Báo Cáo về Mô Hình CARTE: Tiền Huấn Luyện và Chuyển Giao với Việc Học trên Dữ Liệu Bảng

Nguyễn Hữu Lộc - 23C15031 Lê Trường Vũ - 23C15042

09/02/2025

Mục lục

1 Giới thiệu sơ lược

Các mô hình học máy tiền huấn luyện (pre-trained model) hiện nay đã trở nên phổ biến hơn nhiều và góp phần đáng kể vào sự phát triển của lĩnh vực trí tuệ nhân tạo trên nhiều loại dữ liệu khác nhau cả về mặt học thuật lẫn thương mại, nhất là trong các mảng về hình ảnh hay văn bản. Những mô hình này có thể được tải xuống từ các kho lưu trữ như HuggingFace, mang theo một lượng lớn thông tin ngầm và các phép biến đổi phức tạp, giúp tận dụng sức mạnh của học sâu ngay cả khi dữ liệu huấn luyện có quy mô nhỏ. Chính nhờ cách tiếp cận này, các mô hình nền tảng (foundation model), đặc biệt là các mô hình ngôn ngữ lớn (LLM), đã bùng nổ và định hình lại nhiều lĩnh vực.

Tuy nhiên, đối với dữ liệu dạng bảng thì dù loại dữ liệu này có ý nghĩa quan trọng trong môi trường doanh nghiệp và tổ chức nhưng tới hiện tại vẫn chưa có một mô hình nào có thể tạo nền tảng như vậy và trở ngại chính đến từ việc tích hợp dữ liệu từ nhiều bảng khác nhau. Đôi khi, các bảng có thể không chứa bất kỳ thông tin liên quan nào với nhau, và khi có, việc kết nối chúng lại trở thành một bài toán phức tạp trong lĩnh vực nghiên cứu cơ sở dữ liệu. Những thách thức điển hình bao gồm việc tìm mối tương ứng giữa các cột (đối sánh sơ đồ - schema matching) hoặc xử lý các nguồn dữ liệu có cách đặt tên khác nhau cho cùng một đối tượng (đối sánh đối tượng - entity matching). Do sự khác biệt này, việc huấn luyện trước trên dữ liệu bảng vẫn chưa khả thi, khiến phương pháp học sâu chưa thực sự đáng cân nhắc so với các phương pháp khác dựa trên cây quyết định.

Trong nghiên cứu này, nhóm đã đề xuất kiến trúc học máy CARTE (Context-Aware Representation of Table Entries) có thể học từ nhiều bảng dữ liệu mà không cần phải đối sánh sơ đồ hay đối sánh chuỗi ký tự. Cách tiếp cận cốt lõi của nhóm nghiên cứu là biểu diễn bảng dưới dạng đồ thị, trong đó các cột và giá trị trong bảng được ánh xạ thành các vector nhúng. Mô hình này được huấn luyện trước trên một cơ sở tri thức quy mô lớn, giúp nó có khả năng học hỏi từ một lượng lớn các đối tượng và mối quan hệ. Từ đó, mô hình có thể được tinh chỉnh cho các tác vụ cụ thể, ví dụ như những trường hợp hạn chế về dữ liệu huấn luyện, học từ nhiều bảng cùng lúc hay giúp bổ sung thông tin từ các nguồn dữ liệu thiếu liên kết.Thực nghiệm cho thấy CARTE mang lại sự cải thiện đáng kể về hiệu suất, vượt trội hơn hẳn so với 42 phương pháp cơ sở (baseline), bao gồm cả các phương pháp dựa trên cây quyết định và các kỹ thuật trích xuất đặc trưng khác, và thậm chí có hiệu quả cao đối với các bảng chứa dữ liệu dạng chuỗi văn bản, vốn rất phổ biến trong thực tế nhưng lại ít được đề cập trong các bộ dữ liệu chuẩn dành cho học máy.

2 Các Nghiên Cứu Liên Quan

Dữ liệu bảng đóng vai trò quan trọng trong nhiều ứng dụng thực tế, do đó, đã có nhiều phương pháp học sâu được phát triển để xử lý loại dữ liệu này. Tuy nhiên, trong hầu hết các trường hợp, các phương pháp này vẫn chưa thể vượt qua được những mô hình dựa trên cây quyết định. Mặc dù một số nghiên cứu đã chỉ ra rằng mạng nơ-ron có thể hoạt động tốt trên một số loại bảng nhất định, và nhiều kiến trúc hứa hẹn vẫn liên tục được nghiên cứu, nhưng để thực sự tạo vượt qua được thì các mô hình học sâu trên dữ liệu bảng cần phải tạo được một bước ngoặc lớn, chẳng hạn như khả năng tận dụng tri thức nền tảng tương tự như các mô hình ngôn ngữ lớn (LLM) đã làm.

Học chuyển giao trong dữ liệu bảng chủ yếu tập trung vào các trường hợp mà tập dữ liệu đích có cùng cấu trúc cột với tập dữ liệu nguồn. Một số phương pháp tiếp cận đã được đề xuất mô hình được huấn luyện trước trên một lượng dữ liệu dạng bảng lớn hơn nhưng không gán nhãn, hoặc sử dụng các bộ dữ liệu liên quan để thu hẹp khoảng cách giữa học sâu và mô hình cây quyết định, đặc biệt là trong bối cảnh dữ liệu y tế.

Một số phương pháp có thể xử lý các bảng với cột khác nhau bằng cách ánh xạ dữ liệu sang không gian đặc trưng chung, điển hình là XTab, nhưng vẫn chưa đạt hiệu suất tốt hơn so với các mô hình cây hiện đại. Ngoài ra, cũng có các nghiên cứu hướng đến việc biểu diễn từng dòng dữ liệu dưới dạng vector nhúng để học trên nhiều bảng, đặc trưng có Transtab đã vượt qua một số mô hình cơ sở như là XGBoost. Tuy nhiên những phương pháp này có thể mở rộng cho nhiều ứng dụng khác nhau hay không.

Một số tiến bộ đã đạt được trong việc tiền huấn luyện mô hình trên dữ liệu bảng, chẳng hạn như TabPFN – một mô hình Transformer được huấn luyện trên dữ liệu tổng hợp, giúp cải thiện hiệu suất trên các tập dữ liệu nhỏ. Tuy nhiên, nó chưa có cơ chế xử lý tốt các cột dạng phân loại, vốn là một điểm mạnh của các mô hình cây. Các mô hình ngôn ngữ lớn (LLM) cũng đã được thử nghiệm cho dữ liệu bảng, chẳng hạn như TabLLM, trong đó dữ liệu bảng được chuyển thành những chuỗi token để ứng dụng vào việc tinh chỉnh các mô hình LLM. Tuy nhiên, do khó khăn trong việc xử lý số liệu trong các mô hình này, chúng vẫn chưa đủ tính cạnh tranh so với TabPFN hay cả những phương pháp dựa theo cây quyết định.

Một thách thức quan trọng của dữ liệu bảng là sự xuất hiện của các giá trị rời rạc, thường được biểu diễn dưới dạng chuỗi văn bản. Một số nghiên cứu đã phát triển cách tiếp cận dựa trên biểu diễn chuỗi để hỗ trợ việc học từ loại dữ liệu này. Chẳng hạn như mô hình TableVectorizer chuyển đổi các cột khác nhau thành dạng ma trận số phù hợp cho việc học máy, hoặc mô hình KEN sử dụng phương pháp nhúng trong đồ thị tri thức để mã hóa thông tin từ nguồn như Wikipedia. Tuy nhiên, những phương pháp này thường yêu cầu ánh xạ chính xác từng giá trị của cột sang một thực thể cụ thể liên kết đến Wikipedia, tạo ra nhu cầu phải xử lý thêm tác vụ đối sánh đối tượng (entity matching).

Các mô hình thống kê truyền thống thường yêu cầu dữ liệu được tập hợp trong một bảng nhất quán, một bài toán thuộc lĩnh vực tích hợp dữ liệu. Một trong những thách thức quan trọng của lĩnh vực này là đối sánh sơ đồ (tìm cột tương ứng giữa các nguồn dữ liệu khác nhau) và đối sánh đối tượng (liên kết các chuỗi ký tự với đối tượng thực tế). Các phương pháp học sâu, đặc biệt là các mô hình dựa trên cơ chế tự chú ý, đã được áp dụng để hỗ trợ chuẩn hóa và tích hợp dữ liệu, giúp tự động hóa nhiều tác vụ mà trước đây cần đến sự can thiệp thủ công.

Tuy nhiên, nghiên cứu của nhóm nghiên cứu hướng đến một mục tiêu khác: thay vì dựa vào việc đối sánh rõ ràng các đối tượng hoặc sơ đồ, nhóm nghiên cứu tìm cách khai thác cấu trúc ngầm của dữ liệu để hỗ trợ các bài toán học máy mà không yêu cầu bất kỳ thao tác thủ công nào trong việc tìm kiếm nguồn dữ liệu liên quan. Đây là một vấn đề cấp thiết bởi hiện tại, có nhiều nhóm nghiên cứu đang nỗ lực xây dựng các kho dữ liệu bảng quy mô lớn nhưng do quy mô nhỏ của hầu hết các bảng dữ liệu hiện có và sự khác biệt lớn giữa các tập dữ liệu bảng.

3 Cách Mô Hình CARTE Học từ Dữ Liệu Bảng

CARTE có thể học trên nhiều bảng dữ liệu khác nhau nhờ vào hai yếu tố chính: một cách biểu diễn mới cho đối tượng dạng bảng với cấu trúc đồ thị và một kiến trúc mạng nơ-ron sâu có khả năng nắm bắt ngữ cảnh(context) của bảng. Trong đó, cách biểu diễn đồ thị giúp đồng bộ hóa nhiều bảng dữ liệu vào cùng một không gian biểu diễn, làm cho việc tiền huấn luyện trên các dữ liệu nền tảng thiếu liên kết trước đây trở nên khả thi. Đồng thời, mạng nơ-ron sâu có khả năng nhận thức ngữ cảnh giúp truyền tải các thông tin nền sáng các tác vụ.

3.1 Biểu Diễn Đồ Thị của Đối Tượng Bảng

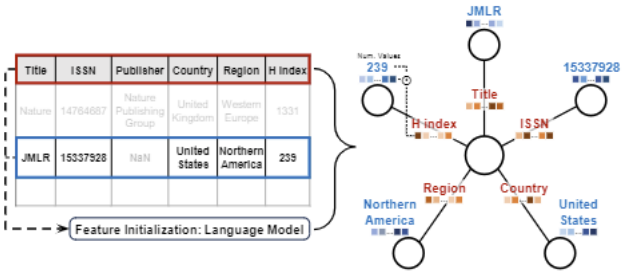
Việc sử dụng đồ thị là yếu tố quan trọng để giúp khái quát hóa các đối tượng bảng (tabular entity). Một đồ thị G bao gồm tập hợp các nút và cạnh, trong đó các nút biểu diễn đối tượng và các cạnh mô tả quan hệ giữa chúng. Đồ thị là công cụ mạnh mẽ để biểu diễn thông tin quan hệ giữa các đối tượng, và học sâu trên đồ thị đã được chứng minh là có tiềm năng lớn trong các tác vụ liên quan cơ sở dữ liệu có quan hệ (relational database).

CARTE tổ chức mỗi dòng dữ liệu trong bảng dưới dạng một đồ thị nhỏ. Giả sử bảng có k cột, CARTE biểu diễn dòng dữ liệu thứ i dưới dạng một đồ thị $G_i(X, E)$ trong đó X và E lần lượt là đặc trưng của nút và cạnh, được biểu diễn trong không gian R^d. Cấu trúc của $G_i(X, E)$ có dạng một đồ thị hình sao với $k - p_i$ nút lá, trong đó p_i là số cột có giá trị bị thiếu trong dòng i. Trên mỗi đồ thị con này, các nút lá được gán nhãn theo giá trị ô dữ liệu và tên cột tương ứng, có thể thấy thông qua Hình 1.

Để biến đồ thị này thành đầu vào cho mạng nơ-ron, CARTE sử dụng một mô hình ngôn ngữ để khởi tạo các đặc trưng. Cụ thể là với các giá trị phân loại và tên cột, CARTE dùng mô hình ngôn ngữ để chuyển đổi giá trị thành dữ liệu dạng nhúng có d chiều. Với các giá trị số, đặc trưng của nút được khởi tạo bằng cách nhân giá trị số với nhúng của tên cột tương ứng. Ví dụ, nếu một ô chứa giá trị 239 trong cột "H index", đặc trưng của nút đó sẽ là $239 \times E_{Hindex}$. Nút trung tâm của đồ thị được khởi tạo bằng giá trị trung bình của các nút lá và sẽ đóng vai trò là nút "readout". Trong mạng Neural Graph (Graph Neural Network - GNN), "node readout" (đọc nút) là một quá trình tổng hợp thông tin từ các nút (nodes) của đồ thị để tạo ra một biểu diễn đặc trưng có ý nghĩa cho toàn bộ đồ thị hoặc một phần của nó.

Ưu điểm của cách biểu diễn đồ thị là cách tiếp cận này mang lại nhiều lợi ích. Giữ ngữ cảnh của dữ liệu bảng: Trong dữ liệu bảng, mỗi giá trị chỉ có ý nghĩa khi được xem xét trong ngữ cảnh của tên cột. Ví dụ như ở Hình 1, một dòng dữ liệu gồm các giá trị "JMLR", "15337928", "239" sẽ khó hiểu nếu không có tên cột đi kèm. Tuy nhiên, khi biết rằng chúng thuộc các cột "Title", "ISSN" và "H index", ta có thể nhận ra rằng đây là một tạp chí học thuật. CARTE khai thác ngữ cảnh này thông qua đồ thị, giúp mô hình học tốt hơn.

Vì các bảng có thể có số lượng cột khác nhau hoặc trật tự cột khác nhau, cách biểu diễn đồ thị của CARTE giúp kết nối dữ liệu mà không cần phải dữ



Hình 1: CARTE biểu diễn mỗi hàng của bảng dưới dạng một đồ thị hình sao (graphlet). Các nút lá chứa giá trị ô và tên cột tương ứng. Các đặc trưng nút được khởi tạo bằng mô hình ngôn ngữ, trong đó giá trị số được điều chỉnh bằng đặc trưng cột. Nút trung tâm, ban đầu là trung bình của các nút lá, đóng vai trò tổng hợp thông tin của graphlet.

liệu tên cột. Điều này mở ra khả năng học trên nhiều bảng mà không cần đối sánh sơ đồ. Bên cạnh đó, việc CARTE sử dụng mô hình ngôn ngữ để xử lý các giá trị dạng văn bản, danh mục và tên cột, giúp giảm bớt công đoạn tiền xử lý dữ liệu như mã hóa phân loại hay loại bỏ trùng lặp. Ngoài ra, CARTE có thể xử lý các từ vựng mở rộng, cho phép nhận diện những biến thể ngôn ngữ như "North America" và "Northern America" mà không cần ánh xạ thủ công.

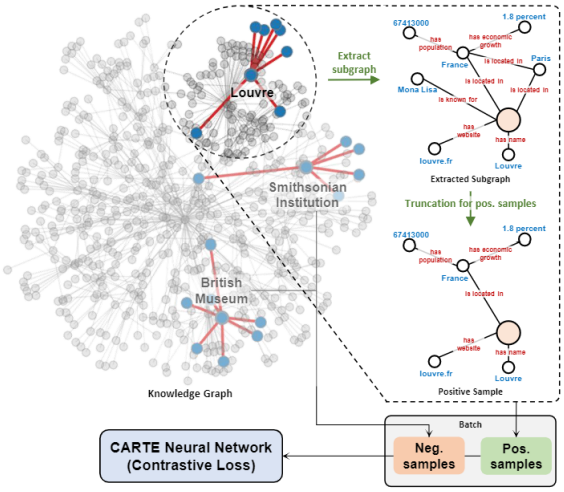
Nhờ những đặc điểm trên, cách tiếp cận đồ thị của CARTE giúp khái quát hóa dữ liệu từ các bảng không đồng nhất, đưa tất cả các bảng vào cùng một miền biểu diễn mà không cần phải thực hiện đối sánh sơ đồ hay đối sánh đối tượng. Điều này tạo điều kiện cho việc học trên nhiều bảng, mở ra cơ hội tiền huấn luyện và học chuyển giao trên dữ liệu bảng một cách hiệu quả.

3.2 Tiền Huấn Luyện Mô Hình từ Cơ Sở Tri Thức Lớn

CARTE được tiền huấn luyện trên YAGO3, một cơ sở tri thức lớn được xây dựng từ Wikidata và các nguồn khác, chứa thông tin thực tế về thế giới. YAGO lưu trữ dữ liệu dưới dạng đồ thị tri thức, bao gồm các bộ ba (head, relation, tail). Ví dụ, bộ ba ("Louvre", "is located in", "Paris") trong Hình 2 là một mẫu dữ liệu có thể tìm thấy trong YAGO. Phiên bản hiện tại của YAGO chứa hơn 18,1 triệu bộ ba với 6,3 triệu thực thể.

Trong phần này, nhóm sẽ mô tả quá trình tiền huấn luyện của mô hình CARTE, được tóm tắt trong Hình 2. Từ đồ thị tri thức, các đồ thị con (graphlets) phù hợp sẽ được trích xuất để xử lý làm đầu vào cho CARTE. Để thực hiện học tự giám sát với hàm mất mát tương phản (contrastive loss), nhóm thêm vào batch các phiên bản bị cắt ngắn của các graphlets đã chọn để đó mô hình học cách tổng hợp thông tin dựa trên ngữ cảnh được cung cấp.

Từ đồ thị tri thức lớn YAGO, nhóm xây dựng các graphlets nhỏ hơn có thể dùng làm đầu vào cho CARTE. Để tạo một graphlet phù hợp cho một thực thể, nhóm trích xuất tiểu đồ của nó trong phạm vi k quan hệ liên kết (k-hop



Hình 2: CARTE tiền huấn luyện bằng cách tạo graphlet từ một đồ thị tri thức lớn. Những graphlet này được đưa vào mạng nơ-ron để huấn luyện theo phương pháp tự giám sát, giúp mô hình học cách tổng hợp thông tin từ các bảng thông qua quan hệ giữa các cột.

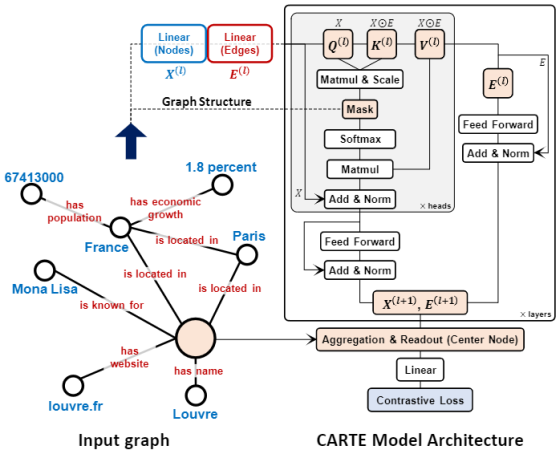
relations). Để duy trì cấu trúc được trình bày trong Hình 1, đồng thời tận dụng thêm thông tin từ nhiều bước nhảy, nhóm đặt k=2, nhưng giới hạn số lượng quan hệ 1-hop và 2-hop lần lượt là 100 và 10.

Các graphlets từ bảng dữ liệu trong Hình 1 sử dụng một token làm node trung tâm (center node) cho mỗi hàng, trong khi các graphlets từ đồ thị tri thức có thể sử dụng tên thực thể (ví dụ: "Louvre"). Để tránh sự khác biệt này, nhóm sử dụng một token làm node trung tâm, kèm theo một node bổ sung chứa tên thực thể và được kết nối bằng quan hệ "has name". Cuối cùng, tương tự như đã đề cập ở phần 3.1, nhóm khởi tạo các node và cạnh bằng cách sử dụng embedding từ mô hình ngôn ngữ FastText.

Để tạo một batch mẫu với kích thước N_b, đầu tiên nhóm chọn các thực thể từ YAGO sẽ được đưa vào batch và tạo các graphlets tương ứng. Trong đó, 90% N_b được lấy từ các thực thể có ít nhất 6 quan hệ 1-hop, 10% còn lại được lấy từ các thực thể khác. Lý do chính cho việc lấy mẫu như vậy là do phần lớn thực thể trong YAGO chỉ có một hoặc hai quan hệ 1-hop, trong khi dữ liệu dạng bảng thường có nhiều hơn (tương ứng với nhiều cột).

Để áp dụng học tự giám sát với mất mát tương phản, nhóm tạo ra các mẫu dương bằng cách rút gọn cấu trúc thông qua việc xóa một phần ngẫu nhiên (từ 30% đến 70%) số cạnh trong graphlets gốc. Hình 2 minh họa một graphlet của thực thể "Louvre" cùng với phiên bản bị cắt ngắn của nó.

Hình 3 mô tả kiến trúc của CARTE. Về cơ bản, CARTE sử dụng mô hình Transformer encoder truyền thống của và sửa đội thành một mạng đồ thị với cơ chế chú ý (graph attentional network). Một thành phần quan trọng trong kiến



Hình 3: CARTE sử dụng các đồ thị làm đầu vào, với đặc trưng là các nút và cạnh được dùng trong lớp self-attention. Các lớp này cập nhật đặc trưng nút dựa trên thông tin cạnh và sử dụng attention mask để duy trì cấu trúc đồ thị. Lớp Tổng hợp & Đọc trích xuất đặc trưng từ nút trung tâm (center node), và đầu ra cuối cùng được dùng để tính toán cho hàm mất mát đối lập (contrastive loss).

trúc của CARTE là lớp tự chú ý (self-attention) tính toán mức độ quan trọng của các node và cạnh trong đồ thị. Trong mô hình đồ thị, cơ chế chú ý điều chỉnh tầm quan trọng của các node lân cận đối với một node nhất định. Đối với dữ liệu bảng, điều này có nghĩa là xác định mức độ quan trọng của từng giá trị trong hàng, với ngữ cảnh được bổ sung bởi thông tin cột.

Chúng ta sẽ đi sâu vào cơ chế tập trung (attention) của CARTE, được thiết kế để nắm bắt ngữ cảnh và mối quan hệ giữa các thành phần. Để đảm bảo tính nhất quán trong ký hiệu, chúng ta ký hiệu vector bằng một mũi tên phía trên ($\vec{A}$), ma trận bằng chữ in đậm ($\mathbf{A}$), và các đại lượng vô hướng (scalar) bằng chữ thường (A). Để giúp việc đọc hiểu dễ dàng hơn, chúng ta chỉ xem xét một lớp tập trung đơn đầu (single-head attention), nhưng cách tiếp cận này có thể mở rộng thành nhiều đầu (multi-head) bằng cách ghép nối hoặc trung bình các kết quả tập trung.

Xem xét một đồ thị có N nút thì ta có $\vec{X}_i^{(l)} \in \mathbb{R}^d$ là đặc trưng của nút thứ i, với $\vec{X}_i j^{(l)} \in \mathbb{R}^d$ là đặc trưng của cạnh hướng từ j tới i. Trong mô hình CARTE, mỗi đồ thị con (graphlet) luôn bao gồm một nút trung tâm, được ký hiệu là $i = 1$.

Cơ chế tập trung dựa trên ba thành phần chính: truy vấn (query), khóa (key), và giá trị (value). Truy vấn đại diện cho thông tin cần quan tâm—trong trường hợp này là các nút. Vì vậy, chúng ta sử dụng cách tiếp cận phổ biến và chỉ dựa vào thông tin của nút để xây dựng truy vấn. Ngược lại, cặp khóa-giá trị cần phản ánh những thông tin mà các nút lân cận có thể cung cấp. Do đó, ta

$$\text{Query:} \quad \vec{Q}_i = \vec{X}_i^{(l)} \cdot \mathbf{W_Q} \tag{1}$$

$$\text{Key:} \quad \vec{K}_{ij} = (\vec{X}_i^{(l)} \odot \vec{E}_{ij}^{(l)}) \cdot \mathbf{W_K} \tag{2}$$

$$\text{Value:} \quad \vec{V}_{ij} = (\vec{X}_i^{(l)} \odot \vec{E}_{ij}^{(l)}) \cdot \mathbf{W_V} \tag{3}$$

Hình 4: 3 thành phần chính của cơ chế attention

$$A_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \text{ where } e_{ij} = \frac{\vec{Q}_i \cdot \vec{K}_{ij}^T}{\sqrt{d}}$$

Hình 5: Công thức Attention Score theo phương pháp scaled dot-product attention

đưa thông tin cạnh vào quá trình xây dựng khóa và giá trị. Cụ thể, ba thành phần này được xác định như Hình 4.

Các yếu tố W_Q, W_K, W_V trong Hình 4 là các trọng số để huấn luyện và dấu $\odot$ đánh dấu phép nhân từng phần tử (element-wise multiplication) được lấy cảm hứng từ một số nghiên cứu khác về phương pháp nhúng đồ thị tri thức. Dựa trên công thức tính attention score theo phương pháp scaled dot-product attention, trọng số tập trung của nút j với nút i, ký hiệu là A_ij được tính như Hình 5.

Ở đây, tổng chỉ xét trên các nút lân cận của i, điều này tương ứng với một bước "che mặt nạ" (masking) để bảo toàn cấu trúc đồ thị. Việc đưa loại quan hệ (chẳng hạn như nhãn cột trong bảng dữ liệu) vào tính toán attention giúp mô hình tái diễn giải ngữ cảnh chính xác hơn. Chẳng hạn, một thực thể như "George Bush" có thể chỉ tổng thống thứ 41 hoặc 43 của Mỹ, hoặc một tàu sân bay mang tên này. Ý nghĩa của nó phụ thuộc vào mối quan hệ đi kèm (ví dụ: "George Bush", "con của", "George Bush"). Nếu thay bằng mối quan hệ khác như "cha của", thì cách xác định thực thể sẽ thay đổi hoàn toàn.

Đầu ra của các lớp attention được biểu diễn thông qua các đặc trưng đã được biến đổi của nút và cạnh. Cụ thể, đặc trưng mới của nút i ở tầng $l+1$ được tính như Hình 6. Ở các tầng cuối cùng, mô hình sử dụng một lớp attention nhưng không cập nhật thông tin cạnh. Sau đó, một lớp tổng hợp (readout layer) sẽ trích xuất biểu diễn của nút trung tâm. Đối với giai đoạn tiền huấn luyện (pretraining), đầu ra từ mô hình sẽ được xử lý để tính toán hàm mất mát tương phản (contrastive loss).

Để thiết kế hàm mất mát tương phản không giám sát (self-supervised contrastive loss), chúng ta áp dụng khung phương pháp từ một nghiên cứu khác. Cụ thể, đồ thị con gốc (original graphlet) và một phiên bản bị cắt bớt (truncation) được xem là mẫu dương (positives), trong khi các đồ thị con khác trong cùng một lô được coi là mẫu âm (negatives). Quá trình học dựa trên độ tương đồng cosin giữa các đầu ra của mạng, sau đó được đưa vào hàm mất mát InfoNCE, một kỹ thuật phổ biến trong học biểu diễn tương phản để tối ưu hóa khoảng

$$\text{transformed entry} \qquad \vec{X}_i^{(l+1)} = \sigma_X\big(\sum_j A_{ij} \cdot \vec{V}_{ij}\big)$$

$$\text{transformed relation} \qquad \vec{E}_{ij}^{(l+1)} = \sigma_E(E_{ij}^{(l)} \cdot \mathbf{W_E})$$

Hình 6: Kết quả của lớp attention.

cách giữa các mẫu dương và âm.

3.3 Tinh Chỉnh cho Tác Vụ

Trong một tác vụ đầu ra cụ thể, quá trình tinh chỉnh CARTE chỉ sử dụng lại một phần của kiến trúc đã được huấn luyện trước (như minh họa trong Hình 3). Cụ thể, các lớp ban đầu xử lý nút và cạnh (khối màu xanh và đỏ trong Hình 3) cùng với lớp "Tổng hợp & Đọc ra" được giữ lại. Cách tiếp cận này có sự khác biệt so với nhiều phương pháp tinh chỉnh thông thường, nhưng nó xuất phát từ hành vi của mạng nơ-ron đồ thị.

Thực tế, các thực thể trong bảng đầu ra có cấu trúc đồ thị đơn giản hơn so với giai đoạn huấn luyện ban đầu. Thứ nhất, chúng thường có dạng hình sao (Hình 1). Thứ hai, các bảng đầu ra có ít sự biến đổi về cấu trúc đồ thị hơn và số lượng biến rời rạc cũng thấp hơn so với dữ liệu ban đầu từ YAGO. Nếu kiến trúc quá sâu, các đặc trưng phân biệt có thể bị làm mờ trong biểu diễn đầu ra, dẫn đến hiện tượng làm mịn quá mức. Do đó, chúng tôi áp dụng một quy ước phổ biến trong mô hình đồ thị là số lớp attention sẽ tương đương với số bậc quan hệ tối đa, trong trường hợp này là $k = 1$. Đối với phân lớp cuối cùng, các lớp tuyến tính được gắn trực tiếp vào mô hình. Khi tinh chỉnh mô hình cơ sở, chúng tôi xem xét hai phương thức suy luận khác nhau.

3.3.1 Suy luận trên dữ liệu bảng đơn

Đây là kịch bản phổ biến, trong đó một bảng duy nhất được cung cấp với biến mục tiêu cần dự đoán. Trước khi chuyển đổi các thực thể bảng thành đồ thị, chúng tôi áp dụng biến đổi lũy thừa (power transform) lên các biến giá trị số nhằm cải thiện độ ổn định của mô hình. Biến đổi này đã được chứng minh hiệu quả trong nhiều nghiên cứu trước đây và giúp quá trình tinh chỉnh CARTE ổn định hơn. Ngoài ra, chúng tôi áp dụng chiến lược bagging, hay còn gọi là boostrap aggregating, trong đó nhiều mô hình được huấn luyện song song với các tập dữ liệu khác nhau được chia nhỏ theo chiến lược. Kết quả dự đoán cuối cùng được tính bằng cách lấy trung bình đầu ra của các mô hình này.

3.3.2 Học chuyển giao từ một bảng nguồn sang bảng đích

Chúng tôi cũng sử dụng CARTE trong các bài toán học chuyển giao, nơi có một bảng nguồn X_S có thể hỗ trợ dự đoán trên bảng đích X_T. Đáng chú ý, bảng nguồn thường có số lượng mẫu huấn luyện lớn hơn so với bảng đích và việc tinh chỉnh CARTE cần được thực hiện đồng thời trên cả hai bảng. Nhờ vào biểu

diễn đồ thị, mô hình có thể tinh chỉnh đồng thời mà không cần sự tương ứng giữa các cột của hai bảng nhưng kết quả đầu ra y_S và y_T của bảng nguồn X_S và bảng đích X_T cần phải tương đồng.

Để đảm bảo điều này, chúng tôi thực hiện một bước chuyển đổi trên y_S để phù hợp với kỳ vọng toán học bậc nhất của y_T bằng cách sử dụng phép biến đổi lũy thừa như trước đó, nhưng lần này áp dụng phép biến đổi nghịch đảo. Nếu bản chất của biến mục tiêu giữa bảng nguồn và bảng đích khác nhau (phân loại hoặc hồi quy), chúng tôi điều chỉnh y_S như sau:

- Nếu y_T thuộc tác vụ phân loại, nhóm sẽ nhị phân hóa biến hồi quy trọng bảng nguồn X_S.
- Nếu y_T là thuộc tác vụ hồi quy, nhóm sẽ sử dụng biến phân loại của bảng nguồn X_S, mã hóa nó dưới dạng 0,1 và chuẩn hóa theo phương pháp chuẩn.

Quá trình tinh chỉnh CARTE tiếp tục bằng cách lấy các tập dữ liệu với tỷ lệ cố định các dòng từ bảng nguồn và bảng đích (trong đó mỗi tập có kích thước 64, với 8 dòng từ bảng đích). Nhóm sẽ áp dụng chiến lược dừng sớm trên tập xác thực của bảng đích, và dựa phương pháp bagging để tạo ra nhiều mô hình con dựa trên các tập xác thực khác nhau và tính trung bình kết quả dự đoán.

Thường thì, quá trình dừng sớm diễn ra trước khi tất cả các điểm dữ liệu của bảng nguồn được sử dụng. Điều này giúp tránh hiện tượng overfitting vào dữ liệu nguồn, vì dữ liệu nguồn có thể không quan trọng bằng dữ liệu đích đối với việc dự đoán y_T. Các siêu tham số được sử dụng vẫn giữ nguyên như trong thiết lập với một bảng đơn lẻ.

Vì nhóm đã chọn bảng nguồn khá linh hoạt từ dữ liệu có liên quan yếu, nên không phải lúc nào mô hình học theo cặp (pairwise learner) cũng cải thiện so với mô hình học trên một bảng đơn lẻ, đặc biệt khi bảng nguồn không cung cấp đủ thông tin liên quan. Để khắc phục, nhóm kết hợp mô hình học theo cặp với mô hình học đơn lẻ bằng cách tổng hợp (ensemble) dự đoán của cả hai cách thông qua hàm softmax. Trọng số trong softmax được xác định dựa trên điểm dự đoán từ tập xác thực nội bộ của các mô hình thành phần. Để điều chỉnh nhiệt độ (temperature) của softmax, trọng số sẽ được chia cho độ lệch chuẩn giữa các mô hình, giúp cân bằng ảnh hưởng của từng mô hình trong tổ hợp.

3.3.3 Học chung trên nhiều bảng

Mấu chốt của học chuyển giao, như đã đề cập, là cần phải tìm được bảng nguồn phù hợp. Nếu có nhiều bảng từ cùng một lĩnh vực, CARTE có thể tận dụng tất cả để chọn lọc thông tin hữu ích nhất cho tác vụ học chuyển giao. Trong trường hợp này, bài toán được đặt ra với một bảng đích X_T và một tập hợp các bảng nguồn $\{X_{S,1}, ..., X_{S,m}\}$.

Quy trình huấn luyện như sau:

1. Xây dựng mô hình học đơn lẻ trên X_T.

2. Xây dựng từng mô hình học cặp giữa bảng đích X_T và từng bảng nguồn $X_{S,i}$ bằng phương pháp học cặp đã mô tả ở trên.

Tuy nhiên, không phải mọi cặp bảng đều mang lại thông tin hữu ích như nhau. Do đó, để tìm tổ hợp dữ liệu tối ưu, chúng tôi sử dụng chiến lược tương tự: kết hợp (ensemble) các mô hình học cặp cùng với mô hình học đơn lẻ. Nếu tất cả các bảng nguồn đều tạo ra dự đoán tốt, chúng sẽ được kết hợp với trọng số bằng nhau. Ngược lại, nếu một bảng nguồn vượt trội hơn, kết quả dự đoán sẽ được neo vào bảng đó.

4 Nghiên Cứu Thực Nghiệm

4.1 Cài Đặt Thực Nghiệm

Chúng tôi sử dụng 51 tập dữ liệu dạng bảng, trong đó có 40 tập dành cho bài toán hồi quy và 11 tập dành cho bài toán phân loại. Các tập dữ liệu này được thu thập từ nhiều nguồn khác nhau, chủ yếu từ các nghiên cứu trước đây về học máy và các cuộc thi trên Kaggle. Chúng bao gồm nhiều chủ đề liên quan đến xã hội và kinh doanh, chẳng hạn như tai nạn, bầu cử, tiền lương, thực phẩm, nhà hàng, v.v. Chúng tôi chọn các tập dữ liệu đại diện cho các ứng dụng khoa học dữ liệu hiện đại—các bảng có cột ý nghĩa và giá trị rời rạc—khác với nhiều tập dữ liệu từ UCI. Danh sách cụ thể của các tập dữ liệu được cung cấp trong Phụ lục B.

Chúng tôi đánh giá nhiều phương pháp khác nhau, với các ký hiệu viết tắt như sau:

- CatBoost: Một thuật toán cây tăng cường độ dốc (gradient boosting) phổ biến để học trên dữ liệu dạng bảng. Các đặc trưng dạng văn bản được xử lý dưới dạng dữ liệu phân loại, mã hóa bằng phương pháp mã hóa phân loại của CatBoost—một phiên bản cải tiến của phương pháp mã hóa mục tiêu.

- TabVec: Sử dụng bộ TableVectorizer từ thư viện Skrub để chuyển đổi bảng có dữ liệu dạng chuỗi thành mảng số. Các cột có ít giá trị phân loại được mã hóa one-hot, trong khi các cột có nhiều giá trị phân loại được mã hóa bằng bộ mã hóa Gamma-Poisson, giúp trích xuất danh mục tiềm ẩn từ các chuỗi con. Với các mô hình không dựa trên cây, các giá trị thiếu được thay thế bằng trung bình đối với đặc trưng số và được coi là một danh mục riêng đối với đặc trưng phân loại. Với mô hình mạng nơ-ron, dữ liệu được chuẩn hóa về khoảng [0, 1] bằng phương pháp min-max.

- XGB, HGB, RF: Các mô hình cây quyết định, bao gồm XGBoost, HistGradientBoosting và Random Forest (từ thư viện scikit-learn).

- MLP và ResNet: Mạng nơ-ron đa tầng (MLP) và phiên bản mở rộng của nó với các lớp chuẩn hóa (layer norm/batch norm) và kết nối tắt (skip-connections).

- Ridge và Logistic: Các mô hình tuyến tính, bao gồm hồi quy Ridge và hồi quy Logistic, được áp dụng cho các bài toán hồi quy và phân loại.
- S-LLM: Dựa trên ý tưởng từ TabLLM, chúng tôi thử nghiệm cách mã hóa từng hàng dữ liệu bằng một mô hình ngôn ngữ lớn (LLM). Cụ thể, mỗi hàng được biểu diễn dưới dạng một câu và mã hóa bằng mô hình intfloat/e5-small-v2 từ HuggingFace. Khác với TabLLM, đầu ra được chuyển vào mô hình XGB để học cả bài toán hồi quy và phân loại. Với dữ liệu số, chúng tôi thử nghiệm hai cách xử lý: hoặc nối trực tiếp vào đặc trưng đầu vào (CN), hoặc chuyển đổi thành chuỗi trước khi đưa vào LLM (EN).
- TabPFN: Một mô hình transformer được huấn luyện trước trên dữ liệu tổng hợp, giúp dự đoán cho các tập dữ liệu nhỏ chỉ với một lần suy luận. Các đặc trưng văn bản được xử lý dưới dạng phân loại và mã hóa bằng phương pháp mã hóa mục tiêu.

4.2 Kết Quả trên Bảng Đơn

4.3 Học Đa Bảng

5 Bàn Luận và Kết Luận