

BÀI TOÁN PHÂN TÍCH CẢM XÚC DỰA TRÊN KHÍA CẠNH CHO BỘ DỮ LIỆU NHÀ HÀNG

Trần Trung Hiếu^{1,*}, Lâm Gia Huy^{1,*}, Võ Đình Tứ^{1,*},
Lưu Thanh Sơn^{1,†}, and Nguyễn Văn Kiệt^{1,†}

¹Trường Đại Học Công Nghệ Thông Tin
Đại Học Quốc Gia Thành Phố Hồ Chí Minh
^{*}{18520754, 18520832, 18521589}@gm.uit.edu.vn
[†]{sonlt,kietnv}@uit.edu.vn

Tóm tắt nội dung Phân tích tự động trong các đánh giá của khách hàng trực tuyến đang là một trong những chủ đề nghiên cứu nóng gần đây. Khi số lượng bài đánh giá ngày càng nhiều, điều cần thiết là phải phát triển một mô hình phân tích hiệu quả có khả năng trích xuất các khía cạnh của sản phẩm và xác định cảm xúc cho các khía cạnh đó. Nếu làm tốt được điều này, lượng thông tin thu được sẽ góp phần cải thiện hiệu suất kinh doanh một cách đáng kể. Một bài toán cơ bản và là nền tảng giải quyết vấn đề trên chính là bài toán phân tích cảm xúc dựa theo khía cạnh (Aspect-based sentiment analysis). Bài toán này tập trung vào việc nhận dạng các khía cạnh và từ đó đưa ra đánh giá xu hướng cảm xúc đối với phía cạnh đó. Trong bài báo này chúng tôi trình bày và giải quyết hai bài toán con của bài toán phân tích cảm xúc theo khía cạnh: phát hiện loại khía cạnh (Aspect Category Detection) và phân loại cảm xúc của khía cạnh (Aspect Polarity Classification). Bộ dữ liệu chúng tôi sử dụng trong bài báo này có tên là Bộ dữ liệu nhà hàng tiêu chuẩn Việt Nam. Kết quả thực nghiệm cho thấy rằng mô hình PhoBERT cho kết quả tốt nhất trên cả hai bài toán con với 87.63% cho F1-score trên nhiệm vụ phát hiện loại khía cạnh và 82.92% cho F1-score cho nhiệm vụ phân loại cảm xúc của khía cạnh.

Keywords: Phân tích cảm xúc dựa theo khía cạnh · Phát hiện loại khía cạnh · Phân loại cảm xúc của khía cạnh.

1 Giới thiệu

Ngày nay, với sự phát triển nhanh chóng của các nội dung do người dùng tạo ra trên internet, phân tích cảm xúc tự động trong các đánh giá của khách hàng trực tuyến đã trở thành một chủ đề nghiên cứu nóng gần đây. Với số lượng đa dạng các loại sản phẩm và dịch vụ được đánh giá trên nhiều trang web, điều cần thiết là phải phát triển một mô hình phân tích cảm xúc hiệu quả có khả năng trích xuất các khía cạnh của sản phẩm mà người dùng đề cập và xác định cảm xúc mà khách hàng đánh giá về sản phẩm đó. Từ đó có thể khai thác thông tin này để lên kế hoạch kinh doanh phù hợp hơn, cải thiện chất lượng sản phẩm cũng như dịch vụ đi kèm để phù hợp hơn với người

tiêu dùng. Đó là lý do vì sao bài toán phân tích cảm xúc là một chủ đề được quan tâm rất nhiều.

Phân tích cảm xúc là lĩnh vực phân tích quan điểm, cảm xúc, đánh giá, thái độ và cách thể hiện cảm xúc của người dùng đối với sản phẩm và các thuộc tính của sản phẩm đó. Phân tích cảm xúc có thể được chia thành ba nhánh con dựa trên các công trình nghiên cứu nổi tiếng, đó là phân tích cảm xúc dựa trên tài liệu, phân tích cảm xúc dựa trên câu và phân tích cảm xúc dựa trên khía cạnh. Phân tích cảm xúc dựa trên tài liệu và phân tích cảm xúc dựa trên câu tập trung nhiều hơn vào xu hướng của cảm xúc tổng thể của một tài liệu hoặc câu. Trong khi đó phân tích cảm xúc dựa theo khía cạnh (ABSA) giúp thu thập thông tin cảm xúc toàn diện bằng cách liệt kê ra các thực thể, thuộc tính đi kèm và khuynh hướng cảm xúc cụ thể đến từng thực thể, thuộc tính giúp cung cấp nền tảng đáng tin cậy cho việc đưa ra từng quyết định cụ thể.

Nhiệm vụ ABSA được chia thành hai bài toán tổ hợp lớn: phân loại thông tin khía cạnh được trích xuất và cảm xúc (Aspect Term Sentiment Analysis - ATSA) và phân loại loại khía cạnh và cảm xúc (Aspect Category Sentiment Analysis - ACSA). Trong bài báo này chúng tôi sẽ trình bày cũng như giải quyết bài toán tổ hợp ACSA bằng cách thực hiện hai nhiệm vụ con: phát hiện loại khía cạnh (Aspect Category Detection - ACD) và phân loại cảm xúc của khía cạnh (Aspect Polarity Classification - APC). Các định nghĩa cụ thể của hai nhiệm vụ con này sẽ được chúng tôi nêu rõ ở phần 3.

Để giải quyết hai bài toán vừa nêu, chúng tôi tiến hành thực nghiệm trên bộ dữ liệu có tên là Bộ dữ liệu nhà hàng tiêu chuẩn Việt Nam. Chúng tôi đã áp dụng các mô hình máy học truyền thống (Complement Naive Bayes, Logistic Regression, Support Vector Machine), các mô hình học sâu (Text-CNN, Bi-GRU) và các mô hình Transformer (XLM-R, viBert_FPT, PhoBERT) để so sánh hiệu suất trên bộ dữ liệu.

Phần còn lại của bài báo sẽ được tổ chức như sau: Phần 2 sẽ trình bày các công trình cụ thể liên quan đến bài toán ABSA. Bộ dữ liệu được thực nghiệm và cách tiếp cận bài toán sẽ được nêu rõ trong phần 3. Phần 4 sẽ là quá trình chuẩn bị dữ liệu, đề xuất các kịch bản thực nghiệm và đưa ra các kết quả mô hình. Cuối cùng, phần 5 sẽ đưa ra những kết luận và những mục tiêu tiếp theo mà chúng tôi định hướng trong tương lai.

2 Công trình liên quan

Trong suốt thập kỷ qua, phân tích cảm xúc dựa theo khía cạnh được xem là một trong những phần quan trọng nhất trong thương mại điện tử vì nó có ứng dụng tiềm năng để phân tích phản hồi của người dùng trực tuyến dựa trên các quan điểm và nhận xét của họ [1], [2].

Vào năm 2004, M Hu và B đã đề xuất vấn đề tổng hợp ý kiến người dùng về các tính năng của sản phẩm được bán trực tuyến [3]. Vào năm 2009, [4] đề xuất hệ khuyến nghị dựa trên hồi quy, sử dụng các bình luận đánh giá của người dùng. [5] dựa trên cấu trúc ngữ pháp của mệnh đề, Tun Thura Thet và các cộng sự đã đề xuất một phương pháp tiếp cận ngôn ngữ để tính toán điểm cảm xúc cho một mệnh đề đối với các khía cạnh khác nhau của một bộ phim, năm 2010, Tương tự, như vậy ở năm kế tiếp (2011) [6] đã đề xuất mô hình SLDA và ASUM cho việc trích xuất các cặp khía cạnh, tình cảm trong các bình luận về thiết bị điện tử và nhà hàng. Năm 2017, [7] sử dụng Latent Dirichlet Allocation để mô hình hóa các khía cạnh có trong bình luận của người dùng cùng các kỹ thuật xử lý ngôn ngữ tự nhiên để tìm ra mối quan hệ giữa các từ và khía cạnh, sau đó áp dụng mô hình máy học truyền thống Native Bayes để dự đoán cảm xúc cho khía cạnh tương ứng có trong câu bình luận về khách sạn. Các nghiên cứu trên hầu

hết đã lột tả được quá trình đi lên của bài toán Phân tích cảm xúc dựa theo khía cạnh bằng cách tiếp cận ngôn ngữ với máy học truyền thống trên nhiều bộ dữ liệu sẵn có trên thế giới. Vì ngôn ngữ là đa dạng, nên cách xử lý cho mỗi ngôn ngữ là khác nhau. Năm 2010, Việt Nam bắt đầu cho ra những nghiên cứu đầu tiên về phân tích cảm xúc ở mức độ câu. [8] sử dụng phương pháp rule-based để xây dựng đánh giá tự động về ý kiến của người dùng ở cấp độ câu cho hai miền dữ liệu về máy tính xách tay và máy tính để bàn, các tác giả đã sử dụng khung GATE để xây dựng nên các thành phần của xử lý ngôn ngữ tự nhiên. Nối tiếp đó là vô vàn các nghiên cứu liên quan được hình thành. Năm 2016, câu lạc bộ Xử lý Ngôn Ngữ và Tiếng Nói Tiếng Việt (VLSP) ở Việt Nam đã chia sẻ tác vụ "Phân tích cảm xúc" với nhiệm vụ phân loại văn bản thành một trong ba cảm xúc phân cực: Tiêu cực, Trung tính, và Tích cực¹. Tập dữ liệu chứa các bình luận của các bài báo được thu thập từ các trang web. Sau đó hai năm, câu lạc bộ VLSP đã chính thức đưa ra nhiệm vụ chung đầu tiên về "Phân tích cảm xúc dựa trên khía cạnh" cho hai miền dữ liệu cụ thể là nhà hàng và khách sạn² ở mức độ document. Từ kho dữ liệu ở mức độ document, một kho dữ liệu chuẩn ở mức độ câu đã được hình thành trong [9], phục vụ cho hai bài toán con ACD và APD. Kho dữ liệu chuẩn này đóng vai trò quan trọng để phát triển các phương pháp đối với các ngôn ngữ có tài nguyên thấp như Tiếng Việt. Bên cạnh xây dựng dữ liệu, [9] các tác giả cũng đã thực nghiệm bằng các phương pháp học giám sát, và thu được các kết quả đáng mong đợi khi sử dụng BERT. Một trong hai bộ dữ liệu từ kho dữ liệu chuẩn là "The Vietnamese benchmark restaurant datasets" được chúng tôi sử dụng để thực thi cho hai bài toán ACD và APC. Trong đó APD được chúng tôi lấy ý tưởng theo [10] - thực hiện thêm khía cạnh hoặc các từ thể hiện khía cạnh vào sau văn bản tương ứng để thực hiện bài toán.

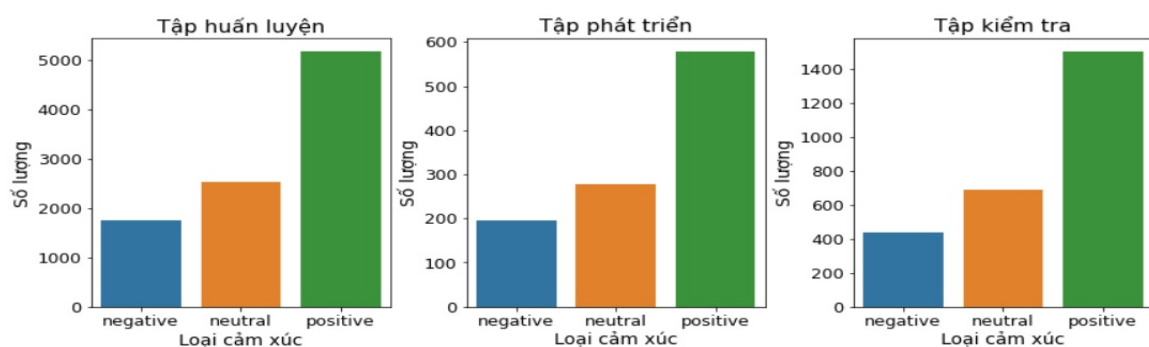
3 Bộ dữ liệu và cách tiếp cận

3.1 Bộ dữ liệu

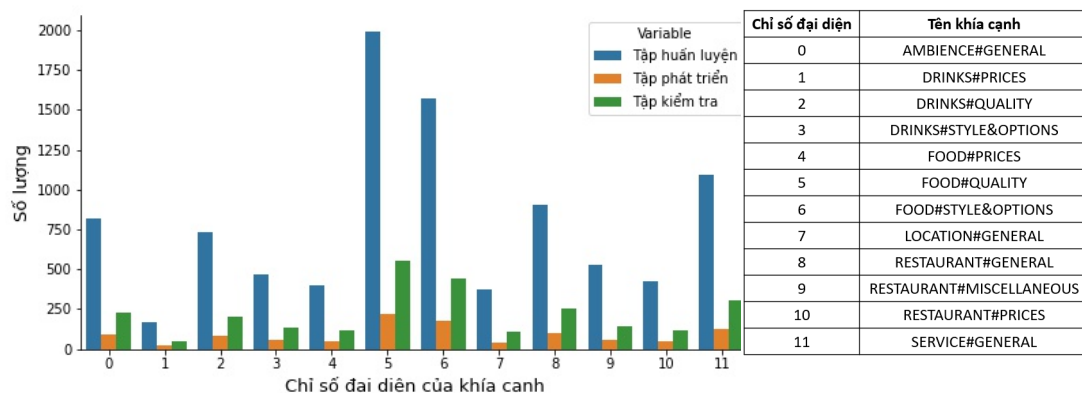
Bộ dữ liệu mà chúng tôi sử dụng có tên là Bộ dữ liệu nhà hàng tiêu chuẩn Việt Nam [9] do Đặng Văn Thìn và các cộng sự xây dựng tương tự như Subtask2 của SemEval-2016 Task 5 [11]. Bộ dữ liệu chứa các bình luận của người dùng, các khía cạnh được người dùng nhắc đến và cảm xúc đối với từng khía cạnh cụ thể. Trong bộ dữ liệu, 6 loại thực thể (Entity) đã kết hợp với 5 loại thuộc tính (Attribute) để tạo ra 12 loại khía cạnh khác nhau (Entity#Attribute) được mô tả cụ thể trong hình 2. Cảm xúc của người được thể hiện trong hình 1 bao gồm: tiêu cực (negative), trung tính (neutral) và tích cực (positive). Bộ dữ liệu được chia thành ba phần gồm: tập huấn luyện (7028 bình luận, 9458 khía cạnh), tập phát triển (771 bình luận, 1053 khía cạnh) và tập kiểm tra (1938 bình luận, 2629 khía cạnh).

¹ <https://vlsp.org.vn/vlsp2016/eval/sa>

² <https://vlsp.org.vn/vlsp2018/eval/sa>



Hình 1: Thống kê số lượng cảm xúc.



Hình 2: Thống kê số lượng khía cạnh.

3.2 Mô tả bài toán

Bài toán phân tích cảm xúc dựa theo khía cạnh được mô tả tổng quát trong bảng 1:

Bảng 1: Bài toán tổng quát.

Đầu vào	Đầu ra
Giá có mấy chỗ hơi mắc..	RESTAURANT#PRICES - netative (Nhà hàng # Giá tiền - Tiêu cực)

Chúng tôi tiếp cận bài toán phân tích cảm xúc theo khía cạnh thông qua hai bài toán con gồm: phát hiện loại khía cạnh (Aspect Category Detection) và phân loại cảm xúc của khía cạnh (Aspect Polarity Classification). Chi tiết về mỗi bài toán con sẽ được chúng tôi trình bày sau đây.

3.2.1 Phát hiện loại khía cạnh

Phát hiện loại khía cạnh là một trong hai nhiệm vụ chính của bài toán phân tích cảm

xúc dựa theo khía cạnh. Cho một bình luận của người dùng cùng với một danh sách các khía cạnh, nhiệm vụ của bài toán phát hiện loại khía cạnh là phát hiện ra rằng khía cạnh nào thực sự được người dùng nhắc đến bên trong bình luận. Bài toán phát hiện loại khía cạnh thuộc dạng bài toán phân loại đa nhãn. Cụ thể về bài toán được mô tả như sau:

Cho tập các câu bình luận D (d_1, d_2, \dots, d_m) và tập các khía cạnh cho trước L (l_1, l_2, \dots, l_n) với $n = 12$. Với mỗi khía cạnh $l_i \in L$ có dạng Entity#Attribute với Entity là các thực thể (Vị trí (LOCATION), Thức uống (DRINKS), ...) và Attribute là các thuộc tính tương ứng với thực thể (giá (PRICES), chất lượng (QUALITY), ...). Mỗi câu bình luận $d_i \in D$ sẽ chứa một hoặc nhiều khía cạnh l_i . Nhiệm vụ của chúng tôi là sẽ xác định những khía cạnh l_i được nhắc đến trong bình luận d_i .

Bảng 2: Một số ví dụ cho bài toán phát hiện loại khía cạnh.

Đầu vào	Đầu ra
Giá có mấy chỗ hơi mát.	RESTAURANT#PRICES (Nhà hàng # Giá tiền)
Quán đông lắm, gọi món phải đợi hơi lâu, ko biết phải tại mình đi trễ ko, nhưng mà nghe nói ở đây giờ nào cũng đông cũng phải ngồi đợi một lúc hết.	RESTAURANT#GENERAL (Nhà hàng # Chung), SERVICE#GENERAL (Dịch vụ # Chung)
Thức uống bên này lúc nào cũng làm mình hài lòng, ngoại trừ giá hơi cao thì không có gì phàn nàn.	DRINKS#QUALITY (Thức uống # Chất lượng), DRINKS#PRICES (Thức uống # Giá tiền)

3.2.2 Phân loại cảm xúc của khía cạnh

Ở nhiệm vụ phân loại cảm xúc của khía cạnh, chúng tôi thực hiện xác định cảm xúc tương ứng cho từng khía cạnh được thể hiện trong câu bình luận bằng cách thực hiện đưa khía cạnh vào cuối câu bình luận tương ứng [12]. Sau đó sẽ thực hiện việc xác định cảm xúc của khía cạnh như xác định cảm xúc cho một câu bình luận tổng quát. Bài toán lúc này sẽ thuộc dạng bài toán phân loại đa lớp. Cụ thể bài toán được biểu diễn như sau:

Cho một tập hợp các câu bình luận D (d_1, d_2, \dots, d_m). Tương ứng với mỗi câu bình luận $d_i \in D$ là một tập các khía cạnh L_i ($l_{i1}, l_{i2}, \dots, l_{in}$) mà trong đó khía cạnh $l_{ik} \in L_i$ được nhắc đến trong bình luận d_i . Tập S (s_1, s_2, s_3) là tập hợp các trường hợp cảm xúc có thể có của người dùng khi nói đến khía cạnh l_{ik} . Nhiệm vụ của chúng tôi là sẽ kết hợp câu bình luận d_i và tập các khía cạnh L_i sao cho với mỗi khía cạnh $l_{ik} \in L_i$ sẽ xác định một cảm xúc $s \in S$ tương ứng theo cách sau:

- Từ một câu bình luận $d_i \in D$, tiến hành nhân bản thành n câu bình luận với n là số lượng khía cạnh có trong tập L_i . Kết quả sau khi nhân bản sẽ được tập D_i ($d_{i1}, d_{i2}, \dots, d_{in}$) tương ứng với câu bình luận d_i .
- Thực hiện nối từng khía cạnh $l_{ik} \in L_i$ vào tương ứng với từng câu bình luận nhân bản $d_{ik} \in D_i$ để tạo ra câu bình luận mới có mang ý nghĩa của khía cạnh. Trong đó i là chỉ số biểu thị của câu bình luận và k là chỉ số biểu thị của khía cạnh. Kết quả thu được sẽ là tập I_i ($I_{i1}, I_{i2}, \dots, I_{in}$)
- Cuối cùng với mỗi $I_{ik} \in I_i$, tiến hành xác định cảm xúc $s \in S$ tương ứng.

Với mỗi $I_{ik} \in I_i$ tương ứng với khía cạnh thứ k của câu d_i , mô hình phân loại cảm xúc của chúng tôi sẽ chỉ đưa ra một cảm xúc tương ứng với khía cạnh thứ k . Các khía

cạnh trước khi được nối vào câu sẽ được chúng tôi dịch sang Tiếng Việt như ở bảng 3 để phù hợp hơn với mô hình phân loại. Ngoài ra, không chỉ thực nghiệm cách kết hợp khía cạnh vào cuối câu bình luận, chúng tôi cũng thực nghiệm việc nối khía cạnh vào đầu câu để kiểm tra xem liệu rằng việc đưa khía cạnh vào đầu hay vào cuối câu mang lại hiệu quả tốt hơn.

Bảng 3: Chuyển đổi khía cạnh sang dạng Tiếng Việt.

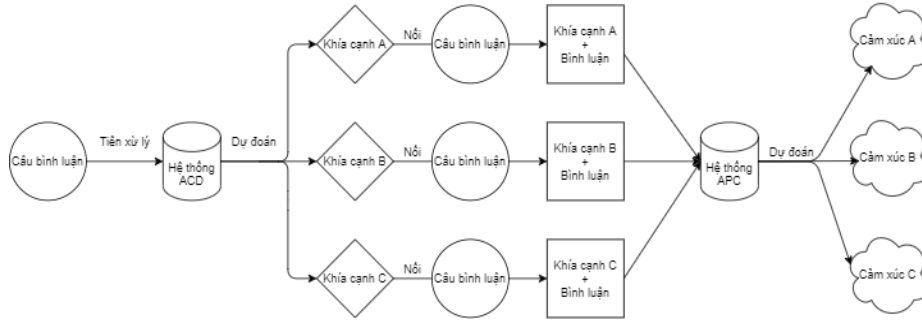
Khía cạnh ban đầu	Khía cạnh ở dạng Tiếng Việt
AMBIENCE#GENERAL	không gian tổng quát
DRINKS#PRICES	đồ uống giá tiền
DRINKS#QUALITY	đồ uống chất lượng
DRINKS#STYLE&OPTIONS	đồ uống loại
FOOD#PRICES	thức ăn giá tiền
FOOD#QUALITY	thức ăn chất lượng
FOOD#STYLE&OPTIONS	thức ăn loại
LOCATION#GENERAL	vị trí tổng quát
RESTAURANT#GENERAL	nhà hàng tổng quát
RESTAURANT#MISCELLANEOUS	nhà hàng điều khoản khác
RESTAURANT#PRICES	nhà hàng giá tiền
SERVICE#GENERAL	phục vụ tổng quát

Bảng 4: Một số ví dụ cho bài toán phát hiện cảm xúc của khía cạnh (Thêm khía cạnh vào đầu câu bình luận).

Đầu vào	Thức uống bên này lúc nào cũng làm mình hài lòng, ngoại trừ giá hơi cao thì không có gì phàn nàn. DRINKS#QUALITY (Thức uống # Chất lượng), DRINKS#PRICES(Thức uống # Giá tiền)	
Xử lý đầu vào	thức uống chất lượng Thức uống bên này lúc nào cũng làm mình hài lòng, ngoại trừ giá hơi cao thì không có gì phàn nàn.	thức uống giá tiền Thức uống bên này lúc nào cũng làm mình hài lòng, ngoại trừ giá hơi cao thì không có gì phàn nàn.
Đầu ra	Positive (Tích cực)	Negative (Tiêu cực)

Hình 3 mô tả ý tưởng để kết hợp hai bài toán con với nhau nhằm giải quyết bài toán lớn ban đầu là phân tích cảm xúc dựa theo khía cạnh. Sau khi phát hiện được các khía cạnh ở nhiệm vụ phát hiện loại khía cạnh, hệ thống của chúng tôi sẽ tiếp tục kết hợp các khía cạnh này với bình luận đầu vào để từ đó đưa ra các cảm xúc tương ứng với khía cạnh.

Một điểm cần lưu ý là trong quá trình thực nghiệm các khía cạnh mà chúng tôi sử dụng để huấn luyện cho mô hình phân loại cảm xúc dựa vào khía cạnh sẽ được lấy từ bộ khía cạnh GOLD để thuận tiện hơn cho việc thực nghiệm và đánh giá.



Hình 3: Tổng quan các bước xử lý bài toán phân tích cảm xúc dựa theo khía cạnh.

3.3 Các mô hình sử dụng

3.3.1 Máy học truyền thống

Logistic Regression (LR) [13]: là một trong những mô hình phân loại cơ bản thường được sử dụng cho các bài toán phân loại nhị phân. Đối với các bài toán phân loại, chúng ta phải chuyển đổi văn bản thành vectơ trước khi lắp chúng vào mô hình đào tạo. Bên cạnh bài toán con sử dụng đầu vào nhị phân như phát hiện loại khía cạnh thì nhiệm vụ phân loại cảm xúc của khía cạnh của chúng tôi có sử dụng bài toán phân loại nhiều lớp nên sẽ có dùng thêm cơ chế one vs res của Logistic Regression.

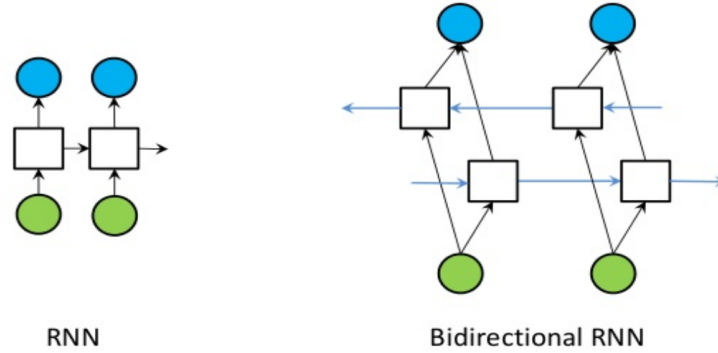
Complement Naive Bayes (CNB) [14]: Là adaptation của thuật toán Multinomial Naive Bayes (MNB) đặc biệt thích hợp cho các tập dữ liệu không cân bằng. Cụ thể, CNB sử dụng thống kê từ phần bổ sung của mỗi lớp để tính trọng số của mô hình. Các nhà tạo ra CNB theo kinh nghiệm của họ cho thấy rằng các ước tính tham số cho CNB ổn định hơn so với các ước lượng cho MNB. Hơn nữa, CNB thường xuyên làm tốt hơn MNB về các nhiệm vụ phân loại văn bản.

Support Vector Machine (SVM) [15]: là một thuật toán phân loại dựa trên khái niệm siêu phẳng, phân chia dữ liệu thành các lớp khác nhau. Thuật toán này thường được sử dụng cho các bài toán phân loại và hồi quy. Tương tự như thuật toán Logistic Regression, SVM chuyển đổi dữ liệu văn bản thành dạng vectơ trước khi phù hợp với mô hình đào tạo.

3.3.2 Học sâu

Text-CNN [16]: là một kiến trúc mạng nơ-ron nhiều lớp sử dụng cho các bài toán phân loại. Text-CNN sử dụng các lớp Convolution (CONV) để trích xuất thông tin có giá trị từ dữ liệu, cụ thể trong bài toán của chúng tôi sẽ trích xuất những đặc trưng về khía cạnh cũng như tình cảm để thực hiện phân loại.

Bi-GRU [17]: chính là sự kết hợp của hai GRU để thực hiện học ngữ cảnh ở cả hai chiều của một văn bản. GRU là phiên bản cải tiến của mạng thần kinh tái tạo tiêu chuẩn (RNN). Nó sử dụng hai cổng cập nhật (update) và cổng đặt lại (reset) để giải quyết vấn đề Vanishing gradient mà các mạng RNN đơn giản thường mắc phải. Ngoài ra, nó giúp lưu trữ những thông tin cần thiết để tránh trường hợp mất mát thông tin khi văn bản quá dài. Đây chính là một sự kết hợp khá tối ưu cho bài toán phân loại của chúng tôi.



Hình 4: Mô hình Bi-GRU.

3.3.3 Transformer

XLM-RoBERTa (XLM-R) [18]: Mô hình XLM-R dựa trên kiến trúc Transformer. Chúng tôi đề xuất mô hình XLM-R để thực hiện bài toán là vì mô hình này không chỉ hoạt động tốt trong việc thu thập thông tin ngữ nghĩa theo ngữ cảnh/khía cạnh mà còn có thể giải quyết những khó khăn do các vấn đề ngôn ngữ hỗn hợp gây ra, cụ thể bài toán chúng ta ở đây là phân loại cảm xúc ở dạng ngôn ngữ Tiếng Việt thực tế đời thường, nên khó tránh khỏi có các loại ngôn ngữ khác. Mô hình được xây dựng bằng cách sử dụng³

viBert_FPT [19]: là một mô hình joint syntactic-semantic embedding, không chỉ sử dụng thông tin cú pháp từ trình phân tích cú pháp Lexicalized Tree-Adjoining Grammar để làm phong phú thêm các word embeddings mà còn tạo ra các biểu diễn phân tán cho chính các cấu trúc cú pháp. Word embeddings từ mô hình của họ tốt hơn Skip-gram embeddings trong một số thử nghiệm mô hình phân loại tình cảm của nhiều công trình nghiên cứu trước đó, cùng với độ tương đồng từ của nó. Việc embeddings cấu trúc cú pháp giúp cải thiện trình phân tích cú pháp phụ thuộc dựa trên quá trình chuyển đổi một cách rõ ràng.

PhoBERT [20]: là một pre-trained được huấn luyện dành riêng cho tiếng Việt. PhoBERT được huấn luyện trên khoảng 20GB dữ liệu bao gồm khoảng 1GB Vietnamese Wikipedia corpus và 19GB còn lại lấy từ Vietnamese news corpus. Đây là một lượng dữ liệu uy tín để huấn luyện một mô hình như BERT. Một điểm nữa cũng quan trọng cần chú ý đó là: PhoBERT sử dụng RDRSegmenter của⁴ để tách từ cho dữ liệu đầu vào trước khi qua BPE Encoder và chỉ sử dụng Masked Language Model để train, bỏ đi phần Next Sentence Prediction.

4 Thực nghiệm và kết quả

4.1 Chuẩn bị dữ liệu

Trước khi tiến hành chạy mô hình, chúng tôi thực hiện tiền xử lý dữ liệu theo các bước sau đây:

³ <https://huggingface.co/transformers/>

⁴ <https://github.com/vncorenlp/VnCoreNLP>

Title Suppressed Due to Excessive Length 9

Bảng 5: Các bước tiền xử lý dữ liệu.

Bước	Cách xử lý	Ví dụ
1	Chuẩn hóa unicode	-
2	Chuẩn hóa gõ dấu Tiếng Việt	hoà, thuý -> hòa, thúy
3	Xóa mã html, url, tag, hashtag	Món này ăn khá ổn đấy @TrungLe. -> Món này ăn khá ổn đấy.
4	Đưa từ về dạng chuẩn	ngonnnn ghê -> ngon ghê
5	Chuyển emotion, emoji về dạng Tiếng Việt	:) -> cười nhẹ
6	Chuẩn hóa giá tiền	100k, 2000 đồng -> giá tiền
7	Đưa về dạng chữ viết thường	Món Bánh mì ngon ghê. -> món bánh mì ngon ghê.
8	Tách từ bằng VNCoreNLP	có mùi hôi hải sản. -> có mùi hôi hải _sản.
9	Xóa stopwords	món bánh _mì này rất ngon. -> món bánh _mì ngon.
10	Xóa các kí tự không cần thiết	có món : bánh _mì , bánh _xèo, -> có món bánh _mì bánh _xèo
11	Xóa khoảng trắng thừa	chủ quán _nhiệt _tình -> chủ quán nhiệt _tình

Ở đây chúng tôi sẽ thực nghiệm trên hai kịch bản: có tiền xử lý dữ liệu và không tiền xử lý dữ liệu. Mục đích của việc thực nghiệm này là để quan sát xem rằng liệu các bước tiền xử lý dữ liệu mà chúng tôi đã đề cập ở bảng 5 có thực sự mang lại hiệu quả hay không. Bởi vì mô hình PhoBERT được huấn luyện trên một dữ liệu đã được tách từ bằng VNCoreNLP, do đó mặc dù ở phần thực nghiệm không áp dụng tiền xử lý dữ liệu thì PhoBERT vẫn được áp dụng bước tách từ bằng VNCoreNLP. Đây là ngoại lệ duy nhất ở kịch bản không tiền xử lý dữ liệu.

4.2 Thông số mô hình

Chúng tôi thực hiện mô hình CNB với thông số $\alpha = 1.0$, $\text{fit_prior} = \text{True}$ và $\text{norm} = \text{False}$. Đối với Logistic Regression, thông số $C = 1.0$, $\text{max_Iter} = 100$ và $\text{solver} = \text{'lbfgs'}$. Trong khi đó, mô hình SVM được sử dụng với $C = 1.0$, $\text{max_Iter} = -1$ và thông số $\text{kernel} = \text{'rbf'}$. Đối với mô hình học sâu Text-CNN và Bi-GRU, chúng tôi sử dụng 10 epoch, $\text{batch_size} = 256$, số lượng unit = 128, hàm kích hoạt sigmoid, $\text{sequence_length} = 30$ nếu có bước tiền xử lý dữ liệu và $\text{sequence_length} = 50$ nếu không có bước tiền xử lý. Cuối cùng, chúng tôi thực hiện các mô hình Bert với số lượng epoch = 7, $\text{train_batch_size} = 16$ và $\text{eval_batch_size} = 16$. Riêng mô hình XLM-R có $\text{train_batch_size} = 32$.

Bên cạnh đó, để khởi tạo ma trận biểu diễn từ cho các mô hình học sâu, chúng tôi đã sử dụng một số bộ biểu diễn từ đã được đào tạo trước đó trên kho dữ liệu tiếng Việt như BPEmb Subword embedding (BPEmb)⁵, Wikipedia Word2vec (w2v_wiki)⁶, Multi embedding (Multi)⁷ và fastText⁸. Các thông tin cụ thể về số lượng từ sử dụng trong các bộ biểu diễn từ sẽ được trình bày trong bảng 6 dưới đây

⁵ <https://github.com/bheinzerling/bpemb>

⁶ <https://github.com/sonvx/word2vecVN>

⁷ <https://github.com/vietnlp/etnlp>

⁸ <https://github.com/skhanhnguyen/fasttext-vi>

10 Authors Suppressed Due to Excessive Length
 Bảng 6: Thông kê số lượng từ trong các bộ biểu diễn từ.

Bộ biểu diễn từ	Số lượng từ vựng	Tiền xử lý		Không tiền xử lý	
		Số lượng từ có trong bộ biểu diễn từ	Số lượng từ không nằm trong bộ biểu diễn từ	Số lượng từ có trong bộ biểu diễn từ	Số lượng từ không nằm trong bộ biểu diễn từ
fastText	200000	3888	257	4259	522
w2v_wiki	231486	3395	750	3549	1232
BPEmb	200000	2684	1461	2815	1966
Multi	21591	2126	2019	2268	2513

4.3 Kết quả

Chúng tôi đã thực nghiệm hai bài toán con trên các kịch bản khác nhau như: không tiền xử lý và có tiền xử lý dữ liệu, thêm khía cạnh vào đầu hoặc cuối câu. Việc huấn luyện mô hình trên các phương pháp máy học truyền thống, học sâu và Transformer cho kết quả khác nhau. Kết quả thu được sẽ được trình bày sau đây.

4.3.1 Phát hiện loại khía cạnh .

Bảng 7: Kết quả phát hiện loại khía cạnh ở kịch bản không tiền xử lý và kịch bản có tiền xử lý dữ liệu.

Phương pháp	Mô hình	Phát hiện loại khía cạnh (Không tiền xử lý)			Phát hiện loại khía cạnh (Có tiền xử lý)		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Truyền thống	CNB	90.33	23.09	36.78	82.62	44.12	57.53
	LR	86.44	46.33	60.33	80.49	51.96	63.15
	SVM	85.91	50.78	63.83	82.08	54.89	65.79
Học sâu + Word Embedding	Text-CNN + fastText	83.09	73.83	78.19	81.14	73.49	77.13
	Text-CNN + w2v_wiki	80.73	65.50	72.32	78.93	65.54	71.61
	Text-CNN + BPEmb	81.94	69.38	75.14	80.14	68.62	73.93
	Text-CNN + Multi	85.29	72.12	78.15	82.53	70.25	75.90
	Bi-GRU + fastText	83.60	74.25	78.65	80.12	73.56	76.7
	Bi-GRU + w2v_wiki	82.39	69.23	75.24	80.47	72.23	76.13
	Bi-GRU + BPEmb	82.22	73.68	77.71	81.51	73.60	77.35
	Bi-GRU + Multi	82.02	74.78	78.23	79.47	72.00	75.55
Transformer	XLNet	87.58	86.61	87.09	83.66	75.73	79.50
	viBert FPT	85.34	81.02	83.12	82.10	74.32	78.02
	PhoBERT	88.21	87.07	87.63	84.10	80.49	82.25

Bảng 7 trình bày kết quả thực nghiệm bài toán phát hiện loại khía cạnh trên hai kịch bản: không tiền xử lý và có tiền xử lý dữ liệu.

Dựa vào kết quả thu được có thể thấy rằng ở phương pháp máy học truyền thống, SVM là phương pháp huấn luyện thu được kết quả tốt nhất với 63.83% F1-score ở kịch bản không tiền xử lý và 65.79% F1-score ở kịch bản có tiền xử lý dữ liệu. Ngoài ra, việc sử dụng các bước tiền xử lý dữ liệu thực sự mang lại hiệu quả đối với các phương pháp máy học truyền thống khi nó làm tăng F1-score trên cả ba mô hình CNB, LR và SVM.

Đối với phương pháp học sâu, mô hình Bi-GRU khi kết hợp với fastText mang lại kết quả F1-score cao nhất với 78.65%. fastText cũng là bộ vector biểu diễn từ mang lại kết quả tốt nhất khi đạt hiệu suất cao nhất ở 3 trên 4 kịch bản thực nghiệm học sâu. Kết quả tổng quan trên mô hình Bi-GRU cũng cao hơn so với Text-CNN khi có F1-score cao hơn ở 6 trên 8 trường hợp. Việc tiền xử lý dữ liệu ở cả Text-CNN và Bi-GRU đều làm giảm hiệu suất của mô hình. Trường hợp duy nhất mà tiền xử lý dữ liệu mang lại kết quả tốt là trên mô hình Bi-GRU khi kết hợp với bộ w2v_wiki nhưng chỉ cải thiện được 0.89% F1-score.

Ở các mô hình Transformer, kết quả tốt nhất trên nhiệm vụ phát hiện loại khía cạnh thuộc về mô hình PhoBERT với 87.63% khi không tiền xử lý dữ liệu. Mô hình đa ngôn ngữ XLM-R có F1-score chỉ thấp hơn 0.54% so với kết quả tốt nhất. Tuy chỉ đạt được 83.12% F1-score và là mô hình Bert cho kết quả thấp nhất trong số ba mô hình được áp dụng nhưng viBert_FPT vẫn có hiệu suất vượt trội hơn so với các phương pháp máy học và học sâu. Các bước tiền xử lý dữ liệu không mang lại hiệu quả khi đều làm giảm hiệu suất của các mô hình Bert. F1-score giảm 5.38% đối với PhoBERT, 5.1% đối với viBert_FPT và 7.59% đối với mô hình XLM-R.

Qua các phân tích trên, chúng ta có thể thấy rằng đối với bài toán phát hiện loại khía cạnh thì việc áp dụng các bước tiền xử lý ở phần 4.1 thực sự chỉ mang lại hiệu quả trên các phương pháp máy học truyền thống. Đối với các mô hình học sâu và Transformer, việc áp dụng các bước tiền xử lý này làm giảm đáng kể hiệu suất của mô hình.

Bảng 8: Kết quả phát hiện loại khía cạnh bằng Transformer trong [21].

Loại mô hình	Mô hình	Precision	Recall	F1-score
Đa ngôn ngữ	mBERT	81.39	76.43	78.78
	mDistilBert	80.35	76.07	78.16
	XLM-R	82.98	81.40	82.18
Đơn ngôn ngữ	viBert4news	79.26	77.48	78.36
	viBert_FPT	80.65	79.12	79.88
	vELECTRA_FPT	83.08	79.54	81.27
	PhoBERT	85.06	87.49	86.53

So sánh với các kết quả được đề cập trong bảng 8 có thể thấy rằng kết quả các mô hình của chúng tôi ở kịch bản có tiền xử lý dữ liệu đều thấp hơn so với mô hình tương tự. Tuy nhiên ở kịch bản không có tiền xử lý dữ liệu, hiệu suất của các mô hình đã có sự cải thiện so với kết quả trong [21]. Mô hình XLM-R của chúng tôi đã có cải thiện 4.91% trên F1-score. Ngoài ra, viBert_FPT và PhoBERT cũng được cải thiện lần lượt 3.24% và 1.1% F1-score. Đây là một kết quả tốt và có sự cải thiện so với công trình trước đây.

4.3.2 Phân loại cảm xúc của khía cạnh

Bảng 9 mô tả kết quả thực nghiệm bài toán phân loại cảm xúc của khía cạnh khi thêm khía cạnh vào sau câu ở cả hai kịch bản: không có tiền xử lý và có tiền xử lý dữ liệu.

Dựa vào bảng kết quả có thể thấy LR là phương pháp máy học có kết quả tốt nhất với 64.42% F1-score ở kịch bản không tiền xử lý và 62.11% F1-score ở kịch bản có tiền xử lý dữ liệu. Ngoài ra, việc sử dụng các bước tiền xử lý dữ liệu chỉ cải thiện duy nhất

F1-score ở mô hình CNB mà không cải thiện trên LR và SVM. Nhìn chung việc tiền xử lý dữ liệu không mang lại hiệu quả.

Ở phương pháp học sâu, mô hình Text-CNN khi kết hợp với BPEmb mang lại kết quả F1-score cao nhất với 71.70%. BPEmb cũng là bộ vector biểu diễn từ mang lại kết quả tốt nhất khi đạt hiệu suất cao nhất ở 3 trên 4 kịch bản thực nghiệm học sâu. Mô hình Bi-GRU và Text-CNN có hiệu suất tương đương trong nhiệm vụ này. Việc tiền xử lý dữ liệu đều làm giảm hiệu suất của các mô hình học sâu.

Trên các mô hình Transformer, kết quả tốt nhất trên nhiệm vụ phân loại cảm xúc của khía cạnh thuộc về mô hình PhoBERT với 82.58% khi không tiền xử lý dữ liệu. Mô hình XLM-R và viBert_FPT lần lượt đạt 80.66% và 74.35% F1-score khi không tiền xử lý dữ liệu. Tương tự như trên nhiệm vụ phát hiện loại khía cạnh, hiệu suất của cả ba mô hình Bert vượt trội hơn so với các phương pháp khác đã áp dụng. Các bước tiền xử lý dữ liệu làm giảm đáng kể hiệu suất của ba mô hình Bert. F1-score trên PhoBERT, XLM-R, viBert_FPT lần lượt giảm 10.54%, 12.66% và 8.8%.

Bảng 9: Kết quả phân loại cảm xúc (khi thêm khía cạnh vào phía sau câu) ở kịch bản không tiền xử lý và kịch bản có tiền xử lý dữ liệu

Phương pháp	Mô hình	Phân loại cảm xúc - Thêm khía cạnh vào sau câu (Không tiền xử lý)			Phân loại cảm xúc - Thêm khía cạnh vào sau câu (Có tiền xử lý)		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Truyền thống	CNB	79.21	42.88	41.41	69.62	52.89	55.52
	Logistic Regression	69.85	61.94	64.42	67.96	59.55	62.11
	SVM	72.78	58.65	61.82	70.31	56.04	59.15
Học sâu + Word Embedding	Text-CNN + fastText	70.58	72.02	70.99	66.07	65.71	65.88
	Text-CNN + w2v_wiki	71.20	67.29	68.76	66.38	61.67	63.39
	Text-CNN + BPEmb	72.28	71.23	71.70	66.78	63.42	64.72
	Text-CNN + Multi	71.28	72.10	71.66	66.87	63.91	65.13
	Bi-GRU + fastText	69.38	70.38	69.85	64.28	64.35	64.31
	Bi-GRU + w2v_wiki	71.12	71.65	71.30	67.37	65.46	66.13
	Bi-GRU + BPEmb	72.17	71.35	71.45	66.40	66.10	66.18
	Bi-GRU + Multi	69.42	70.65	69.96	64.68	63.39	63.97
Transformer	XLM-R	80.58	80.79	80.66	68.58	67.51	68.00
	viBert_FPT	75.08	73.97	74.35	65.39	65.74	65.55
	PhoBERT	82.61	82.56	82.58	72.20	71.95	72.04

Bảng 10: Kết quả phân loại cảm xúc (khi thêm khía cạnh vào phía đầu câu) ở kịch bản không tiền xử lý và kịch bản có tiền xử lý dữ liệu.

Phương pháp	Mô hình	Phát hiện cảm xúc - Thêm khía cạnh vào đầu câu (Không tiền xử lý)			Phát hiện cảm xúc - Thêm khía cạnh vào đầu câu (Tiền xử lý)		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Truyền thống	CNB	74.92	42.83	40.99	69.67	51.85	54.20
	LR	69.71	61.59	64.08	68.54	58.79	61.50
	SVM	74.16	59.26	62.61	69.90	55.71	58.77
Học sâu + Word Embedding	Text-CNN + fastText	72.46	69.67	70.85	67.72	62.64	64.46
	Text-CNN + w2v_wiki	71.05	68.12	69.34	66.35	62.55	64.03
	Text-CNN + BPEmb	71.99	69.92	70.73	67.61	62.59	64.44
	Text-CNN + Multi	73.04	71.28	71.99	67.13	64.23	65.37
	Bi-GRU + fastText	69.69	68.97	69.18	66.07	63.61	64.61
	Bi-GRU + w2v_wiki	72.87	70.92	71.65	67.16	65.87	66.45
	Bi-GRU + BPEmb	71.13	70.83	70.79	66.22	64.98	65.49
	Bi-GRU + Multi	71.53	69.20	70.14	62.76	63.25	62.95
Transformer	XLM-R	81.90	81.48	81.63	69.57	68.39	68.90
	viBert FPT	76.26	75.85	75.98	66.44	66.18	66.28
	PhoBERT	83.20	82.70	82.92	71.00	70.73	70.81

Kết quả thực nghiệm bài toán phân loại cảm xúc của khía cạnh khi thêm khía cạnh vào đầu câu ở cả hai kịch bản: không có tiền xử lý và có tiền xử lý dữ liệu được thể hiện trong bảng 10.

Tương tự như ở bảng 9, LR vẫn là phương pháp máy học có kết quả tốt nhất với 64.08% F1-score ở kịch bản không tiền xử lý và 61.50% F1-score ở kịch bản có tiền xử lý dữ liệu. Việc tiền xử lý dữ liệu tổng quan không mang lại hiệu quả cho phương pháp máy học.

Đối với phương pháp học sâu, mô hình Text-CNN khi kết hợp với Multi mang lại kết quả F1-score cao nhất với 71.99%. Hiệu suất của bộ vector biểu diễn từ Multi và w2v_wiki tốt hơn so với hai bộ còn lại nhưng không nhiều. Mô hình Bi-GRU và Text-CNN có hiệu suất tương đương. Việc tiền xử lý dữ liệu đều làm giảm hiệu suất của các mô hình.

Đối với các mô hình Transformer, hiệu suất của cả ba mô hình Bert vượt trội hơn so với các phương pháp khác đã áp dụng. Kết quả tốt nhất trên nhiệm vụ phân loại cảm xúc của khía cạnh thuộc về mô hình PhoBERT với 82.92% khi không tiền xử lý dữ liệu. Các bước tiền xử lý dữ liệu làm giảm lần lượt 12.11%, 12.73%, 9.7% F1-score trên mô hình PhoBERT, XLM-R và viBert_FPT.

Qua phân tích từ bảng 9 và bảng 10, chúng ta thấy rằng trong tất cả các mô hình được áp dụng chỉ có duy nhất mô hình CNB là nhận được sự cải thiện F1-score khi áp dụng tiền xử lý dữ liệu. Đối với các mô hình còn lại thì việc áp dụng tiền xử lý dữ liệu đều làm giảm hiệu suất của mô hình trên F1-score. Vì vậy chúng ta có thể kết luận rằng đối với bài toán phân loại cảm xúc của khía cạnh thì việc áp dụng các bước tiền xử lý ở phần 4.1 không mang lại hiệu quả đối với cả ba phương pháp: học máy truyền thống, học sâu và transformer.

Mặt khác, nếu so sánh kết quả giữa kịch bản thêm khía cạnh vào đầu câu và thêm vào cuối câu để giải quyết bài toán phân loại cảm xúc của khía cạnh thì việc chọn kịch bản thêm khía cạnh vào đầu bình luận mang lại hiệu quả trên các mô hình Transformer. Cụ thể chỉ có duy nhất một trường hợp mà kết quả ở kịch bản thêm khía cạnh vào đầu bình luận không cải thiện hệ số F1-score so với kịch bản thêm khía cạnh vào cuối câu. Trong 5 trường hợp còn lại của các mô hình Transformer, kết quả F1-score luôn được cải thiện so với khi thêm khía cạnh vào cuối bình luận. Trái lại là đối với các phương pháp máy học truyền thống, việc thêm khía cạnh vào đầu câu chỉ mang lại sự cải thiện F1-score trên 1 trong 6 trường hợp. Vì vậy, việc thêm khía cạnh vào cuối câu để giải quyết bài toán sẽ phù hợp hơn đối với các phương pháp học máy. Cuối cùng, trên các phương pháp học sâu thì kịch bản thêm khía cạnh vào đầu và vào cuối câu bình luận mang lại hiệu suất tương đương.

5 Kết luận và hướng phát triển

Trong bài báo này, chúng tôi đã giới thiệu phương pháp để giải quyết bài toán phân tích cảm xúc dựa theo khía cạnh thông qua hai bài toán con: phát hiện loại khía cạnh và phân loại cảm xúc của khía cạnh. Các phương pháp máy học truyền thống (CNB, Logistic Regression, SVM), học sâu (Text-CNN, Bi-LSTM) và Transformer (PhoBERT, XLM-R, viBert_FPT) được chúng tôi thực nghiệm trên hai bài toán con và thu được kết quả khả quan.

Đối với nhiệm vụ phát hiện loại khía cạnh, kết quả tốt nhất mà chúng tôi đạt được là 87.63% F1-score trên mô hình PhoBERT ở kịch bản không áp dụng các bước tiền xử lý dữ liệu. Kết quả này mang lại sự cải thiện trên F1-score so với công trình trước đó [21] (87.63% so với 86.53%). Đối với phương pháp học sâu, mô hình Bi-GRU có hiệu suất cao hơn so với Text-CNN. Thêm vào đó, bộ vector biểu diễn từ fastText giúp mang lại hiệu suất cao hơn các bộ còn lại khi kết hợp với các phương pháp học sâu. Mặt khác, việc thực nghiệm hai kịch bản tiền xử lý dữ liệu giúp chúng tôi nhận thấy rằng các bước tiền xử lý đã đề cập chỉ mang lại hiệu quả trong các phương pháp máy học truyền thống.

Đối với nhiệm vụ phân loại cảm xúc của khía cạnh, mô hình PhoBERT cũng cho kết quả tốt nhất với giá trị F1-score là 82.92%. Kết quả này nằm trong kịch bản thêm khía cạnh vào đầu câu và không áp dụng các bước tiền xử lý. Đối với hai phương pháp học sâu là Text-CNN và Bi-GRU, chúng có hiệu suất tương đương. Đối với các bộ vector biểu diễn từ, BPEmb là bộ mang lại hiệu quả cao nhất trong kịch bản thêm khía cạnh vào phía sau câu. Mặt khác, việc áp dụng các bước tiền xử lý dữ liệu làm giảm hiệu suất của các mô hình phân loại. Đối với kịch bản thêm khía cạnh vào đầu câu chỉ thực sự mang lại hiệu quả đối với các mô hình transformer trong khi việc thêm khía cạnh vào cuối câu giúp cải thiện hiệu suất của các mô hình máy học truyền thống.

Trong tương lai, chúng tôi sẽ tìm cách để cải thiện hiệu suất của các mô hình hiện tại thông qua việc tinh chỉnh các mô hình Bert. Ngoài ra chúng tôi sẽ tìm hiểu về cách tiếp cận học đa tác vụ để từ đó có thể giải quyết bài toán phân tích cảm xúc dựa theo khía cạnh. Sau đó, so sánh hiệu suất của phương pháp học đa tác vụ so với hiệu suất kết hợp của hai nhiệm vụ con mà chúng tôi đã thực hiện. Mặt khác, với mỗi mô hình, chúng tôi sẽ thực hiện và đánh giá mức độ hiệu quả của từng bước tiền xử lý dữ liệu riêng biệt. Cuối cùng, chúng tôi sẽ tiến hành xây dựng module phân tích khía cạnh dựa theo cảm xúc để có thể ứng dụng trong lĩnh vực kinh doanh nhà hàng tại Việt Nam.

Acknowledgment

Bảng 11: Bảng phân công công việc

Thành viên	Công việc
Trần Trung Hiếu	Tìm hiểu bài toán, code, chạy mô hình, viết báo cáo phần 3.1, 4.3, 5
Lâm Gia Huy	Tìm hiểu bài toán, code, chạy mô hình, viết báo cáo phần 2, 3.2, 4.1, 4.2.
Võ Đình Tứ	Tiền xử lý dữ liệu, viết báo cáo phần tóm tắt nội dung, 1, 3.3

Tài liệu

1. Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.
2. Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230, 2017.
3. Míngqíng Hu and Bing Liu. Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760, 2004.
4. Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: improving rating predictions using review text content. In *WebDB*, volume 9, pages 1–6. Citeseer, 2009.
5. Tun Thura Thet, Jin-Cheon Na, and Christopher SG Khoo. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of information science*, 36(6):823–848, 2010.
6. Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824, 2011.
7. Isidoros Perikos, Konstantinos Kostas, Foteini Grivokostopoulou, and Ioannis Hatzilygeroudis. A system for aspect-based opinion mining of hotel reviews. In *International Conference on Web Information Systems and Technologies*, volume 2, pages 388–394. SCITEPRESS, 2017.
8. Binh Thanh Kieu and Son Bao Pham. Sentiment analysis for vietnamese. In *2010 Second International Conference on Knowledge and Systems Engineering*, pages 152–157. IEEE, 2010.
9. Dang Van Thin, Ngan Luu-Thuy Nguyen, Tri Minh Truong, Lac Si Le, and Duy Tin Vo. Two new large corpora for vietnamese aspect-based sentiment analysis at sentence level. *Transactions on Asian and Low-Resource Language Information Processing*, 20(4):1–22, 2021.
10. Mengting Hu, Shiwan Zhao, Honglei Guo, Renhong Cheng, and Zhong Su. Learning to detect opinion snippet for aspect-based sentiment analysis. *arXiv preprint arXiv:1909.11297*, 2019.
11. Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30, 2016.

12. Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6280–6285, 2019.
13. Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. Stockholm, Sweden, 1999.
14. Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.
15. Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.
16. Y Kim. Convolutional neural networks for sentence classification. new york university. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2014.
17. Dongyang Wang, Junli Su, and Hongbin Yu. Feature extraction and analysis of natural language processing for deep learning english language. *IEEE Access*, 8:46335–46345, 2020.
18. Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.
19. Hoang-Vu Dang and Phuong Le-Hong. A combined syntactic-semantic embedding model based on lexicalized tree-adjoining grammar. *Computer Speech & Language*, 68:101202, 2021.
20. Dat Quoc Nguyen and Anh Tuan Nguyen. Phobert: Pre-trained language models for vietnamese. *arXiv preprint arXiv:2003.00744*, 2020.
21. Dang Van Thin, Lac Si Le, Vu Xuan Hoang, and Ngan Luu-Thuy Nguyen. Investigating monolingual and multilingual bertmodels for vietnamese aspect category detection. *arXiv preprint arXiv:2103.09519*, 2021.