

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Nguyễn Thị Thuý Hiền**

**MẠNG TÍCH CHẬP ĐỒ THỊ  
NÂNG CAO NGŨ NGHĨA VÀ CÚ PHÁP CHO  
PHÂN TÍCH QUAN ĐIỂM MỨC KHÍA CẠNH  
VÀ ÁP DỤNG VÀO DỮ LIỆU TIẾNG VIỆT**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Khoa học máy tính**

**HÀ NỘI - 2023**

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Thị Thuý Hiền

**MẠNG TÍCH CHẬP ĐỒ THỊ  
NÂNG CAO NGŨ NGHĨA VÀ CÚ PHÁP CHO  
PHÂN TÍCH QUAN ĐIỂM MỨC KHÍA CẠNH  
VÀ ÁP DỤNG VÀO DỮ LIỆU TIẾNG VIỆT**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Khoa học máy tính**

**Cán bộ hướng dẫn: PGS.TS. Hà Quang Thụy**

**Cán bộ đồng hướng dẫn: ThS. Nguyễn Thị Cẩm Vân**

**HÀ NỘI - 2023**

**VIETNAM NATIONAL UNIVERSITY, HANOI**  
**UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**Nguyen Thi Thuy Hien**

**SYNTACTIC AND SEMANTIC ENHANCED  
GRAPH CONVOLUTIONAL NETWORK FOR  
ASPECT-BASED SENTIMENT ANALYSIS  
AND APPLIED FOR VIETNAMESE**

**Major: Computer Science**

**Supervisor: Assoc. Prof. Ha Quang Thuy**

**Co-Supervisor: Master Nguyen Thi Cam Van**

**HANOI - 2023**

# TÓM TẮT

Khai phá quan điểm mức khía cạnh là bài toán trích xuất thông tin hướng mục tiêu, đã và đang nhận được sự quan tâm lớn của cộng đồng NLP. Phân lớp quan điểm khía cạnh là một trong những nhiệm vụ quan trọng liên quan đến bài toán này. Những phương pháp đầu tiên cho nhiệm vụ phân lớp quan điểm khía cạnh chủ yếu dựa vào mạng nơ ron. Tiếp theo đó, nhiều nghiên cứu bắt đầu sử dụng cơ chế chú ý với mạng RNN để xem xét tương quan ngữ nghĩa giữa khía cạnh và ngữ cảnh câu (Bahdanau và cộng sự - 2014), tuy nhiên cơ chế này dễ đưa ra dự đoán sai do ảnh hưởng bởi nhiễu trong câu. Gần đây, nhờ việc cải thiện độ chính xác của các trình phân tích cú pháp mà các phương pháp sử dụng mạng tích chập đồ thị dựa trên cây phụ thuộc ngày càng được chú ý (Zhang và cộng sự - 2019, Huang và Carley - 2019). Phương pháp này vẫn tồn tại nhược điểm khi các văn bản đánh giá quan điểm không tuân theo kiến trúc ngữ pháp chung nào.

Để khắc phục các hạn chế đã đề cập, nhiều nghiên cứu đã bắt đầu tích hợp cả thông tin cấu trúc cú pháp và tương quan ngữ nghĩa của câu để xác định xu hướng quan điểm (Li và cộng sự - 2021). Thực nghiệm cho thấy việc kết hợp hai khía cạnh này thực sự bổ sung cho nhau và cải thiện hiệu suất mô hình một cách đáng kể. Mô hình Mạng tích chập đồ thị nâng cao ngữ nghĩa và cú pháp được đề xuất bởi Zheng Zhang và cộng sự (2022) đã tận dụng cả cấu trúc cú pháp và cơ chế chú ý để phân loại quan điểm. Dựa trên ý tưởng đó, khoá luận đã áp dụng cho bộ dữ liệu tiếng Việt UIT-ViSFD bởi Phan và cộng sự (2021). Bộ dữ liệu gồm 11122 bản ghi, chứa các đánh giá về điện thoại di động, được thu thập từ trang thương mại điện tử. Kết quả đánh giá cho thấy mô hình thực sự mang lại hiệu quả khả quan khi phân loại quan điểm trên tập dữ liệu này.

**Từ khóa:** Khai phá quan điểm mức khía cạnh, Khai phá quan điểm, Mạng tích chập đồ thị, UIT-ViSFD

# LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc nhất đến Thầy giáo, Phó giáo sư Tiến sĩ Hà Quang Thụy và Cô giáo, Thạc sĩ Nguyễn Thị Cẩm Vân. Thầy và cô đã luôn tận tình giúp đỡ, chỉ bảo và tạo động lực giúp em hoàn thiện đề tài một cách tốt nhất trong suốt thời gian vừa qua.

Tiếp theo, em xin gửi lời cảm ơn sâu sắc đến các thầy cô Trường đại học Công nghệ - Đại học Quốc gia Hà Nội. Các thầy cô đã chia sẻ nhiều kinh nghiệm cũng như truyền đạt nhiều kiến thức quý giá, giúp em hoàn thiện hơn trên con đường sự nghiệp của mình.

Em cũng xin gửi lời cảm ơn sâu sắc đến thầy cô, anh chị và các bạn trong Phòng thí nghiệm Khoa học dữ liệu và Công nghệ Tri thức (DS&KTLab), chị Phan Thị Huyền Trang (Yeungnam University, Hàn Quốc) vì đã tạo điều kiện, định hướng và giúp đỡ em rất nhiều trong học tập và cũng như khóa luận này.

Cuối cùng, em xin gửi lời cảm ơn chân thành đến gia đình, bạn bè, tập thể lớp K64-CACLC4, những người luôn sẵn sàng chia sẻ và giúp đỡ em trên giảng đường đại học và trong cuộc sống.

Em xin chân thành cảm ơn!

Sinh viên

**Nguyễn Thị Thuý Hiền**

# LỜI CAM ĐOAN

Tôi xin cam đoan kết quả trình bày trong khóa luận hoàn toàn là tôi thực hiện dưới sự hướng dẫn của PSG. TS Hà Quang Thụy và ThS. Nguyễn Thị Cẩm Vân.

Tôi xin khẳng định tất cả các tham khảo từ những nghiên cứu liên quan đều được chú thích rõ ràng và nêu rõ nguồn gốc trong mục tài liệu tham khảo. Một lần nữa tôi xin khẳng định tôi không có sao chép tài liệu, công trình nghiên cứu của người khác mà không có trích dẫn cụ thể trong mục tài liệu tham khảo.

Tôi xin chịu hoàn toàn trách nhiệm về lời cam đoan này.

Sinh viên

**Nguyễn Thị Thuý Hiền**

# MỤC LỤC

<b>TÓM TẮT</b> . . . . .	<b>iv</b>
<b>LỜI CẢM ƠN</b> . . . . .	<b>v</b>
<b>LỜI CAM ĐOAN</b> . . . . .	<b>vi</b>
<b>MỤC LỤC</b> . . . . .	<b>vii</b>
<b>DANH SÁCH THUẬT NGỮ</b> . . . . .	<b>ix</b>
<b>DANH SÁCH HÌNH VẼ</b> . . . . .	<b>x</b>
<b>DANH SÁCH BẢNG BIỂU</b> . . . . .	<b>xi</b>
<b>MỞ ĐẦU</b> . . . . .	<b>1</b>
<b>1 KHAI PHÁ QUAN ĐIỂM MỨC KHÍA CẠNH DỰA TRÊN MẠNG TÍCH CHẬP ĐỒ THỊ</b> . . . . .	<b>3</b>
1.1 Khai phá quan điểm mức khía cạnh . . . . .	3
1.1.1 Định nghĩa khai phá quan điểm mức khía cạnh . . . . .	4
1.1.2 Nhiệm vụ cơ bản trong phân tích quan điểm mức khía cạnh . . . . .	4
1.1.3 Phương pháp giải quyết bài toán phân tích quan điểm mức khía cạnh . . . . .	6
1.2 Mạng tích chập đồ thị . . . . .	8
1.2.1 Định nghĩa đồ thị . . . . .	8
1.2.2 Mạng tích chập đồ thị . . . . .	10
1.2.3 Phân loại mạng GCN . . . . .	11
1.3 Mạng tích chập đồ thị cho khai phá quan điểm mức khía cạnh . . . . .	13
1.4 Bài toán khóa luận . . . . .	15
1.5 Tóm tắt chương 1 . . . . .	16

<b>2</b>	<b>HAI MÔ HÌNH MẠNG TÍCH CHẬP ĐỒ THỊ NỀN TẢNG</b>	<b>17</b>
2.1	Mô hình mạng tích chập đồ thị đặc tả khía cạnh	17
2.1.1	Giới thiệu	17
2.1.2	Kiến trúc mô hình	18
2.2	Mô hình mạng tích chập đồ thị kép	22
2.2.1	Giới thiệu	22
2.2.2	Kiến trúc mô hình	23
2.3	Ý tưởng khóa luận	26
2.4	Tóm tắt chương 2	27
<b>3</b>	<b>MÔ HÌNH KHOÁ LUẬN</b>	<b>28</b>
3.1	Mô hình khoá luận	28
3.2	Tiền xử lý và chuẩn hoá	28
3.3	Mô hình nhúng từ	29
3.4	Bộ mã hoá	31
3.4.1	Mã hoá ngữ nghĩa	31
3.4.2	Mã hoá cú pháp	33
3.4.3	Lớp GCN	34
3.5	Tóm tắt chương 3	36
<b>4</b>	<b>THỰC NGHIỆM VÀ ĐÁNH GIÁ</b>	<b>37</b>
4.1	Kế hoạch thực nghiệm	37
4.2	Phần mềm và phần cứng	37
4.3	Bộ dữ liệu tiếng Việt UIT-ViSFD và UIT-ViSD4SA	39
4.4	Cài đặt môi trường thực nghiệm	41
4.4.1	Tiền xử lý dữ liệu	41
4.4.2	Kịch bản thực nghiệm	42
4.5	Đánh giá kết quả thực nghiệm	42
4.5.1	Độ đo đánh giá	42
4.5.2	Cài đặt cấu hình	43
4.6	Kết quả thực nghiệm	45
	<b>KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU TIẾP THEO</b>	<b>47</b>
	<b>TÀI LIỆU THAM KHẢO</b>	<b>48</b>



# DANH SÁCH THUẬT NGỮ

STT	Tên Tiếng Anh	Tên viết tắt	Tên Tiếng Việt
1	Natural processing language	NLP	Xử lý ngôn ngữ tự nhiên
2	Aspect-Based Sentiment Analysis	ABSA	Phân tích quan điểm mức khía cạnh
3	Graph Convolutional Network	GCN	Mạng tích chập đồ thị
4	Aspect-specific Graph Convolutional Network	ASGCN	Mạng tích chập đồ thị đặc tả khía cạnh
5	Dual Graph Convolutional Network	DualGCN	Mạng tích chập đồ thị kép
6	Syntactic and Semantic Enhanced Graph Convolutional Network	SSEGCN	Mạng tích chập đồ thị nâng cao ngữ nghĩa và cú pháp
7	Pre-trained Language Models	PLM	Mô hình ngôn ngữ được huấn luyện trước
8	Part-of Speech	PoS	Nhãn từ loại
9	Token		Đơn vị mã hoá
10	Pooling		Tổng hợp
11	Transformer		Bộ biến đổi

# DANH SÁCH HÌNH VẼ

1.1	Phân loại nhiệm vụ trong ABSA . . . . .	5
1.2	Mạng tích chập đồ thị GCN . . . . .	9
1.3	Ví dụ về lớp GCN [11] . . . . .	11
1.4	Phân loại mạng GCN cho ABSA . . . . .	13
1.5	Cấu trúc phân loại mạng GCN cho ABSA. a) Mạng GCN dựa trên tri thức đơn; b) Mạng GCN dựa trên tri thức kép; c) Mạng GCN dựa trên đa tri thức [6] . . . . .	14
2.1	Kiến trúc mạng tích chập đồ thị đặc tả khía cạnh [11] . . . . .	19
2.2	Kiến trúc mạng tích chập đồ thị kép [4] . . . . .	23
3.1	Kiến trúc mô hình áp dụng [13] . . . . .	29
4.1	Ma trận nhầm lẫn trên bộ dữ liệu UIT-ViSFD . . . . .	45
4.2	Ma trận nhầm lẫn trên bộ dữ liệu UIT-ViSD4SA . . . . .	46

# DANH SÁCH BẢNG BIỂU

4.1	Danh sách công cụ và phần mềm . . . . .	38
4.2	Các trường dữ liệu trong UIT-ViSFD . . . . .	39
4.3	Các danh mục khía cạnh thuộc về chủ đề điện thoại di động trong UIT-ViSFD . . . . .	40
4.4	Thống kê số lượng dữ liệu theo khía cạnh và theo nhãn quan điểm trong 3 tập dữ liệu huấn luyện, xác thực, kiểm thử . . . . .	40
4.5	Kết quả thực nghiệm trên bộ dữ liệu UIT-ViSFD . . . . .	45
4.6	Kết quả thực nghiệm trên bộ dữ liệu UIT-ViSD4SA . . . . .	45
4.7	Kết quả so sánh với các baseline khác trên bộ dữ liệu UIT-ViSFD . . . . .	46
4.8	Kết quả so sánh với các baseline khác trên bộ dữ liệu UIT-ViSD4SA . . . . .	46

# MỞ ĐẦU

Phân tích quan điểm mức khía cạnh là nhiệm vụ phân tích quan điểm chi tiết, đã và đang nhận được sự quan tâm lớn trong cộng đồng NLP những năm gần đây. Cùng với sự phát triển và ngày càng phổ biến của các nền tảng thương mại điện tử, việc nắm bắt và hiểu rõ ý kiến của người dùng về sản phẩm, dịch vụ cụ thể sẽ giúp doanh nghiệp định hình chiến lược marketing hữu ích cũng như cải thiện chất lượng sản phẩm. Bên cạnh đó, số lượng lớn văn bản phi cấu trúc có được từ đánh giá của khách hàng là biển thông tin tiềm năng để khai thác và phân tích.

Mục tiêu chính của phân tích quan điểm mức khía cạnh là xác định các yếu tố quan điểm trong câu. Những nghiên cứu đầu tiên trong ABSA chỉ tập trung vào việc xác định một yếu tố quan điểm đơn lẻ. Trong những năm gần đây, nhiệm vụ ABSA kết hợp đã được nghiên cứu và thực nghiệm nhằm nắm bắt thông tin quan điểm mức khía cạnh một cách hoàn thiện hơn. Bên cạnh việc trích xuất nhiều yếu tố quan điểm, nhiệm vụ ABSA kết hợp còn cung cấp mối quan hệ phụ thuộc và sự tương ứng giữa chúng.

Bài toán khoá luận tập trung vào phân lớp quan điểm khía cạnh, mục tiêu chính là xác định xu hướng quan điểm của một khía cạnh cụ thể trong câu. Khía cạnh ở đây có thể là thuật ngữ khía cạnh hay danh mục khía cạnh. Bất kể là loại khía cạnh nào, bài toán cần tìm cách khai thác mối liên hệ giữa khía cạnh và ngữ cảnh câu để phân loại quan điểm.

Nhiều bộ dữ liệu điểm chuẩn và phương pháp đã được đề xuất để ứng dụng cho bài toán phân tích quan điểm mức khía cạnh. Những nghiên cứu đầu tiên tập trung vào mục tiêu mô hình hoá sự tương tác giữa khía cạnh và ngữ cảnh trong câu. Tiếp theo sau đó, một số phương pháp đã tận dụng cơ chế chú ý để nắm bắt mối liên hệ giữa các thành phần trong câu đối với một khía cạnh cụ thể. Một hướng nghiên cứu khác tập trung vào mô hình hoá cấu trúc cú pháp của câu để dự đoán vị trí cú pháp giữa khía cạnh và ý kiến tương ứng thường biểu thị định hướng quan điểm. Nhiều nghiên cứu đã tận dụng cây cú pháp làm đặc trưng cho bài toán phân loại quan điểm. Nhờ có sự phát triển của

mạng nơon học sâu, cây cú pháp với độ chính xác cao đã cải thiện hiệu quả của các mô hình phân loại dựa trên cây cú pháp. Trong đó, phương pháp sử dụng mạng tích chập đồ thị là một trong những phương pháp tiên tiến được nghiên cứu gần đây. Phương pháp này là mở rộng của các mô hình học sâu cho dữ liệu đồ thị. Bên cạnh việc thiết kế các mô hình phục vụ cho các nhiệm vụ riêng biệt, sự ra đời của mô hình ngôn ngữ được huấn luyện trước như BERT cũng đã cải thiện hiệu suất đáng kể.

Khoá luận tiến hành nghiên cứu và ứng dụng phương pháp sử dụng mạng tích chập đồ thị cho bài toán phân tích quan điểm mức khía cạnh. Nghiên cứu này đã được sử dụng rộng rãi những năm gần đây, khảo sát của Phan và cộng sự (2023) [6] là một trong những khảo sát tốt và cập nhật nhất về phương pháp này.

Cụ thể, khoá luận ứng dụng Mô hình Mạng tích chập đồ thị nâng cao ngữ nghĩa và cú pháp, gọi tắt là SSEGCN, được đề xuất năm 2022 bởi Zheng Zhang và cộng sự [13]. Mô hình này sử dụng cả thông tin cú pháp và ngữ nghĩa dựa trên cây đồ thị để dự đoán quan điểm. Bên cạnh cơ chế tự chú ý, các tác giả cũng đề xuất cơ chế chú ý nhận thức khía cạnh để học được không chỉ quan hệ ngữ nghĩa liên quan đến khía cạnh mà cả ngữ nghĩa toàn cục của câu. Ngoài ra, ma trận mặt nạ cú pháp cũng được xây dựng để nắm bắt thông tin cú pháp toàn diện. Các biểu diễn nút cũng được nâng cao hơn bằng mạng tích chập đồ thị trên ma trận điểm chú ý.

Với phương pháp nêu trên, khoá luận tiến hành chạy thực nghiệm trên bộ dữ liệu tiếng Việt UIT-ViSFD bởi Phan và cộng sự (2021) [7].

Khoá luận được trình bày theo các chương sau:

**Chương 1. Khai phá quan điểm mức khía cạnh dựa trên mạng tích chập đồ thị:** Chương này sẽ trình bày khái quát về bài toán khai phá quan điểm mức khía cạnh, phương pháp sử dụng mạng tích chập đồ thị để giải quyết bài toán này và phát biểu bài toán khoá luận.

**Chương 2. Hai mô hình mạng tích chập đồ thị nền tảng:** Giới thiệu hai mô hình dựa trên mạng tích chập đồ thị cho nhiệm vụ phân loại quan điểm khía cạnh.

**Chương 3. Mô hình khoá luận:** Giới thiệu mô hình phân lớp quan điểm mức khía cạnh tiếng Việt dựa trên mạng nơon đồ thị và áp dụng với dữ liệu tiếng Việt.

**Chương 4. Thực nghiệm và đánh giá:** Trình bày về kịch bản thực nghiệm của mô hình, kết quả, đánh giá mô hình và đưa ra nhận xét.

**Phần kết luận:** Tóm lược kết quả đạt được, hạn chế trong khoá luận và định hướng phát triển khoá luận trong tương lai.

# Chương 1

## KHAI PHÁ QUAN ĐIỂM MỨC KHÍA CẠNH DỰA TRÊN MẠNG TÍCH CHẬP ĐỒ THỊ

Ở chương đầu tiên, khoá luận sẽ trình bày tổng quát về bài toán khai phá quan điểm mức khía cạnh, phương pháp sử dụng mạng tích chập đồ thị để giải quyết nhiệm vụ này và phát biểu bài toán khoá luận.

### 1.1 Khai phá quan điểm mức khía cạnh

Các phương tiện xã hội ngày càng phát triển dẫn đến sự sản sinh một lượng lớn các văn bản bày tỏ quan điểm và ý kiến về các sản phẩm, dịch vụ. Trong đó, thông tin từ văn bản được phân thành hai loại chính: thông tin dữ kiện và thông tin quan điểm [2]. Thông tin dữ kiện bày tỏ khách quan về các thực thể và thuộc tính tương ứng, trong khi thông tin quan điểm bày tỏ ý kiến chủ quan của con người, thể hiện đánh giá hay quan điểm đối với thực thể. Thông tin quan điểm bao gồm thông tin dữ kiện về người đưa ra quan điểm, đối tượng quan điểm và quan điểm đối với đối tượng đó. Đối tượng quan điểm có thể là thực thể hoặc các khía cạnh của thực thể tương ứng. Phân tích quan điểm mức khía cạnh là quy trình thu thập, xử lý, phân tích, suy luận và tổng hợp các quan điểm chủ quan của các thực thể nằm trong văn bản ở mức khía cạnh.

### 1.1.1 Định nghĩa khai phá quan điểm mức khía cạnh

Theo Zhang và cộng sự [12], một quan điểm là một bộ năm bao gồm  $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ . Trong đó,  $e_i$  là một thực thể với thực thể có thể là một sản phẩm, dịch vụ, sự kiện, vấn đề, con người hay tổ chức;  $a_{ij}$  là một khía cạnh tương ứng của thực thể  $e_i$ , khía cạnh chính là các thuộc tính của thực thể;  $s_{ijkl}$  là quan điểm được đưa ra bởi người  $h_k$  tại thời điểm  $t_l$ . Bài toán khai phá quan điểm mức khía cạnh đối với tài liệu  $d$  quy về tìm ra tất cả các bộ năm quan điểm xuất hiện trong  $d$ . Trong thực tế, để bài toán trở nên đơn giản, có thể xem thực thể là một khía cạnh "chung" và gọi chung cả khía cạnh và thực thể là khía cạnh. Tuy vậy khi xác định quan điểm cần ngầm hiểu khía cạnh đó thuộc về thực thể nào. Ngoài ra, người đưa ra quan điểm và thời gian quan điểm được đưa ra cũng được hiểu ngầm định. Do đó, bài toán khai phá quan điểm mức khía cạnh thường quy về tìm cặp (*đích quan điểm, quan điểm*). Trong đó, *đích quan điểm* có thể là thuật ngữ khía cạnh (a) hoặc danh mục khía cạnh (c), *quan điểm* có thể là thuật ngữ ý kiến đối với đích quan điểm (o) hoặc phân lớp quan điểm (p). Cụ thể:

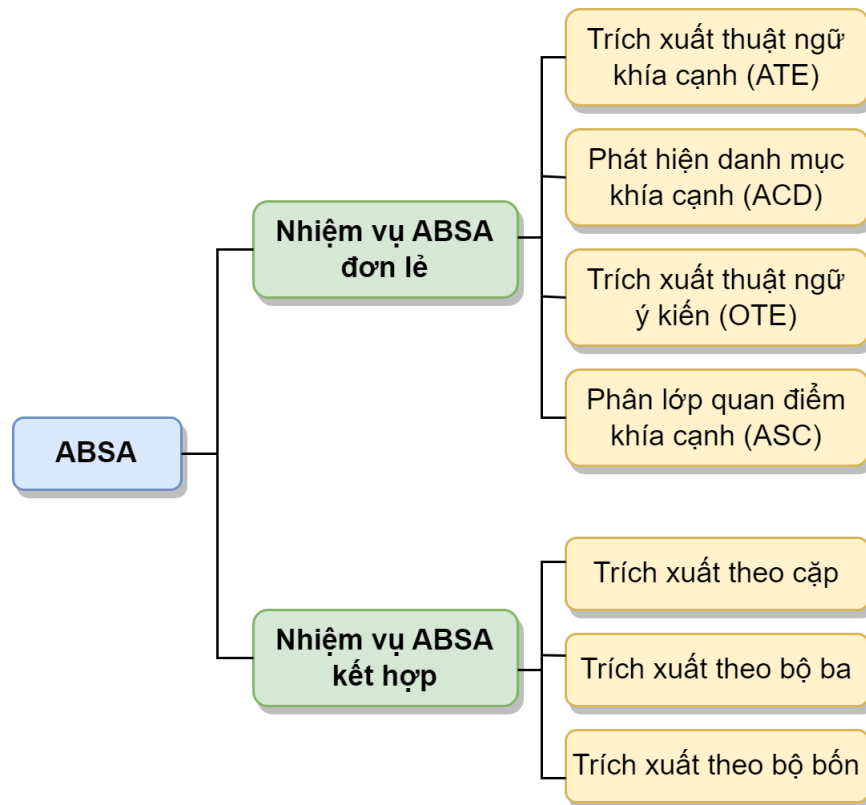
- **Danh mục khía cạnh (c):** Thuộc một trong tập các danh mục khía cạnh  $C$  được định nghĩa trước tùy theo lĩnh vực nghiên cứu. Ví dụ, "pin" hay "camera" là các danh mục khía cạnh cho lĩnh vực điện thoại di động. Một khía cạnh chỉ thuộc về một danh mục duy nhất.
- **Thuật ngữ khía cạnh (a):** Là khía cạnh xuất hiện trực tiếp trong câu, cũng chính là từ hoặc cụm từ nằm trong câu. Ví dụ, "loa" trong câu "loa nghe hơi nhỏ" là một khía cạnh mà qua đó quan điểm được thể hiện. Khi khía cạnh không được thể hiện rõ, có thể sử dụng nhãn đặc biệt "null".
- **Thuật ngữ ý kiến (o):** Là thái độ được bộc lộ đối với khía cạnh. Cùng với ví dụ như trên, "hơi nhỏ" là thuật ngữ ý kiến dành cho khía cạnh "loa".
- **Xu hướng quan điểm (p):** Mô tả quan điểm đối với khía cạnh, thường thuộc một trong ba phân loại tích cực, tiêu cực, trung lập.

### 1.1.2 Nhiệm vụ cơ bản trong phân tích quan điểm mức khía cạnh

Nhìn chung, để giải quyết bài toán ABSA cần thực hiện một số nhiệm vụ cơ bản. Các nhiệm vụ này có thể được phân thành nhiệm vụ đơn lẻ hoặc kết hợp. Nhiệm vụ đơn lẻ liên quan đến xác định các yếu tố quan điểm riêng rẽ, cụ thể bao gồm trích xuất thuật ngữ khía cạnh, phát hiện danh mục khía cạnh, trích xuất thuật ngữ ý kiến và phân loại

quan điểm khía cạnh. Nhiệm vụ kết hợp thực hiện trích xuất đồng thời từ hai yếu tố quan điểm trở lên và xác định mối quan hệ giữa chúng, ví dụ đồng thời phát hiện danh mục khía cạnh và phân lớp quan điểm cho danh mục đó (trích xuất theo cặp).

Hình vẽ 1.1 mô tả phân loại nhiệm vụ trong ABSA, bao gồm các nhiệm vụ ABSA đơn lẻ chỉ liên quan đến một yếu tố quan điểm hay nhiệm vụ ABSA kết hợp với từ hai yếu tố quan điểm trở lên.



Hình 1.1: Phân loại nhiệm vụ trong ABSA

Chương này chỉ đề cập đến một số nhiệm vụ ABSA đơn lẻ cơ bản, làm nền tảng để phát triển các nhiệm vụ sau này. Các nhiệm vụ ABSA kết hợp liên quan đến tác vụ trích xuất đồng thời nhiều yếu tố quan điểm và mối quan hệ giữa chúng.

### Nhiệm vụ ABSA đơn lẻ

- **Trích xuất thuật ngữ khía cạnh (ATE):** Đây là nhiệm vụ nền tảng của phân tích quan điểm mức khía cạnh. Nhiệm vụ này yêu cầu trích xuất tất cả các khía cạnh được biểu thị quan điểm trong câu. Ví dụ, với câu "Pin ổn, giá lại còn rẻ, tuy nhiên loa nghe hơi nhỏ" thì "pin", "giá" và "loa" là ba khía cạnh cần trích xuất. Khi câu đề cập đến quan điểm chung chung, ví dụ "mọi thứ đều ok" thì khía cạnh ở đây là "tổng quát". Mặc dù nhiệm vụ này xem chung cả thực thể và khía cạnh là một, cần



lưu ý khía cạnh được trích xuất thuộc về thực thể nào.

- **Phát hiện danh mục khía cạnh (ACD):** Nhiệm vụ này yêu cầu xác định các danh mục khía cạnh được đề cập đến trong câu, trong đó các danh mục thuộc về một bộ danh mục được định nghĩa trước và thường thuộc về lĩnh vực cụ thể. So sánh với nhiệm vụ ATE, ACD có lợi hơn bởi vì ATE dự đoán từng thuật ngữ khía cạnh riêng lẻ trong khi danh mục dự đoán của ACD có thể được coi là dự đoán tổng hợp. Ngoài ra ACD có thể xác định mục tiêu ý kiến kể cả khi chúng không được đề cập rõ ràng.
- **Trích xuất thuật ngữ ý kiến (OTE):** Nhiệm vụ này nhằm xác định ý kiến được thể hiện đối với khía cạnh. Vì ý kiến và khía cạnh thường đi cùng với nhau, chỉ trích xuất thuật ngữ ý kiến mà bỏ qua xem xét khía cạnh tương ứng thường vô nghĩa. Vì thế, tùy thuộc vào thuật ngữ khía cạnh xuất hiện ở đầu vào hay đầu ra mà OTE có thể được chia thành hai nhiệm vụ: (1) Đồng trích xuất ý kiến quan điểm (AOCE) hay (2) Trích xuất từ ý kiến đối với mục tiêu (TOWE). AOCE cố gắng dự đoán thuật ngữ khía cạnh và ý kiến cùng lúc. Mặc dù hai yếu tố quan điểm được đề cập, AOCE vẫn là nhiệm vụ ABSA đơn lẻ vì mối quan hệ phụ thuộc giữa hai yếu tố không được xem xét. Mặt khác, TOWE có mục tiêu trích xuất thuật ngữ ý kiến tương ứng với một khía cạnh cụ thể trong văn bản.
- **Phân loại quan điểm khía cạnh (ASC):** Bài toán này yêu cầu dự đoán xu hướng quan điểm cho một khía cạnh cụ thể trong một câu. Nói chung, khía cạnh có thể là thuật ngữ khía cạnh hoặc danh mục khía cạnh, dẫn đến hai bài toán trong ASC: phân loại quan điểm dựa trên thuật ngữ khía cạnh và phân loại quan điểm dựa trên danh mục khía cạnh. Đối với nhiệm vụ xác định quan điểm cho thuật ngữ khía cạnh, có thể khai thác thêm thông tin vị trí của khía cạnh trong câu. Mặc dù giữa hai bài toán này có một số khác biệt, chúng đều quy về câu hỏi làm thế nào để khai thác mối liên hệ giữa khía cạnh (thuật ngữ/ danh mục) và ngữ cảnh câu. Trên thực tế, một số nghiên cứu xem xét đồng thời hai nhiệm vụ này và giải quyết chúng với cùng một mô hình.

### 1.1.3 Phương pháp giải quyết bài toán phân tích quan điểm mức khía cạnh

Các phương pháp chính được sử dụng cho nhiệm vụ ABSA cho đến hiện tại có thể phân thành bốn loại: dựa trên tri thức, dựa trên học máy, dựa trên kết hợp và được sử dụng rộng rãi gần đây là dựa trên mạng tích chập đồ thị.

## **Dựa trên tri thức**

Các phương pháp dựa trên tri thức sử dụng cơ sở tri thức làm kỹ thuật chính để xác định các đặc trưng quan trọng cho phân tích quan điểm. Cơ sở tri thức thường được định nghĩa là kho lưu trữ thông tin, bao gồm các bộ quy tắc, quan hệ và giả định mà hệ thống máy tính có thể sử dụng để xác định quan điểm của các khía cạnh. Các phương pháp dựa trên tri thức phụ thuộc mạnh mẽ vào việc xây dựng các cơ sở tri thức cho các biểu diễn đầu vào.

Về ưu điểm, các phương pháp dựa trên tri thức thường sử dụng các kỹ thuật tương đối đơn giản, dẫn đến các phương pháp ABSA có tính minh bạch cao. Bên cạnh đó, phương pháp dựa trên tri thức không yêu cầu huấn luyện.

Về nhược điểm, xây dựng tri thức đòi hỏi thời gian đáng kể và hiệu suất của các phương pháp dựa trên tri thức thường thấp.

## **Dựa trên học máy**

Các phương pháp học máy sử dụng quy trình huấn luyện làm kỹ thuật chính để trích xuất các mẫu từ tập dữ liệu để phân biệt giữa các lớp quan điểm. Các phương pháp này phụ thuộc rất nhiều vào việc xây dựng các tập dữ liệu huấn luyện là các vector đặc trưng và các nhãn chính xác tương ứng của chúng.

Về ưu điểm, các phương pháp học máy học các mẫu đặc trưng của miền từ văn bản và cải thiện dần hiệu suất. Điều này dẫn đến hiệu suất của các phương pháp học máy thường cao hơn so với các phương pháp dựa trên tri thức.

Tuy nhiên, các phương pháp học máy thường yêu cầu tập dữ liệu huấn luyện lớn. Ngoài ra, phương pháp này cũng phụ thuộc vào dữ liệu huấn luyện dành riêng cho miền. Các mô hình đạt được kết quả tốt hiện tại đang tuân theo giả định dữ liệu huấn luyện và kiểm thử đến từ cùng phân phối (cùng miền). Điều này dẫn đến cần huấn luyện lại mô hình khi phân phối của dữ liệu thay đổi. Tuy vậy, việc này thường rất tốn kém và có thể bất khả thi khi phải thu thập thêm một lượng lớn dữ liệu được gán nhãn, đặc biệt nhiệm vụ ABSA lại yêu cầu chú thích ở mức chi tiết.

## **Dựa trên kết hợp**

Các phương pháp kết hợp thường được sử dụng để huấn luyện các mô hình ngôn ngữ lớn chứa nhiều ngôn ngữ khác nhau. Chúng giải quyết vấn đề ngôn ngữ lớn bằng cách sử dụng cơ sở tri thức để bổ sung thêm kiến thức cho các mô hình học máy nhằm bù đắp cho việc thiếu dữ liệu huấn luyện. Các nghiên cứu này là sự kết hợp của cả phương pháp học máy và phương pháp dựa trên tri thức sử dụng các cơ chế khác nhau. Hai cơ

chế phổ biến được sử dụng là một cơ sở tri thức để xác định các đặc trưng để huấn luyện các thuật toán học máy và các mô hình học máy, dựa trên tri thức riêng biệt dưới dạng phân loại tuần tự hoặc song song.

Về ưu điểm, các phương pháp kết hợp tích hợp ngữ cảnh của các từ quan điểm dựa trên phân tích tri thức với sự linh hoạt của các phương pháp học máy để loại bỏ sự mơ hồ khi phân loại quan điểm. Ngoài ra, phương pháp kết hợp có thể khắc phục những hạn chế của cả hai phương pháp kết hợp và khai thác lợi ích của chúng. Do đó, các phương pháp này có thể tích hợp hiệu suất cao của các phương pháp học máy và tính ổn định của các phương pháp dựa trên tri thức để cải thiện hiệu suất của chúng.

Về nhược điểm, việc tích hợp các mô hình học máy và dựa trên tri thức để xây dựng một phương pháp kết hợp thường khó khăn. Ngoài ra, hiệu suất cũng phụ thuộc vào tri thức nhận được từ các mô hình dựa trên tri thức.

### **Dựa trên mạng tích chập đồ thị**

Các phương pháp dựa trên GCN được xây dựng bằng cách sử dụng các thuật toán học sâu trên các cấu trúc đồ thị. Các phương pháp dựa trên GCN về cơ bản bắt đầu bằng cách xây dựng cấu trúc dữ liệu đồ thị để biểu diễn các yếu tố và mối quan hệ về các khía cạnh, mục tiêu và quan điểm trong văn bản. Các biểu diễn đồ thị này sau đó được đưa vào các lớp tích chập của GCN.

Ưu điểm của phương pháp này là chúng thể hiện tốt nếu đồ thị phù hợp với nhiệm vụ, trực quan và dễ diễn giải.

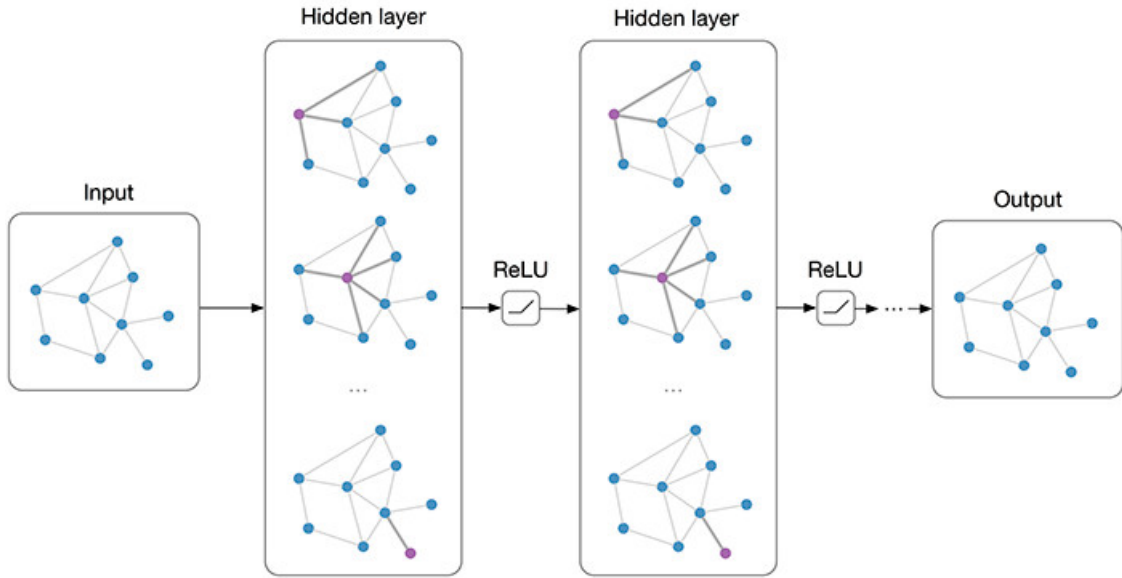
Về nhược điểm, hiệu suất của các phương pháp dựa trên GCN phụ thuộc nhiều vào cấu trúc đồ thị và trọng số cạnh. Ngoài ra, hiệu suất cũng khá thấp nếu đồ thị không phù hợp với nhiệm vụ.

## **1.2 Mạng tích chập đồ thị**

### **1.2.1 Định nghĩa đồ thị**

Một đồ thị  $G = (V, E, A)$  bao gồm  $m$  nút  $V$ , một tập  $n$  cạnh  $E$  và ma trận kề  $A \in \mathbb{R}^{m \times m}$  chứa các trọng số của cạnh  $A_{ij}$ . Giá trị của  $A_{ij}$  được xác định như sau:

$$A_{ij} = \begin{cases} a & \text{nếu } v_i, v_j \in V; e_{ij} \in E \\ 1 & \text{nếu } v_i = v_j \\ 0 & \text{các trường hợp còn lại} \end{cases} \quad (1.1)$$



Hình 1.2: Mạng tích chập đồ thị GCN

Trong đó:  $a$  là trọng số cạnh và  $a = 1$  nếu đồ thị không có trọng số. Một nút  $v_i$  có một tập các nút kề  $N(i)$  được xác định bởi  $N(i) = \{v_j \in V | (v_i, v_j) \in E\}$ . Đồ thị  $G$  có một ma trận đặc trưng nút  $N \in \mathbb{R}^{m \times d_n}$ , với mỗi hàng  $n \in N$  biểu thị một vector đặc trưng của nút  $v_i \in V$ . Cho  $D$  là ma trận chéo và là ma trận bậc của  $A$ , ta có  $D_{ii} = \sum_j A_{ij}$ . Khi đó  $L = D - A$  là ma trận Laplacian của  $A$ .

### Toán tử cho các đặc trưng nút trên đồ thị

Có một số toán tử cơ bản khác nhau cho các đặc trưng nút, trong đó, toán tử lọc được sử dụng làm thuộc tính chính để phân lớp trong mạng tích chập đồ thị. Toán tử lọc trên đồ thị là một phép toán cục bộ được áp dụng cho các đặc trưng nút với hai loại chính: lọc tần số và lọc nút.

*Biến đổi Fourier đồ thị:* Biến đổi Fourier đồ thị của một đặc trưng nút  $n \in N$  được định nghĩa:

$$\hat{n}(e_l) = \langle n, u_l \rangle = \sum_{i=1}^m n(i)u_l(i) \quad (1.2)$$

Trong đó:  $e_l$  là vector riêng của ma trận  $L$ ,  $u_l$  là giá trị riêng tương ứng của  $e_l$ . Khi đó, biến đổi Fourier đồ thị nghịch đảo là:

$$\hat{n}(i) = \sum_{l=1}^m \hat{n}(e_l)u_l(i) \quad (1.3)$$

*Lọc tần số*: Toán tử lọc tần số thường là một tích chập của đặc trưng nút bộ lọc trong miền phổ. Tích chập trong miền phổ giống như phép biến đổi Fourier nghịch đảo của phép nhân giữa các biểu diễn phổ của các đặc trưng hai nút. Sau đó, việc lọc tần số của đặc trưng  $n$  tại nút  $v_i$  được định nghĩa là:

$$(n *_G y)(i) = \sum_{l=1}^m \hat{n}(e_l) \hat{y}(e_l) u_l(i) \quad (1.4)$$

Trong đó:  $(n *_G y)(i)$  là lọc tần số của một đặc trưng nút  $n$  trên đồ thị  $G$  với bộ lọc  $y$ ,  $\hat{n}(e_l) \hat{y}(e_l)$  là các lọc trên miền phổ.

*Lọc nút*: Toán tử lọc nút thường là một tích chập của đặc trưng nút lọc trong miền đỉnh. Bộ lọc này được định nghĩa là sự kết hợp tuyến tính của các đặc trưng nút của các nút lân cận. Sau đó, bộ lọc đỉnh của đặc trưng  $n$  tại nút  $v_i$  được định nghĩa là:

$$(n *_G y)(i) = W_{i,i} n(i) + \sum_{j \in N(i,K)} W_{i,j} n(j) \quad (1.5)$$

Trong đó:  $W_{i,i}, W_{i,j}$  là các trọng số kết hợp,  $N(i, K)$  là các lân cận cục bộ K-hop của nút  $v_i$  trong đồ thị  $G$ .

### 1.2.2 Mạng tích chập đồ thị

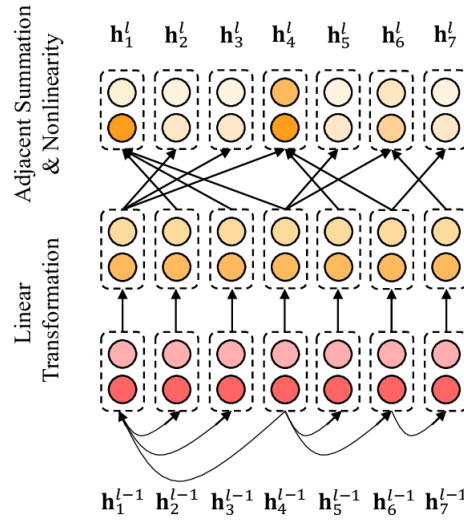
Mạng tích chập đồ thị (GCN) được đề xuất bởi Kipf và cộng sự [3], là một loại mạng nơron được thiết kế để làm việc với dữ liệu có cấu trúc đồ thị. GCN đã và đang được ứng dụng rộng rãi trong nhiều lĩnh vực như thị giác máy, xử lý ngôn ngữ tự nhiên, hệ thống khuyến nghị,... GCN cho phép dự đoán các đặc trưng cho các nút, các kết nối giữa chúng hoặc toàn bộ đồ thị và khái quát hóa thành các đồ thị vô hình, qua đó nắm bắt thông tin cấu trúc cục bộ và toàn cục. GCN cũng cho phép tích hợp nhiều lớp tích chập để cải thiện chất lượng của các biểu diễn nút. Đây cũng là giải pháp tiềm năng để giải quyết các vấn đề phức tạp trên đồ thị, chẳng hạn như khi dữ liệu gốc không có cấu trúc dựa trên đồ thị, như văn bản và hình ảnh, và dữ liệu có cấu trúc dựa trên đồ thị phức tạp và đa dạng liên kết giữa các cạnh.

Ý tưởng của GCN là tạo ra mạng nơron tích chập làm việc trên đồ thị thay vì lưới thông thường. Mô hình GCN hoạt động bằng cách truyền thông tin giữa các nút lân cận trong đồ thị. Nó sử dụng cơ chế truyền thông báo để cập nhật lặp lại các biểu diễn của nút dựa trên biểu diễn của các nút lân cận. Điều này tương tự như cách CNN kết hợp các bộ lọc trên một lưới thông thường để trích xuất các đặc trưng.

Cho đồ thị với  $n$  nút, đồ thị có thể được biểu diễn dưới dạng ma trận kề  $A \in \mathbb{R}^{n \times n}$ . Một phần tử  $A_{ij}$  trong  $A$  chỉ ra nút thứ  $i$  có kết nối với nút thứ  $j$  hay không. Cụ thể,  $A_{ij} = 1$  nếu nút thứ  $i$  có kết nối với nút thứ  $j$ , ngược lại  $A_{ij} = 0$ . Thêm vào đó, ma trận kề  $A$  có thể được coi là đầu ra rời rạc cuối cùng của trình phân tích cú pháp phụ thuộc. Kí hiệu  $h_i^l$  là đầu ra của lớp thứ  $l$  cho nút  $i$ , khi đó  $h_i^0$  biểu diễn trạng thái ban đầu của nút  $i$ . Với GCN gồm  $L$  lớp,  $l \in [1, 2, \dots, L]$  thì  $h_i^L$  là trạng thái cuối cùng của nút  $i$ . Đối với nút thứ  $i$  ở lớp thứ  $l$ , biểu diễn trạng thái ẩn của nút đó  $h_i^l$  được cập nhật bởi phép toán:

$$h_i^l = \sigma\left(\sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l\right) \quad (1.6)$$

Trong đó:  $W^l$  là trọng số biến đổi tuyến tính,  $b$  là độ lệch và  $\sigma$  là hàm phi tuyến (ví dụ ReLU).



Hình 1.3: Ví dụ về lớp GCN [11]

Vì quá trình tích chập đồ thị chỉ mã hóa thông tin của các nút lân cận trực tiếp, một nút trong đồ thị chỉ có thể bị ảnh hưởng bởi các nút lân cận trong  $L$  bước của GCN gồm  $L$  lớp.

### 1.2.3 Phân loại mạng GCN

Mạng GCN được phân chia thành hai loại dựa trên hai toán tử lọc cho đặc trưng nút trên đồ thị, bao gồm mạng tích chập đồ thị quang phổ và mạng tích chập đồ thị không gian.

#### Mạng tích chập đồ thị quang phổ

Phương pháp GCN quang phổ được coi là phương pháp bắt đầu từ việc xây dựng

lọc tần số bằng cách sử dụng tích chập đồ thị phổ. Các phương pháp này lấy ý tưởng từ mạng nơ ron tích chập truyền thống (CNN) bằng cách giả định rằng  $N$  là ma trận đặc trưng nút bao gồm nhiều vector đặc trưng. Lớp tích chập của các GCN quang phổ dựa trên CNN được định nghĩa như sau:

$$H_{:,j}^{(k)} = \delta \left( \sum_{i=1}^{f_{k-1}} E D_{i,j}^{(k)} E^T H_{:,i}^{(k-1)} \right), (1 \leq j \leq f_k) \quad (1.7)$$

Trong đó:  $E$  là tập  $n$  cạnh của đồ thị,  $D_{i,j}^{(k)}$  là ma trận chéo và là ma trận bậc của ma trận kề  $A$ , với  $D_{ii} = \sum_j A_{ij}$ ,  $k$  là thứ tự của các lớp,  $H^{(k-1)} \in R^{m \times f_{k-1}}$  là đặc trưng nút đầu vào, và  $H^{(0)} = N.f_{k-1}$  là số lượng đặc trưng đầu vào.

Các phương pháp GCN quang phổ phụ thuộc vào các vector riêng cụ thể của ma trận Laplacian. Do đó, các phương pháp này có một số hạn chế. Đầu tiên, các vector riêng phụ thuộc rất nhiều vào cấu trúc đồ thị, nghĩa là nếu đồ thị có bất kỳ thay đổi nào liên quan đến các nút hoặc cạnh, thì các giá trị riêng sẽ thay đổi. Thứ hai, các bộ lọc đã học phụ thuộc vào miền phổ, do đó không thể áp dụng chúng cho một cấu trúc đồ thị khác. Cuối cùng, độ phức tạp tính toán của phép phân tách các vector riêng là  $O(m^3)$ .

### Mạng tích chập đồ thị không gian

Các phương pháp GCN không gian được coi là các phương pháp bắt đầu bằng việc xây dựng lọc nút bằng cách sử dụng các kết cấu đồ thị không gian truyền tải và tổng hợp các đặc trưng của nút trong vùng lân cận của nút. Theo cách tiếp cận này, tích chập đồ thị không gian cho nút  $v_i$  ở lớp thứ  $k$  được thiết kế như sau:

$$h_{N(i)}^{(k)} = H_{v_i,:}^{(k)} + \sum_{v_j \in N(i)} H_{v_j,:}^{(k)} \quad (1.8)$$

$$H_{v_i,:}^{(k+1)} = \delta(h_{N(i)}^{(k)} \theta_{|N(i)|}^{(k)}) \quad (1.9)$$

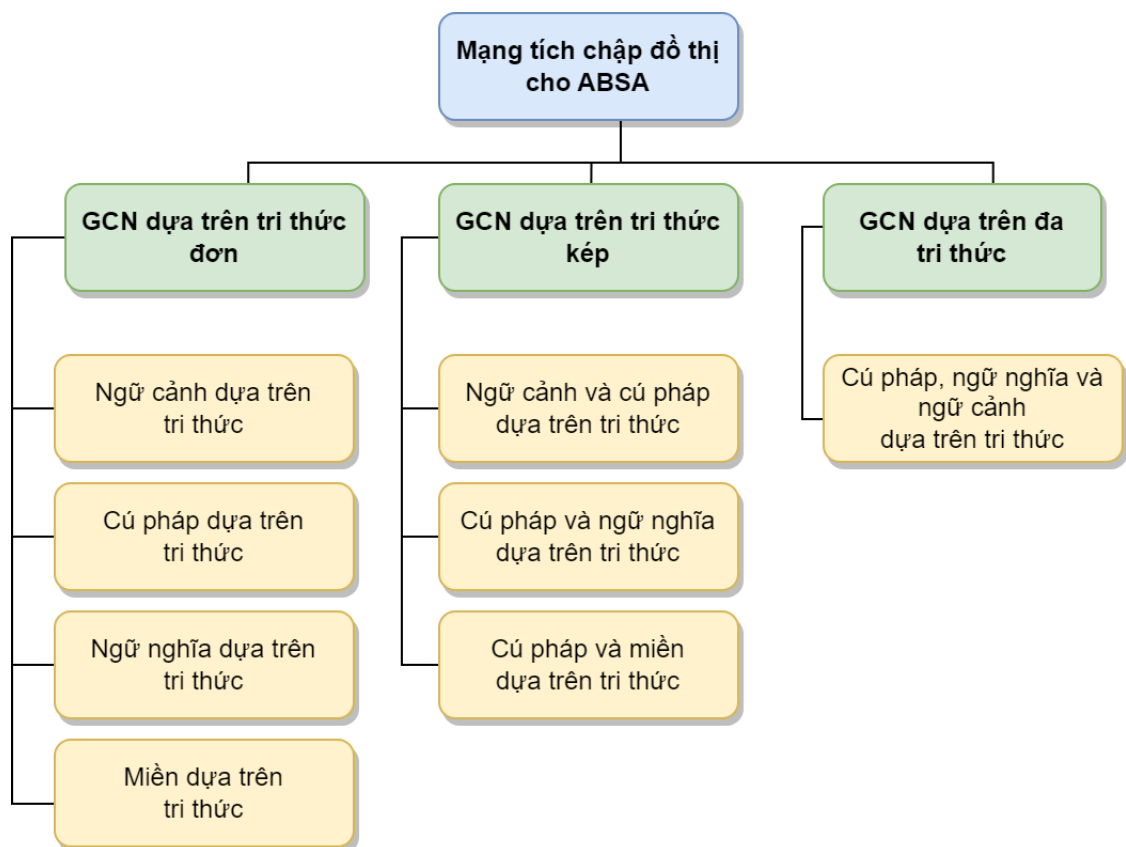
Trong đó:  $\theta_{|N(i)|}^{(k)}$  là ma trận trọng số cho nút với cùng bậc  $|N(i)|$  ở lớp thứ  $k$ .

Các phương pháp GCN không gian đã phát triển nhanh chóng nhờ tính linh hoạt, hiệu quả và tính tổng quát hấp dẫn của chúng, đặc biệt là về ABSA. Các biến thể khác nhau của phương pháp GCN cho ABSA đã được đề xuất, với kết quả đầy hứa hẹn.

### 1.3 Mạng tích chập đồ thị cho khai phá quan điểm mức khía cạnh

Phương pháp sử dụng mạng tích chập đồ thị là phương pháp tiên tiến nhất cho đến hiện tại và mang lại hiệu suất đầy hứa hẹn. Phương pháp này được xây dựng sử dụng các thuật toán học sâu trên cấu trúc đồ thị. Về đặc điểm, các phương pháp dựa trên GCN chủ yếu bắt đầu bằng việc xây dựng cấu trúc dữ liệu đồ thị để biểu diễn các phần tử và mối quan hệ giữa khía cạnh, mục tiêu và quan điểm trong văn bản. Sau đó, các biểu diễn đồ thị được đưa qua các lớp tích chập của GCN.

Hầu hết các phương pháp cho ABSA dựa trên GCN đều thuộc về các phương pháp GCN không gian. Hình 1.4 thể hiện các phân loại mạng GCN ABSA. Sự khác biệt chính giữa ba phân loại này là số lượng loại tri thức được sử dụng trong các lớp GCN.



Hình 1.4: Phân loại mạng GCN cho ABSA

Trong đó:

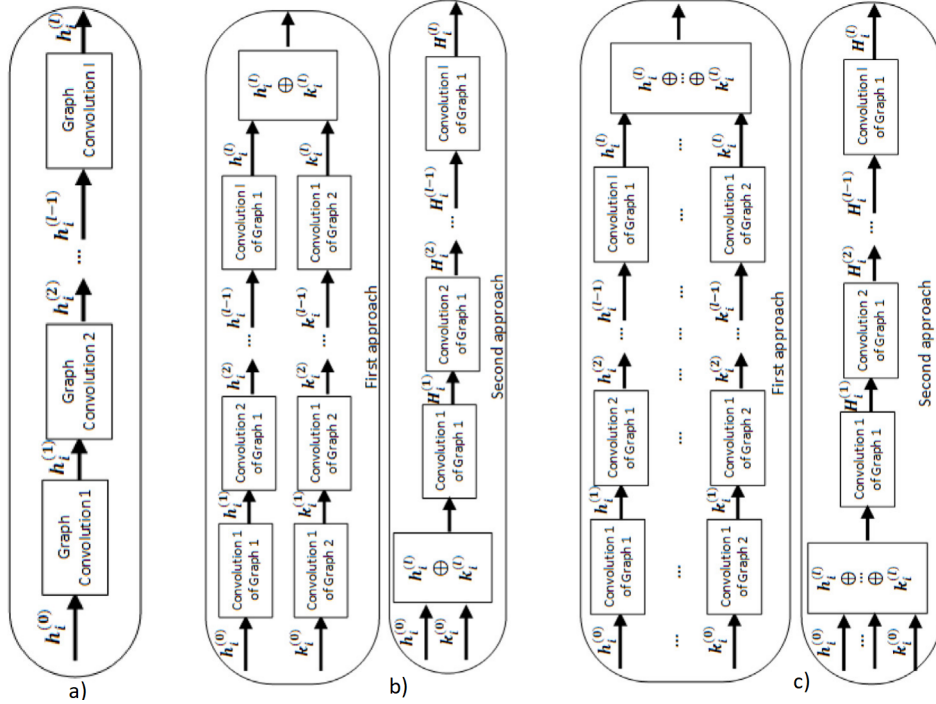
- Tri thức cú pháp là thông tin liên quan đến sự phụ thuộc ngữ pháp giữa các từ hoặc cụm từ trong câu.
- Tri thức ngữ nghĩa là thông tin liên quan đến ý nghĩa chung về cảm xúc giữa các



từ theo ngữ cảnh liên quan đến một khía cạnh cụ thể trong câu.

- Tri thức ngữ cảnh là những thông tin liên quan đến tương tác ngữ cảnh và quan hệ giữa từ chỉ khía cạnh, từ chỉ quan điểm và từ chỉ mục tiêu trong câu.

- Tri thức miền là thông tin liên quan đến một lĩnh vực cụ thể, chẳng hạn như giáo dục, chăm sóc sức khỏe, tài chính và thể thao.



Hình 1.5: Cấu trúc phân loại mạng GCN cho ABSA. a) Mạng GCN dựa trên tri thức đơn; b) Mạng GCN dựa trên tri thức kép; c) Mạng GCN dựa trên đa tri thức [6]

Hình 1.5a cho thấy mô hình GCN dựa trên tri thức đơn với nhiều lớp tích chập trên đồ thị dựa trên tri thức đơn. Điều này có nghĩa là đầu vào của lớp tích chập đầu tiên là biểu diễn đặc trưng nút được tạo bằng cách tích hợp thông tin đặc trưng từ các nút lân cận của nút này trên đồ thị dựa trên tri thức đơn. Đầu ra của lớp tích chập cuối cùng là biểu diễn GCN cuối cùng được sử dụng để dự đoán kết quả phân tích quan điểm.

Hình 1.5b thể hiện mô hình GCN dựa trên tri thức kép với nhiều lớp tích chập. Có thể chia ra hai cách tiếp cận để tạo các biểu diễn đặc trưng nút. Đầu tiên, hai loại biểu diễn đặc trưng nút riêng biệt được tạo ra bằng cách tổng hợp thông tin đặc trưng từ các nút lân cận trên các đồ thị dựa trên tri thức tương ứng. Sau đó, tích hợp hai loại biểu diễn đặc trưng nút này thành các biểu diễn đặc trưng nút chung, được sử dụng làm đầu vào của lớp tích chập đầu tiên. Đầu ra của lớp tích chập cuối cùng là biểu diễn GCN cuối cùng được sử dụng để dự đoán kết quả phân tích quan điểm. Trong cách tiếp cận thứ

hai, đầu tiên hai mô hình GCN dựa trên đơn tri thức với nhiều lớp tích chập song song được xây dựng, như trong hình 1.5a. Sau đó, hai biểu diễn GCN cuối cùng tương ứng với hai GCN dựa trên tri thức đơn được tích hợp vào biểu diễn đặc trưng nút chung được sử dụng để dự đoán kết quả phân tích quan điểm.

Hình 1.5c cho thấy mô hình GCN dựa trên đa tri thức với nhiều lớp tích chập. Chúng được xây dựng tương tự với mô hình GCN dựa trên tri thức kép với hai cách tiếp cận để tạo các biểu diễn đặc trưng nút. Tuy nhiên, khác với mô hình GCN dựa trên tri thức kép, cách tiếp cận này sử dụng nhiều hơn hai loại tri thức.

## 1.4 Bài toán khóa luận

Phân loại quan điểm khía cạnh là một trong những nhiệm vụ cơ bản của ABSA. Đây cũng chính là bài toán mà khóa luận cần giải quyết. Nhìn chung, nhiệm vụ này yêu cầu xác định quan điểm của từng khía cạnh cụ thể trong câu, cho trước câu và khía cạnh. Cụ thể, bài toán được phát biểu như sau:

### Đầu vào

Cho cặp câu đánh giá và khía cạnh tiếng Việt  $(s, a)$ , trong đó  $s = \{w_1, w_2, \dots, w_n\}$  và  $a = \{a_1, a_2, \dots, a_m\}$ ,  $a$  là một từ hoặc cụm từ xuất hiện trong  $s$ .

### Đầu ra

Xác định xu hướng quan điểm tương ứng cho mỗi khía cạnh đã đề cập ở đầu vào  $p = \{p_1, p_2, \dots, p_m\}$ .

### Phương pháp

Khoá luận dự định áp dụng mô hình Mạng tích chập đồ thị nâng cao ngữ nghĩa và cú pháp được đề xuất bởi ZhengZhang và cộng sự [13] cho bộ dữ liệu tiếng Việt UIT-ViSFD.

## 1.5 Tóm tắt chương 1

Chương 1 vừa trình bày tổng quan về khai phá quan điểm mức khía cạnh, trong đó có các thuật ngữ liên quan và các nhiệm vụ đơn lẻ cần giải quyết. Bên cạnh đó, chương 1 cũng đề cập đến phương pháp áp dụng mạng tích chập đồ thị cho bài toán phân lớp quan điểm khía cạnh, giới thiệu cấu trúc mạng GCN cũng như đề cập các hạn chế còn tồn tại, các nghiên cứu được đưa ra để khắc phục các hạn chế đó cho bài toán phân lớp. Cuối cùng, phát biểu bài toán khoá luận và đưa ra định hướng giải quyết cho bài toán này.

## Chương 2

# HAI MÔ HÌNH MẠNG TÍCH CHẬP ĐỒ THỊ NỀN TẢNG

Trong chương 2, khóa luận sẽ trình bày hai mô hình dựa trên mạng tích chập đồ thị dựa trên tri thức kép cho nhiệm vụ phân loại quan điểm khía cạnh, đó là mô hình mạng tích chập đồ thị đặc tả khía cạnh (ASGCN) và mô hình mạng tích chập đồ thị kép (DualGCN).

### 2.1 Mô hình mạng tích chập đồ thị đặc tả khía cạnh

#### 2.1.1 Giới thiệu

Mô hình mạng tích chập đồ thị đặc tả khía cạnh, viết tắt là ASGCN, được đề xuất bởi Zhang và cộng sự (2019) [11]. Đây là mô hình dựa trên tri thức kép bao gồm cú pháp và ngữ cảnh. Mô hình này có khả năng giải quyết hai hạn chế tồn tại từ các nghiên cứu trước đó. Đầu tiên là vấn đề liên quan đến các mô hình dựa trên cơ chế chú ý. Mặc dù các mô hình này đầy hứa hẹn cho nhiệm vụ khai phá quan điểm, chúng lại không đủ để nắm bắt phụ thuộc cú pháp giữa từ ngữ cảnh và khía cạnh trong câu. Do đó, chỉ sử dụng mỗi cơ chế chú ý có thể đưa ra dự đoán sai cho một khía cạnh, do sử dụng sai từ quan điểm vì không nắm bắt được thứ tự cú pháp. Cho ví dụ "Pin ỏn còn màn hình hiển thị thì tệ", các mô hình dựa trên chú ý thường xác định "tệ" là từ ý kiến cho khía cạnh "pin", nhưng lại không đúng trong trường hợp này. Mặc dù He và cộng sự (2018) [1] có đề xuất một vài ràng buộc cú pháp trên trọng số chú ý, ảnh hưởng của cấu trúc cú pháp vẫn chưa được khai thác triệt để. Vấn đề thứ hai cho thấy rằng ý kiến đối với khía cạnh cụ thể có thể được biểu thị bởi nhiều từ không kề nhau, dẫn đến khi đưa ra dự đoán có thể bỏ sót

từ làm cho kết quả bị sai lệch. Mạng nơ ron tích chập CNN được cài đặt để khai thác các cụm từ gồm nhiều từ mô tả khía cạnh dựa trên ý tưởng quan điểm của khía cạnh thường được xác định bởi cụm từ khoá thay vì từ đơn lẻ. Tuy nhiên, các mô hình dựa trên CNN chỉ có thể cảm nhận các đặc trưng nhiều từ dưới dạng các từ liên tiếp với các phép toán tích chập trên các chuỗi từ, nhưng không đủ để xác định quan điểm được mô tả bởi nhiều từ không nằm cạnh nhau. Ví dụ, cho câu "nhân viên nên phục vụ nhiệt tình hơn", các mô hình dựa trên CNN có thể chỉ phát hiện "nhiệt tình" là cụm từ mô tả cho khía cạnh "nhân viên", bỏ qua từ "nên" vốn có tác động đến quan điểm đối với khía cạnh.

Nhờ cấu trúc GCN nhiều lớp với mỗi lớp mã hoá và cập nhật biểu diễn nút trong đồ thị, mô hình ASGCN có thể xử lý hai vấn đề được đề cập ở trên. ASGCN đầu tiên sử dụng một lớp mạng bộ nhớ ngắn hạn-dài hạn tạm thời hai chiều (Bi-LSTM) để nắm bắt thông tin ngữ cảnh theo thứ tự từ trong câu. Tiếp sau đó, để thu được đặc trưng khía cạnh cụ thể, một cấu trúc tích chập đồ thị nhiều lớp được cài đặt trên lớp LSTM, theo sau là cơ chế che giấu để lọc ra các từ không phải khía cạnh, chỉ giữ các đặc trưng cụ thể theo khía cạnh mức cao. Các đặc trưng cụ thể theo khía cạnh được đưa trở lại đầu ra LSTM để truy xuất các đặc trưng thông tin liên quan đến khía cạnh, sau đó được sử dụng để dự đoán quan điểm dựa trên khía cạnh.

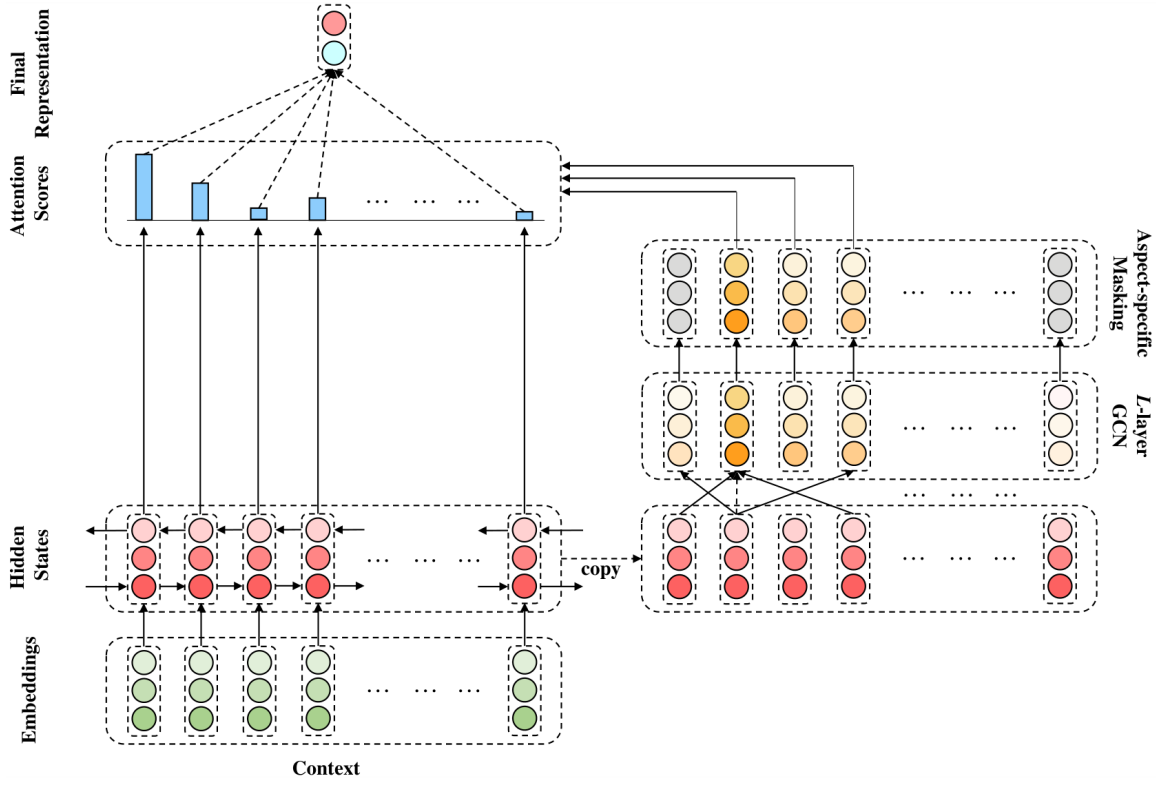
## 2.1.2 Kiến trúc mô hình

### Lớp nhúng và LSTM hai chiều

Cho câu  $c$  gồm  $n$  từ:  $c = \{w_1^c, w_2^c, \dots, w_{\tau+1}^c, \dots, w_{\tau+m}^c, \dots, w_n^c\}$  chứa khía cạnh  $m$  từ tương ứng bắt đầu từ mã tách từ thứ  $\tau + 1$ , đầu tiên nhúng mỗi mã tách từ sang một không gian vector giá trị thực chiều thấp, với ma trận nhúng  $E \in \mathbb{R}^{|V| \times d_e}$ , trong đó  $|V|$  là kích thước của từ vựng và  $d_e$  là chiều của từ nhúng. ASGCN sử dụng các vector nhúng Word2Vec. Từ nhúng của câu được đưa vào một lớp LSTM hai chiều để tạo ra vector trạng thái ẩn  $H^c = \{h_1^c, h_2^c, \dots, h_{\tau+1}^c, \dots, h_{\tau+m}^c, \dots, h_{n-1}^c, h_n^c\}$ , trong đó  $h_t^c \in \mathbb{R}^{2d_h}$  biểu thị vector trạng thái ẩn tại bước thời gian  $t$  và  $d_h$  là chiều của đầu ra vector trạng thái ẩn bởi LSTM một chiều.

### Lớp trích xuất đặc trưng đặc tả khía cạnh

Khác với bài toán phân lớp quan điểm chung, phân lớp quan điểm mức khía cạnh có mục tiêu đánh giá quan điểm đối với một khía cạnh cụ thể, cũng vì thế còn được gọi là chiến lược trích xuất đặc trưng đặc tả khía cạnh. ASGCN sử dụng nhiều lớp GCN trên cây phụ thuộc cú pháp kết hợp với cơ chế che giấu khía cạnh ở lớp trên cùng để đạt được mục tiêu này.



Hình 2.1: Kiến trúc mạng tích chập đồ thị đặc tả khía cạnh [11]

### Tích chập đồ thị trên cây phụ thuộc

Với mục tiêu giải quyết các hạn chế còn tồn tại của các cách tiếp cận trước đó, mạng tích chập đồ thị được tận dụng trên cây phụ thuộc cú pháp của câu. Đặc biệt, sau khi xây dựng cây phụ thuộc của câu sẽ thu được một trận kề  $A \in \mathbb{R}^{n \times n}$  dựa theo các từ trong câu. Điều quan trọng cần lưu ý là cây phụ thuộc là đồ thị có hướng. Mặc dù GCN thường không xem xét các hướng, nhưng chúng có thể được điều chỉnh theo kịch bản nhận biết hướng. Theo đó, hai biến thể của ASGCN được Zhang và cộng sự (2019) [11] đề xuất, ASGCN-DG trên các đồ thị phụ thuộc không có hướng và ASGCN-DT liên quan đến các cây phụ thuộc có hướng. Trên thực tế, sự khác biệt duy nhất giữa ASGCN-DG và ASGCN-DT nằm ở ma trận kề của chúng: Ma trận kề của ASGCN-DT thưa hơn nhiều so với ma trận kề của ASGCN-DG. Cài đặt như vậy phù hợp với hiện tượng các nút cha mẹ bị ảnh hưởng rộng rãi bởi các nút con của chúng. Hơn nữa, theo ý tưởng tự lặp trong Kipf và Welling (2017) [3], mỗi từ được đặt liền kề với chính nó, tức là các giá trị đường chéo của A đều là 1.

Các biến thể ASGCN được thực hiện theo kiểu đa lớp trên đầu ra LSTM hai chiều, tức là  $H^0 = H^c$  để giúp các nút nhận biết ngữ cảnh. Sau đó, biểu diễn của từng nút được cập nhật với phép toán tích chập đồ thị với hệ số chuẩn hóa như sau:

$$\tilde{h}_i^l = \sum_{j=1}^n A_{ij} W^l g_j^{l-1} \quad (2.1)$$

$$h_i^l = \text{ReLU}(\tilde{h}_i^l / (d_i + 1) + b^l) \quad (2.2)$$

Trong đó:  $g_j^{l-1} \in \mathbb{R}^{2d_h}$  là biểu diễn token thứ  $j$  phát triển từ lớp GCN trước đó,  $h_i^l \in \mathbb{R}^{2d_h}$  là sản phẩm của lớp GCN hiện tại, và  $d_i = \sum_{j=1}^n A_{ij}$  là bậc của token thứ  $i$  trong cây phụ thuộc. Trọng số  $W^l, b^l$  là các tham số huấn luyện được.

Đáng chú ý là  $h_i^l$  không được đưa vào ngay lập tức lớp GCN liên tiếp mà tiến hành chuyển đổi nhận biết vị trí ngay từ đầu:

$$g_i^l = \mathcal{F}(h_i^l) \quad (2.3)$$

Trong đó:  $\mathcal{F}(\cdot)$  là hàm gán trọng số vị trí, đã được áp dụng trong các nghiên cứu trước đây để tăng cường độ quan trọng của từ ngữ cạnh gần với khía cạnh. Bằng cách đó, có thể giảm nhiễu và sai lệch phát sinh một cách tự nhiên từ quá trình phân tích cú pháp phụ thuộc. Cụ thể, hàm  $\mathcal{F}(\cdot)$  là:

$$q_i = \begin{cases} 1 - \frac{\tau+1-i}{n}, & 1 \leq i < \tau + 1 \\ 0, & \tau + 1 \leq i \leq \tau + m \\ 1 - \frac{i-\tau-m}{n}, & \tau + m < i \leq n \end{cases} \quad (2.4)$$

$$\mathcal{F}(h_i^l) = q_i h_i^l \quad (2.5)$$

Trong đó:  $q_i \in \mathbb{R}$  là trọng số vị trí của token thứ  $i$ . Đầu ra cuối cùng của lớp GCN thứ  $L$  là  $H^L = \{h_1^L, h_2^L, \dots, h_{\tau+m}^L, \dots, h_{n-1}^L, h_n^L\}$ ,  $h_t^L \in \mathbb{R}^{2d_h}$

*Che giấu đặc tả khía cạnh*

Lớp này thực hiện che giấu vector trạng thái ẩn của các từ không phải khía cạnh và giữ trạng thái của các từ khía cạnh không đổi:

$$h_t^L = 0, 1 \leq t < \tau + 1, \tau + m < t \leq n \quad (2.6)$$

Đầu ra của lớp che giấu không này là đặc trưng hướng khía cạnh  $H_{mask}^L = \{0, \dots, h_{\tau+1}^L, \dots, h_{\tau+m}^L, \dots, 0\}$ . Thông qua tích chập đồ thị, các đặc trưng  $H_{mask}^L$  thu được ngữ cảnh xung quanh khía cạnh, xem xét cả phụ thuộc cú pháp và các quan hệ đa từ tầm xa.

## Chú ý nhận thức khía cạnh

Dựa trên đặc trưng hướng khía cạnh, một biểu diễn của các vector trạng thái ẩn  $H^c$  được tạo ra thông qua một cơ chế chú ý dựa trên truy xuất mới. Ý tưởng là trích xuất các đặc trưng cần thiết, có liên quan ngữ nghĩa đến các từ khía cạnh từ các vector trạng thái ẩn, và thiết lập trọng số chú ý dựa trên truy xuất cho mỗi từ ngữ cảnh. Trọng số chú ý được tính toán như sau:

$$\mathcal{B}_t = \sum_{i=1}^n h_t^{cT} h_i^L = \sum_{i=\tau+1}^{\tau+m} h_t^{cT} h_i^L \quad (2.7)$$

$$\lrcorner = \frac{\exp(\mathcal{B}_t)}{\sum_{i=1}^n \exp(\mathcal{B}_i)} \quad (2.8)$$

Ở đây, tích chấm được sử dụng để đo mức độ liên quan về ngữ nghĩa giữa các từ thành phần khía cạnh và các từ trong câu sao cho việc che giấu đặc tả khía cạnh, tức là che dấu bằng không, có thể có hiệu lực. Biểu diễn cuối cùng của dự đoán có công thức:

$$r = \sum_{t=1}^n \lrcorner h_t^c \quad (2.9)$$

## Phân lớp quan điểm

Sau khi có được biểu diễn  $r$ , biểu diễn này được đưa vào một lớp kết nối đầy đủ, theo sau là lớp chuẩn hoá softmax để thu được phân phối xác suất  $p \in \mathbb{R}^{d_p}$  trên không gian quyết định phân cực:

$$p = \text{softmax}(W_p r + b_p) \quad (2.10)$$

Trong đó:  $d_p$  tương tự như chiều của nhãn quan điểm,  $W_p \in \mathbb{R}^{d_p \times 2d_h}$  và  $b_p \in \mathbb{R}^{d_p}$  là trọng số và độ lệch có thể học được.

## Huấn luyện

Mô hình được huấn luyện sử dụng thuật toán xuống đồi đạo hàm tiêu chuẩn với hàm mất mát cross-entropy và bộ điều chỉnh  $L_2$ :

$$Loss = - \sum_{(c, \hat{p}) \in C} \log p_{\hat{p}} + \lambda ||\theta||_2 \quad (2.11)$$



Trong đó:  $C$  biểu thị tập dữ liệu,  $\hat{p}$  là nhãn và  $p_{\hat{p}}$  nghĩa là phần tử thứ  $\hat{p}$  của  $p$ ,  $\theta$  biểu thị tất cả tham số huấn luyện được, và  $\lambda$  là hệ số chính quy hoá  $L_2$ .

## 2.2 Mô hình mạng tích chập đồ thị kép

### 2.2.1 Giới thiệu

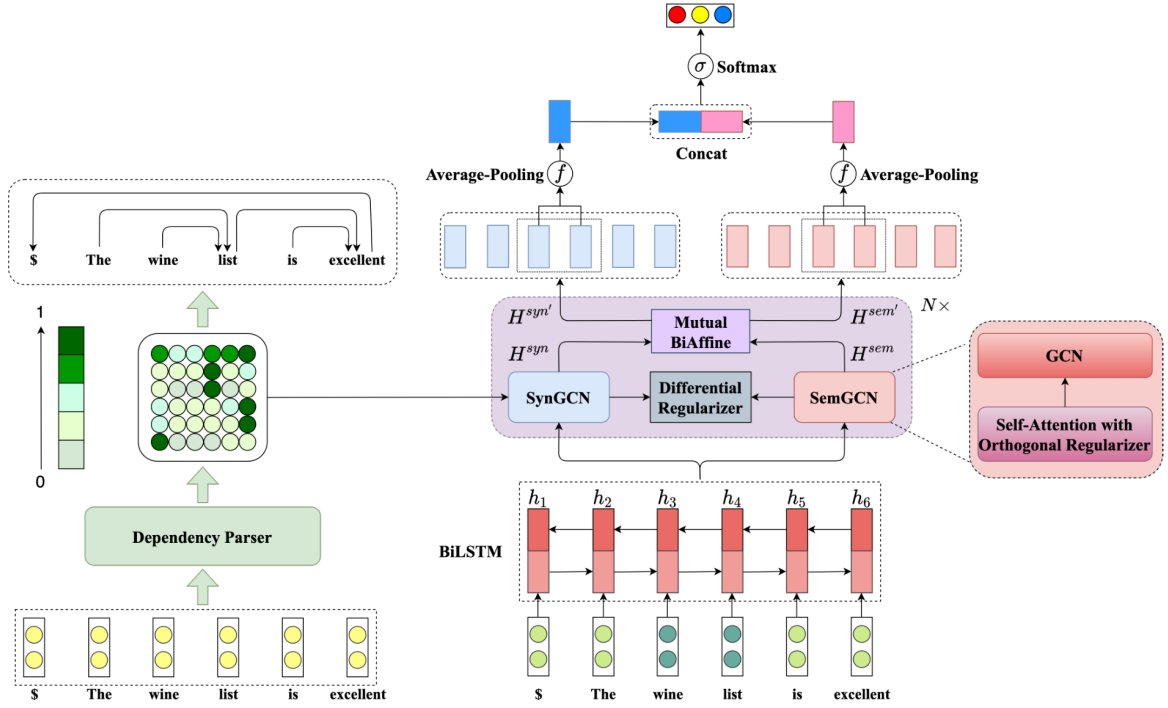
Mô hình mạng tích chập đồ thị kép được đề xuất bởi Li và cộng sự (2021) [4], được viết tắt là DualGCN. Mô hình DualGCN là mô hình dựa trên tri thức kép, bao gồm ngữ nghĩa và cú pháp. Mô hình này cũng sử dụng mạng tích chập đồ thị trên cây cú pháp để mô hình hoá quan hệ cấu trúc cú pháp của các từ trong câu. Tuy nhiên tại thời điểm đó, tồn tại hai vấn đề liên quan đến việc áp dụng phương pháp này với bài toán phân lớp quan điểm. Thứ nhất, các trình phân tích cú pháp phụ thuộc chưa thực sự cho ra kết quả chính xác. Thứ hai, GCN không hoạt động tốt trên các văn bản không tuân theo cấu trúc ngữ pháp chung nào. Điều này khá thực tế vì hầu hết các văn bản thể hiện quan điểm do người dùng tạo ra, đặc biệt là các văn bản trực tuyến thường không sử dụng ngữ pháp đầy đủ, có nhiều lỗi chính tả hay các kí hiệu đặc biệt. Bên cạnh đó, các đối tượng cũng được bày tỏ quan điểm bằng nhiều cách, nhiều cụm từ khác nhau.

Mô hình DualGCN đã có những cải tiến nhằm khắc phục hai vấn đề vừa đề cập. Mô hình này xem xét đồng thời tính bổ sung của cấu trúc cú pháp và tương quan ngữ nghĩa. Để giải quyết sự thiếu chính xác của các trình phân tích cú pháp phụ thuộc, ma trận xác suất của tất cả các cung phụ thuộc từ trình phân tích cú pháp phụ thuộc được sử dụng để xây dựng mạng tích chập đồ thị dựa trên cú pháp (SynGCN). Ý tưởng đằng sau cách tiếp cận này là do ma trận xác suất biểu diễn các phụ thuộc giữa các từ chứa thông tin cú pháp phong phú so sánh với đầu ra rời rạc cuối cùng của trình phân tích cú pháp phụ thuộc. Tiếp đó, nhằm khắc phục việc thiếu nhạy cảm với ngữ pháp của các văn bản đánh giá, mạng tích chập đồ thị dựa trên tương quan ngữ nghĩa được xây dựng tận dụng cơ chế tự chú ý (SemGCN). Cách tiếp cận này xuất phát từ ý tưởng rằng ma trận chú ý được định hình bởi việc tự học, cũng được xem như một đồ thị có hướng với cạnh có trọng số, có thể biểu thị tương quan ngữ nghĩa giữa các từ. Sau khi đã có được hai mô đun liên quan đến cú pháp và ngữ nghĩa là SynGCN và SemGCN, chúng được nối với nhau nhờ mô đun BiAffine [8].

Hai bộ điều chỉnh cũng được đề xuất cho mô hình DualGCN để nâng cao hiệu suất mô hình. Bộ điều chỉnh trực giao khuyến khích SemGCN học các tương quan về ngữ nghĩa ít trùng lặp hơn cho mỗi từ. Nguyên do là các thuật ngữ liên quan đến ngữ nghĩa

của mỗi từ không được trùng nhau, nên phân phối xác suất chú ý trên các từ là trực giao. Trong khi bộ điều chỉnh vi sai khuyến khích SemGCN học các đặc trưng ngữ nghĩa mà SynGCN không nắm bắt được.

## 2.2.2 Kiến trúc mô hình



Hình 2.2: Kiến trúc mạng tích chập đồ thị kép [4]

### Lớp nhúng và mã hoá

Cho cặp câu-khía cạnh  $(s, a)$ , với  $s = \{w_1, w_2, \dots, w_n\}$ ,  $a = \{a_1, a_2, \dots, a_m\}$  là một khía cạnh. Mô hình sử dụng bộ mã hoá câu BiLSTM và BERT để trích xuất biểu diễn trạng thái ẩn.

Với bộ mã hoá BiLSTM: Trích xuất từ nhúng của câu  $x = \{x_1, x_2, \dots, x_n\}$  từ bảng tra cứu nhúng  $E \in \mathbb{R}^{|V| \times d_e}$ , trong đó  $|V|$  là kích thước của bộ từ vựng và  $d_e$  là chiều của từ nhúng. Đưa từ nhúng của câu vào bộ mã hoá BiLSTM để thu được vector trạng thái ẩn  $H = \{h_1, h_2, \dots, h_n\}$ , với  $h_i \in \mathbb{R}^{2d}$  là trạng thái ẩn tại thời gian  $t$  từ BiLSTM. Chiều của vector trạng thái ẩn  $d$  là đầu ra của LSTM một chiều.

Với bộ mã hoá BERT: Biểu diễn cặp câu-khía cạnh dưới dạng "[CLS] câu [SEP] khía cạnh [SEP]" để thu được biểu diễn ẩn nhận thức khía cạnh của câu. Hơn nữa, để khớp các biểu diễn dựa trên từ của BERT với kết quả của phụ thuộc cú pháp trên từ, Li và cộng sự mở rộng các phụ thuộc của một từ thành tất cả các từ phụ của nó. Sau

đó, biểu diễn trạng thái ẩn của câu lần lượt được đưa vào môđun SynGCN và SemGCN. Sau đó, môđun BiAffine được sử dụng để tạo luồng thông tin hiệu quả. Cuối cùng, kết hợp tất cả các biểu diễn của nút khía cạnh từ môđun SynGCN và SemGCN thông qua pooling và nối để tạo ra biểu diễn khía cạnh cuối cùng.

### GCN dựa trên cú pháp - SynGCN

Môđun SynGCN lấy mã hoá cú pháp làm đầu vào. Để mã hoá thông tin cú pháp, Li và cộng sự tận dụng ma trận xác suất của tất cả các cung phụ thuộc từ phân tích cú pháp phụ thuộc. So sánh với đầu ra rời rạc cuối cùng của phân tích cú pháp phụ thuộc, ma trận xác suất phụ thuộc có thể nắm bắt thông tin cú pháp phong phú bằng việc cung cấp tất cả cấu trúc cú pháp ẩn. Vì thế, ma trận xác suất phụ thuộc được sử dụng để giảm thiểu các lỗi phân tích cú pháp phụ thuộc.

Với mã hoá cú pháp của ma trận kề  $A^{syn} \in \mathbb{R}^{n \times n}$ , môđun SynGCN lấy vector trạng thái ẩn  $H$  từ BiLSTM làm biểu diễn nút khởi đầu trong đồ thị cú pháp. Biểu diễn đồ thị cú pháp  $H^{syn} = \{h_1^{syn}, h_2^{syn}, \dots, h_n^{syn}\}$  thu được từ môđun SynGCN sử dụng phép tính (1). Ở đây,  $h_i^{syn} \in \mathbb{R}^d$  là biểu diễn ẩn của nút thứ  $i$ . Với khía cạnh, sử dụng  $\{h_{a_1}^{syn}, h_{a_2}^{syn}, \dots, h_{a_n}^{syn}\}$  để chỉ biểu diễn trạng thái ẩn tương ứng.

### GCN dựa trên ngữ nghĩa - SemGCN

SemGCN thu được ma trận chú ý dưới dạng ma trận kề thông qua cơ chế tự chú ý. Một mặt, tự chú ý có thể nắm bắt thuật ngữ liên quan đến ngữ nghĩa của mỗi từ trong một câu, điều này linh hoạt hơn so với cấu trúc cú pháp. Mặt khác, SemGCN có thể đáp ứng những đánh giá trực tuyến vốn không nhạy với thông tin cú pháp.

#### Tự chú ý

Tự chú ý tính toán điểm chú ý theo cặp giữa mỗi phần tử trong câu. Trong mô hình DualGCN, ma trận điểm chú ý  $A^{sem} \in \mathbb{R}^{n \times n}$  được tính toán sử dụng lớp tự chú ý. Sau đó ma trận điểm chú ý  $A^{sem}$  được lấy làm ma trận kề của môđun SemGCN:

$$A^{sem} = softmax\left(\frac{QW^Q \times (KW^K)^T}{\sqrt{d}}\right) \quad (2.12)$$

Trong đó:  $Q$  và  $K$  đều bằng biểu diễn đồ thị của lớp trước đó của môđun SemGCN,  $W^Q$  và  $W^K$  là ma trận trọng số có thể học được. Thêm vào đó,  $d$  là chiều của đặc trưng nút đầu vào. Chỉ sử dụng duy nhất một đầu tự chú ý để thu được ma trận điểm chú ý cho câu. Tương tự môđun SynGCN, môđun SemGCN thu được biểu diễn đồ thị  $H^{sem}$ . Hơn nữa, ký hiệu  $\{h_{a_1}^{sem}, h_{a_2}^{sem}, \dots, h_{a_n}^{sem}\}$  được dùng để chỉ biểu diễn ẩn của tất cả các nút

khía cạnh.

### Mô đun BiAffine

Để trao đổi thông tin liên quan giữa hai mô đun SynGCN và SemGCN một cách hiệu quả, biến đổi BiAffine được sử dụng làm cầu nối. Quá trình được xây dựng như sau:

$$H^{syn/} = softmax(H^{syn}W_1(H^{semT}))H^{sem} \quad (2.13)$$

$$H^{sem/} = softmax(H^{sem}W_2(H^{synT}))H^{syn} \quad (2.14)$$

Trong đó:  $W_1, W_2$  là các tham số huấn luyện được.

Cuối cùng, pooling trung bình và phép nối được áp dụng cho các nút khía cạnh của mô đun SynGCN và SemGCN. Do đó, biểu diễn đặc trưng cuối cùng thu được là :

$$h_a^{syn} = f(h_{a_1}^{syn}, h_{a_2}^{syn}, \dots, h_{a_n}^{syn}) \quad (2.15)$$

$$h_a^{sem} = f(h_{a_1}^{sem}, h_{a_2}^{sem}, \dots, h_{a_n}^{sem}) \quad (2.16)$$

$$r = [h_a^{syn}, h_a^{sem}] \quad (2.17)$$

Trong đó:  $f(.)$  là hàm tổng hợp trung bình được áp dụng trên các biểu diễn nút khía cạnh. Sau đó, biểu diễn  $r$  thu được được đưa qua lớp tuyến tính, tiếp theo là hàm softmax để tạo ra phân phối xác suất quan điểm  $p$ :

$$p(a) = softmax(W_p r + b_p) \quad (2.18)$$

Trong đó:  $W_p, b_p$  là trọng số và độ lệch có thể học được.

### Bộ điều chỉnh

Để cải thiện biểu diễn ngữ nghĩa, có hai bộ điều chỉnh được áp dụng cho mô đun SemGCN: bộ điều chỉnh trực giao và bộ điều chỉnh vi phân.

#### Bộ điều chỉnh trực giao

Theo trực giác, các thành phần liên quan của mỗi từ nên ở các vùng khác nhau trong một câu, do đó, sự phân bố điểm chú ý hiếm khi trùng lặp. Vì vậy, bộ điều chỉnh trực giao được đề xuất để khuyến khích tính trực giao giữa các vector điểm chú ý của tất cả các từ. Cho ma trận điểm chú ý  $A^{sem} \in \mathbb{R}^{n \times n}$ , bộ điều chỉnh trực giao có công thức:

$$R_O = ||A^{sem}A^{semT} - I||_F \quad (2.19)$$

Trong đó,  $I$  là ma trận nhận dạng. Chỉ số dưới  $F$  biểu thị Frobenious norm. Tóm lại, mỗi phần tử không chéo nhau của  $A^{sem}A^{semT}$  được tối thiểu hoá để duy trì tính trực giao của ma trận  $A^{sem}$ .

#### *Bộ điều chỉnh vi phân*

Hai loại biểu diễn đặc trưng học được từ mô đun SynGCN và SemGCN được kì vọng biểu diễn thông tin khác biệt, dựa trên cây phụ thuộc cú pháp và mối tương quan ngữ nghĩa. Vì vậy, một bộ điều chỉnh vi phân giữa hai ma trận kề của SynGCN và SemGCN được áp dụng. Bộ điều chỉnh này chỉ hạn chế với  $A^{sem}$ :

$$R_D = \frac{1}{||A^{sem} - A^{syn}||_p} \quad (2.20)$$

## **2.3 Ý tưởng khóa luận**

Nhờ độ chính xác của trình phân tích cú pháp hiện tại mà phương pháp áp dụng mạng tích chập đồ thị trên cây phụ thuộc đã cải thiện hiệu suất đáng kể. Bên cạnh đó, cách tiếp cận tận dụng cả kiến trúc cú pháp và tương quan ngữ nghĩa cũng đang được sử dụng rộng rãi nhờ sự bổ sung thông tin cho nhau. Dựa trên ý tưởng đó, khoá luận dự định áp dụng mô hình mạng tích chập đồ thị sử dụng tri thức kép để giải quyết bài toán khoá luận. Mô hình được đề xuất bởi ZhengZhang và cộng sự [13], viết tắt là SSEGCN. Mô hình SSEGCN được phát triển từ hai mô hình nền tảng vừa đề cập [11] [4]. Hai bộ biểu diễn tri thức ngữ nghĩa và cú pháp của mô hình này sẽ có sự thay đổi so với hai mô hình nền tảng ở trên. Khoá luận sẽ tiến hành thực nghiệm trên bộ dữ liệu tiếng Việt.

## 2.4 Tóm tắt chương 2

Chương 2 vừa giới thiệu hai mô hình dựa trên GCN cho bài toán phân lớp quan điểm, đó là ASGCN và DualGCN. Hai mô hình này đều khắc phục được một số hạn chế của các nghiên cứu trước đó, đồng thời đề xuất các ý tưởng mới góp phần nâng cao độ chính xác của mô hình phân lớp.

## Chương 3

# MÔ HÌNH KHOÁ LUẬN

Chương 3 sẽ trình bày mô hình phân lớp quan điểm mức khía cạnh được áp dụng cho dữ liệu tiếng Việt và các mô đun tương ứng.

### 3.1 Mô hình khoá luận

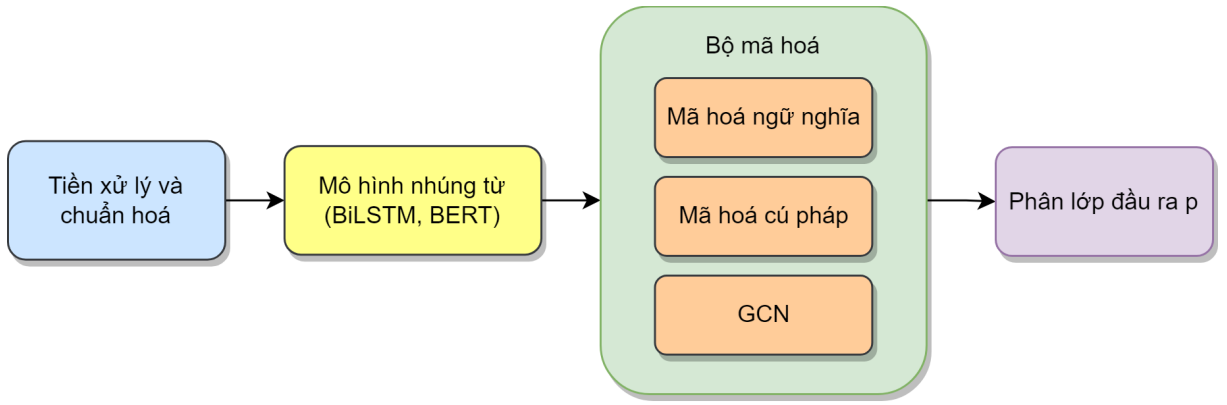
Mô hình được ứng dụng để giải quyết bài toán khoá luận gồm ba thành phần chính:

- Khởi tiên xử lý và chuẩn hoá tiền hành xử lý dữ liệu đầu vào để trích xuất thêm các chú thích cần thiết và đưa dữ liệu về định dạng phù hợp.
- Mô hình nhúng từ có vai trò ánh xạ cặp câu, khía cạnh  $(s, a)$  từ không gian từ sang không gian vector và trích xuất trạng thái ẩn để lấy thông tin ngữ cảnh.
- Bộ mã hoá tiếp tục sử dụng đầu ra từ mô hình nhúng từ làm đầu vào, mã hoá các đặc trưng liên quan đến ngữ nghĩa (sử dụng cơ chế chú ý), cú pháp (sử dụng ma trận mặt nạ cú pháp) và đưa qua một lớp GCN.

Hình 3.1 thể hiện kiến trúc tổng quan của khoá luận. Khoá luận ứng dụng mô hình dựa trên mạng tích chập đồ thị được đề xuất bởi Zhang và cộng sự [13].

### 3.2 Tiền xử lý và chuẩn hoá

Dữ liệu đầu vào bao gồm câu và khía cạnh tương ứng trong câu đó. Để phục vụ cho việc dự đoán, đầu tiên, khối này tiến hành trích xuất các chú thích cần thiết cho câu đầu vào tiếng Việt  $s$ , bao gồm: mã tách từ, nhãn từ loại và phân tích cú pháp phụ thuộc. Đối với tiếng Việt, các mã tách từ có thể nhiều hơn một âm tiết do đặc trưng của ngôn ngữ. Với phân tích cú pháp phụ thuộc, trích xuất nhãn quan hệ phụ thuộc và vị trí từ mà từ hiện tại



Hình 3.1: Kiến trúc mô hình áp dụng [13]

phụ thuộc vào hay biến đổi. Các từ được đánh số từ một đến  $n$  với  $n$  là số lượng mã tách từ. Riêng từ có nhãn root có vị trí từ phụ thuộc vào là 0. Hiện nay, đã có nhiều thư viện hỗ trợ trích xuất các chú thích ngôn ngữ nêu trên cho tiếng Việt một cách nhanh chóng và chính xác, chẳng hạn như VnCoreNLP [10], Spacy cho tiếng Việt hay Underthesea. Khoá luận sử dụng thư viện Underthesea với ngôn ngữ Python để có được các chú thích cần thiết.

Tiếp đó, ma trận kề  $A$  của mỗi câu đầu vào tiếng Việt  $s$  được xây dựng với  $A \in \mathbb{R}^{n \times n}$ . Trong  $A$ , nếu hai nút  $v_i, v_j$  có kết nối với nhau thì  $A_{ij} = 1$  và ngược lại,  $A_{ij} = 0$ . Xem mỗi mã tách từ là một nút trong đồ thị cây cú pháp phụ thuộc, với câu đầu vào tiếng Việt, các mã tách từ được kí hiệu  $s = \{w_1, w_2, \dots, w_n\}$ . Một từ điển có khoá là các nút và giá trị là các nút có kết nối với nó dựa trên ma trận kề  $A$  cũng được xây dựng. Để tìm đường đi ngắn nhất giữa hai nút có kết nối với nhau, một ma trận đường đi ngắn nhất cũng được tạo ra và sử dụng để làm đặc trưng cho mỗi câu dữ liệu đầu vào.

### 3.3 Mô hình nhúng từ

Khoá luận sử dụng biểu diễn từ PhoW2V [5] để ánh xạ từ không gian từ sang không gian vector. PhoW2V cung cấp các bộ nhúng cấp độ từ và âm tiết Word2Vec được huấn luyện trước cho tiếng Việt trên kho văn bản lớn. Khoá luận lựa chọn bộ từ nhúng cấp độ từ với chiều vector nhúng là 300. Gọi  $|V|$  là kích thước bộ từ vựng,  $d_e$  là chiều của vector từ nhúng, ở đây là 300. Cho đầu vào là câu tiếng Việt  $s = \{w_1, w_2, \dots, w_n\}$  với các khía cạnh tương ứng  $a = \{a_1, a_2, \dots, a_m\}$  cũng chính là thành phần của câu  $s$ . Bước nhúng từ tiến hành ánh xạ mỗi từ trong  $s$  thành một vector giá trị thực chiều thấp với ma trận nhúng  $E \in \mathbb{R}^{|V| \times d_e}$ . Khi đó, câu  $s$  thu được kết quả các từ nhúng tương ứng với mỗi từ đầu vào  $x = \{x_1, x_2, \dots, x_n\}$ .



Mặt khác, PhoW2V lại không cho biết thông tin ngữ cảnh của các từ trong câu. Do đó, BiLSTM và BERT được sử dụng kết hợp để trích xuất các biểu diễn ngữ cảnh ẩn. BiLSTM tích hợp thông tin ngữ cảnh trong phần nhúng từ bằng cách theo dõi các phụ thuộc dọc theo chuỗi từ. BiLSTM lấy đầu vào là từ nhúng được trích xuất ở trên  $x = \{x_1, x_2, \dots, x_n\}$  theo đúng thứ tự đó. Lớp LSTM đầu tiên học các biểu diễn trạng thái ẩn  $\{\vec{h}_1^0, \vec{h}_2^0, \dots, \vec{h}_n^0\}$  theo chiều xuôi dọc theo từ nhúng  $x$ . Điều này cho phép nắm bắt thông tin ngữ cảnh theo chiều xuôi. Với cấu trúc tương tự, lớp LTSM theo hướng ngược lại sẽ học các biểu diễn  $\{\vec{h}_n^0, \vec{h}_{n-1}^0, \dots, \vec{h}_1^0\}$  trên  $x$ . Cuối cùng, nối các biểu diễn song song tương ứng được mô hình hóa bởi cả LSTM xuôi và ngược thành các biểu diễn chiều cao hơn  $H = \{h_1^0, h_2^0, \dots, h_n^0\}$  với  $h_i^{2d}$ , trong đó chứa chuỗi con  $h_a = \{h_{a_1}^0, h_{a_2}^0, \dots, h_{a_m}^0\}$  là trạng thái ẩn của các khía cạnh tương ứng. Theo cách làm này, có thể thu được thông tin ngữ cảnh giữa từ thể hiện ý kiến và khía cạnh.  $H$  cũng được sử dụng để làm biểu diễn nút ban đầu trong đồ thị.

Bộ mã hóa BERT là một loại mô hình mạng nơ ron, dựa trên kiến trúc máy biến áp và được đào tạo trước trên một kho văn bản lớn để tìm hiểu các cách biểu diễn từ phong phú theo ngữ cảnh. BERT xây dựng các máy biến áp sử dụng cơ chế chú ý để nắm bắt mối quan hệ giữa các từ thông qua bộ mã hóa đầu vào và bộ giải mã đầu ra. Không giống như các mô hình biểu diễn văn bản khác biểu diễn từng từ một, BERT có thể biểu diễn toàn bộ câu dựa trên các máy biến áp. Do đó, BERT có thể nắm bắt hiệu quả ngữ nghĩa ẩn giữa các từ khi tạo biểu diễn vector.

Đối với bộ mã hóa BERT của mô hình nhúng từ, đầu vào của bộ mã hóa BERT được tạo ra bằng cách nối câu với khía cạnh cần dự đoán quan điểm. Câu và khía cạnh được phân tách bằng mã thông báo đặc biệt, cụ thể là "[SEP]". Bằng cách này, mô hình có thể tính đến ngữ cảnh của câu cũng như khía cạnh đang được thảo luận. Cụ thể hơn, đầu vào có dạng "[CLS] câu [SEP] khía cạnh [SEP]", trong đó "[CLS]" đại diện cho một mã thông báo đặc biệt được sử dụng để biểu thị bắt đầu chuỗi đầu vào, câu chính là câu đầu vào và khía cạnh là khía cạnh hoặc thực thể quan tâm. Mã thông báo "[SEP]" cuối cùng được sử dụng để tách khía cạnh khỏi câu và để biểu thị phần cuối của chuỗi đầu vào. Định dạng đầu vào này cho phép bộ mã hóa BERT tìm hiểu các biểu diễn theo ngữ cảnh của cả câu và khía cạnh.

## 3.4 Bộ mã hoá

### 3.4.1 Mã hoá ngữ nghĩa

Bộ mã hoá ngữ nghĩa sử dụng cơ chế chú ý để nắm bắt thông tin ngữ nghĩa, bao gồm tự chú ý  $A_{self}$  và chú ý nhận thức khía cạnh  $A_{asp}$ . Cơ chế chú ý là một cách phổ biến để nắm bắt các tương tác giữa các từ khía cạnh và ngữ cảnh. Trong đó, tự chú ý sẽ xem xét ngữ nghĩa toàn cục của câu, trong khi chú ý nhận thức khía cạnh sẽ đánh giá mối tương quan ngữ nghĩa giữa các từ trong câu với khía cạnh cụ thể. Bộ mã hoá ngữ nghĩa kết hợp chú ý nhận thức khía cạnh và tự chú ý để có các đặc trưng ngữ nghĩa tốt hơn. Ở đây, bộ mã hoá ngữ nghĩa xây dựng  $p$  ma trận kể chú ý và  $p$  là một siêu tham số. Khoá luận lựa chọn  $p = 5$ .

#### Tự chú ý

Tự chú ý là một cơ chế trong học sâu, cho phép tính toán tầm quan trọng của mỗi từ trong câu với các từ còn lại. Cơ chế này thường được sử dụng trong ngữ cảnh các mô hình dựa trên máy biến áp như BERT hay GPT. Tự chú ý đã được sử dụng trong nhiều tác vụ xử lý ngôn ngữ tự nhiên như dịch máy, phân loại văn bản hay mô hình hoá ngôn ngữ. Thực nghiệm cho thấy tự chú ý thực sự mang lại hiệu quả trong việc nắm bắt các phụ thuộc lâu dài cũng như cải thiện hiệu suất của các mô hình này.

Tự chú ý có thể được xem là ánh xạ một truy vấn  $Q$  và một cặp khoá  $K$  - giá trị  $V$  với một đầu ra, trong đó truy vấn, khoá, giá trị và đầu ra đều là vector. Đầu ra được tính toán dưới dạng tổng trọng số của các giá trị, trong đó trọng số được gán cho từng giá trị được tính toán bằng hàm tương thích của truy vấn với khóa tương ứng. Cụ thể:

Cho chuỗi đầu vào  $Q = \{q_1, q_2, \dots, q_n\}$ , với  $q_i$  là vector  $d$  chiều. Đầu tiên, thực hiện biến đổi tuyến tính các vector đầu vào, thu được:

$$K = \{k_1, k_2, \dots, k_n\} = QW_K \quad (3.1)$$

$$V = \{v_1, v_2, \dots, v_n\} = QW_V \quad (3.2)$$

$$Q = \{q_1, q_2, \dots, q_n\} = QW_Q \quad (3.3)$$

Trong đó:  $W_K, W_V, W_Q$  là các ma trận trọng số có thể học được.

Sau đó, điểm tự chú ý cho mỗi cặp vector truy vấn-khoá được tính toán:

$$score(q_i, k_j) = \frac{q_i * k_j}{\sqrt{d}} \quad (3.4)$$

Trong đó:  $*$  biểu thị tích vô hướng và  $d$  là chiều của vector truy vấn và vector khoá.

Tiếp đó, các điểm được chuẩn hoá sử dụng hàm softmax:

$$attention(q_i) = softmax(score(q_i, K)) \quad (3.5)$$

Cuối cùng, tổng trọng số của vector giá trị được tính toán sử dụng điểm trọng số:

$$self - attention(q_i) = sum_j(attention(q_i)_j * v_j) \quad (3.6)$$

Trong đó:  $sum_j$  biểu thị tổng trên tất cả các vector giá trị và  $attention(q_i)_j$  là điểm chú ý của vector khoá thứ  $j$  tương ứng với vector truy vấn thứ  $i$ .

### Tự chú ý nhiều đầu

Tự chú ý nhiều đầu là phần mở rộng của tự chú ý, trong đó chuỗi đầu vào được chiếu  $h$  lần thành nhiều biểu diễn được gọi là "head". Trên mỗi phiên bản chiếu của truy vấn, khoá và giá trị, hàm chú ý được thể hiện song song, thu được giá trị đầu ra  $d_v$  chiều. Sau đó chúng được nối với nhau và được chiếu lần nữa, thu được kết quả cuối cùng.

Chú ý nhiều đầu cho phép mô hình cùng tham gia vào thông tin từ các không gian con biểu diễn khác nhau ở các vị trí khác nhau. Với một đầu chú ý duy nhất, tính trung bình sẽ hạn chế điều này.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (3.7)$$

Trong đó,  $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

Cụ thể,  $A_{self}$  ở đây được xây dựng sử dụng cơ chế tự chú ý  $p$  đầu để nắm bắt sự tương tác giữa hai từ tùy ý trong một câu.

$$A_{self}^i = \frac{QW^Q \times (KW^K)^T}{\sqrt{d}} \quad (3.8)$$

Trong đó:  $Q, K$  đều bằng  $H$  được tạo ra bởi lớp mã hoá.  $W^Q \in \mathbb{R}^{d \times d}$  là các trọng số học được.

### Chú ý nhận thức khía cạnh

Không giống như nhiệm vụ phân loại quan điểm ở cấp độ câu, phân loại quan điểm dựa trên khía cạnh nhằm đánh giá quan điểm của một thuật ngữ khía cạnh cụ thể trong câu ngữ cảnh của nó và do đó yêu cầu mô hình hóa mối tương quan ngữ nghĩa cụ thể dựa trên các thuật ngữ khía cạnh khác nhau. Cơ chế chú ý nhận thức khía cạnh được đề xuất, coi thuật ngữ khía cạnh là truy vấn để tính toán chú ý để tìm hiểu các đặc trưng liên quan đến khía cạnh:

$$A_{asp}^i = \tanh(H_a W^a \times (K W^k)^T) + b \quad (3.9)$$

Trong đó:  $K$  bằng  $H$  được tạo bởi lớp mã hóa.  $W^a \in \mathbb{R}^{d \times d}$  và  $W^k \in \mathbb{R}^{d \times d}$  là các trọng số có thể học được. Phép gộp trung bình được áp dụng trên  $h_a$  và sau đó sao chép nó  $n$  lần để thu được  $H_a \in \mathbb{R}^{n \times d}$  dưới dạng biểu diễn khía cạnh. Lưu ý rằng sử dụng chú ý nhận thức khía cạnh  $p$  đầu để thu được ma trận điểm chú ý cho một câu,  $A_{asp}^i$  chỉ ra rằng nó có được thông qua đầu chú ý thứ  $i$ .

Sau khi đã thu được ma trận điểm tự chú ý và chú ý nhận thức khía cạnh, chúng được nối với nhau:

$$A^i = A_{asp}^i + A_{self}^i \quad (3.10)$$

Trong đó:  $A^i \in \mathbb{R}^{n \times n}$  được sử dụng làm đầu vào cho tính toán của bộ mã hoá cú pháp tiếp theo. Đối với mỗi  $A^i$ , nó đại diện cho một đồ thị được kết nối đầy đủ.

### 3.4.2 Mã hoá cú pháp

Phần này giới thiệu ma trận mặt nạ cú pháp, sau đó che dấu từng đồ thị kết nối đầy đủ theo các khoảng cách cú pháp khác nhau. Xem cây phụ thuộc cú pháp là một đồ thị vô hướng và mỗi mã tách từ là một nút. Khi đó, khoảng cách giữa nút  $v_i$  và  $v_j$  trong cây phụ thuộc cú pháp được định nghĩa là  $d(v_i, v_j)$ . Vì có nhiều đường dẫn giữa các nút trên cây phụ thuộc cú pháp, xem khoảng cách của chúng chính là đường dẫn ngắn nhất  $D$ :

$$D(i, j) = \min d(v_i, v_j) \quad (3.11)$$

Ở phần trước, cơ chế chú ý  $p$  đầu thu được  $p$  ma trận kề. Vì vậy, đặt số lượng ma trận mặt nạ cú pháp dựa trên khoảng cách cú pháp khác nhau bằng số lượng đầu chú ý. Khi khoảng cách cú pháp tương đối nhỏ, mô hình của có thể học thông tin cục bộ. Ngược lại, nếu khoảng cách cú pháp tương đối lớn, thông tin cấu trúc toàn cục sẽ được

xem xét. Việc tính toán ma trận mặt nạ cú pháp  $M^k$  với ngưỡng  $k$  được xây dựng như sau:

$$M_{ij}^k = \begin{cases} 0, & D(i, j) \leq k \\ -\infty, & otherwise \end{cases} \quad (3.12)$$

Trong đó  $k \in [1, p]$ . Cụ thể thực nghiệm sử dụng  $k = 5$ . Để có được thông tin toàn cục và đặc trưng cục bộ, phạm vi chú ý bị hạn chế bởi các khoảng cách cú pháp khác nhau.

$$M = \{M^1, \dots, M^k, \dots, M^p\} \quad (3.13)$$

$$A_{mask}^i = softmax(A^i + M^i) \quad (3.14)$$

Tương tự, ma trận mặt nạ cú pháp dựa trên khoảng cách  $i$  được ký hiệu là  $A_i \in \mathbb{R}^{n \times n}$

### 3.4.3 Lớp GCN

Bộ mã hoá cú pháp thu được  $p$  ma trận mặt nạ cú pháp khác nhau, do đó cần có  $p$  phép tích chập đồ thị trên  $A_{mask} \in \mathbb{R}^{p \times n \times n}$ . Kí hiệu  $h^{l-1}$  là trạng thái đầu vào và  $h^l$  là trạng thái đầu ra của lớp thứ  $l$ ,  $h^0$  là đầu ra của lớp mã hoá câu. Mỗi nút trong lớp GCN thứ  $l$  được cập nhật dựa theo biểu diễn ẩn của các nút kề nó:

$$h_i^l = \sigma\left(\sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l\right) \quad (3.15)$$

Trong đó:  $W^l$  là trọng số biến đổi tuyến tính,  $b^l$  là thuật ngữ độ lệch,  $\sigma$  là hàm phi tuyến. Biểu diễn đầu ra cuối cùng của lớp GCN thứ  $l$  là  $H^l = \{h_1^l, h_2^l, \dots, h_n^l\}$ .

Sau khi tổng hợp biểu diễn nút từ mỗi lớp mã hoá sẽ thu được biểu diễn đặc trưng cuối cùng. Các từ không phải khía cạnh của biểu diễn đầu ra được học bởi lớp GCN cũng được che giấu để có được biểu diễn thuật ngữ khía cạnh. Hơn nữa, một tổng hợp trung bình để giữ lại hầu hết thông tin trong biểu diễn thuật ngữ khía cạnh  $h_a^l$ :

$$h_a^l = f(h_{a_1}^l, h_{a_2}^l, \dots, h_{a_m}^l) \quad (3.16)$$

Trong đó:  $f(.)$  là hàm gộp trung bình được áp dụng trên biểu diễn khía cạnh tăng cường bởi lớp GCN. Sau đó,  $h_a^l$  được đưa qua một lớp tuyến tính, theo sau là hàm softmax để thu được phân phối xác suất của không gian quyết định cực quan điểm:

$$p(a) = \text{softmax}(W_p h_a^l + b_p) \quad (3.17)$$

Trong đó:  $W_p, b_p$  là tham số và độ lệch có thể học được

### **Huấn luyện mô hình**

Hàm mất mát cross-entropy được sử dụng làm hàm mục tiêu:

$$L(\theta) = - \sum_{(s,a) \in \mathcal{D}} \sum_{c \in \mathcal{C}} \log p(a) \quad (3.18)$$

Trong đó:  $\mathcal{D}$  chứa tất cả các cặp câu-khía cạnh và  $a$  biểu thị khía cạnh xuất hiện trong câu  $s$ .  $\theta$  biểu thị tất cả các tham số huấn luyện được và  $\mathcal{C}$  là tập hợp tất cả cực quan điểm.

### **3.5 Tóm tắt chương 3**

Chương ba vừa trình bày mô hình đề xuất để phân loại quan điểm mức khía cạnh cho dữ liệu tiếng Việt. Cụ thể, có ba mô đun chính trong mô hình là tiền xử lý và chuẩn hoá, mô hình nhúng từ và bộ mã hoá. Chương ba đề cập đầu vào và đầu ra của từng mô đun cũng như cấu trúc tương ứng của chúng.

## Chương 4

# THỰC NGHIỆM VÀ ĐÁNH GIÁ

Chương bốn trình bày các bước và công cụ hỗ trợ triển khai thực nghiệm để đánh giá hiệu năng mô hình được đề xuất ở chương ba cho nhiệm vụ phân lớp quan điểm. Mô hình được huấn luyện từ đầu với bộ dữ liệu tiếng Việt UIT-ViSFD. Bộ dữ liệu cũng được gán thêm nhãn vị trí của khía cạnh và từ quan điểm thể hiện trong câu. Chi tiết cụ thể về kịch bản thực nghiệm và tập dữ liệu được trình bày chi tiết trong các mục dưới đây.

### 4.1 Kế hoạch thực nghiệm

Khóa luận tiến hành thực nghiệm với mô hình được đề xuất ở chương ba trên bộ dữ liệu tiếng Việt UIT-ViSFD. Nhãn ban đầu của bộ dữ liệu chỉ bao gồm danh mục khía cạnh và quan điểm tương ứng. Tuy nhiên để phục vụ cho bài toán và ứng dụng được với mô hình đề xuất, khóa luận cũng tiến hành trích xuất thêm vị trí của danh mục và từ thể hiện ý kiến trong câu.

**Độ đo đánh giá:** sử dụng các độ đo cho mô hình phân lớp nhiều lớp là precision, recall và F1 (macro-average và weighted-average).

### 4.2 Phần mềm và phần cứng

Danh sách phần mềm và công cụ sử dụng cho việc thực nghiệm được trình bày trong bảng 4.1.

Mô hình được huấn luyện sử dụng Google Colab với cài đặt cấu hình như sau:



Bảng 4.1: Danh sách công cụ và phần mềm

STT	Tên phần mềm	Vai trò	Nguồn
1	Jupyter Notebook	Jupyter là một nền tảng tính toán khoa học mã nguồn mở, với khả năng nổi bật cho phép tương tác trực tiếp với từng dòng code, hỗ trợ hơn 40 ngôn ngữ lập trình, trong đó tập trung vào 3 ngôn ngữ là Julia, Python và R	<a href="https://jupyter.org/">https://jupyter.org/</a>
2	Google Colab	Cho phép thực thi Python trên nền tảng đám mây, không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp	<a href="https://colab.research.google.com">https://colab.research.google.com</a>
3	Pytorch	PyTorch là một framework học máy dựa trên thư viện Torch, được sử dụng trong lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên, do Meta AI phát triển và ngày nay là một phần của Linux Foundation	<a href="https://pytorch.org/">https://pytorch.org/</a>
4	Mô hình nâng cao ngữ nghĩa và cú pháp cho phân lớp quan điểm mức khía cạnh	Mô hình cho phân lớp quan điểm mức khía cạnh được áp dụng ở chương ba	<a href="https://github.com/zhangzheng1997/SSEGCN-ABSA">https://github.com/zhangzheng1997/SSEGCN-ABSA</a>

- Số lượng CPU: 2
- Vendor ID: GenuineIntel
- CPU Family: 6
- Model: 79
- Tên Model: Intel(R) Xeon(R) CPU @ 2.20GHz
- Microcode: 0x1

- Cpu MHz: 2200.000
- Dung lượng Cache: 56320 KB
- Số nhân CPU: 1
- Cpuid level: 13
- Bogomips: 4400.00
- Address Sizes: 46 bits vật lý và 48 bits ảo
- 1 GPU Tesla T4

### 4.3 Bộ dữ liệu tiếng Việt UIT-ViSFD và UIT-ViSD4SA

UIT-ViSFD là bộ dữ liệu được đề xuất bởi Phan và cộng sự [7], gồm các đánh giá của người dùng về điện thoại di động. Bộ dữ liệu gồm 11,122 bản ghi, được chia thành ba phần: tập huấn luyện, tập xác thực và tập kiểm thử. Tập dữ liệu được sử dụng cho các tác vụ ABSA liên quan đến danh mục khía cạnh và xu hướng quan điểm. Trong đó có 10 danh mục khía cạnh liên quan đến điện thoại di động được định nghĩa trước và 3 nhãn quan điểm bao gồm tích cực, trung lập, tiêu cực. Dưới đây là thống kê về cấu trúc dữ liệu cũng như tóm tắt số lượng dữ liệu.

Bảng 4.2: Các trường dữ liệu trong UIT-ViSFD

STT	Trường dữ liệu	Mô tả	Kiểu dữ liệu
1	<i>comment</i>	Bình luận của người dùng	Chuỗi
2	<i>n_star</i>	Đánh giá theo sao về sản phẩm	Số nguyên
3	<i>date_time</i>	Thời gian đánh giá	Chuỗi
4	<i>label</i>	Nhãn quan điểm cho danh mục khía cạnh	Chuỗi

Bảng 4.3: Các danh mục khía cạnh thuộc về chủ đề điện thoại di động trong UIT-ViSFD

STT	Danh mục	Mô tả
1	MÀN HÌNH	Kích thước, màu sắc, công nghệ hiển thị
2	CAMERA	Chất lượng camera, độ rung, độ trễ, tiêu điểm, màu sắc hình ảnh
3	TÍNH NĂNG	Vân tay, kết nối wifi, cảm biến, nhận diện khuôn mặt
4	PIN	Dung lượng, chất lượng pin
5	HIỆU NĂNG	Dung lượng ram, chip xử lý, hiệu năng sử dụng, độ mượt
6	LƯU TRỮ	Dung lượng bộ nhớ, khả năng mở dung lượng
7	THIẾT KẾ	Phong cách, thiết kế, vỏ điện thoại
8	GIÁ	Đánh giá về giá sản phẩm
9	TỔNG QUÁT	Nhận xét chung của khách hàng về điện thoại
10	DỊCH VỤ VÀ PHỤ KIỆN	Dịch vụ bán hàng, bảo hành, nhân viên, phụ kiện điện thoại

Bảng 4.4: Thống kê số lượng dữ liệu theo khía cạnh và theo nhãn quan điểm trong 3 tập dữ liệu huấn luyện, xác thực, kiểm thử

Khía cạnh	Huấn luyện			Xác thực			Kiểm thử			Tổng
	Tích cực	Trung lập	Tiêu cực	Tích cực	Trung lập	Tiêu cực	Tích cực	Trung lập	Tiêu cực	
PIN	2,027	349	1,228	303	51	150	554	92	368	5,122
CAMERA	1,231	288	627	172	36	88	346	71	171	3,030
THIẾT KẾ	999	77	302	135	12	40	274	28	96	1,963
TÍNH NĂNG	785	198	1,659	115	33	233	200	52	459	3,734
TỔNG QUÁT	3,627	290	949	528	34	127	1,004	83	294	6,936
HIỆU NĂNG	2,253	391	1,496	327	45	210	602	116	454	5,894
GIÁ	609	391	316	72	144	36	162	328	79	2,882
MÀN HÌNH	514	56	379	62	12	47	136	17	116	1,339
DỊCH VỤ VÀ PHỤ KIỆN	1,401	107	487	199	13	78	199	27	167	2,678
DUNG LƯỢNG	59	107	21	11	1	2	18	3	6	132
<b>Tổng</b>	13,505	2,903	7,464	1,924	381	1,011	3,495	817	2,210	

Bộ dữ liệu UIT-ViSD4SA [9] là bộ dữ liệu được gán nhãn vị trí cho khía cạnh và từ thể hiện quan điểm dựa trên bộ dữ liệu UIT-ViSFD. Tuy nhiên, hai bộ dữ liệu này có nhãn khía cạnh và quan điểm lệch nhau.

## 4.4 Cài đặt môi trường thực nghiệm

### 4.4.1 Tiền xử lý dữ liệu

Cả ba tập dữ liệu bao gồm huấn luyện, xác thực và kiểm thử được đưa vào tiền xử lý. Đầu tiên, các trường dữ liệu không dùng đến như `n_star` và `data_time` được loại bỏ. Tiếp đó, tiến hành gán nhãn vị trí cho khía cạnh và từ thể hiện quan điểm trong câu. Việc gán nhãn vị trí dựa trên khía cạnh và nhãn quan điểm đã có sẵn, được thực hiện thủ công.

Ví dụ, cho câu **"Pin trâu . Giá rẻ. Cầu hình ngon . Mượt mà . Máy chơi game rất tốt . Nhân viên tư vấn nhiệt tình"** với nhãn gốc từ tập dữ liệu là **"BATTERY#Positive; PERFORMANCE#Positive; PRICE#Positive; SERACC#Positive"**. Giả sử với danh mục BATTERY, trích xuất vị trí tương ứng với việc tìm vị trí bắt đầu và kết thúc của "Pin trâu" trong câu trên. Câu được đánh vị trí từ 0. Khi đó thu được nhãn mới chứa vị trí là **"[0, 8, 'BATTERY#POSITIVE']"**. Tương tự với các nhãn còn lại thu được kết quả nhãn **"[[0, 8, 'BATTERY#POSITIVE'], [11, 17, 'PRICE#POSITIVE'], [19, 66, 'PERFORMANCE#POSITIVE'], [69, 96, 'SERACC#POSITIVE']]"**.

Sau khi đã gán nhãn vị trí, bộ dữ liệu được trích xuất các chú thích cần thiết từ trường dữ liệu *comment* sử dụng gói thư viện Underthesea của Python. Các bước bao gồm: đưa dữ liệu về lowercase, tách từ, trích xuất chú thích từ loại PoS, trích xuất chú thích phân tích cú pháp phụ thuộc và thay đổi định dạng nhãn quan điểm. Kết quả cuối cùng được lưu thành file json với các trường dữ liệu dưới đây:

- **token:** Chứa các mã tách từ.
- **pos:** Chứa các nhãn từ loại.
- **head:** Chỉ số head của token hiện tại có được trong quan hệ head-dependent từ cây phụ thuộc. Đánh dấu chỉ số câu bắt đầu từ 1, ROOT có head = 0.
- **deprel:** Chứa các nhãn quan hệ phụ thuộc giữa head-dependent.
- **aspects:** Chứa các thành phần của khía cạnh, bao gồm:
  - **term:** thuật ngữ khía cạnh
  - **from:** vị trí bắt đầu của khía cạnh và ý kiến tương ứng được trích xuất
  - **to:** vị trí kết thúc của khía cạnh và ý kiến tương ứng được trích xuất
  - **polarity:** xu hướng quan điểm

Ví dụ một mẫu dữ liệu sau khi xử lý:

```
{  
    "token":["máy","mượt","pin","trâu"],  
    "pos":["N","V","V","N"],  
    "head":[0,1,1,1],  
    "deprel":["root","nmod","nmod","nmod"],  
    "aspects":  
        [{"term":["performance"],  
            "from":0,  
            "to":2,  
            "polarity":"positive",  
            "term":["battery"],  
            "from":11,  
            "to":13,  
            "polarity":"positive"}]  
}
```

#### 4.4.2 Kịch bản thực nghiệm

Khoá luận tiến hành thực nghiệm huấn luyện trên mô hình ngay từ đầu. Tập dữ liệu sử dụng là tập dữ liệu UIT-ViSFD sau khi đã được tiền xử lý ở các bước trên. Khoá luận sử dụng toàn bộ tập dữ liệu để huấn luyện, giữ nguyên phân chia tập huấn luyện, xác thực và kiểm thử như tập dữ liệu gốc.

### 4.5 Đánh giá kết quả thực nghiệm

#### 4.5.1 Độ đo đánh giá

Sử dụng các độ đo accuracy, precision, recall, F1, macro-average và weighted-average. Đây là các chỉ số đánh giá phổ biến được dùng cho mô hình phân lớp, trong đó macro-average và weighted-average được sử dụng cho bài toán phân lớp nhiều lớp.

Gọi True Positive (TP) là số lượng nhãn mà mô hình dự đoán khớp nhãn thực tế là positive, True Negative (TN) là số lượng nhãn mà mô hình dự đoán khớp nhãn thực tế là negative, False Positive (FP) là số lượng nhãn mà mô hình dự đoán là positive nhưng nhãn thực tế lại là negative và False Negative (FN) là số lượng nhãn mà mô hình dự đoán là negative nhưng nhãn thực tế lại là positive.

Khi đó:

- Accuracy tính tỉ lệ giữa số nhãn được dự đoán đúng và tổng số nhãn dự đoán.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

- Precision cho biết trong tất cả trường hợp dự đoán là positive thì có tỉ lệ bao nhiêu trường hợp dự đoán đúng.

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

- Recall đo lường tỷ lệ dự đoán chính xác các trường hợp positive trên toàn bộ các nhãn thực tế thuộc nhóm positive.

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

- Điểm  $F_1$  là trung bình điều hòa giữa precision và recall. Do đó  $F_1$  đại diện hơn trong việc đánh giá độ chính xác trên đồng thời precision và recall.

$$F_1 = \frac{(2 \times P \times R)}{(P + R)} \quad (4.4)$$

- Trong trường hợp phân lớp nhiều lớp, các phương pháp tính trung bình được áp dụng để tính điểm  $F_1$ . Macro-average được tính bằng cách sử dụng trung bình cộng (còn gọi là trung bình không trọng số) của tất cả các điểm  $F_1$  cho mỗi lớp. Weighted-average tính trung bình cộng điểm  $F_1$  nhân với tỉ lệ support tương ứng của mỗi lớp (còn gọi là trung bình có trọng số), với support là số lượng tương ứng của mỗi lớp trong tập dữ liệu.

#### 4.5.2 Cài đặt cấu hình

Mô hình được huấn luyện trên CPU với các siêu tham số:

- Hàm tối ưu hóa: Adam
- Trình khởi tạo tham số ban đầu: xavier initializer

- Tỷ lệ học: 0.002
- Bộ điều chỉnh: L2
- Alpha, beta: 0.25, 0.25
- Số lượng epoch: 20
- Kích thước batch: 16

### **Mô hình nhúng từ**

Khoá luận sử dụng vector nhúng PhoW2V [5] được huấn luyện trước trên bộ dữ liệu tiếng Việt ở cấp độ từ. Chiều của vector nhúng là 300.

### **Bộ mã hoá**

Với mã hoá ngữ nghĩa, sử dụng tự chú ý nhiều đầu với số đầu là 5. Đây cũng chính là số lượng ma trận mặt nạ cú pháp và số phép tích chập đồ thị trên lớp GCN. Với mã hoá cú pháp, sử dụng đồ thị vô hướng. Với lớp GCN, sử dụng 1 lớp GCN với 50 lớp ẩn. Tỷ lệ dropout cho lớp này là 0.1.

## 4.6 Kết quả thực nghiệm

Bảng 4.5: Kết quả thực nghiệm trên bộ dữ liệu UIT-ViSFD

	<b>Precision</b>	<b>Recall</b>	<b>F1</b>	<b>support</b>
<b>positive</b>	0.7654	0.9714	0.8562	4192
<b>negative</b>	0.8766	0.5832	0.7004	2279
<b>neutral</b>	0.8182	0.1589	0.2662	453
<b>accuracy</b>			0.7904	6924
<b>macro avg</b>	0.8201	0.5712	<b>0.6076</b>	6924
<b>weighted avg</b>	0.8055	0.7904	0.7663	6924

		Predicted labels		
True labels		4072	112	8
		942	1329	8
		306	75	72

Hình 4.1: Ma trận nhầm lẫn trên bộ dữ liệu UIT-ViSFD

Bảng 4.6: Kết quả thực nghiệm trên bộ dữ liệu UIT-ViSD4SA

	<b>Precision</b>	<b>Recall</b>	<b>F1</b>	<b>support</b>
<b>positive</b>	0.7610	0.9684	0.8522	4333
<b>negative</b>	0.8672	0.5592	0.6799	2289
<b>neutral</b>	0.7547	0.0950	0.1688	421
<b>accuracy</b>			0.7832	7043
<b>macro avg</b>	0.7943	0.5409	<b>0.5670</b>	7043
<b>weighted avg</b>	0.7951	0.7832	0.7554	7043

### Nhận xét:

Hiện tại, kết quả đạt được khi sử dụng mô hình khoá luận ở chương ba cho bộ dữ liệu UIT-ViSFD có được precision là 0.7943, recall là 0.5409 và F1 là 0.5670 với độ đo macro-average. Khi so sánh với các baseline khác, kết quả này đang cao hơn so với các mô hình Naive Bayes, SVM, Random Forest, CNN, LSTM và đang thấp hơn so với BiLSTM. Đối với bộ dữ liệu UIT-ViSD4SA, kết quả của khoá luận đang là 0.7943 cho



		Predicted labels		
True labels		4196	128	9
		1005	1280	4
		313	68	40

Hình 4.2: Ma trận nhầm lẫn trên bộ dữ liệu UIT-ViSD4SA

Bảng 4.7: Kết quả so sánh với các baseline khác trên bộ dữ liệu UIT-ViSFD

	<b>Precision</b>	<b>Recall</b>	<b>F1 (macro-avg)</b>
<b>Naive Bayes</b>	0.4907	0.3043	0.3756
<b>SVM</b>	0.1609	0.2335	0.1969
<b>Random Forest</b>	0.1711	0.2456	0.2017
<b>CNN</b>	0.3334	0.2292	0.2716
<b>LSTM</b>	0.5651	0.4839	0.5213
<b>BiLSTM</b>	0.6582	0.6053	<b>0.6306</b>
<b>Mô hình khoá luận</b>	0.8201	0.5712	0.6076
			0.7663 (weighted-avg)

Bảng 4.8: Kết quả so sánh với các baseline khác trên bộ dữ liệu UIT-ViSD4SA

	<b>Precision</b>	<b>Recall</b>	<b>F1 (macro-avg)</b>
<b>BiLSTM-CRF</b>	0.4684	0.4546	0.4570
<b>Mô hình khoá luận</b>	0.7943	0.5409	<b>0.5670</b>

precision, 0.5409 cho recall và 0.5670 cho F1. Kết quả này hiện đang cao hơn so với mô hình BiLSTM-CRF được sử dụng cho cùng bộ dữ liệu.

# KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU TIẾP THEO

## KẾT QUẢ ĐẠT ĐƯỢC VÀ HẠN CHẾ

Thực nghiệm cho thấy kết quả đạt được khi áp dụng mô hình khoá luận với bộ dữ liệu tiếng Việt UIT-ViSFD đang đứng thứ hai so với các baseline khác. Nguyên nhân ở đây là do chênh lệch nhãn đang lớn, cụ thể nhãn neutral đang quá ít so với hai nhãn còn lại, thực chất là do khi trích xuất vị trí của khía cạnh, một khía cạnh được đề cập trong câu có thể nằm ở nhiều vị trí với nhãn đối nghịch nhau (tích cực và tiêu cực) dẫn đến đánh giá chung thì nhãn sẽ là trung lập.

## ĐỊNH HƯỚNG TƯƠNG LAI

Khoá luận tiến hành xử lý lại dữ liệu để cải thiện việc lệch nhãn và nghiên cứu cải tiến mô hình nhằm mang lại kết quả tốt hơn.

# TÀI LIỆU THAM KHẢO

- [1] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, “Effective attention modeling for aspect-level sentiment classification,” in *Proceedings of the 27th International Conference on Computational Linguistics*, Aug. 2018, pp. 1121–1131.
- [2] N. Hà Nam, N. T. Thành, and H. Q. Thụy, “*Giáo trình Khai phá dữ liệu*”. NXB Giáo dục, 2013.
- [3] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2017.
- [4] R. Li, H. Chen, F. Feng, Z. Ma, X. Wang, and E. Hovy, “Dual graph convolutional networks for aspect-based sentiment analysis,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Aug. 2021, pp. 6319–6329.
- [5] A. T. Nguyen, M. H. Dao, and D. Q. Nguyen, “A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4079–4085.
- [6] H. T. Phan, N. T. Nguyen, and D. Hwang, “Aspect-level sentiment analysis: A survey of graph convolutional network methods,” *Information Fusion*, vol. 91, pp. 149–172, 2023.
- [7] L. Phan, P. Phúc, K. Nguyen, S. Huynh, L. Thanh Nguyen, T. Huynh, and K. Nguyen, “Sa2sl: From aspect-based sentiment analysis to social listening system for business intelligence,” 05 2021.
- [8] H. Tang, D. Ji, C. Li, and Q. Zhou, “Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification,” in *Proceedings of the 58th An-*

*nual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, Jul. 2020, pp. 6578–6588.

- [9] K. N. T. Thanh, S. H. Khai, P. P. Huynh, L. P. Luc, D.-V. Nguyen, and K. N. Van, “Span detection for aspect-based sentiment analysis in vietnamese,” in *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*. Association for Computational Linguistics, 11 2021, pp. 318–328.
- [10] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, “VnCoreNLP: A Vietnamese natural language processing toolkit,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, Jun. 2018, pp. 56–60.
- [11] C. Zhang, Q. Li, and D. Song, “Aspect-based sentiment classification with aspect-specific graph convolutional networks,” 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)*. Association for Computational Linguistics, Nov. 2019, pp. 4568–4578.
- [12] L. Zhang and B. Liu, “Sentiment analysis and opinion mining,” in *Encyclopedia of Machine Learning and Data Mining*, 2012.
- [13] Z. Zhang, Z. Zhou, and Y. Wang, “SSEGCN: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022, pp. 4916–4925.