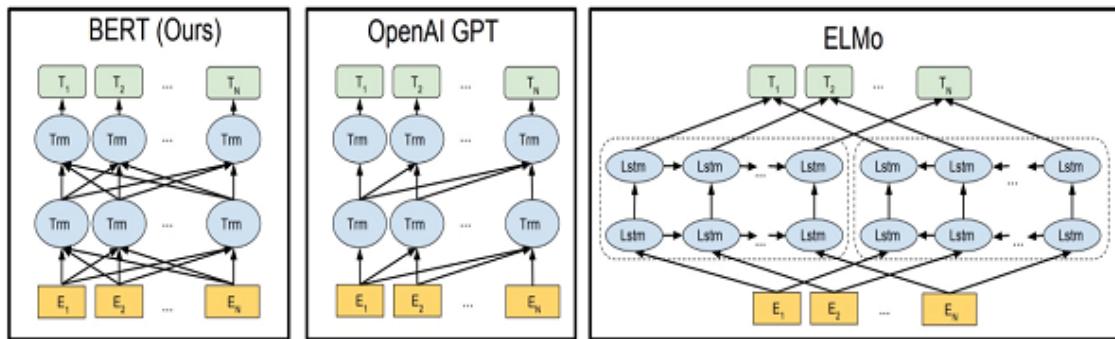
Kiến trúc của mô hình BERT là một kiến trúc đa tầng gồm nhiều lớp Bidirectional Transformer encoder. Trái ngược với các mô hình directional (các mô hình chỉ đọc dữ liệu theo 1 chiều duy nhất - trái → phải, phải → trái) đọc dữ liệu theo dạng tuần tự, Encoder đọc toàn bộ dữ liệu trong 1 lần, việc này làm cho BERT có khả năng huấn luyện dữ liệu theo cả hai chiều, qua đó mô hình có thể học được ngữ cảnh (context) của từ tốt hơn bằng cách sử dụng những từ xung quanh nó (phải và trái).



**Hình 2.3:** Kiến trúc mô hình BERT

Trong **Hình 2.4** là hình ảnh so sánh giữa BERT và hai mô hình là OpenAI GPT

và ELMo.



**Hình 2.4:** So sánh trực quan giữa BERT, OpenAI và ELMo

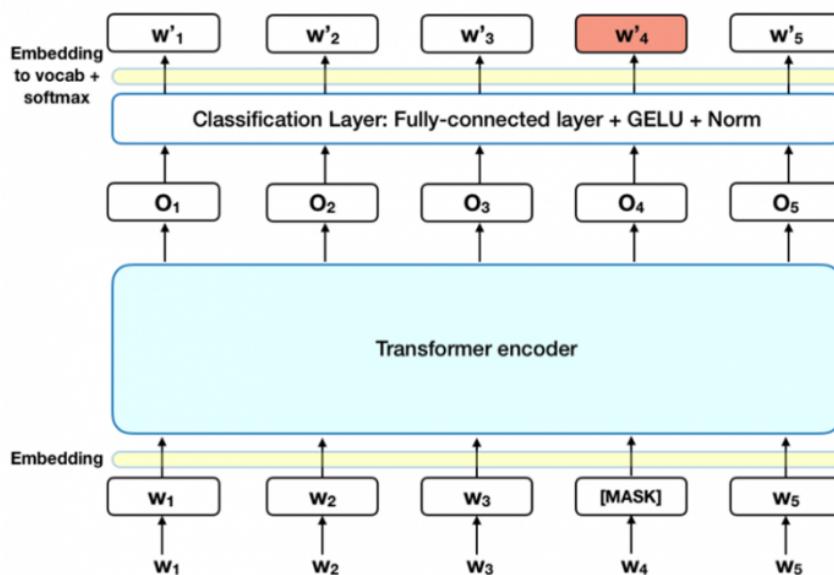
Mô hình BERT được tạo ra từ các tác vụ: Masked Language Model (MLM) và Next Sentence Prediction (dự đoán câu tiếp).

### 2.3.1 Masked Language Model (MLM)

Mô hình ngôn ngữ có mặt nạ (Masked Language Model-MLM) là một mẫu ngẫu nhiên của các mã thông báo (token) trong chuỗi đầu vào được chọn và thay thế bằng mã thông báo đặc biệt [MASK]. Sau đó chúng ta chỉ dự đoán các mã thông báo được giấu đi đó.

Mục tiêu MLM là một mất mát cross-entropy trong việc dự đoán các mã thông báo được che giấu.

Trong đó, việc tinh chỉnh là một quá trình sử dụng một mô hình mạng đã được huấn luyện cho một nhiệm vụ nhất định để thực hiện một nhiệm vụ tương tự.



**Hình 2.5:** Sơ đồ kiến trúc BERT cho tác vụ Masked Language Model (MLM) [5].

**Hình 2.5** là sơ đồ huấn luyện BERT theo tác vụ Masked Language Model (MLM).

Trước khi đưa vào BERT, thì 15% số từ trong chuỗi được thay thế ngẫu nhiên bởi token [MASK], khi đó mô hình sẽ dự đoán từ được thay thế bởi [MASK] với context là các từ không bị thay thế bởi [MASK]. Mask ML gồm các bước xử lý sau:

1. Thêm một lớp phân loại với đầu vào là đầu ra của Encoder.
2. Nhân các vector đầu ra với ma trận embedding để đưa chúng về không gian từ vựng (vocabulary dimensional).
3. Tính toán xác suất của mỗi từ trong tập từ vựng sử dụng hàm softmax.

Hàm lỗi (loss function) của BERT chỉ tập trung vào đánh giá các từ được đánh dấu [MASKED] mà bỏ qua những từ còn lại, do đó mô hình hội tụ chậm hơn so với các mô hình directional, nhưng chính điều này giúp cho mô hình hiểu ngữ cảnh tốt hơn.(Trên thực tế, con số 15% không phải là cố định mà có thể thay đổi theo mục đích của bài toán.)

### 2.3.2 Next Sentence Prediction (NSP)

Trong chiến lược này, thì mô hình sử dụng một cặp câu là dữ liệu đầu vào và dự đoán câu thứ 2 là câu tiếp theo của câu thứ 1 hay không. Trong quá trình huấn luyện, 50% lượng dữ liệu đầu vào là cặp câu trong đó câu thứ 2 thực sự là câu tiếp theo của câu thứ 1, 50% còn lại thì câu thứ 2 được chọn ngẫu nhiên từ tập dữ liệu.

Một số nguyên tắc được đưa ra khi xử lý dữ liệu như sau:

- Đánh dấu các vị trí đầu câu thứ nhất bằng token [CLS] và vị trí cuối các câu bằng token [SEP].
- Các token trong từng câu được đánh dấu là A hoặc B.
- Chèn thêm vector embedding biểu diễn vị trí của token trong câu (chi tiết về vector embedding này có thể tìm thấy trong bài báo về Transformer).

Các bước xử lý trong Next Sentence Prediction (NSP):

- Toàn bộ câu đầu vào được đưa vào Transformer.
- Chuyển vector đầu ra của [CLS] về kích thước 2x1 bằng một classification layer.
- Tính toán xác suất IsNextSequence bằng softmax.

**Hình 2.6** là một ví dụ đầu vào của mô hình BERT.

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	E <sub>[CLS]</sub>	E <sub>my</sub>	E <sub>dog</sub>	E <sub>is</sub>	E <sub>cute</sub>	E <sub>[SEP]</sub>	E <sub>he</sub>	E <sub>likes</sub>	E <sub>play</sub>	E <sub>##ing</sub>	E <sub>[SEP]</sub>
Segment Embeddings	E <sub>A</sub>	E <sub>A</sub>	E <sub>A</sub>	E <sub>A</sub>	E <sub>A</sub>	E <sub>A</sub>	E <sub>B</sub>	E <sub>B</sub>	E <sub>B</sub>	E <sub>B</sub>	E <sub>B</sub>
Position Embeddings	E <sub>0</sub>	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	E <sub>8</sub>	E <sub>9</sub>	E <sub>10</sub>

**Hình 2.6:** Ví dụ đầu vào của một mô hình BERT.

Từ **Hình 2.6** ta có các thông tin sau: Positional embeddings: vị trí token trong câu, tối đa 512 tokens. Token embeddings: các token của xâu đầu vào. Token đầu tiên là [CLS]. Token kết thúc câu là [SEP]. Trong task phân loại, đầu ra của Transformer (hidden state cuối cùng) ứng với token này là giá trị phân loại. Segment embeddings: phân biệt 2 câu trong trường hợp đầu vào là cặp câu, câu A là các giá trị 0, câu B là các giá trị 1.

## 2.4 Mô hình RoBERTa

Những năm gần đây, các mô hình ngôn ngữ được đào tạo trước đã rất phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên, một trong những mô hình đào tạo trước phổ biến nhất là BERT. Chính vì vậy Facebook đã công bố một dự án kế thừa lại kiến trúc và thuật toán của mô hình BERT có tên là RoBERTa [5] được giới thiệu giữa năm 2019.

Mô hình RoBERTa [5] sẽ huấn luyện lại BERT trên những bộ dữ liệu lớn cho các ngôn ngữ khác ngoài ngôn ngữ như tiếng anh.

Đã có nhiều mô hình đào tạo trước cho một số ngôn ngữ riêng huấn luyện trên RoBERTa. Đại diện có thể là RobBERT [6] cho ngôn ngữ Hà Lan.

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
<b>RoBERTa</b>						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	<b>94.6/89.4</b>	<b>90.2</b>	<b>96.4</b>
<b>BERT<sub>LARGE</sub></b>						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
<b>XLNet<sub>LARGE</sub></b>						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

**Hình 2.7:** Kết quả của mô hình RoBERTa.

**Hình 2.7** là kết quả của mô hình RoBERTa trên bộ dữ liệu lớn.

Trong bài báo tác giả cho biết RoBERTa lặp lại các tác vụ huấn luyện từ mô hình BERT, nhưng có thay đổi đó là huấn luyện mô hình lâu hơn, với batch size lớn hơn và trên nhiều dữ liệu hơn.

Ngoài ra để nâng cao độ chính xác trong biểu diễn từ thì RoBERTa đã loại bỏ tác vụ dự đoán câu tiếp theo (Next Sentence Prediction) và huấn luyện trên các câu văn dài hơn. Đồng thời mô hình cũng thay đổi linh hoạt kiểu masking (tức là ẩn đi một số từ ở câu đầu ta bằng token <mask>) áp dụng cho dữ liệu huấn luyện. Ngoài ra, RoBERTa còn làm tốt hơn BERT trong các tác vụ riêng lẻ theo chuẩn General Language Understanding Evaluation (GLUE).

## 2.5 Mô hình PhoBERT

Đầu tiên, PhoBERT [7] là một mô hình đào tạo trước được huấn luyện cho ngôn ngữ đơn ngữ, tức là mô hình chỉ huấn luyện dành riêng cho 1 ngôn ngữ duy nhất, ngôn ngữ mà mô hình PhoBERT hướng tới là tiếng Việt. Việc huấn luyện mô hình sẽ dựa trên kiến trúc và cách tiếp cận giống như RoBERTa của Facebook giới thiệu giữa năm 2019.

Mô hình PhoBERT cũng có 2 phiên bản là: PhoBERT base với 12 transformers block và PhoBERT large với 24 transformers block tương ứng với cùng một kiến trúc của BERT base và BERT large.

PhoBERT được huấn luyện trên khoảng 20GB dữ liệu bao gồm khoảng 1GB bộ Vietnamese Wikipedia corpus và 19GB còn lại lấy từ bộ Vietnamese news corpus bằng cách xóa các bài báo tương tự và trùng lặp khỏi kho dữ liệu. Đây là một lượng dữ liệu khá ổn để huấn luyện một mô hình.

Ngoài ra, mô hình PhoBERT sử dụng gói RDRSegmenter của VnCoreNLP để tách từ cho dữ liệu đầu vào trước khi qua BPE encoder.

POS tagging (word-level)		NER (word-level)		NLI (syllable- or word-level)	
Model	Acc.	Model	F <sub>1</sub>	Model	Acc.
RDRPOSTagger (Nguyen et al., 2014) [♣]	95.1	BiLSTM-CNN-CRF [♦]	88.3	BiLSTM-max (Conneau et al., 2018)	66.4
BiLSTM-CNN-CRF (Ma and Hovy, 2016) [♣]	95.4	VnCoreNLP-NER (Vu et al., 2018)	88.6	mBiLSTM (Artetxe and Schwenk, 2019)	72.0
VnCoreNLP-POS (Nguyen et al., 2017)	95.9	VNER (Nguyen et al., 2019b)	89.6	multilingual BERT (Wu and Dredze, 2019)	69.5
jPTDP-v2 (Nguyen and Verspoor, 2018) [★]	95.7	BiLSTM-CNN-CRF + ETNLP [♣]	91.1	XLM <sub>MLM+TLM</sub> (Conneau and Lample, 2019)	76.6
jointWPD (Nguyen, 2019)	96.0	VnCoreNLP-NER + ETNLP [♣]	91.3	XLM-R <sub>base</sub> (Conneau et al., 2020)	75.4
XLM-R <sub>large</sub> (our result)	96.3	XLM-R <sub>large</sub> (our result)	92.0	XLM-R <sub>large</sub> (Conneau et al., 2020)	79.7
PhoBERT-base	96.7	PhoBERT-base	93.6	PhoBERT-base	78.5
PhoBERT-large	<b>96.8</b>	PhoBERT-large	<b>94.7</b>	PhoBERT-large	<b>80.0</b>

**Hình 2.8:** Kết quả của mô hình PhoBERT.

### Hình 2.8 là kết quả của mô hình phoBERT.

Do tiếp cận theo tư tưởng của mô hình RoBERTa, vì vậy PhoBERT chỉ sử dụng Mô hình ngôn ngữ có mặt nạ (Masked Language Model) để huấn luyện và sẽ bỏ đi tác vụ Dự đoán câu tiếp theo (Next Sentence Prediction) trong quá trình huấn luyện.

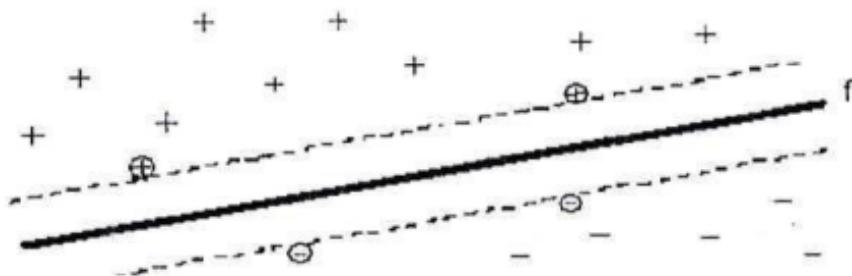
## 2.6 Thuật toán SVM

SVM [8] là viết tắt của cụm từ Support Vector Machine. Đây là một thuật toán khá hiệu quả trong lớp các bài toán phân loại nhị phân và dự báo của học có giám sát (supervised learning). Thuật toán này có ưu điểm là hoạt động tốt đối với những mẫu dữ liệu có kích thước lớn và thường mang lại kết quả vượt trội so với lớp các thuật toán khác trong học có giám sát. Phương pháp này thực hiện phân lớp dựa trên nguyên lý cực tiểu hóa rủi ro có cấu trúc SRM (Structural Risk Minimization).

### 2.6.1 Ý tưởng của phương pháp

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

Mục đích của phương pháp SVM là tìm được khoảng cách biên lớn nhất, điều này được minh họa trong :



**Hình 2.9:** Siêu phẳng phân chia dữ liệu học thành 2 lớp + và - với khoảng cách biên lớn nhất.

### 2.6.2 Nội dung của phương pháp

SVM thực chất là một thuật toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f trên không gian F sao cho sai số phân loại là thấp nhất.

Cho tập mẫu  $(x_1, y_1), (x_2, y_2), \dots (x_i, y_i)$  với  $x_i \in R^n$ , thuộc vào hai lớp nhãn:  $y_i \in \{-1, 1\}$  là nhãn lớp tương ứng của các  $x_i$  (-1 biểu thị lớp I, 1 biểu thị lớp II).

Ta có phương trình siêu phẳng chứa vectơ  $\vec{x}_i$  trong không gian:  $\vec{x}_i \cdot \vec{w} + b = 0$

$$\text{Đặt } f(\vec{X}_i) = \text{sign}(\vec{X}_i \cdot \vec{W}_i + b) = \begin{cases} +1 & \text{if } \vec{X}_i \cdot \vec{W}_i + b > 0 \\ -1 & \text{if } \vec{X}_i \cdot \vec{W}_i + b < 0 \end{cases}$$

Như vậy,  $f(\vec{X}_i)$  biểu diễn sự phân lớp của  $\vec{X}_i$  vào hai lớp như đã nêu. Ta nói  $y_i=+1$  nếu  $\vec{X}_i \in$  lớp I và  $y_i=-1$  nếu  $\vec{X}_i \in$  lớp II. Khi đó, để có siêu phẳng f ta sẽ phải giải bài toán sau:

Bài toán cần giải chính là ta đi tìm min  $\|\vec{W}\|$  với W thỏa mãn điều kiện sau:

$$y_i \cdot \text{sign}(\vec{X}_i \cdot \vec{W}_i + b) \geq 1 \forall i \in \overline{1, n}$$

Bài toán SVM có thể giải bằng kỹ thuật sử dụng hàm đối ngẫu Lagrange để biến đổi về thành dạng đẳng thức. Một đặc điểm thú vị của SVM là mặt phẳng quyết định chỉ phụ thuộc các Support Vector và nó có khoảng cách đến mặt phẳng quyết định là  $\frac{1}{\|\vec{W}\|}$ . Cho dù các điểm khác bị xóa đi thì thuật toán vẫn cho kết quả giống như ban đầu. Đây chính là điểm nổi bật của phương pháp SVM so với các phương pháp khác vì tất cả các dữ liệu trong tập huấn luyện đều được dùng để tối ưu hóa kết quả.

Tóm lại, trong trường hợp nhị phân - phân tách tuyến tính, việc phân lớp được thực hiện qua hàm quyết định  $f(\vec{X}_i) = \text{sign}(\vec{X}_i \cdot \vec{W}_i + b)$  hàm này thu được bằng việc thay đổi vectơ chuẩn w, đây là vectơ để cực đại hóa viền chức năng.

## Kết chương

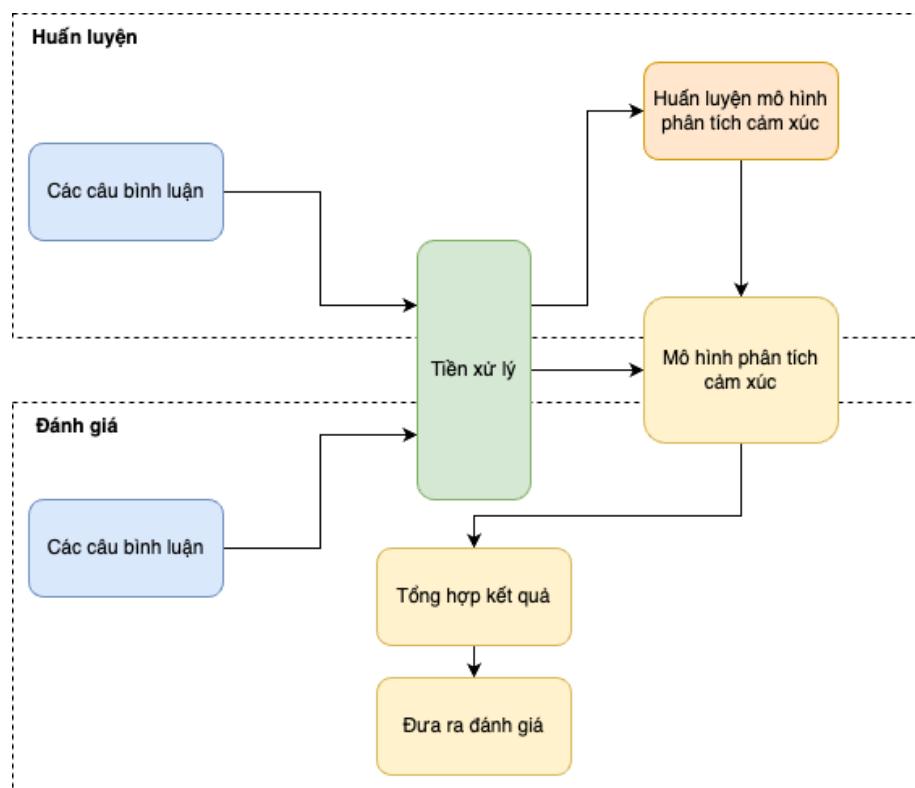
Chương này đã trình bày tổng quan về một số cơ sở lý thuyết được sử dụng trong đồ án. Bên cạnh đó Chương 2 còn nêu ra ngữ cảnh của bài toán và một số nghiên cứu nổi bật trong phân tích sắc thái. Từ những cơ sở lý thuyết này sẽ làm tiền đề cho phần đề xuất của đồ án trong chương tiếp theo - Chương 3.

## CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

Trong Chương 2 em đã trình bày về cơ sở lý thuyết của các phương pháp liên quan tới bài toán phân tích cảm xúc trong ĐATN. Trong Chương này em sẽ khái quát về giải pháp ĐATN hướng tới, cùng với những nội dung có trong giải pháp.

### 3.1 Tổng quan giải pháp

Quy trình thực hiện đánh giá sản phẩm theo mô hình phân tích cảm xúc được triển khai như sau:



**Hình 3.1:** Quy trình thực hiện đánh giá sản phẩm

Từ **Hình 3.1** em đưa ra giải pháp cho ĐATN này như sau:

Tại bước huấn luyện, dữ liệu là các câu văn mang tính phản hồi và đánh giá. Đầu tiên các câu văn này sẽ được đưa vào tiền xử lý dữ liệu, tách từ tiếng việt. Sau bước tiền xử lý dữ liệu và tách từ, các câu văn sẽ được đưa vào mô hình phân tích cảm xúc để huấn luyện mô hình. Trong ĐATN này, em hướng tới sử dụng hai mô hình cả học máy là SVM và học sâu là dùng PhoBERT.

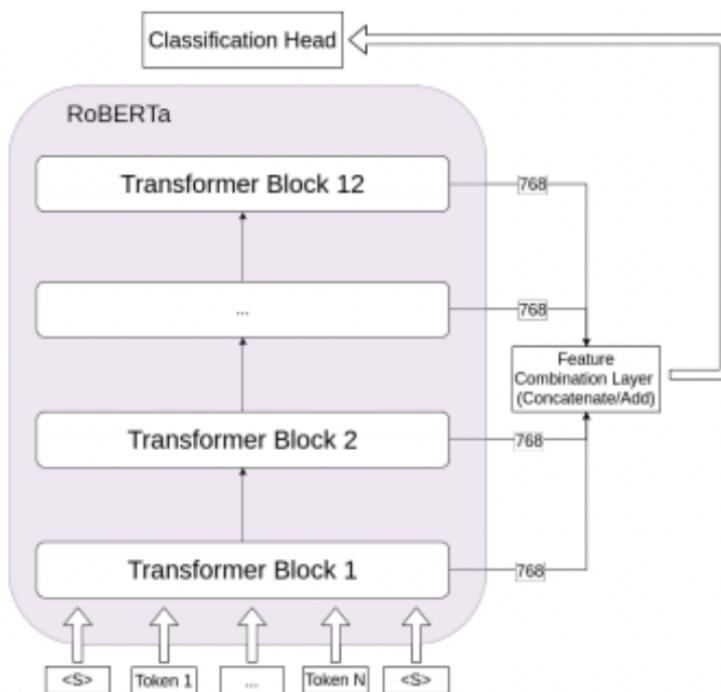
Tại bước dự đoán, các câu bình luận cũng được tiền xử lý dữ liệu, tiếp theo các câu văn sẽ được đưa qua mô hình phân tích cảm xúc đã được huấn luyện trước đó. Kết quả thu được sẽ tổng hợp theo các lớp tích cực (positive), trung bình (neutral), tiêu cực (negative). Cuối cùng là đưa ra đánh giá.

## 3.2 Mô hình đánh giá bằng PhoBERT

Kiến trúc PhoBERT như một mạng xương sống với một số sửa đổi. Đầu ra của mỗi khối Transformer thể hiện một mức ngữ nghĩa khác nhau cho các đầu vào. Trong các thử nghiệm của mình, em sử dụng các kết hợp đầu ra khác nhau của các khối Transformer đó. Mô hình kiến trúc chung được thể hiện trong **Hình 3.1**. Các tính năng được kết hợp trên các đầu ra của nhiều khối biến áp bằng cách ghép hoặc thêm vào được đưa vào đầu phân loại. Đầu phân loại đơn giản là một mạng được bảo vệ hoàn toàn không có các lớp ẩn.

### 3.2.1 Tinh chỉnh mô hình

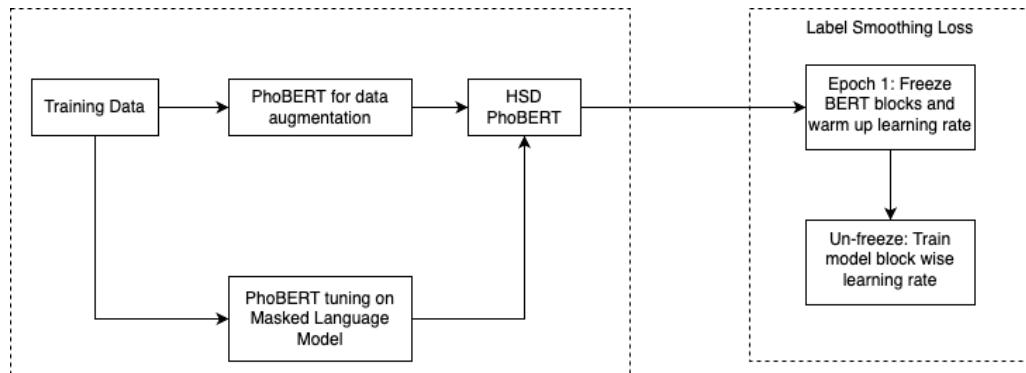
Mô hình được đào tạo trước phù hợp với tập dữ liệu lớn hơn nhiều của một miền hoàn toàn khác. Do đó, mặc dù nó có thể hoạt động rất tốt nói chung, nhưng mô hình vani được đào tạo trước sẽ hoạt động kém hiệu quả ở một nhiệm vụ cụ thể của chúng ta. Điều này đòi hỏi em phải điều chỉnh mô hình cho phù hợp với nhu cầu của mình. Do đó, với trọng số hiện có của PhoBERT như một điểm khởi đầu, em đào tạo mô hình của mình với dữ liệu đào tạo dành riêng cho miền của em về nhiệm vụ tạo mô hình ngôn ngữ được che giấu Masked Language Model (MLM).



**Hình 3.2:** Minh họa kiến trúc [9]

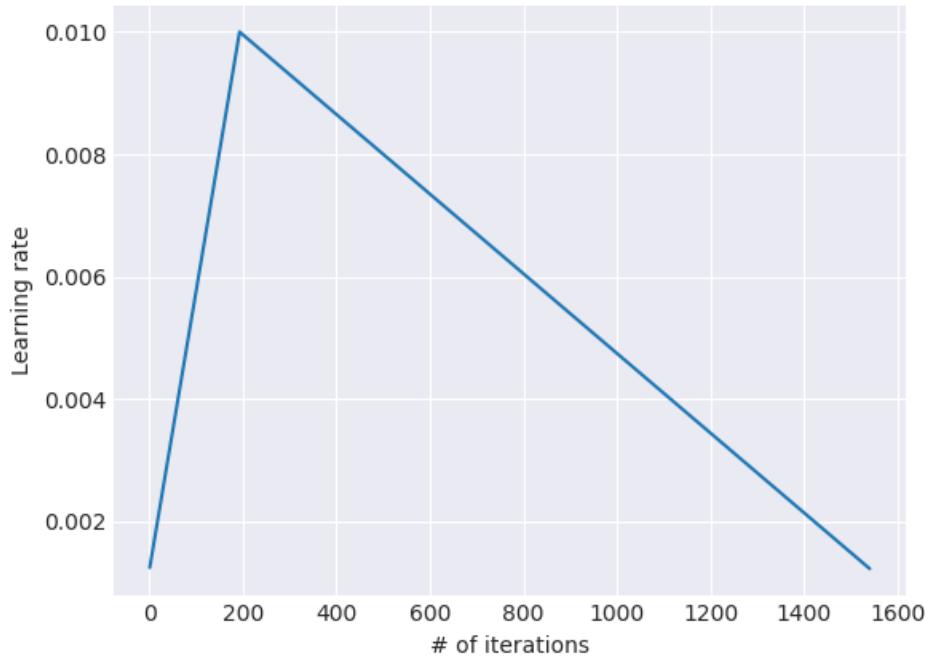
Theo hình trên, ta có đầu vào là một câu được mã hoá và cung cấp cho RoBERTa base, mỗi 1 khối Transformer Block sẽ tạo ra 1 vector 768-D. Các vector này sẽ nối lại với nhau thành 1 vector dài cho đầu phân loại.

**Hình 3.3** là chiến lược tinh chỉnh cho bài toán.



**Hình 3.3:** Chiến lược tinh chỉnh

Hơn nữa, để một mô hình lớn như vậy được đào tạo thành công mà không quên việc khởi tạo tốt của nó (do gradient giảm dần quá xa so với nó) hoặc không hội tụ (do các mô hình học sâu không tốt trong việc lan truyền qua các lớp xa hơn), em sẽ khởi động sơ đồ lập kế hoạch tỷ lệ học tập. Bắt nguồn từ báo cáo dưới tên "tỷ lệ học tập tam giác nghiêng" [10] trong **Hình 3.4**.



**Hình 3.4:** Tỷ lệ học tập tam giác nghiêng

Tỷ lệ học tập tam giác nghiêng (Slanted Triangular Learning Rates (STLR)) là một kịch bản học tập có tỷ lệ, trước tiên làm tăng tỷ lệ học tập một cách tuyến tính và sau đó giảm tuyến tính. Mục đích chính của phương pháp này là làm cho mô

hình hội tụ nhanh hơn cho một vùng thích hợp của không gian tham số trong quá trình bắt đầu đào tạo.

### 3.2.2 Chiến lược tối ưu hoá

Mỗi lớp trong mạng RoBERTa nắm bắt các mức ngữ cảnh khác nhau. Cụ thể, các lớp thấp hơn tạo ra các bản nhúng toàn cục cho các từ trong khi các bản nhúng từ các lớp trên thì không cụ thể hơn.

Do muôn bảo vệ toàn bộ kiến thức trong khi điều chỉnh các biểu diễn ngữ cảnh cho mô hình phân loại. Nên trong Epoch đầu tiên, em chỉ giữ lại các lớp được kết nối đầy đủ chịu trách nhiệm phân loại chuỗi văn bản và phần RoBERTa bị đóng băng. Điều này cho phép mô hình tìm hiểu một quyết định phù hợp cho nhiệm vụ. Sau các Epoch này, em giải phóng tất cả các lớp và đặt tốc độ học tập khác nhau cho các lớp khác nhau: lớp càng sâu, tỷ lệ học tập càng tăng. Điều này ngăn mô hình quên ý nghĩa toàn cục hữu ích của các từ trong khi buộc nó phải học ngữ cảnh của miền.

### 3.2.3 Chiến lược làm mịn nhãn

Để một mô hình lớn như vậy phù hợp với một tập dữ liệu tương đối nhỏ, mô hình có xu hướng trở nên mất tự tin về hiệu suất của nó, đi đến mặt tối của việc trang bị quá nhiều. Để tránh điều này, em sử dụng tính năng làm mịn nhãn, làm mềm các nhãn trung thực (one-hot ground truth labels). Cụ thể, đối với mô hình xác suất đầu ra  $y_k$  của K lớp, thay vì gắn nhãn sự thật cơ bản ta sẽ đi thay nhãn bằng mã hóa một nóng (one-hot):

$$y'_k = \begin{cases} 1 & \text{if } k = j \text{ for some } j \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

Cân bằng lại một chút phân phối mục tiêu để nó trở nên ít bị "phân quyền" hơn bằng cách làm mịn các xác suất với:

$$y'_k = y_k \cdot (1 - \alpha) + \alpha / K \quad (3.2)$$

Đối với một số tham số làm mịn  $\alpha$ . Kết quả là, kỹ thuật này dạy cho mô hình có một số sự không chắc chắn trong các dự đoán của nó và giảm mức độ nghiêm trọng của việc overfitting. Hơn nữa, vì đang tinh chỉnh trên một mô hình được đào tạo trước, vectơ xác suất đầu ra ban đầu của mô hình gần với một one-hot. Điều

này dẫn đến sự không ổn định về số nếu nhãn thực mới cũng là một one-hot, vì với entropy chéo (cross-entropy) là hàm mất mát, tổn thất sẽ trở thành  $1\log 0 = -\infty$ .

Do đó, đang được sử dụng ở đây, làm mịn nhãn đóng vai trò như một nhiễu loạn nhỏ trong phương pháp số, làm cho quá trình đào tạo ổn định hơn, giúp mô hình hội tụ tốt hơn.

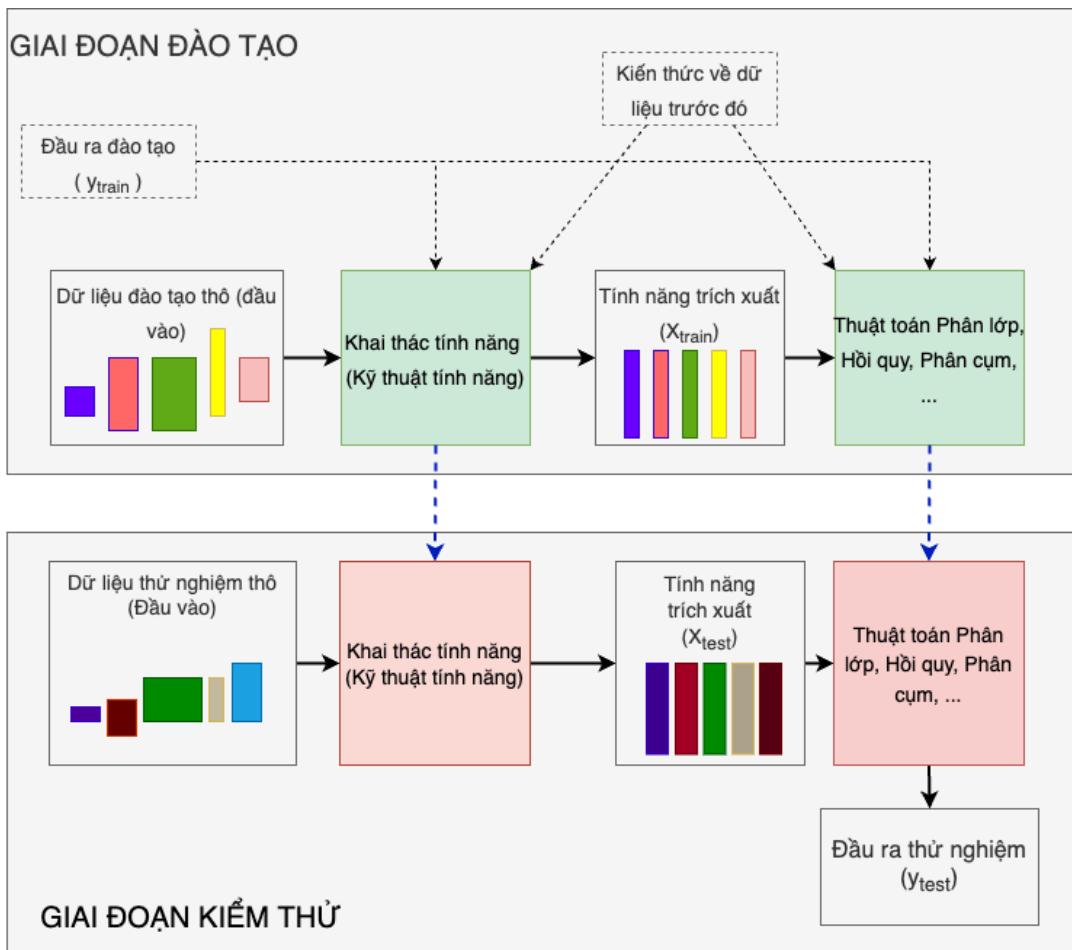
### 3.3 Mô hình đánh giá bằng SVM

Các phương pháp Support Vector Machine (SVM) như Hard Margin, Soft Margin, Kernel đều xây dựng cho bài toán Binary Classification (2 lớp). Để mở rộng các mô hình này áp dụng cho các bài toán multi-class classification, tức có nhiều classes dữ liệu khác nhau là sử dụng nhiều binary classifiers và các kỹ thuật như one-vs-one hoặc one-vs-rest. Trong ĐATN hướng tới mô hình end-to-end để giải quyết.

#### 3.3.1 Mô hình end-to-end

Softmax Regression là mở rộng của Logistic Regression cho bài toán multi-class classification, có thể được coi là một layer của Neural Networks. Các bộ phận lớp cho kết quả cao nhất thường là một Neural Network với rất nhiều layers và layer cuối là một softmax regression, đặc biệt là các Convolutional Neural Networks. Các layer trước thường là kết hợp của các Convolutional layers và các nonlinear activation functions và pooling, các layer trước layer cuối là một công cụ giúp trích chọn đặc trưng của dữ liệu (Feature extraction), layer cuối là softmax regression.

Sự hiệu quả của Softmax Regression nói riêng và Convolutional Neural Networks nói chung là cả bộ trích chọn đặc trưng (feature extractor) và bộ phân lớp (classifier) được huấn luyện đồng thời.



**Hình 3.5:** Mô hình chung cho các bài toán học máy.

Trong **Hình 3.5** là mô hình chung cho một bài toán học máy.

Theo hình trên thì giai đoạn đào tạo sẽ được chia làm 2 khối chính là trích xuất tính năng và phân loại / hồi quy / phân cụm. Trong ĐATN này khối trích xuất tính năng em sử dụng TF-IDF, còn phần còn lại là sử dụng hàm Kernel Linear để làm bộ phân lớp.

### 3.3.2 Hàm số Kernel Linear

Hàm số Kernel Linear là một trường hợp đơn giản của Kernel và bẳng tích vô hướng của hai vector:

$$k(x, z) = x^T \cdot z \quad (3.3)$$

### 3.3.3 Hàm mất mát của Multi-class Support Vector Machine

Hàm mất mát cho bài toán phân loại nhiều lớp bằng SVM như sau:

$$L(X, y, W) = \frac{1}{N} \sum_{n=1}^N \sum_{j \neq y_n} \max(0, 1 - W_{y_n}^T \cdot X_n + W_j^T \cdot X_n) + \frac{\lambda}{2} \cdot \|W\|_F^2 \quad (3.4)$$

Tiếp theo em sẽ tiến hành tối ưu hoá hàm mất mát bằng cách dùng Gradient Descent để tối ưu. Cách tính gradient như sau:

$$\frac{\partial}{\partial W_{y_n}} \max(0, 1 - W_{y_n}^T \cdot x_n + W_j^T \cdot x_n) = \begin{cases} 0 & \text{if } 1 - W_{y_n}^T \cdot x_n + W_j^T \cdot x_n < 0 \\ -x_n & \text{if } 1 - W_{y_n}^T \cdot x_n + W_j^T \cdot x_n > 0 \end{cases} \quad (3.5)$$

$$\frac{\partial}{\partial W_y} \max(0, 1 - W_{y_n}^T \cdot x_n + W_j^T \cdot x_n) = \begin{cases} 0 & \text{if } 1 - W_{y_n}^T \cdot x_n + W_j^T \cdot x_n < 0 \\ x_n & \text{if } 1 - W_{y_n}^T \cdot x_n + W_j^T \cdot x_n > 0 \end{cases} \quad (3.6)$$

**Kết chương** Chương 3 đưa ra đề xuất về giải pháp cho bài toán phân tích cảm xúc mà ĐATN hướng tới. Từ những đề xuất này chương tiếp theo sẽ trình bày về thực nghiệm của ĐATN - Chương 4.

## CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

Chương này là thực nghiệm cho các đề xuất ở Chương 3 nên trong chương này sẽ trình bày về các tham số đánh giá kết quả, phương pháp thực hiện thí nghiệm và các kết quả đạt được.

### 4.1 Các tham số đánh giá

Đối với bài toán phân loại sẽ có các cách đánh giá như: (i) ma trận hỗn loạn (confusion matrix), (ii) các độ đo cơ bản Accuracy, Precision and Recall, (iii) F1-score.

Khi thực hiện bài toán phân loại, có 4 trường hợp của dự đoán có thể xảy ra:

- True Positive (TP): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng)
- True Negative (TN): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng)
- False Positive (FP): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai) – Type I Error
- False Negative (FN): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai) – Type II Error

Bốn trường hợp trên thường được biểu diễn dưới dạng ma trận hỗn loạn (confusion matrix).

**Accuracy:** Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. Bên dưới là công thức tính của Accuracy:

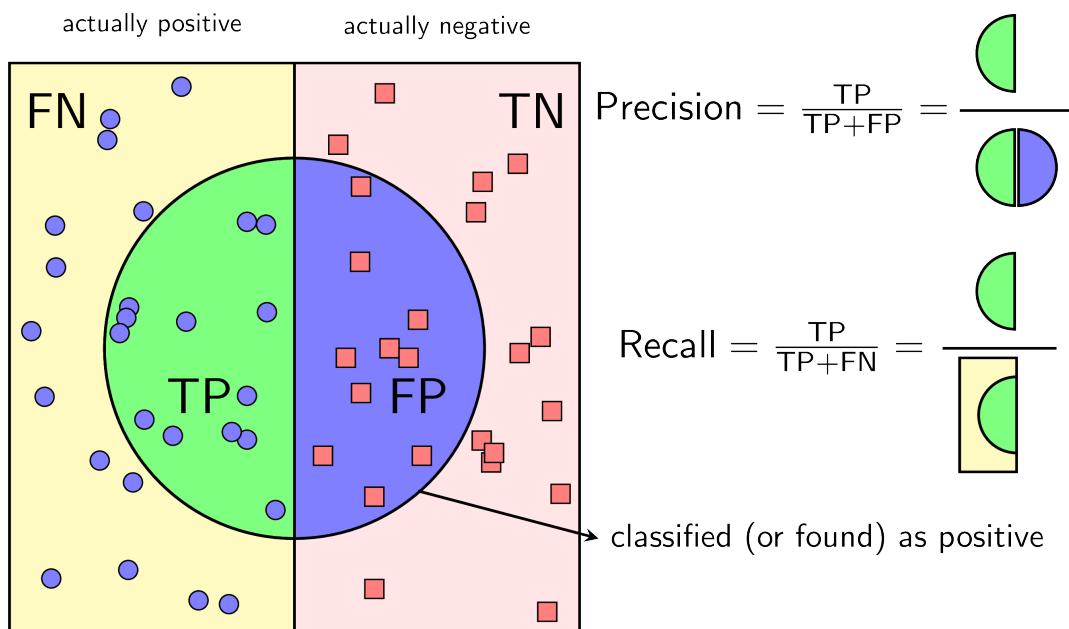
$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}} \quad (4.1)$$

**F1 score:** là harmonic mean của precision và recall. F1 score có giá trị nằm trong khoảng (0,1]. F1 càng cao, bộ phân lớp càng tốt. F1 score có công thức:

$$\frac{2}{F_1} = \frac{1}{\text{precision}} + \frac{1}{\text{recall}} \quad (4.2)$$

hay

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4.3)$$

**Hình 4.1:** Công thức Precision và Recall

**Precision:** được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive ( $TP + FP$ ).

**Recall:** được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive ( $TP + FN$ ).

## 4.2 Phương pháp thí nghiệm

Em tiến hành thí nghiệm theo quy trình như sau: (i) Thực hiện chuẩn bị dữ liệu, (ii) Chuẩn bị môi trường, (iii) Xử lý dữ liệu, (iv) Huấn luyện mô hình PhoBert, (v) Huấn luyện mô hình SVM.

### 4.2.1 Thực hiện chuẩn bị dữ liệu

Em thực hiện xây dựng một đoạn code bằng Selenium để thu thập những câu bình luận trên trang thế giới di động, sau khi thu thập được 3000 câu sẽ đánh 1 nhãn trong 3 nhãn POS(Positive), NEG (Negative) và NEU(Neutral) cho các câu bình luận.

- Cách cài đặt selenium : pip install selenium
- Thiết lập selenium trên chrome driver

Dưới đây là đoạn code thiết lập selenium cho chrome driver nhằm thiết lập cho việc thu thập dữ liệu:

```

from selenium import webdriver

def init_browser(type):
    if type == "chrome":
        # headless chrome
        options = webdriver.ChromeOptions()
        options.add_argument('headless')
        options.add_argument('--no-sandbox')
        options.add_argument('--disable-dev-shm-usage')
        options.add_argument('disable-web-security')
        options.add_argument('allow-running-insecure-content')
        options.add_argument('disable-extensions')
        options.add_argument('--ignore-certificate-errors')
        options.add_argument('--ignore-ssl-errors')
        options.add_argument('--log-level=3')
        options.add_argument('--silent')
        browser = webdriver.Chrome(executable_path='/usr/bin/chromedriver',options=options)
    else:
        raise Exception("Browser is not supported!")
    return browser

```

**Hình 4.2:** Code thiết lập để lấy dữ liệu bình luận trên trang thế giới di động

Một số ví dụ trong dữ liệu thu thập được:

" POS: sản phẩm thiết kế đẹp mỏng nhẹ nhỏ gọn , dễ cầm tay khi sử dụng , giá thành lại rẻ phù hợp cho mọi đối tượng , màu sắc thì rõ , bắt mắt , lướt web cũng nhanh nữa

NEG: quá tệ . sac gần 2 năm bị phồng pin lên rồi nứt ra . nguy hiểm cho người tiêu dùng

NEU: sao không xem được lịch âm nhỉ có ai bí cách xem lịch âm không xem mại không bit cài ntn "

Sử dụng bộ dữ liệu VLSP 2016 [2] cho bài toán: "Sentiment Analysis for Vietnamese Language" gồm 3 nhãn: Positive( 2050), Negative(2050), Neutral(2050).

Một số ví dụ trong bộ dữ liệu VLSP 2016:

" Pos: Đẳng cấp Philips, máy đẹp, pin bền. Đóng và giao hàng rất chuyên nghiệp

Pos: Tốt Giá vừa túi tiền đẹp và sang

Neg: Lâu lâu bị lỗi, màn hình cảm ứng không nhạy, chất lượng camera kém.

Neg: pin nhanh tụt, chỉ được xài 1 ngày.

Neu: Pin khá hơn tí thì tốt nhỉ... "

Cuối cùng em tiến hành gộp bộ dữ liệu lại thành một bộ dữ liệu lớn cho bài toán của em.

#### 4.2.2 Chuẩn bị môi trường

Mô hình được xây dựng và thực nghiệm trên Google Colab có cấu hình như sau:

- Hệ điều hành: Ubuntu 18.04.5.
- GPU Tesla T4.
- CUDA.
- RAM: 15GB.

Trong quá trình training các mô hình học máy và học sâu. Có sử dụng một số các kỹ thuật để trích chọn đặc trưng cũng như tăng độ chính xác của mô hình được thể hiện trong bảng bên dưới:

Method	Kỹ thuật sử dụng	Một số tham số
SVM	TF-IDF Grid Search Hyperparameter Tuning 10-Fold Cross Validation	Penalty = l2 loss= squared_hinge
PhoBERT	Early stopping VinAI tokenizer Focal loss Dropout	Learning rate: 5e-6 Batch size: 8 Optimizer: Adam Epoches: 20 Patience: 3(number of epochs to wait after last time validation loss improved)

**Bảng 4.1:** Kỹ thuật sử dụng và tham số

### 4.2.3 Tiền xử lý dữ liệu

Trước khi đưa vào mô hình để huấn luyện, dữ liệu sẽ đưa qua các bước tiền xử lý để chuẩn hoá dữ liệu:

- Loại bỏ các ký tự đặc biệt và thay bằng dấu cách
- Thay các emoji, emoticon bằng từ mang nghĩa của nó
- Áp dụng từ điển để đồng nhất cách đặt dấu (“òa” => “oà”)
- Áp dụng từ điển các từ viết tắt có ý nghĩa cảm xúc (từ điển tự tạo) (“hjhj” => “hihi”)
- Tách từ bằng thư viện pyvi
- Cuối cùng, dữ liệu được áp dụng các kỹ thuật trích chọn đặc trưng như: vectorizer, tfidf transformer, tokenizer... tùy theo mô hình trước khi tới bước tiếp theo

#### 4.2.4 Huấn luyện mô hình PhoBERT

Sau khi đã chuẩn bị dữ liệu và tiền xử lý dữ liệu xong, em tiến hành huấn luyện mô hình PhoBERT trên tập dữ liệu đã chuẩn bị. Trước khi huấn luyện em tiến hành cài đặt và sử dụng một số thư viện cho quá trình huấn luyện như:

- transformers: pip install transformers
- torch: pip install torch
- sklearn: pip install sklearn
- argparse: pip install argparse

Sau khi cài đặt xong sẽ xác định các đối số đầu vào cho mô hình bằng *argparse* được thể hiện trong đoạn code dưới đây:

```
parser = argparse.ArgumentParser()
parser.add_argument('--weights', type=str, default='phoBERT', help='choose [phoBERT]')
parser.add_argument('--epochs', type=int, default=20)
parser.add_argument('--batch_size', type=int, default=8, help='total batch size for all GPUs')
parser.add_argument('--device', default='0', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
parser.add_argument('--data', type=str, default='datasets', help='path to data folder')
parser.add_argument('--optimizer', type=str, default='AdamW', help='choose Optimizer [AdamW]')
parser.add_argument('--lr', type=float, default=5e-6)
parser.add_argument('--iteration', type=int, default=200)
parser.add_argument('--patience', type=int, default=3, help='how long( num epochs) to wait after last time validation loss improved.')
parser.add_argument('--checkpoint', type=str,
                    default='/content/drive/MyDrive/Colab Notebooks/sentiment_analysis_vietnamese/checkpoints/phoBERT/phoBERT_best.pth',
                    help='path to checkpoint')
parser.add_argument('--checkpoints', type=str, default='checkpoints', help=' Path to save checkpoints dir')

opt = parser.parse_args()
configs = dict()
configs["weights"] = opt.weights
configs["epochs"] = opt.epochs
configs["batch_size"] = opt.batch_size
configs["optimizer"] = opt.optimizer
configs["lr"] = opt.lr
```

**Hình 4.3:** Code cho thiết lập đầu vào của mô hình PhoBERT

Mô hình được huấn luyện trên bộ dữ liệu là 3 lần. Theo như thiết lập cho mô hình thì mỗi lần chạy tối đa số epoch là 20, nhưng mô hình đạt giá trị sớm nên sau khi huấn luyện được từ 4 epoch đến 6 epoch thì mô hình đã dừng lại.

#### 4.2.5 Huấn luyện mô hình với SVM

Trong phần 3.3.1 em đã trình bày là sẽ sử dụng TF-IDF cho khôi Feature Extraction và sử dụng Kernel Linear cho bộ phân lớp. Ngoài ra có thể thấy các kỹ thuật em sử dụng để huấn luyện mô hình được thể hiện trong **Bảng 4.1**.

Từ ý tưởng trên em thực hiện xây dựng mã nguồn cho ý tưởng như sau:

```
#THÊM STOPWORD LÀ NHỮNG TỪ KÉM QUAN TRỌNG
stop_ws = (u'rằng', u'thì', u'là', u'mà', u'lô')
self.experiment = os.path.join('checkpoints', 'SVM')
l_svm = Pipeline([('vect', CountVectorizer(ngram_range=(1,5), stop_words=stop_ws, max_df=0.5, min_df=5)),
                  ('tfidf', TfidfTransformer(use_idf=False, sublinear_tf = True, norm='l2', smooth_idf=True)),
                  ('clf', LinearSVC())])
tuned_parameters = {
    'vect_ngram_range': [(1, 2), (1, 3), (1, 4)],
    'tfidf_use_idf': (True, False),
    'clf_tol': [1, 1e-1, 1e-2, 1e-3]
}
self.l_svm = GridSearchCV(l_svm, tuned_parameters, cv=10, verbose=self.verbose, n_jobs=2)
```

**Hình 4.4:** Code thiết lập cho quá trình huấn luyện mô hình SVM

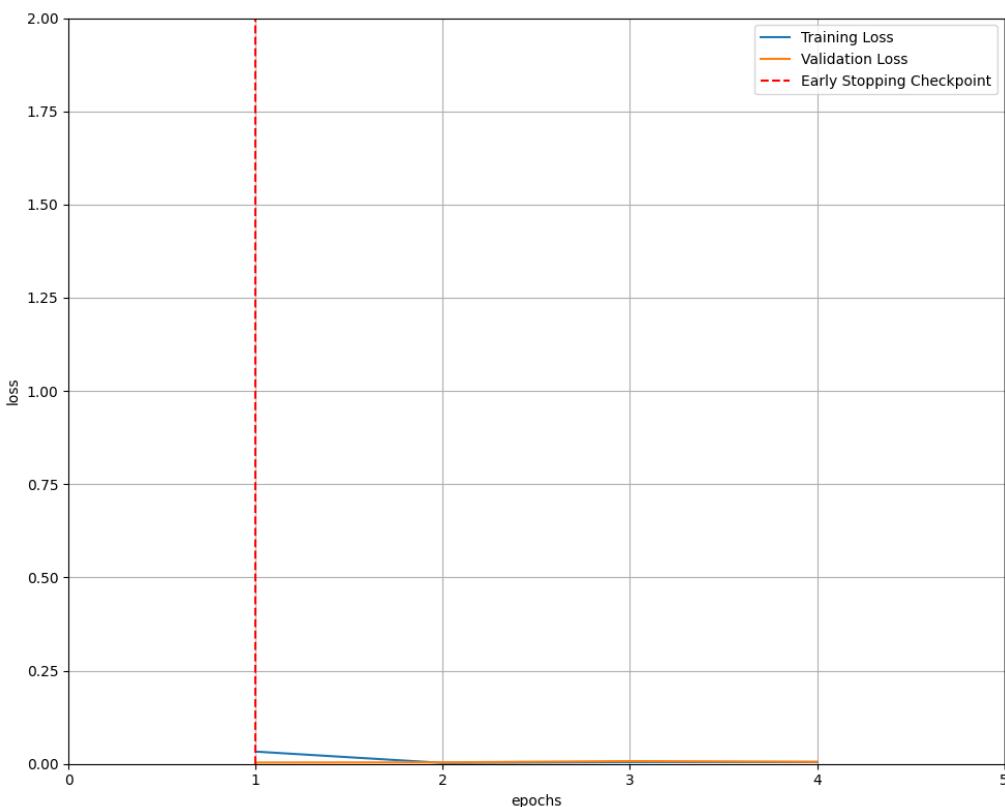
### 4.3 Kết quả của mô hình PhoBERT

Đối với mô hình PhoBERT trên bộ dữ liệu huấn luyện có thời gian huấn luyện và thời gian kiểm tra mô hình như sau:

Method	Training Time	Testing Time	Class	F1 Score	Accuracy
PhoBERT	2810s	15s	Positive	97%	94%
			Negative	97%	
			Neutral	81%	

**Bảng 4.2:** Bảng kết quả của PhoBERT

Từ bảng kết quả trên ta thấy thời gian huấn luyện và thời gian thử nghiệm của PhoBERT khá lâu. Mà đối với bài toán thì ta chú trọng hơn về độ chính xác hơn kết quả. Mô hình PhoBERT cho ta độ chính xác tính bằng F1 score của các lớp cũng ổn định và điểm accuracy và F1 score cho toàn bộ mô hình cũng đạt mức 96% (0.9599561162918266). Tiếp theo là hình ảnh về điểm dừng (Early stopping) khi huấn luyện PhoBERT.

**Hình 4.5:** Early stoping khi training PhoBERT

#### 4.4 Kết quả huấn luyện mô hình SVM

Đối với mô hình huấn luyện sử dụng SVM thì ta có kết quả đánh giá trên Accuracy và F1 Score như sau:

Method	Training time	Testing Time	F1 Score	Accuracy
SVM	200s	0.1s	84%	83%

**Bảng 4.3:** Bảng kết quả của SVM

Từ kết quả ở **Bảng 4.3** ta thấy thời gian huấn luyện và thử nghiệm của một mô hình học máy rất nhanh. Đặc biệt mô hình với SVM (thuật toán phân lớp) cũng cao với lần lượt các tham số đánh giá như F1 đạt 84% trên bộ thử nghiệm và Accuracy là 83%. Đây cũng là một mô hình phân loại tốt.

#### 4.5 So sánh hai mô hình

Từ bảng 4.3 và bảng 4.2 ta nhận thấy thời gian huấn luyện và thời gian thử nghiệm của mô hình sử dụng SVM nhanh hơn so với mô hình bằng PhoBERT. Nhưng đối với một mô hình thì độ chính xác của mô hình mới là điều đánh chú ý. Ở đây ta thấy mô hình PhoBERT cho ta độ chính xác tốt hơn so với mô hình bằng SVM.

## Kết chương

Chương này đã đưa ra các phương pháp đánh giá cùng với các kết quả mà ĐATN đạt được. Trong chương tiếp theo em sẽ trình bày chi tiết về hệ thống đánh giá sản phẩm trên trang thương mại điện tử - Chương 5.

## **CHƯƠNG 5. PHÁT TRIỂN HỆ THỐNG TRANG WEB THU THẬP VÀ ĐÁNH GIÁ SẢN PHẨM**

Chương 5 tập trung trình bày về kiến trúc tổng quan của hệ thống thu thập và đánh giá sản phẩm trên các trang thương mại điện tử và những nội dung liên quan đến xây dựng, triển khai hệ thống bao gồm: chức năng quản lý sản phẩm, chức năng thu thập sản phẩm từ trang thương mại điện tử.

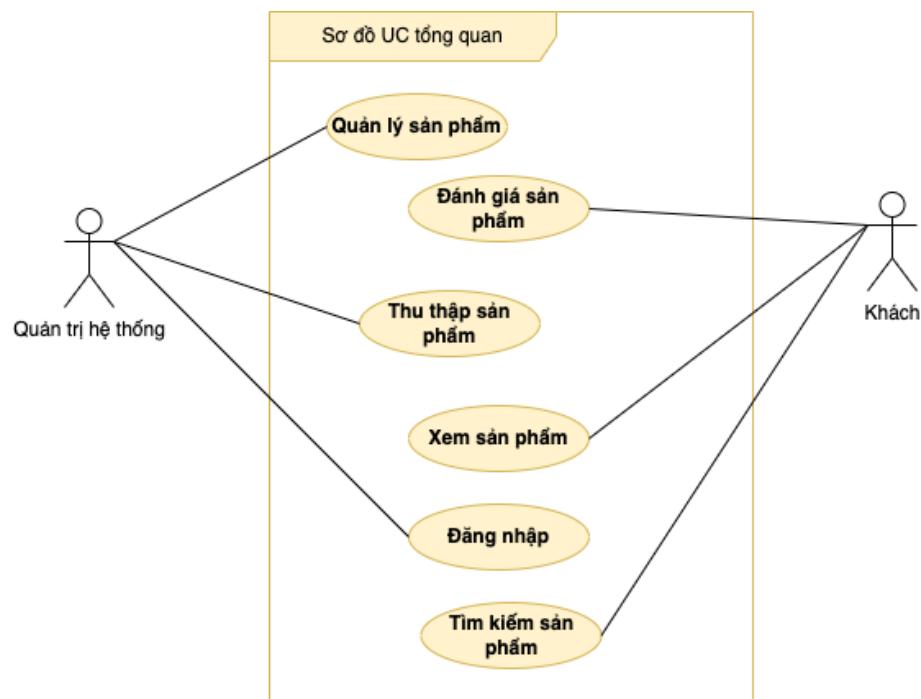
### **5.1 Phân tích yêu cầu**

#### **5.1.1 Tổng quan chức năng**

Hệ thống thu thập và đánh giá sản phẩm trên các trang thương mại điện tử gồm nhiều thành phần và chức năng khác nhau. **Hình 5.1** dưới đây mô tả tổng quan chức năng của toàn bộ hệ thống.

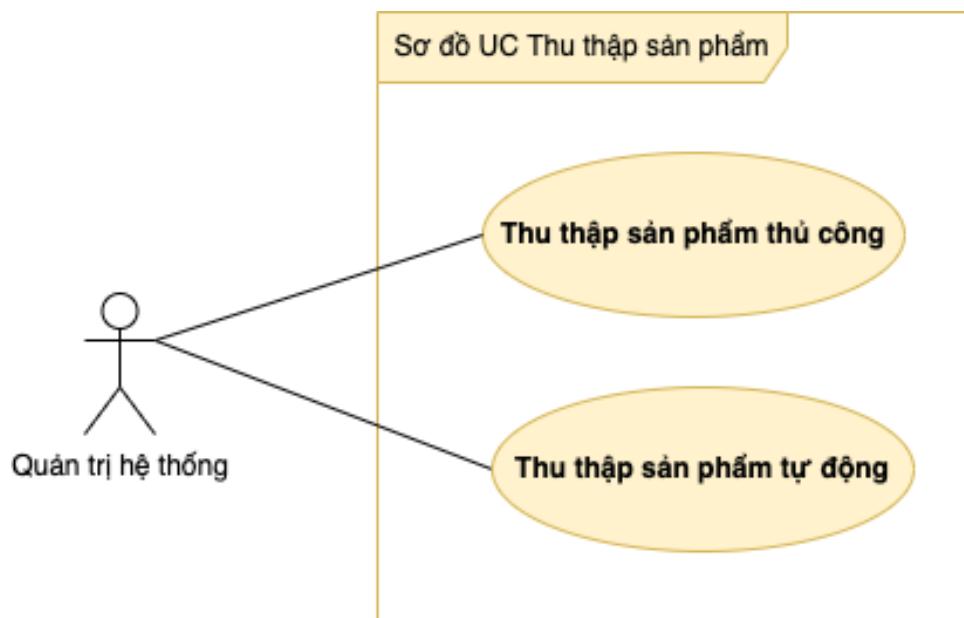
Hệ thống gồm 2 tác nhân, đó là: Quản trị hệ thống và Khách, mỗi tác nhân có một vai trò nhất định. Tuy nhiên, tuỳ mục đích quản lý mà chức năng đăng nhập và thu thập sản phẩm bị ẩn đổi với tác nhân Khách.

- Khách có thể truy cập hệ thống nhằm mục đích đóng góp sản phẩm thông qua chức năng đánh giá sản phẩm bằng link của sản phẩm trên các trang như thế giới di động. Bên cạnh đó, chức năng tìm kiếm sản phẩm giúp khách hàng có thể tìm kiếm các sản phẩm đã được đánh giá trước đó trên hệ thống. Ngoài ra, khách có thể xem toàn bộ đánh giá của sản phẩm trên các trang thương mại khác nhau.
- Quản trị hệ thống là người có quyền cao nhất. Quản trị hệ thống sau khi đăng nhập vào hệ thống thì có thể thu thập sản phẩm từ các trang thương mại điện tử, có thể đặt lệnh để tự động thu thập sản phẩm sau một khoảng thời gian nào đó, hoặc có thể đặt lệnh thủ công để thu thập sản phẩm.

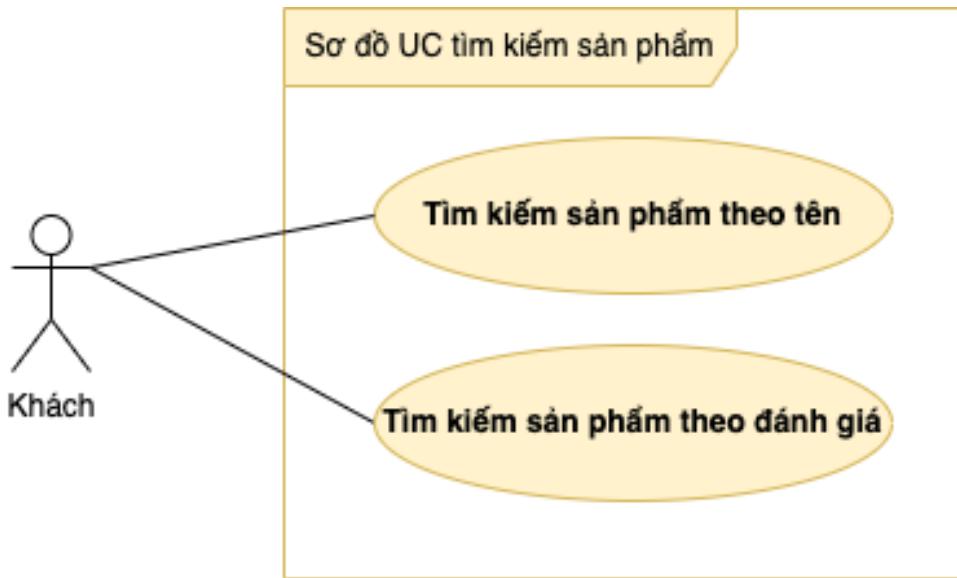
**Hình 5.1:** Biểu đồ use case tổng quan của hệ thống

Sau đây là biểu đồ use case phân rã của một số chức năng chính như: (i) Thu thập sản phẩm, (ii) Tìm kiếm sản phẩm.

**Hình 5.2** là biểu đồ phân rã cho use case thu thập sản phẩm. Với chức năng này, Quản lý hệ thống có thể chọn 1 trong 2 lệnh thu thập sản phẩm đó là: thu thập tự động và thu thập thủ công. Điều đó giúp Quản lý hệ thống dễ dàng thực hiện việc thu thập sản phẩm.

**Hình 5.2:** Biểu đồ use case thu thập sản phẩm

**Hình 5.3** là biểu đồ phân rã cho use case tìm kiếm sản phẩm. Với chức năng này, Khách có thể tìm theo nhiều cách như tìm kiếm theo tên, tìm kiếm theo đánh giá. Điều đó giúp Khách dễ dàng tìm thấy được những sản phẩm phù hợp với điều họ muốn.



**Hình 5.3:** Biểu đồ use case tìm kiếm sản phẩm

### 5.1.2 Đặc tả chức năng

Phần này sẽ đặc tả một số chức năng trong hệ thống thu thập và đánh giá sản phẩm trên các trang thương mại điện tử. **Bảng 5.1** liệt kê số use case được sử dụng. Do kích thước báo cáo có hạn nên em xin phép trình bày về 3 use case chính: (i) Thu thập sản phẩm tự động, (ii) Đánh giá sản phẩm và (iii) Tìm kiếm sản phẩm theo đánh giá.

Mã usecase	Tên usecase
UC001	Đăng nhập
UC002	Quản lý sản phẩm
UC003	Thu thập sản phẩm thủ công
<b>UC004</b>	<b>Thu thập sản phẩm tự động</b>
<b>UC005</b>	<b>Đánh giá sản phẩm</b>
UC006	Xem sản phẩm
UC007	Tìm kiếm sản phẩm theo tên
<b>UC008</b>	<b>Tìm kiếm sản phẩm theo đánh giá</b>

**Bảng 5.1:** Danh sách use case

Đặc tả usecase UC004 được mô tả trong **Bảng 5.2**. Đây là chức năng giúp Quản trị hệ thống có thể thu thập sản phẩm từ các trang thương mại điện tử vào hệ thống một cách tự động thay vì thu thập tay thủ công.

Mã usecase	UC004	Tên usecase	Thu thập sản phẩm tự động
Tác nhân	Quản trị hệ thống		
Tiền điều kiện	Quản trị hệ thống sau khi đăng nhập vào hệ thống và truy cập vào màn hình quản lý sản phẩm		
Luồng xử lý chính	STT	Thực hiện bởi	Hành động
	1	Quản trị viên	Nhấn nút có biểu tượng răng cưa trên màn hình
	2	Hệ thống	Mở giao diện setting
	3	Quản trị viên	Chọn thời gian thu thập (ngày, tuần, tháng) và nhấn nút 'Crawl Product'
	4	Hệ thống	Đếm ngược thời gian
	5	Hệ thống	Thực hiện thu thập dữ liệu
	6	Hệ thống	Thông báo thu thập thành công và cập nhật lại sản phẩm
Luồng xử lý thay thế	STT	Thực hiện bởi	Hành động
	3.a	Quản trị viên	Không nhập đúng thông tin cần thiết (ngày, tuần, tháng) và nhấn nút 'Crawl Product'
Hậu điều kiện	4.a	Hệ thống	Thông báo warning 'Cần phải nhập thời gian thích hợp.'
	Không		

**Bảng 5.2:** Đặc tả usecase "Thu thập sản phẩm tự động"

Đặc tả usecase UC005 được mô tả trong **Bảng 5.3**. Đây là chức năng giúp Khách có thể đóng góp sản phẩm cho hệ thống và có thể nhanh chóng xác định được sản phẩm mình đang muốn mua được đánh giá tốt không.

Mã usecase	UC005	Tên usecase	Đánh giá sản phẩm
<b>Tác nhân</b>	Khách		
<b>Tiền điều kiện</b>	Khách sau khi truy cập vào hệ thống		
<b>Luồng xử lý chính</b>	STT	Thực hiện bởi	Hành động
	1	Khách	Nhấn nút "Đánh giá sản phẩm" trên giao diện
	2	Hệ thống	Mở modal đánh giá sản phẩm (gồm nhập link và chọn nơi bán)
	3	Khách	Nhập link sản phẩm, chọn nơi bán và nhấn nút "Đánh giá"
	4	Hệ thống	Thực hiện thu thập dữ liệu và đánh giá
	5	Hệ thống	Chuyển tới giao diện hiển thị sản phẩm
<b>Luồng xử lý thay thế</b>	STT	Thực hiện bởi	Hành động
	3.a	Khách	Không nhập đúng thông tin cần thiết và nhấn nút "Đánh giá"
	4.a	Hệ thống	Thông báo warning 'Cần phải nhập đúng các thông tin cần thiết'
<b>Hậu điều kiện</b>	Không		

**Bảng 5.3:** Đặc tả usecase "Đánh giá sản phẩm"

Đặc tả usecase UC008 được mô tả trong **Bảng 5.4**. Đây là chức năng giúp Khách có thể tìm kiếm các sản phẩm trong hệ thống theo yêu cầu của Khách như các sản phẩm đánh giá tốt một cách nhanh chóng.

Mã usecase	UC008		Tên usecase	Tìm kiếm sản phẩm theo đánh giá
Tác nhân	Khách			
Tiền điều kiện	Khách sau khi truy cập vào hệ thống			
Luồng xử lý chính	STT	Thực hiện bởi	Hành động	
	1	Khách	Chọn "Tìm kiếm sản phẩm" và chọn 1 trong 3 đánh giá (Tốt, Trung bình, Không tốt) trên sidebar của giao diện	
	2	Hệ thống	Tìm kiếm sản phẩm theo yêu cầu	
Luồng xử lý thay thế	STT	Thực hiện bởi	Hành động	
	3.a	Hệ thống	Thông báo "Không có sản phẩm nào theo yêu cầu."	
Hậu điều kiện	Không			

**Bảng 5.4:** Đặc tả usecase "Tìm kiếm sản phẩm theo đánh giá"

### 5.1.3 Yêu cầu phi chức năng

#### 1. Tính dễ dùng.

Do hệ thống hướng tới tất cả người dùng gồm cả những người có ít kinh nghiệm về máy tính và công nghệ nên hệ thống xây dựng giao diện một cách đơn giản, dễ dùng, đặc biệt là hạn chế các thao tác không cần thiết. Các kí hiệu sử dụng trong hệ thống là kí hiệu được sử dụng phổ biến, dễ hiểu, dễ dùng.

#### 2. Tính dễ bảo trì.

Do hệ thống đang trong quá trình xây dựng và phát triển nên rất cần thiết kế để dễ dàng chỉnh sửa và nâng cấp theo yêu cầu từ phía người dùng.

Hệ thống cần thiết kế tốt để đảm bảo khi xây dựng tính năng mới sẽ không ảnh hưởng tới các tính năng ổn định hiện có.

#### 3. Tính khả thi.

Đối với tính năng đánh giá sản phẩm, hệ thống cần đảm bảo thời gian đưa ra kết quả từ 10 giây đến 20 giây, không để người dùng chờ đợi lâu dẫn tới trải nghiệm không tốt.

## 5.2 Công nghệ sử dụng

### 5.2.1 FrontEnd

#### a) ReactJS

ReactJS [1] là một thư viện JavaScript mã nguồn mở được thiết kế bởi Facebook để tạo ra những ứng dụng web Single Page Application hấp dẫn, nhanh và hiệu quả với mã hóa tối thiểu. Mục đích cốt lõi của ReactJS không chỉ khiến cho trang web trải nghiệm mượt mà mà còn phải nhanh, khả năng mở rộng cao và đơn giản.

Điểm mạnh của ReactJS xuất phát từ việc chia một bộ cục lớn thành các thành phần riêng lẻ (component). Chính vì vậy, thay vì làm việc trên toàn bộ ứng dụng web, ReactJS cho phép một developer có thể chuyển đổi giao diện người dùng phức tạp thành các thành phần đơn giản hơn.

#### Đặc trưng của ReactJS

**Single-way data flow (Luồng dữ liệu một chiều):** Những Framework sử dụng Virtual-DOM như ReactJS khi Virtual-DOM thay đổi, chúng ta không cần thao tác trực tiếp với DOM trên View mà vẫn thấy được sự thay đổi đó. Do Virtual-DOM ngoài đóng vai trò là Model, còn đóng vai trò là View nên mọi sự thay đổi trên Model đã kéo theo sự thay đổi trên View và ngược lại. Có nghĩa là mặc dù chúng ta không tác động trực tiếp vào các phần tử DOM ở View nhưng vẫn thực hiện được cơ chế Data-binding. Điều này làm cho tốc độ ứng dụng tăng lên đáng kể – một lợi thế khi sử dụng Virtual-DOM.

#### b) Ant Design

Ant Design là tập hợp các components của React được xây dựng theo chuẩn thiết kế của Ant UED Team. Ant cung cấp hầu hết các component thông dụng trong ứng dụng web hiện đại, như Layout, Button, Icon, DatePicker, v.v...

### 5.2.2 BackEnd

#### a) NodeJS

**NodeJS** là một môi trường runtime chạy javascript đa nền tảng và có mã nguồn mở, được build dựa trên Chrome's V8 JavaScript engine. Node.js sử dụng mô hình event-driven, non-blocking I/O khiến nó trở nên nhẹ và hiệu quả.

Cha đẻ của Node dựa trên V8 engine, cải tiến một số tính năng chẳng hạn file system API, thư viện HTTP và một số phương thức liên quan đến hệ điều hành. Điều đó có nghĩa là Node.js là một chương trình giúp ta có thể chạy code JavaScript trên máy tính, nói cách khác nó là một JavaScript runtime.

b) Flask

**Flask** là một web framework, nó là một Python module cho phép bạn phát triển các ứng dụng web một cách dễ dàng. Nó có tính mở rộng và là một microframework không bao gồm ORM (Object Relational Manager) hoặc các tính năng tương tự.

c) Firebase

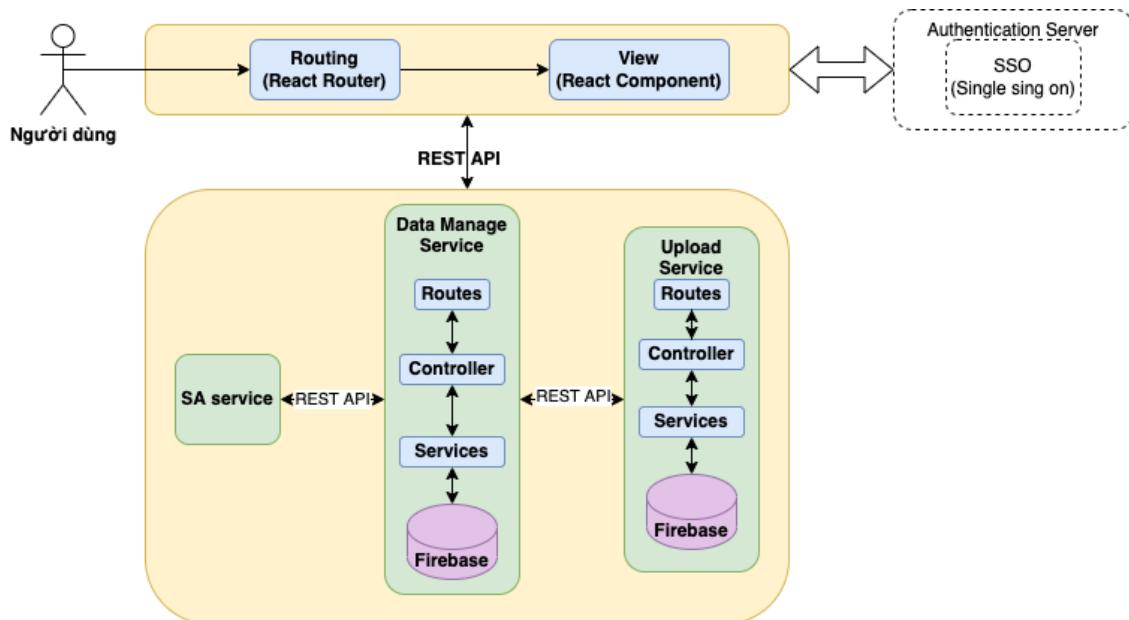
**Firebase** chính là một dịch vụ cơ sở dữ liệu được hoạt động ở trên nền tảng đám mây (Cloud). Đi kèm với đó là một hệ thống máy chủ mạnh mẽ của Google. Hệ thống có chức năng chính là giúp cho người dùng có thể lập trình ứng dụng thông qua cách đơn giản hóa những thao tác với các cơ sở dữ liệu.

### 5.3 Thiết kế kiến trúc

**Hình 5.4** mô tả kiến trúc chung của hệ thống. Bao gồm hai phần chính là Frontend và Backend.

Frontend sử dụng thư viện ReactJS nhằm xây dựng các thành phần giao diện nhận và gửi dữ liệu đến máy chủ thông qua API.

Backend được thiết kế theo mô hình Microservices chia làm 3 dịch vụ chính là: Dịch vụ quản lý và thu thập dữ liệu (Data Manage Service), dịch vụ đánh giá (SA Service), dịch vụ tải lên sản phẩm bằng link (Upload Service).



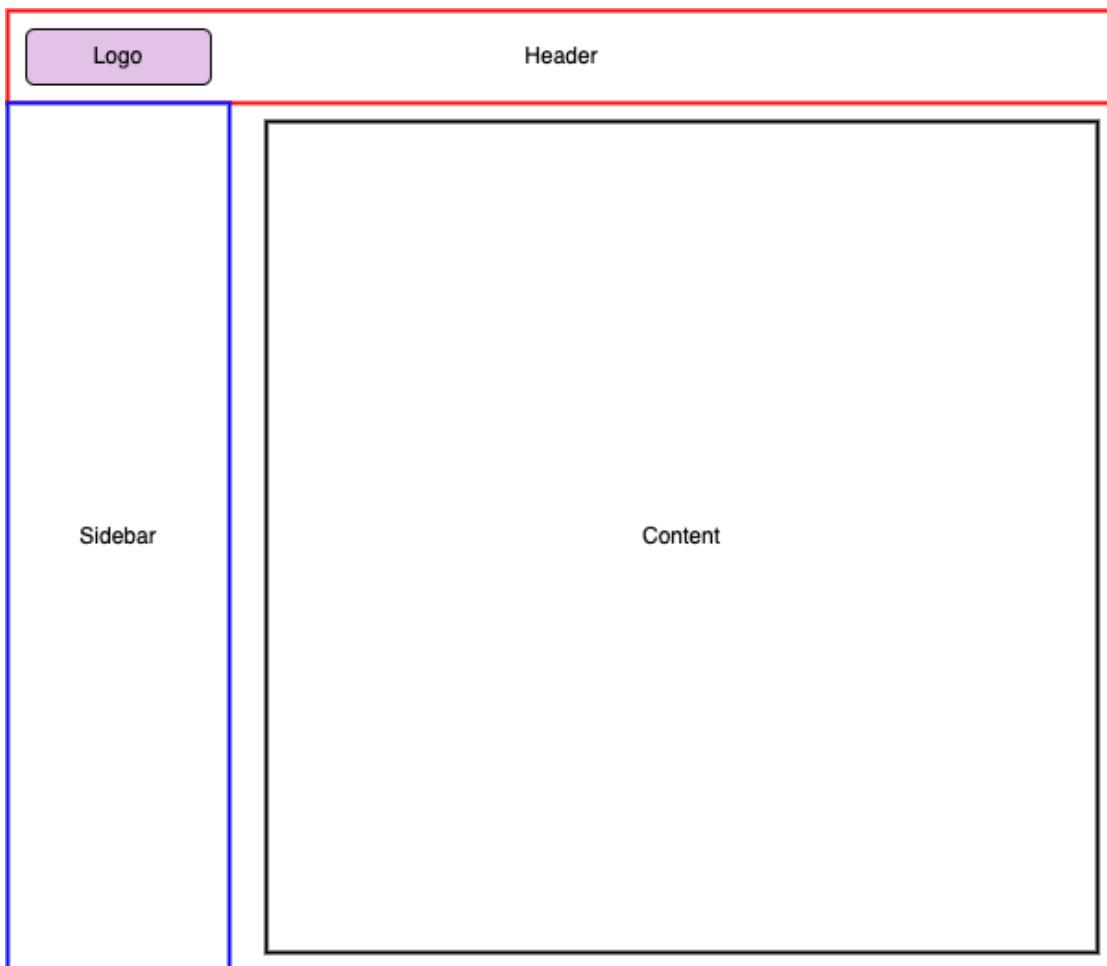
**Hình 5.4:** Kiến trúc chung của hệ thống

Frontend sẽ kết nối tới các dịch vụ của Backend thông qua việc gọi API. Ngoài ra, việc giao tiếp giữa các dịch vụ cũng thông qua API. Nhờ thiết kế như vậy mà các phần được tách biệt và phát triển độc lập với nhau.

Thêm vào đó, để truy cập vào hệ thống thì cần tương tác với dịch vụ xác thực (Authentication Server) đây là dịch vụ cho phép xác thực người dùng.

Chi tiết thiết kế các dịch vụ sẽ được trình bày trong phần 5.5.

#### 5.4 Thiết kế chi tiết giao diện



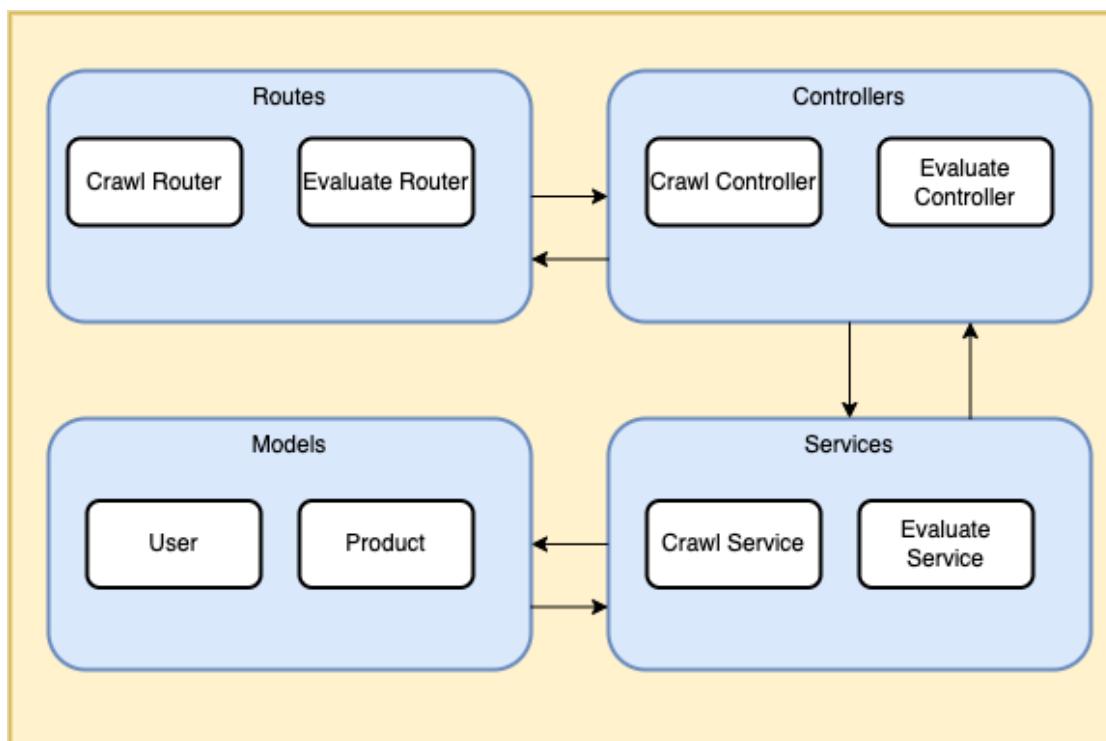
**Hình 5.5:** Thiết kế mockup giao diện

**Hình 5.5** mô tả bố cục giao diện hệ thống bao gồm: Header, Sidebar, Content. Giao diện sử dụng thư viện ReactJS nên được thiết kế thành các thành phần (component).

Đa số giao diện đều có hai thành phần Header, Sidebar chỉ thay đổi phần Content tùy thuộc vào đường dẫn người dùng truy cập để đảm bảo thống nhất về phần giao diện. Header là thanh tiêu đề của trang web gồm logo và một thanh tìm kiếm. Sidebar là thanh bên trái của giao diện, chứa nhiều thành phần nhỏ, mỗi thành phần có một ý nghĩa khác nhau.

## 5.5 Thiết kế chi tiết server

### 5.5.1 Thiết kế kiến trúc backend

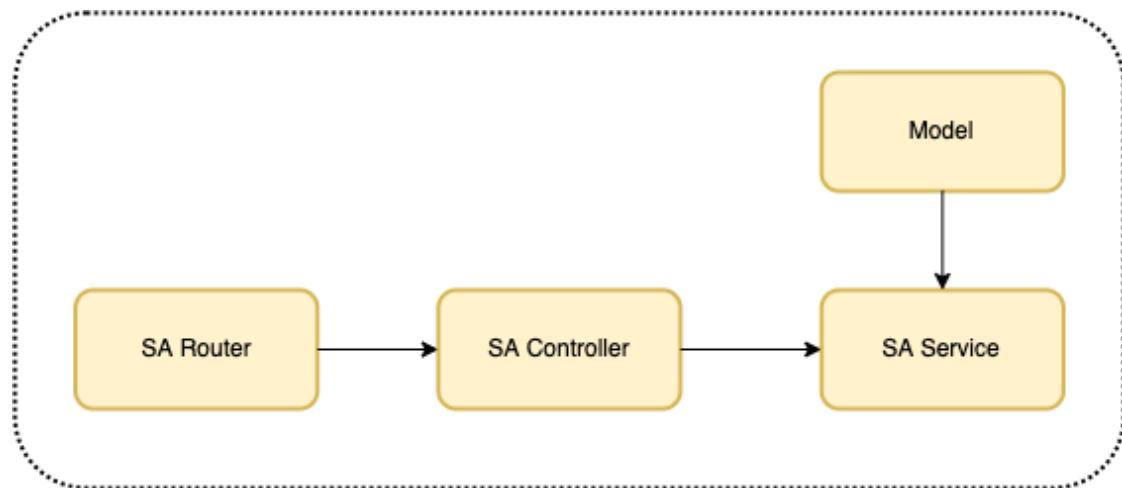


**Hình 5.6:** Thiết kế kiến trúc backend

**Hình 5.6** mô tả kiến trúc backend của hệ thống website. Phần backend được xây dựng bằng thư viện Express của NodeJS và Flask, được chia thành các phần: Routes, Controllers, Services, Models. Phần backend là nơi xử lý request từ client gửi đến.

Khi client gửi yêu cầu (request) đến server, các Router có nhiệm vụ định tuyến, gọi tới các Controllers tương ứng với route nhằm xử lý. Các Controller tiếp nhận yêu cầu từ router tiếp theo là gọi đến Services để xử lý tác vụ. Cuối cùng service sẽ gọi tới Model.

### 5.5.2 Thiết kế kiến trúc dịch vụ đánh giá



**Hình 5.7:** Thiết kế kiến trúc dịch vụ đánh giá

**Hình 5.7** mô tả kiến trúc của dịch vụ đánh giá đối với sản phẩm từ các trang thương mại điện tử. Dịch vụ chia làm 4 thành phần Router, Controller, Service, Model (PhoBERT). Trong đó, chi tiết model được trình bày ở mục.

### 5.5.3 Thiết kế cơ sở dữ liệu

Thiết kế chi tiết cho các collection của hệ thống được trình bày chi tiết trong **Bảng 5.5**

Collection	Trường	Kiểu dữ liệu	Ý nghĩa
User	id	String	Id của người dùng
	username	String	Tên đăng nhập của người dùng
	password	String	Mật khẩu của người dùng
Product	id	String	Id của sản phẩm
	name	String	Tên của sản phẩm
	link	String	Đường dẫn tới sản phẩm trên trang thương mại điện tử
	image	String	Ảnh của sản phẩm
	price	String	Giá bán của sản phẩm
	star	Float	Điểm sao của sản phẩm
	evaluate	String	Loại đánh giá của sản phẩm
	percent	Float	Điểm đánh giá của sản phẩm
	id_shop	Int	Id của trang thương mại

**Bảng 5.5:** Thiết kế chi tiết cơ sở dữ liệu của hệ thống

## 5.6 Thiết kế API

Như đã trình bày các dịch vụ và trang web giao tiếp với nhau thông qua API. Hệ thống cung cấp 9 API, cụ thể được ghi trong **Bảng 5.6**

STT	Ý nghĩa	Phương thức	Địa chỉ
1	Thu thập dữ liệu sản phẩm từ trang tiki	GET	/api/v1/tiki
2	Thu thập dữ liệu sản phẩm từ trang sendo	GET	/api/v1/sendo
3	Thu thập dữ liệu sản phẩm từ trang shopee	GET	/api/v1/shopee
4	Thu thập dữ liệu sản phẩm từ trang thẻ giới di động	GET	/api/v1/tgdd
5	Đánh giá sản phẩm từ trang tiki	GET	/api/v1/tiki/evaluate
6	Đánh giá sản phẩm từ trang sendo	GET	/api/v1/sendo/evaluate
7	Đánh giá sản phẩm từ trang shopee	GET	/api/v1/shopee/evaluate
8	Đánh giá sản phẩm từ trang thẻ giới di động	GET	/api/v1/tgdd/evaluate
9	Đánh giá sản phẩm theo link	POST	/api/v1/evaluate

**Bảng 5.6:** Danh sách API của hệ thống

## 5.7 Xây dựng hệ thống

### 5.7.1 Thư viện và công cụ sử dụng

Trong quá trình thực hiện đồ án em sử dụng một số công cụ hỗ trợ việc xây dựng và phát triển frontend, backend và triển khai hệ thống như **Bảng 5.7**

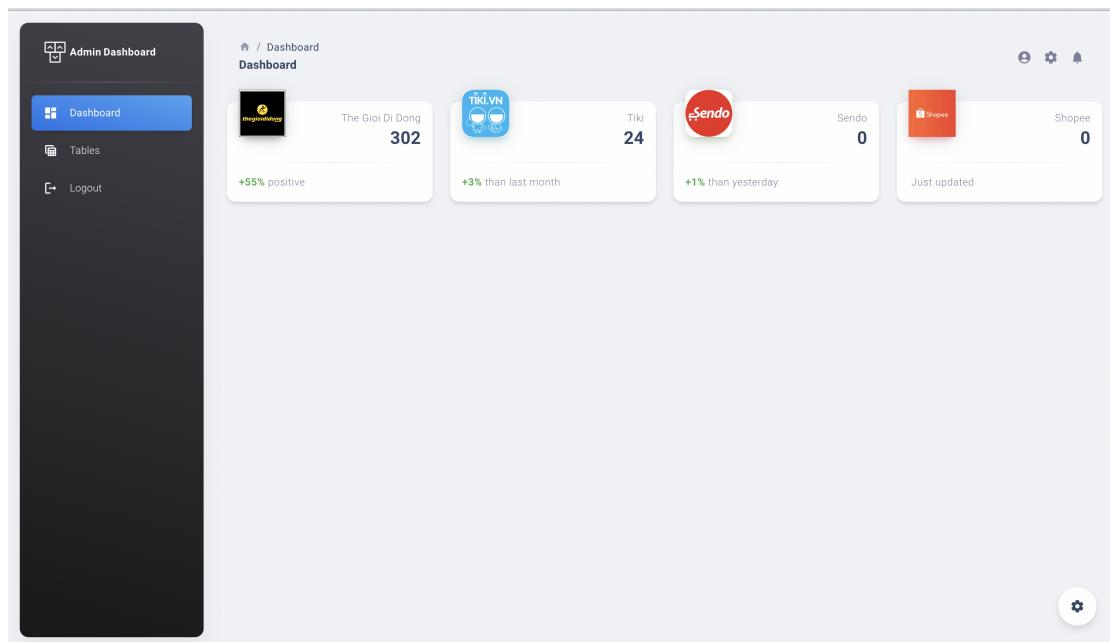
Mục đích	Công cụ	Địa chỉ
<b>IDE</b>	Visual Studio Code	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>
<b>Backend Framework</b>	ExpressJS	<a href="https://expressjs.com/">https://expressjs.com/</a>
<b>Backend Framework</b>	Flask	<a href="https://flask.palletsprojects.com/en/2.1.x/">https://flask.palletsprojects.com/en/2.1.x/</a>
<b>Frontend</b>	ReactJS	<a href="https://reactjs.org/">https://reactjs.org/</a>
<b>Huấn luyện mô hình</b>	Google Colab	<a href="https://colab.research.google.com/">https://colab.research.google.com/</a>
<b>Huấn luyện mô hình</b>	PyTorch	<a href="https://pytorch.org/">https://pytorch.org/</a>
<b>Huấn luyện mô hình</b>	Conda	<a href="https://conda.io/">https://conda.io/</a>

**Bảng 5.7:** Danh sách công cụ và thư viện sử dụng

### 5.7.2 Kết quả phát triển

#### a, Giao diện bên quản lý

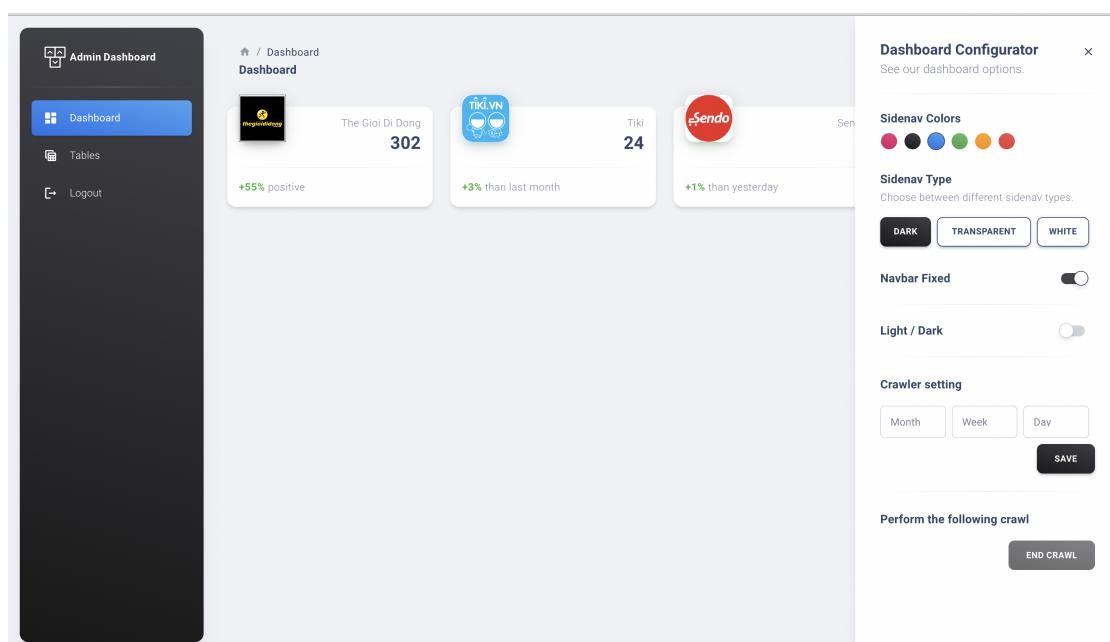
**Hình 5.8** là Giao diện trang tổng quan của màn quản lý



**Hình 5.8:** Giao diện trang tổng quan

Đây là giao diện giúp người quản lý có thể biết trong hệ thống có bao nhiêu sản phẩm của từng trang thương mại điện tử.

**Hình 5.9** là Giao diện của màn cài đặt.



**Hình 5.9:** Giao diện màn cài đặt

Đây là giao diện giúp cho quản lý có thể thay đổi màu của hệ thống và giúp quản lý có thể cài đặt thời gian tự động thu thập dữ liệu từ các trang thương mại điện tử.

**Hình 5.10** là Giao diện của màn quản lý sản phẩm.

PRODUCT	EVALUATE	ACTION
Máy Tính Bảng Samsung Galaxy Tab A7 Lite T225 3GB/32GB - Hàng Chính Hãng	POSITIVE 5 star	Access Link
Máy Tính Bảng Huawei Matepad   Màn Hình 2K Fullview   Hiệu Suất Mạnh Mẽ   Âm Thanh Vòm Sống	POSITIVE 4.7 star	Access Link
Máy Tính Bảng Samsung Galaxy Tab S7 FE LTE T735 (4GB/64GB) - Hàng Chính Hãng	POSITIVE 4.8 star	Access Link
Apple iPad mini (6th Gen) Wi-Fi, 2021	POSITIVE 5 star	Access Link
Apple iPad Pro 12.9 - inch M1 Wi-Fi, 2021	POSITIVE 4.8 star	Access Link
Máy Tính Bảng Huawei Matepad   Màn Hình 2K Fullview   Hiệu Suất Mạnh Mẽ   Âm Thanh Vòm Sống	POSITIVE 4.7 star	Access Link

**Hình 5.10:** Giao diện màn quản lý sản phẩm

Đây là giao diện giúp người quản lý có thể xem hệ thống có những sản phẩm nào. Trong giao diện này có tính năng thu thập sản phẩm thủ công.

### b, Giao diện bên người dùng

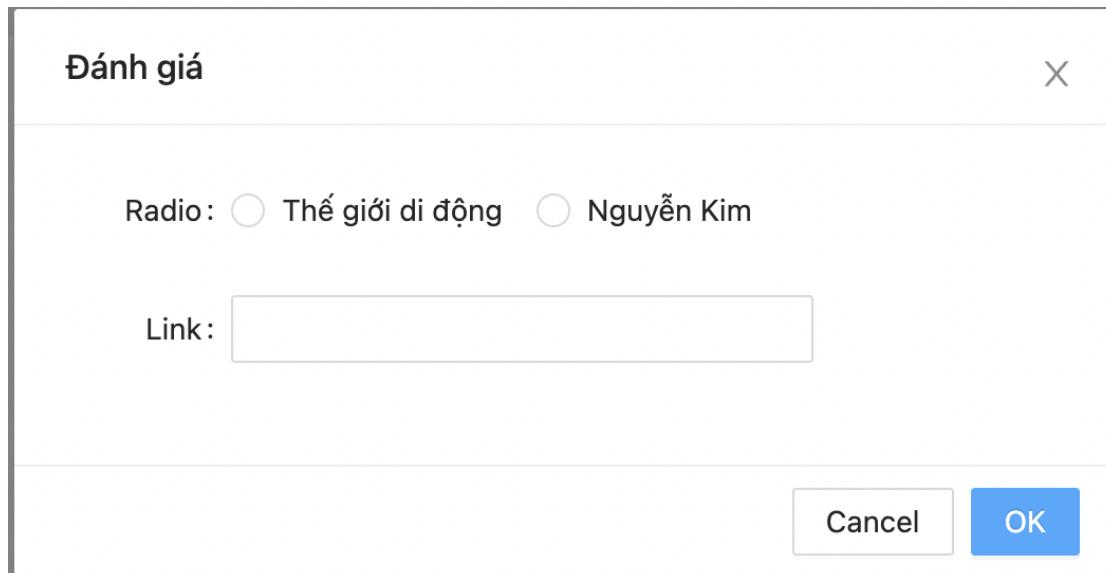
**Hình 5.11** là Giao diện của trang chủ bên người dùng.

Điện thoại Samsung Galaxy S22 5G 128GB	5 star	18990000.0	Add to cart
Máy tính bảng Samsung Galaxy Tab A8 8" T295 (2019)	4 star	3290000.0	Add to cart
Laptop Dell Vostro 3405 R5 3500U/8GB/512GB/Office H&S/Win11 (V4R53500U003W1)	4 star	15990000.0	Add to cart
Laptop Acer TravelMate B3 TMB311 31 C2HB N4020/4GB/128GB/Win11 (NXVNFSV.006)	4 star	4990000.0	Add to cart
Laptop MSI Gaming GE66			
iPhone 13 Pro Max			
Samsung Galaxy S22 Ultra			
Samsung Galaxy Z Fold4			

**Hình 5.11:** Giao diện màn trang chủ

Đây là giao diện trang chủ bên người dùng, giao diện sẽ chứa có sản phẩm của hệ thống và các nút cho chức năng tìm kiếm theo sao, thể loại đánh giá và giá tiền ở phần sidebar. Trên header có nút đánh giá và thanh tìm kiếm.

**Hình 5.12** là Giao diện của modal đánh giá sản phẩm theo đường dẫn.



**Hình 5.12:** Giao diện của modal đánh giá

Đây là popup cho tính năng đánh giá theo đường dẫn mà người dùng nhập vào. Hiện tại hệ thống đang hỗ trợ hai trang thương mại là thế giới di động và nguyễn kim.

**Hình 5.13** là Giao diện của popup chi tiết sản phẩm.



**Hình 5.13:** Giao diện của modal của sản phẩm

Đây là popup chi tiết của sản phẩm. Trên giao diện sẽ có thông tin gồm ảnh, tên, số sao, giá của sản phẩm người dùng chọn xem chi tiết.

### 5.8 Đóng gói và triển khai

Hệ thống đánh giá sản phẩm đã được xây dựng và đóng gói thành các file zip. Trong các file zip gồm có source code và một file hướng dẫn cài đặt thực thi source code.

Cấu trúc của một file hướng dẫn sẽ gồm: (i) Ý nghĩa của source code, (ii) Các bước thực thi source code từ cài đặt tới chạy source.

Tất cả source code của toàn bộ hệ thống từ server tới client đã được lưu trữ trên driver với đường dẫn: [https://husteduvn-my.sharepoint.com/:f/g/personal/do\\_1d176716\\_sis\\_hust\\_edu\\_vn/EiPK1EojSAxCowU9-qgwtNIB8BiaDiX87FFGRltESCW10A?e=LoRr7F](https://husteduvn-my.sharepoint.com/:f/g/personal/do_1d176716_sis_hust_edu_vn/EiPK1EojSAxCowU9-qgwtNIB8BiaDiX87FFGRltESCW10A?e=LoRr7F)

### Kết chương

Chương 5 này đã trình bày chi tiết quá trình thiết kế, xây dựng và phát triển cũng như những công nghệ sử dụng trong quá trình triển khai xây dựng hệ thống đánh giá sản phẩm. Từ những nghiên cứu và thực nghiệm các mô hình cùng với việc phát triển một hệ thống đánh giá sản phẩm, tiếp theo em sẽ trình bày kết luận dành cho toàn bộ đồ án. Phần đó sẽ được trình bày trong chương tiếp theo - Chương 6.

## CHƯƠNG 6. KẾT LUẬN

Trong chương này sẽ trình bày kết luận của đồ án và hướng phát triển trong tương lai.

### 6.1 Kết luận

Trong đồ án này, em đã đề xuất và triển khai hệ thống đánh giá sản phẩm trên các trang thương mại điện tử tích hợp mô hình phân tích cảm xúc bình luận.

Về mặt cơ chế đề xuất, em đã đã xây dựng và hoàn thiện mô hình phân tích cảm xúc bình luận giúp người dùng có cái nhìn tổng quan về sản phẩm và giúp doanh nghiệp có thể thu thập đánh giá về sản phẩm mà họ cung cấp cho người dùng. Cơ chế sử dụng mô hình học sâu - PhoBERT cùng với công cụ tiền xử lý dữ liệu giúp mô hình đánh giá tốt hơn. So với mô hình học sâu - PhoBERT thì mô hình học máy SVM sẽ có độ chính xác thấp hơn.

Về mặt ứng dụng, em đã hoàn thành được hệ thống đánh giá sản phẩm cùng với một số tính năng phù hợp với nhu cầu của người dùng thực tế như: (i) Đánh giá sản phẩm trực tiếp bằng link, (ii) Tìm kiếm sản phẩm đã thu thập và đánh giá trên hệ thống. Việc triển khai và phát triển hệ thống giúp cho giải pháp mà ĐATN đã đưa ra sẽ tiếp cận gần hơn với thực tiễn.

Về vấn đề còn tồn đọng trong ĐATN này là độ chính xác của lớp Neutral vẫn còn thấp hơn so với hai lớp Positive và Negative. Vấn đề này có thể do bộ dữ liệu và bước tiền xử lý của em chưa hoàn hảo nên khi thử nghiệm mô hình thì có một số câu có nhãn Neutral thì bị đánh sai là Negative hoặc Positive.

Ví dụ như câu "May kha dep. Nhưng hình như loa ngoai chưa ôn định. Phân mềm chụp ảnh chưa tối ưu. Dù sao giá vẫn re va hy vọng sẽ co ban cap nhât phân mềm tốt hơn. Cam ơn Thê giới di động vê phong cach phuc vu.". Câu này vừa có khen vừa có chê thì sẽ là Neutral nhưng mô hình lại đánh thành Positive.

### 6.2 Hướng phát triển trong tương lai

Đối với các nghiên cứu tiếp theo, em sẽ tiến hành mở rộng hệ thống và giải quyết các vấn đề còn tồn đọng trong ĐATN này.

Hướng giải quyết cho vấn đề tồn đọng em sẽ hướng tới những việc sau:

- Tiếp tục thu thập bình luận và đánh nhãn cho những câu bình luận để tăng dữ liệu huấn luyện cho mô hình.
- Phát triển một mô hình sửa lỗi chính tả để bước tiền xử lý dữ liệu tốt hơn.

- Tìm cách tinh chỉnh cho mô hình PhoBERT.
- Tiếp tục huấn luyện trên bộ dữ liệu thu thập và xử lý để tăng độ chính xác cho mô hình và đặc biệt là độ chính xác của lớp Neutral.

## TÀI LIỆU THAM KHẢO

- [1] P. Rawat **and** A. N. Mahajan, “Reactjs: A modern web development framework,” *International Journal of Innovative Science and Research Technology*, **jourvol** 5, **number** 11, 2020.
- [2] K. Van Nguyen, V. D. Nguyen, P. X. Nguyen, T. T. Truong **and** N. L.-T. Nguyen, “Uit-vsfc: Vietnamese students’ feedback corpus for sentiment analysis,” **in***2018 10th international conference on knowledge and systems engineering (KSE)* IEEE, 2018, **pages** 19–24.
- [3] H. T. Nguyen, H. V. Nguyen, Q. T. Ngo **and**others, “Vlsp shared task: Sentiment analysis,” *Journal of Computer Science and Cybernetics*, **jourvol** 34, **number** 4, **pages** 295–310, 2018.
- [4] J. Devlin, M.-W. Chang, K. Lee **and** K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [5] Y. Liu, M. Ott, N. Goyal **and**others, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [6] P. Delobelle, T. Winters **and** B. Berendt, “Robbert: A dutch roberta-based language model,” *arXiv preprint arXiv:2001.06286*, 2020.
- [7] D. Q. Nguyen **and** A. T. Nguyen, “Phobert: Pre-trained language models for vietnamese,” *arXiv preprint arXiv:2003.00744*, 2020.
- [8] Z. Yin, J. Liu, M. Krueger **and** H. Gao, “Introduction of svm algorithms and recent applications about fault diagnosis and other aspects,” **in***2015 IEEE 13th International Conference on Industrial Informatics (INDIN)* IEEE, 2015, **pages** 550–555.
- [9] Q. H. Pham, V. A. Nguyen, L. B. Doan, N. N. Tran **and** T. M. Thanh, “From universal language model to downstream task: Improving RoBERTa-based vietnamese hate speech detection,” **in***2020 12th International Conference on Knowledge and Systems Engineering (KSE)* IEEE, 2020. DOI: 10.1109/kse50997.2020.9287406.
- [10] J. Howard **and** S. Ruder, “Universal language model fine-tuning for text classification,” *arXiv preprint arXiv:1801.06146*, 2018.