BỘ GIÁO DỤC VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Vũ Thị Kim Thư

TÊN ĐỀ TÀI LUẬN VĂN

ỨNG DỤNG MÔ HÌNH HỌC SÂU TRANSFORMER CHO BÀI TOÁN KHUYẾN NGHỊ TRÍCH DẪN BÀI BÁO KHOA HỌC

LUẬN VĂN THẠC SĨ MÁY TÍNH

Hà Nội - 2026

BỘ GIÁO DỤC VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Vũ Thị Kim Thư

TÊN ĐỀ TÀI LUẬN VĂN

ỨNG DỤNG MÔ HÌNH HỌC SÂU TRANSFORMER CHO BÀI TOÁN KHUYẾN NGHỊ TRÍCH DẪN BÀI BÁO KHOA HỌC

LUẬN VĂN THẠC SĨ MÁY TÍNH

Ngành: ***Hệ thống thông tin***

Mã số: 8480104

NGƯỜI HƯỚNG DẪN KHOA HỌC:

PGS.TS. Nguyễn Long Giang

Hà Nội - 2026

LỜI CAM ĐOAN

Tôi Vũ Thị Kim Thư xin cam đoan rằng đề tài nghiên cứu trong luận văn mang tên gọi **"Ứng dụng mô hình học sâu Transformer cho bài toán khuyến nghị trích dẫn khoa học"** là thành quả nghiên cứu do chính tôi thực hiện dưới sự hướng dẫn khoa học tận tâm của PGS.TS. Nguyễn Long Giang. Toàn bộ nội dung được triển khai từ quá trình tìm hiểu, tổng hợp và phân tích tài liệu do chính tôi tiến hành. Mọi số liệu, kết quả và nhận định trình bày trong luận văn đều phản ánh trung thực quá trình nghiên cứu và chưa xuất hiện trong bất kỳ công trình nào được công bố trước đây. Tôi hoàn toàn chịu trách nhiệm pháp lý về nội dung của tài liệu này.

Hà Nội, ngày 06 tháng 07 năm 2026

Tác giả luận văn

Vũ Thị Kim Thư

LỜI CẢM ƠN

Trước hết, em xin dành lời tri ân chân thành nhất tới Thầy PGS.TS. Nguyễn Long Giang hướng dẫn khoa học – người đã tận tình chỉ bảo, định hướng và hỗ trợ tôi trong suốt quá trình thực hiện và hoàn thiện luận văn. Những định hướng kịp thời, sự phản biện sắc bén và niềm tin không dao động của thầy chính là chỗ dựa tinh thần giúp em hoàn thiện luận văn này.

Em xin chân thành cảm ơn các Thầy, các chuyên gia và những cá nhân đã hỗ trợ, giúp đỡ tôi về chuyên môn, tài liệu, phương pháp nghiên cứu và kinh nghiệm thực tiễn để luận văn có thể được triển khai và hoàn thành đúng tiến độ.

Xin trân trọng cảm ơn Ban Lãnh đạo Học viện Khoa học và Công nghệ, đặc biệt Phòng Đào tạo cùng các phòng chức năng của Học viện đã tạo điều kiện thuận lợi về cơ sở vật chất, thủ tục hành chính và môi trường học thuật để em có thể yên tâm học tập và nghiên cứu.

Lời cảm ơn cuối cùng, em xin gửi lời cảm ơn chân thành tới gia đình, bạn bè và đồng nghiệp – những người luôn bên cạnh động viên, chia sẻ và hỗ trợ em trong suốt quá trình học tập và thực hiện luận văn.

Hà Nội, ngày 06 tháng 07 năm 2026

Tác giả luận văn

Vũ Thị Kim Thư

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Tiếng Anh đầy đủ	Nghĩa tiếng Việt
BERT	Bidirectional Encoder Representations from Transformers	Biểu diễn mã hóa hai chiều từ Transformer
BiLSTM	Bidirectional Long Short-Term Memory	Bộ nhớ dài-ngắn hạn hai chiều
CB	Content-Based Filtering	Lọc dựa trên nội dung
CF	Collaborative Filtering	Lọc cộng tác
CNN	Convolutional Neural Networks	Mạng nơ-ron tích chập
DualLCR	Dual Local Citation Recommendation	Khuyến nghị trích dẫn cục bộ kép
FNN	Feedforward Neural Networks	Mạng nơ-ron truyền thẳng
GCN	Graph Convolutional Networks	Mạng tích chập đồ thị
GNN	Graph Neural Networks	Mạng nơ-ron đồ thị
GraphSAGE	Graph Sampling and Aggregation	Lấy mẫu và tổng hợp đồ thị
GRU	Gated Recurrent Units	Đơn vị hồi quy có cổng
LSTM	Long Short-Term Memory	Bộ nhớ dài-ngắn hạn
MAP	Mean Average Precision	Độ chính xác trung bình
MRR	Mean Reciprocal Rank	Xếp hạng đối ứng trung bình
NCN	Neural Citation Networks	Mạng nơ-ron trích dẫn
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
RHN	Recurrent Highway Networks	Mạng hồi quy xa lộ
RNN	Recurrent Neural Networks	Mạng nơ-ron hồi quy
SciBERT	Scientific BERT	BERT chuyên biệt cho văn bản khoa học
VGAE	Variational Graph Auto-Encoders	Bộ mã hóa tự động đồ thị biến đổi

DANH MỤC CÁC BẢNG, HÌNH

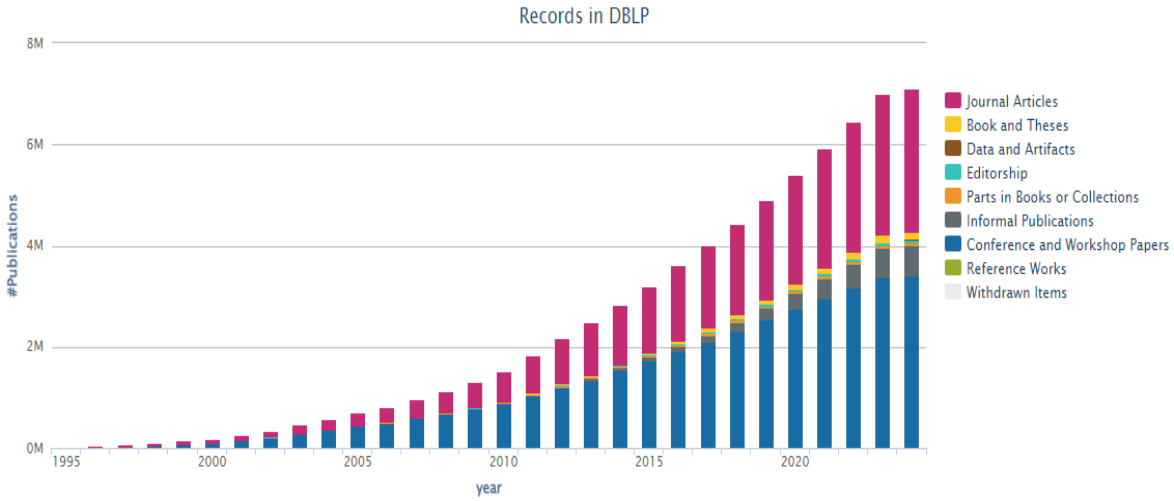
Danh mục hình

Danh mục bảng

MỞ ĐẦU

Hai mươi năm qua, cộng đồng khoa học toàn cầu đang đối mặt với một nghịch lý thú vị: càng có nhiều tri thức được sản sinh, nhà nghiên cứu lại càng khó định hướng trong kho tài liệu khổng lồ đó. Khởi điểm từ năm 1673 khi Hiệp hội Khoa học Hoàng gia Anh ra mắt tạp chí học thuật đầu tiên, nhân loại đã chứng kiến sản lượng xuất bản khoa học tăng trưởng không ngừng — và tốc độ tăng trưởng ngày càng được đẩy nhanh hơn nhờ sự phổ biến của các nền tảng chia sẻ trực tuyến.

Nhìn vào dữ liệu cụ thể, từ đầu thập niên 2000 đến giữa thập niên 2010, tốc độ tăng trưởng bình quân hàng năm của ấn phẩm khoa học và kỹ thuật dao động quanh mức 10%, với tổng số vượt ngưỡng 4 triệu bài vào năm 2018. Riêng với lĩnh vực khoa học máy tính, con số bài báo mới công bố mỗi năm đã tăng gấp ba lần so với đầu thế kỷ XXI. Cơ sở dữ liệu y sinh PubMed — một trong những nguồn tài liệu được tra cứu nhiều nhất thế giới — ghi nhận sự tăng trưởng hàng chục lần chỉ trong vài thập kỷ. Xu hướng này diễn ra tương tự trên hầu khắp các ngành học thuật.



Hình 1.1. Xu hướng tăng trưởng ấn phẩm khoa học được lập chỉ mục trong DBLP (1995 - 2024)

Sự bùng nổ này đặt ra một thách thức cụ thể: nghiên cứu năm 2014 cho thấy nhà khoa học tại các đại học Mỹ và Úc chỉ đọc được trung bình 22 bài báo mỗi tháng — con số không thay đổi so với một thập kỷ trước, trong khi tổng số bài báo cần theo dõi đã tăng đáng kể. Điều này cho thấy khả năng xử lý thông tin của con người đã đạt ngưỡng bão hòa, trong khi lượng tri thức cần tiếp thu cứ liên tục tăng lên.

Trong lĩnh vực viết bài học thuật, trích dẫn tài liệu tham khảo đóng vai trò thiết yếu: đó là cách ghi nhận nguồn gốc ý tưởng, xây dựng nền tảng lý luận cho lập luận mới, đồng thời thể hiện đạo đức khoa học. Song chính vì danh mục tài liệu quá phong phú, việc xác định đúng những công trình cần trích dẫn trong một đoạn văn cụ thể ngày càng tốn nhiều công sức và dễ bỏ sót.

Trước thực trạng đó, các hệ thống gợi ý trích dẫn tự động nổi lên như một hướng giải pháp hứa hẹn. Ý tưởng cốt lõi: thay vì để nhà nghiên cứu tự dò tìm, hệ thống sẽ phân tích ngữ cảnh đang viết và chủ động đề xuất danh sách tài liệu có thể liên quan. Bài toán này lần đầu được McNee và cộng sự hệ thống hóa vào năm 2002, và từ đó trở thành một trong những hướng nghiên cứu được cộng đồng CNTT và Thông tin học quan tâm ngày càng nhiều.

Đặc biệt từ năm 2017, kiến trúc Transformer do Vaswani và cộng sự giới thiệu đã mang lại bước đột phá căn bản cho toàn ngành xử lý ngôn ngữ tự nhiên (NLP). Cơ chế tự chú ý (self-attention) — ưu điểm cốt lõi của Transformer — cho phép mô hình hóa liên hệ giữa mọi cặp từ trong câu mà không bị ràng buộc bởi khoảng cách tuần tự. Từ nền tảng đó, các mô hình tiền huấn luyện như BERT, RoBERTa lần lượt phá vỡ nhiều kỷ lục hiệu năng. Đặc biệt, SciBERT — biến thể được Beltagy và cộng sự (2019) huấn luyện riêng trên ngữ liệu khoa học — cho thấy lợi thế rõ ràng trong các tác vụ liên quan đến văn bản học thuật chuyên ngành.

Song nội dung văn bản chỉ là một chiều thông tin. Trong thực tế, quyết định trích dẫn của một nhà nghiên cứu còn chịu ảnh hưởng của nhiều yếu tố khác: danh tiếng tác giả, uy tín nơi xuất bản, tính mới của công trình, và đặc biệt là vị trí của bài báo trong mạng lưới trích dẫn học thuật. GraphSAGE — mô hình học biểu diễn đồ thị quy nạp (Hamilton và cộng sự, 2017) — mở ra khả năng khai thác chính xác chiều thông tin cấu trúc này, với ưu điểm vượt trội là có thể tổng quát hóa sang bài báo mới mà không cần huấn luyện lại.

1. Mục tiêu luận văn

Khảo sát tổng thể cho thấy các mô hình khuyến nghị trích dẫn hiện tại tồn tại ba khoảng trống chính. Về phía biểu diễn ngôn ngữ: nhiều mô hình vẫn sử dụng kiến trúc BERT tổng quát hoặc BiLSTM từ trước 2019, chưa tận dụng SciBERT được tối ưu hóa cho văn bản học thuật. Về phía biểu diễn đồ thị: GCN transductive phổ biến trong các mô hình kết hợp gặp khó khăn khi bài báo mới

liên tục được bổ sung. Về khai thác siêu dữ liệu: phần lớn hệ thống hiện tại bỏ qua thông tin quan trọng như tác giả, địa điểm, năm công bố — những yếu tố thực tế ảnh hưởng đáng kể đến quyết định trích dẫn.

Xuất phát từ ba khoảng trống đó, luận văn hướng đến ba mục tiêu cụ thể:

- Xây dựng bức tranh tổng quan có hệ thống về các phương pháp học sâu hiện có cho bài toán khuyến nghị trích dẫn — phân tích từ lọc cộng tác, lọc nội dung, lọc đồ thị đến mô hình kết hợp — nhằm làm rõ ưu điểm, hạn chế và xu hướng phát triển của từng hướng.
- Tìm hiểu và phân tích sâu mô hình SciBERT-GraphSAGE — hướng tiếp cận kết hợp SciBERT và GraphSAGE do Đinh Ngọc Thi và cộng sự (2023) [22] công bố — bao gồm kiến trúc, cơ chế hoạt động, chiến lược huấn luyện và nguyên lý tích hợp hai nhánh.
- Phân tích và diễn giải kết quả thực nghiệm trên các bộ dữ liệu chuẩn, so sánh với các mô hình tiên tiến, từ đó đánh giá khách quan về hiệu quả, hạn chế còn tồn tại và tiềm năng ứng dụng thực tiễn.

2. Cấu trúc luận văn

Ngoài phần Mở đầu và Kết luận, luận văn được tổ chức thành ba chương:

- Chương 1 — Tổng quan lý thuyết và cơ sở khoa học của đề tài: Trình bày định nghĩa và các khái niệm cốt lõi của bài toán khuyến nghị trích dẫn; tổng quan bốn nhóm tiếp cận chính kèm phân tích mô hình tiêu biểu; phân tích hạn chế còn tồn tại; và giới thiệu nền tảng lý thuyết về nhúng văn bản, RNN/LSTM/BiLSTM/RHN, Transformer/SciBERT, GraphSAGE cùng các chỉ số đánh giá.
- Chương 2 — Ứng dụng mô hình học sâu Transformer tiên tiến theo tiếp cận kết hợp cho bài toán khuyến nghị trích dẫn bài báo khoa học: Mô tả chi tiết kiến trúc hai nhánh, nguyên lý từng thành phần, cơ chế kết hợp và hàm mục tiêu; thảo luận về thách thức kỹ thuật và ưu-nhược điểm của hướng tiếp cận.
- Chương 3 — Xây dựng chương trình thử nghiệm khuyến nghị trích dẫn bài báo khoa học sử dụng các mô hình đề xuất: Mô tả môi trường thực nghiệm, ba bộ dữ liệu chuẩn, phương pháp đánh giá, kết quả định lượng và thảo luận về ý nghĩa, hạn chế còn lại.

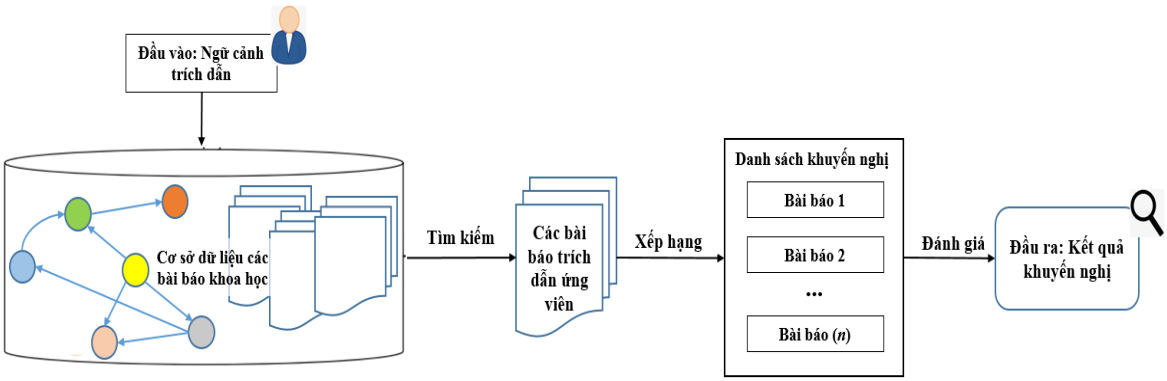
CHƯƠNG 1
TỔNG QUAN LÝ THUYẾT VÀ CƠ SỞ KHOA HỌC CỦA ĐỀ TÀI

Chương 1 trình bày lý thuyết và cơ sở khoa học của đề tài. Nội dung gồm: giới thiệu bài toán khuyến nghị trích dẫn và các khái niệm cơ bản; tổng quan toàn diện các hướng nghiên cứu hiện có (lọc cộng tác, lọc nội dung, lọc đồ thị và mô hình kết hợp); phân tích hạn chế còn tồn tại và hướng giải quyết; và các kiến thức nền tảng cần thiết về phép nhúng văn bản, họ mô hình nơ-ron hồi quy, kiến trúc Transformer, SciBERT và GraphSAGE.

1.1. BÀI TOÁN KHUYẾN NGHỊ TRÍCH DẪN KHOA HỌC

Khuyến nghị trích dẫn (Citation Recommendation) là bài toán tự động xác định, từ một tập hợp các bài báo khoa học, một hoặc nhiều bài báo có thể hoặc được trích dẫn trong một bài báo khác hoặc trong một bản thảo chưa được xuất bản. Bài toán này được McNee và cộng sự chính thức giới thiệu từ năm 2002 và kể từ đó đã thu hút ngày càng nhiều sự quan tâm của cộng đồng nghiên cứu, đặc biệt là khi các phương pháp học sâu mang lại những bước tiến đột phá.

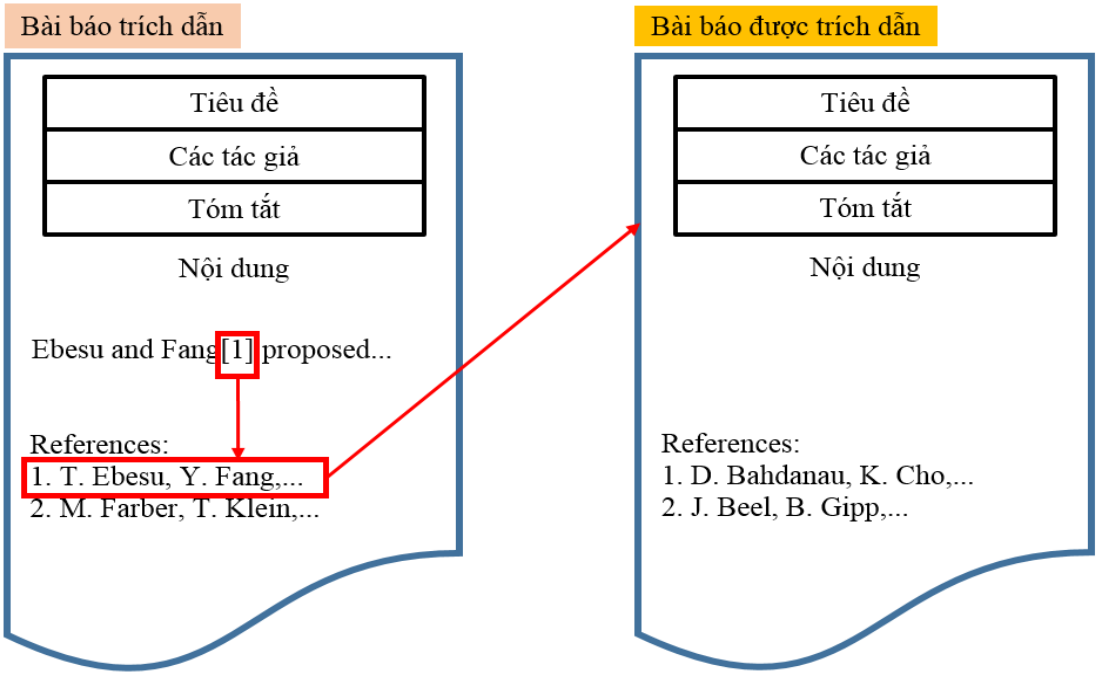
Về hình thức, mô hình khuyến nghị trích dẫn nhận đầu vào là truy vấn q — có thể là ngữ cảnh trích dẫn cụ thể (tại vị trí trích dẫn) hoặc toàn bộ nội dung bài báo — cùng cơ sở dữ liệu $D = \{d_1, d_2, ..., d_n\}$ các bài báo đã xuất bản. Hệ thống tính điểm tương thích $R(q, d_i)$ cho mỗi cặp truy vấn — bài báo ứng viên, sau đó xây dựng danh sách khuyến nghị $C = \{c_1, c_2, ..., c_k\}$ được xếp hạng từ cao đến thấp theo điểm số.



Hình 1.2. Sơ đồ tổng quát luồng xử lý hệ thống khuyến nghị trích dẫn

Các khái niệm cơ bản trong bài toán bao gồm: Trích dẫn (citation) là mối liên kết giữa một tài liệu trích dẫn và một tài liệu được trích dẫn tại vị trí cụ thể, được biểu thị bằng ký hiệu như [1], [2]. Ngữ cảnh trích dẫn (citation context) là phần văn bản xung quanh vị trí trích dẫn, thường bao gồm câu trước và câu

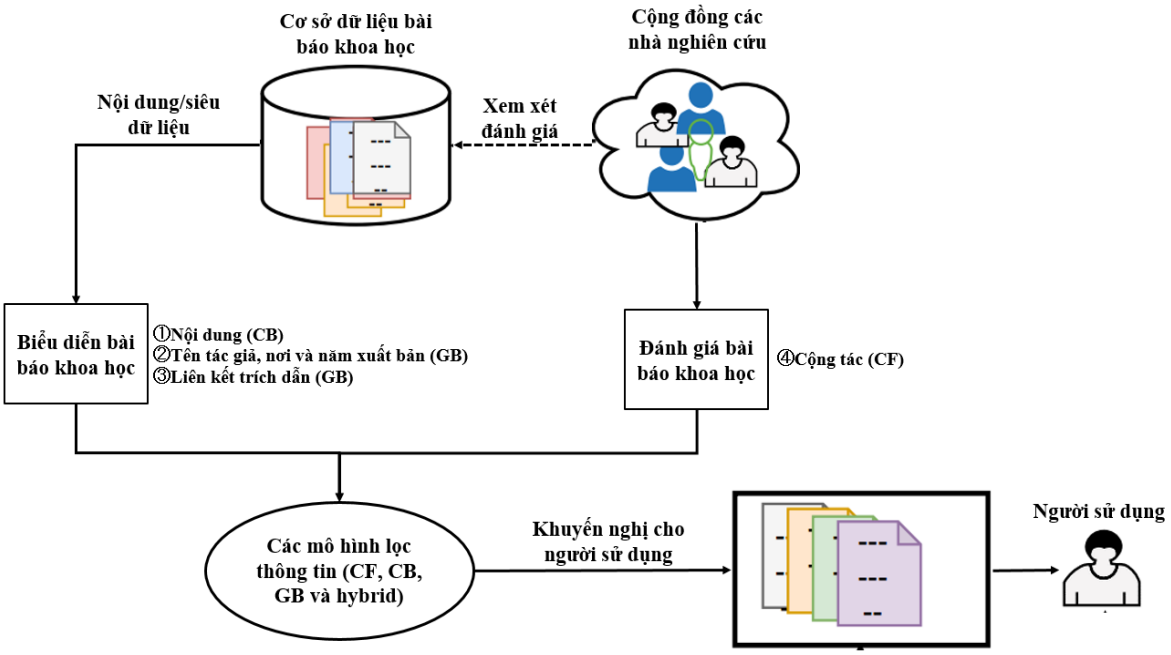
sau (±200 ký tự), cung cấp thông tin về lý do tại sao tài liệu được trích dẫn. Bài báo ứng viên (citation candidates) là tập tài liệu có thể được khuyến nghị, có thể lên đến hàng trăm nghìn hoặc hàng triệu bài. Siêu dữ liệu bài báo (paper metadata) bao gồm tên tác giả, năm công bố, địa điểm xuất bản, từ khóa và liên kết trích dẫn.



Hình 1.3. Ví dụ minh họa ngữ cảnh trích dẫn trong văn bản học thuật

Có hai dạng khuyến nghị trích dẫn chính. Khuyến nghị trích dẫn cục bộ (local citation recommendation) sử dụng một đoạn nhỏ — thường là câu chứa vị trí trích dẫn và ngữ cảnh xung quanh (±200 ký tự) — làm truy vấn. Là loại khuyến nghị nhận biết ngữ cảnh, có tính ứng dụng cao trong việc hỗ trợ viết bài trực tiếp. Khuyến nghị trích dẫn toàn cục (global citation recommendation) dựa trên toàn bộ nội dung bài báo, thích hợp hơn để tìm kiếm tài liệu liên quan ở mức tổng quan. Trong thực tiễn, dạng khuyến nghị cục bộ được quan tâm nhiều hơn và là trọng tâm của luận văn này.

Bài toán đặt ra nhiều thách thức kỹ thuật: không gian ứng viên có thể lên tới hàng trăm nghìn bài báo, đòi hỏi hiệu quả tính toán cao; ngôn ngữ học thuật chuyên ngành phức tạp với nhiều thuật ngữ đặc thù; dữ liệu huấn luyện mất cân bằng nghiêm trọng giữa trích dẫn đúng và không đúng; và bài báo mới công bố chưa có lịch sử trích dẫn tạo ra vấn đề cold-start. Ngoài ra, việc đánh giá chất lượng khuyến nghị cũng phức tạp vì "đúng" trong khuyến nghị trích dẫn có tính chủ quan và đa trị.



Hình 1.4. Sơ đồ phân loại bốn nhóm mô hình theo nguồn thông tin sử dụng

1.2. TỔNG QUAN CÁC VẤN ĐỀ NGHIÊN CỨU LIÊN QUAN

Dựa trên khảo sát toàn diện của Ali và cộng sự (2020), đến năm 2020 đã có khoảng 67 mô hình học sâu cho bài toán khuyến nghị trích dẫn được công bố trong các hội nghị và tạp chí khoa học quốc tế uy tín. Các mô hình này có thể phân loại thành bốn nhóm chính dựa trên loại thông tin được khai thác: lọc cộng tác (Collaborative Filtering — CF), lọc nội dung (Content-Based — CB), lọc dựa vào đồ thị (Graph-Based — GB) và mô hình kết hợp (Hybrid). Mỗi nhóm đã có ưu điểm và hạn chế riêng, phản ánh sự đánh đổi giữa chất lượng khuyến nghị, khả năng mở rộng và yêu cầu dữ liệu.

Bảng 1.1. So sánh bốn nhóm tiếp cận khuyến nghị trích dẫn chính

Phương pháp	Nguồn thông tin sử dụng	Ưu điểm nổi bật	Hạn chế chính
Lọc cộng tác (CF)	Hành vi người dùng, lịch sử trích dẫn, đánh giá cộng đồng	Không cần hiểu nội dung; khai thác trí tuệ tập thể	Cold-start nghiêm trọng; dữ liệu học thuật thưa thớt
Lọc nội dung (CB)	Nội dung văn bản, tóm tắt, từ khóa của bài báo	Cá nhân hóa tốt theo nội dung; không cần lịch sử người dùng	Bỏ qua cấu trúc mạng và siêu dữ liệu; phụ thuộc chất lượng biểu diễn văn bản

Phương pháp	Nguồn thông tin sử dụng	Ưu điểm nổi bật	Hạn chế chính
Lọc đồ thị (GB)	Mạng trích dẫn, quan hệ tác giả-bài báo	Khai thác quan hệ học thuật sâu; ảnh hưởng cộng đồng	Cold-start với bài báo mới; khó mở rộng quy mô; bỏ qua nội dung văn bản
Kết hợp (Hybrid)	Kết hợp nhiều nguồn: văn bản + đồ thị + siêu dữ liệu	Tăng độ chính xác toàn diện; bổ sung hạn chế lẫn nhau	Phức tạp trong tích hợp và tối ưu; chi phí tính toán cao

1.2.1. Mô hình lọc cộng tác

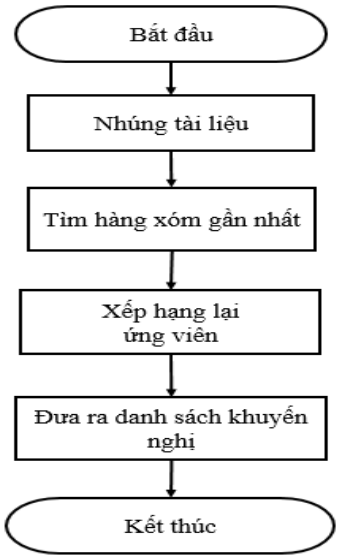
Lọc cộng tác (Collaborative Filtering — CF) khai thác đánh giá và hành vi của cộng đồng học thuật để dự đoán tài liệu trích dẫn. Trong mô hình CF điển hình có danh sách m người dùng $U = \{u_1, u_2, ..., u_m\}$ và n bài báo $P = \{p_1, p_2, ..., p_n\}$. Ma trận xếp hạng tạo ra để biểu diễn sở thích và mô hình sử dụng kỹ thuật "hàng xóm gần nhất" để đề xuất bài báo.

Liu và cộng sự phát triển mô hình CF trên trích dẫn tương đồng — hai bài báo được coi là liên quan cùng trích dẫn một bài báo chung (co-citation) hoặc cùng được trích dẫn bởi một bài báo (bibliographic coupling). Bansal và cộng sự đã đề xuất mô hình sử dụng mạng nơ-ron hồi tiếp sâu GRU với học đa tác vụ — đồng thời huấn luyện cho bài toán khuyến nghị nội dung và dự đoán siêu dữ liệu bài báo — đặc biệt cải thiện hiệu quả trong tình huống cold-start. Sugiyama và Kan xác định bài báo trích dẫn tiềm năng bằng cách khai thác ma trận trích dẫn và sử dụng hệ số tương quan Pearson để tính toán độ tương tự giữa các bài báo.

Jeon và Jung xây dựng mạng lưới trích dẫn tích hợp thông tin ngữ cảnh không gian của các trích dẫn trong bài viết — phân theo vùng xuất hiện như Mở đầu, Phương pháp, Kết quả, Thảo luận hay Kết luận. Họ áp dụng phương pháp CRITIC kết hợp với PageRank có trọng số để đánh giá tầm quan trọng theo từng phần bài báo.

Hạn chế của lọc cộng tác trong bài toán khuyến nghị trích dẫn rất rõ ràng. Thông tin bài báo của cộng đồng nghiên cứu khan hiếm hơn nhiều so với các

lĩnh vực khác như phim hay nhà hàng. Bài báo mới chưa có dữ liệu tương tác sẽ không được khuyến nghị — đây là vấn đề cold-start nghiêm trọng trong bối cảnh học thuật. Cơ sở dữ liệu học thuật thường phân tán và thưa thớt. Vì thế, rất ít nghiên cứu thành công theo hướng CF thuần túy và hầu hết các nghiên cứu hiện đại đều kết hợp CF với các phương pháp khác.



Hình 1.5. Quy trình xử lý của mô hình lọc nội dung

1.2.2. Mô hình lọc nội dung

Mô hình lọc nội dung (Content-Based — CB) phân tích nội dung văn bản của bài báo nhằm tìm các bài có nội dung tương tự. Quy trình chính gồm: (1) nhúng tài liệu — chuyển đổi văn bản thành vector biểu diễn số; (2) tìm hàng xóm gần nhất trong không gian vector dựa trên độ tương tự; (3) xếp hạng lại ứng viên bằng mô hình phân biệt đối xử; (4) khuyến nghị các bài xếp hạng cao nhất.

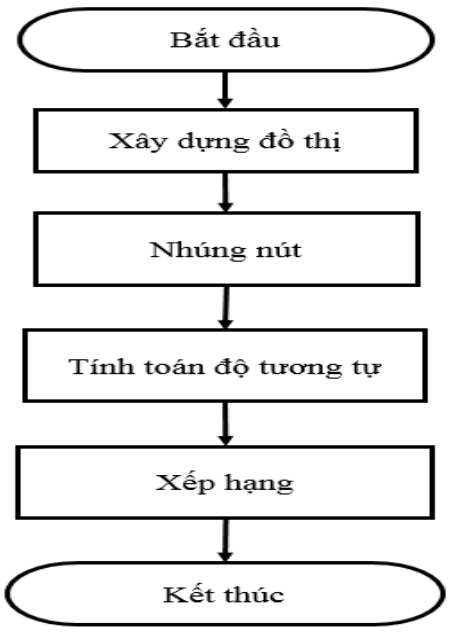
Bhagavatula và cộng sự đề xuất nhúng tài liệu truy vấn vào không gian vector để tìm hàng xóm gần nhất làm ứng viên trích dẫn, rồi xếp hạng lại bằng mô hình phân biệt. Kết quả tốt mà không cần siêu dữ liệu là điểm mạnh của hướng này. Ebesu và Fang đề xuất mô hình NCN (Neural Citation Networks) sử dụng TDNN (Time-Delay Neural Networks) và RNN để xây dựng bộ mã hóa-giải mã trích xuất thông tin từ ngữ cảnh trích dẫn và bài báo ứng viên, sử dụng BM25 để sắp xếp lại danh sách ứng viên. Färber và cộng sự cải tiến NCN năm 2020, có đề xuất phương pháp tiếp cận di truyền kết hợp nhúng và truy xuất thông tin, thu được kết quả tốt hơn trên bộ dữ liệu RefSeer, arXiv CS, nhưng vẫn chưa sử dụng đủ thông tin bài báo và phương pháp nhúng hiện đại nhất.

Gu và cộng sự đề xuất mô hình HAtten (Hierarchical Attention) sử dụng bộ mã hóa văn bản có cấu trúc chú ý phân cấp để nắm bắt thông tin ngữ cảnh đa cấp từ bài báo trích dẫn. Hệ thống hai giai đoạn: giai đoạn tìm nạp dùng Siamese encoder để truy xuất K bài báo ứng viên gần nhất; giai đoạn xếp hạng lại dùng SciBERT để tính điểm chính xác. Zhang và Ma đề xuất MP-BERT4REC xử lý tình huống một ngữ cảnh cần nhiều trích dẫn phù hợp, tối ưu bằng hàm mất mát multi-positive triplet loss với nhiều positive samples trong cùng batch. Cohan và cộng sự giới thiệu SPECTER tích hợp thông tin mạng trích dẫn vào quá trình tiền huấn luyện tài liệu, tạo embedding phản ánh cả ngữ nghĩa và quan hệ học thuật.

Hạn chế cơ bản của lọc nội dung là không xem xét đặc điểm riêng của người dùng và thay đổi sở thích theo thời gian; hiệu quả phụ thuộc lớn vào chất lượng biểu diễn văn bản; bỏ qua thông tin siêu dữ liệu và quan hệ trích dẫn giữa các bài báo — những thứ thực tế có ảnh hưởng lớn đến quyết định trích dẫn.

1.2.3. Mô hình lọc dựa vào đồ thị

Lọc dựa vào đồ thị (Graph-Based — GB) cấu trúc mạng liên kết trích dẫn để khuyến nghị. Quy trình: (1) xây dựng đồ thị — nút là bài báo, cạnh là liên kết trích dẫn; (2) nhúng nút bằng GNN (Graph Neural Networks); (3) tính toán độ tương tự; (4) xếp hạng bài báo.



Hình 1.6. Quy trình xử lý của mô hình lọc dựa vào đồ thị trích dẫn

Các mô hình khai thác mạng thông tin không đồng nhất đã được nghiên cứu và phát triển. Chẳng hạn, Chen và các cộng sự xây dựng các mạng như bài báo–bài báo, bài báo–tác giả và bài báo–thuật ngữ nhằm biểu diễn các mối quan hệ đa dạng trong lĩnh vực học thuật. Yang và cộng sự tập trung huấn luyện mô hình trên mạng không đồng nhất bao gồm các loại nút như bài báo, tác giả và địa điểm công bố. Trong khi đó, Cai và cộng sự đề xuất cách biểu diễn mạng thư mục học thuật để lưu trữ thông tin về tác giả, tóm tắt và nơi xuất bản, từ đó tính toán độ tương đồng để tạo danh sách gợi ý phù hợp. Bên cạnh đó, Pornprasit và cộng sự phát triển thuật toán nhúng mạng trích dẫn ConvCN nhằm mô hình hóa mối quan hệ trích dẫn giữa các bài báo. Cuối cùng, Guo và cộng sự đưa ra mô hình CSCR (Content-Sensitive for Citation Recommendation), bổ sung các liên kết giữa những bài báo có nội dung tương đồng, qua đó giúp giảm thiểu tình trạng dữ liệu thưa thớt.

Ali và các cộng sự đề xuất mô hình GCR-GAN (Global Citation Recommendation sử dụng Generative Adversarial Network) nhằm xây dựng hệ thống gợi ý trích dẫn mang tính cá nhân hóa trên mạng thông tin học thuật không đồng nhất (Heterogeneous Bibliographic Network – HBN). Phương pháp này tận dụng các quan hệ ngữ nghĩa giữa các thực thể trong HBN bằng cách áp dụng SPECTER để biểu diễn nội dung học thuật. Đồng thời, mô hình sử dụng mạng tự mã hóa khử nhiễu (Denoising Auto-encoder) để học các đặc trưng biểu diễn của đồ thị, qua đó nâng cao hiệu quả trong việc đề xuất trích dẫn.

Hạn chế của lọc đồ thị khá rõ ràng. Vấn đề mở rộng (scalability) là trở ngại lớn khi xử lý các mạng trích dẫn hàng triệu nút và cạnh — GCN truyền thống cần toàn bộ đồ thị trong bộ nhớ. Hiện tượng dữ liệu thưa (sparsity) và vấn đề cold-start với bài báo mới công bố làm suy giảm hiệu năng mô hình. Cuối cùng, nội dung văn bản — yếu tố quan trọng để đánh giá sự liên quan về chủ đề — thường bị bỏ qua trong các mô hình đồ thị thuần túy.

1.2.4. Mô hình kết hợp

Nhận thức được hạn chế của từng phương pháp đơn lẻ, từ năm 2020 nhiều công bố đề xuất kết hợp các phương pháp để cải thiện hiệu quả. Đây là hướng tiếp cận mà luận văn tập trung khảo sát và tìm hiểu.

Jeong và các cộng sự giới thiệu mô hình BERT-GCN, trong đó BERT được sử dụng để nắm bắt ngữ cảnh và nội dung văn bản, còn GCN đảm nhiệm việc

biểu diễn các thông tin siêu dữ liệu của bài báo như tác giả, năm công bố và địa điểm xuất bản. Đây được xem là một trong những hướng tiếp cận tiên phong cho thấy hiệu quả rõ rệt khi kết hợp giữa dữ liệu văn bản và cấu trúc đồ thị.

Trong một hướng nghiên cứu khác, Medić và Šnajder đề xuất mô hình DualLCR với kiến trúc gồm hai thành phần tách biệt. Thành phần thứ nhất là mô-đun ngữ nghĩa, áp dụng BiLSTM để xử lý nội dung bài báo, trong khi thành phần thứ hai là mô-đun thông tin học thuật, tập trung khai thác các đặc trưng như tên tác giả, số lượng trích dẫn theo năm và tổng số trích dẫn tích lũy.

Wang và cộng sự phát triển DeepCite kết hợp BERT, CNN đa tỉ lệ và BiLSTM: BERT trích xuất đặc trưng ngữ nghĩa cấp cao; CNN khai thác đặc trưng cục bộ theo nhiều kích thước cửa sổ; BiLSTM nắm bắt phụ thuộc dài hạn trong chuỗi văn bản. Wang và các cộng sự đề xuất một mô hình dựa trên kiến trúc mạng bộ nhớ đầu cuối, trong đó biểu diễn của bài báo và ngữ cảnh trích dẫn được học thông qua BiLSTM. Đồng thời, mô hình còn kết hợp thêm thông tin về tác giả và các mối quan hệ trích dẫn vào không gian vector phân tán nhằm nâng cao khả năng biểu diễn. Bên cạnh đó, Cai và cộng sự giới thiệu mô hình GLNNR (Global-Local Neighborhoods based Network Representation), kết hợp cả cấu trúc mạng (các liên kết trích dẫn) và dữ liệu phi cấu trúc như tên tác giả, địa điểm và năm xuất bản. Cách tiếp cận này giúp học được biểu diễn nút giàu ngữ cảnh và toàn diện hơn.

Çelik và cộng sự giới thiệu CiteBART — mô hình sinh (generative) dựa trên kiến trúc BART có khả năng tạo trực tiếp tiêu đề bài báo từ ngữ cảnh trích dẫn theo hướng end-to-end. Thi và cộng sự đề xuất mô hình RHN-DualLCR tích hợp Recurrent Highway Networks với SciBERT cho khuyến nghị trích dẫn cục bộ. Sau đó nhóm này phát triển tiếp thành mô hình SciBERT-GraphSAGE — đây là mô hình được luận văn tập trung tìm hiểu.

Bảng 1.2. Tổng hợp và đối chiếu các mô hình tiêu biểu

Mô hình	Nhóm PP	Thành phần chính	Ưu điểm	Hạn chế
NCN [6]	CB	TDNN + RNN (Encoder-Decoder)	Bộ mã hóa-giải mã linh hoạt	Chưa dùng đủ thông tin; phép nhúng cũ
DualLCR [8]	CB+CF	BiLSTM + AI2 embedding	Kết hợp ngữ nghĩa và học thuật	BiLSTM và phép nhúng AI2 lỗi thời
BERT-GCN [10]	CB+Graph	BERT + GCN	Khai thác cả văn bản và đồ thị	GCN transductive; BERT tổng quát
HAtten [11]	CB	Hierarchical Attention + SciBERT	Trích xuất thông tin đa tầng	Không dùng cấu trúc mạng trích dẫn
SPECTER [13]	CB	BERT + Triplet Loss (citation)	Embedding học thuật phong phú	Không học quy nạp cho bài mới
CiteBART [12]	CB (Gen)	BART Seq2Seq	Sinh trực tiếp tiêu đề bài báo	Khó kiểm soát độ chính xác
RHN-DualLCR [21]	CB+CF	SciBERT + RHN	SciBERT + RHN hiện đại hơn BiLSTM	Chưa khai thác cấu trúc đồ thị
SciBERT-GraphSAGE [22]	CB+Graph	SciBERT + GraphSAGE	Ngữ nghĩa KH + đồ thị quy nạp	Chi phí tính toán cao

1.3. PHÂN TÍCH HẠN CHẾ VÀ ĐỊNH HƯỚNG NGHIÊN CỨU

Qua khảo sát toàn diện các nghiên cứu hiện có, có thể nhận thấy ba hạn chế cốt lõi cần được giải quyết nhằm nâng cao hiệu quả hệ thống khuyến nghị trích dẫn:

Hạn chế thứ nhất liên quan đến biểu diễn ngôn ngữ học thuật. Nhiều mô hình sử dụng BERT tổng quát hoặc BiLSTM với phép nhúng cũ (AI2, GloVe) cho nội dung văn bản học thuật. Tiêu biểu là mô hình NCN của Ebesu và Fang (2017), phiên bản cải tiến của Färber và cộng sự (2020) — mặc dù đạt kết quả tốt trên bộ dữ liệu RefSeer và arXiv CS, chưa sử dụng được toàn bộ thông tin bài báo (tiêu đề, tác giả, năm xuất bản) và chưa áp dụng các mô hình ngôn ngữ tiên tiến. Khoảng 42% từ vựng trong văn bản khoa học không xuất hiện trong từ điển BERT tổng quát, dẫn đến biểu diễn ngữ nghĩa kém chính xác cho các thuật ngữ chuyên ngành như "ablation study", "perplexity", "nucleotide sequencing".

Hạn chế thứ hai liên quan đến phương pháp học biểu diễn đồ thị. Mô hình BERT-GCN của Jeong và cộng sự (2020) là hướng đi đúng khi kết hợp nội dung và đồ thị, nhưng dùng GCN transductive — không thể tổng quát hóa cho bài báo mới mà không cần huấn luyện lại toàn bộ mô hình. Đây là hạn chế nghiêm trọng trong bối cảnh thực tế có hàng nghìn bài báo mới được công bố mỗi ngày trên ArXiv, PubMed và các cơ sở dữ liệu khác. Mô hình DualLCR của Medić và Šnajder sử dụng phép nhúng AI2 từ năm 2017 không còn là phương pháp tốt nhất cho văn bản khoa học khi đã có SciBERT được huấn luyện chuyên biệt trên hàng triệu bài báo.

Hạn chế thứ ba nằm ở việc các mô hình hiện tại vẫn chưa tận dụng một cách hiệu quả nguồn siêu dữ liệu của bài báo. Trong thực tế, khi tìm kiếm tài liệu để trích dẫn, các nhà nghiên cứu không chỉ quan tâm đến nội dung mà còn cân nhắc đến nhiều yếu tố khác như uy tín của tác giả, số lượng trích dẫn, thời điểm công bố cũng như hội nghị hoặc tạp chí nơi bài báo được xuất bản. Phần lớn các mô hình lọc nội dung bỏ qua hoàn toàn những yếu tố này.

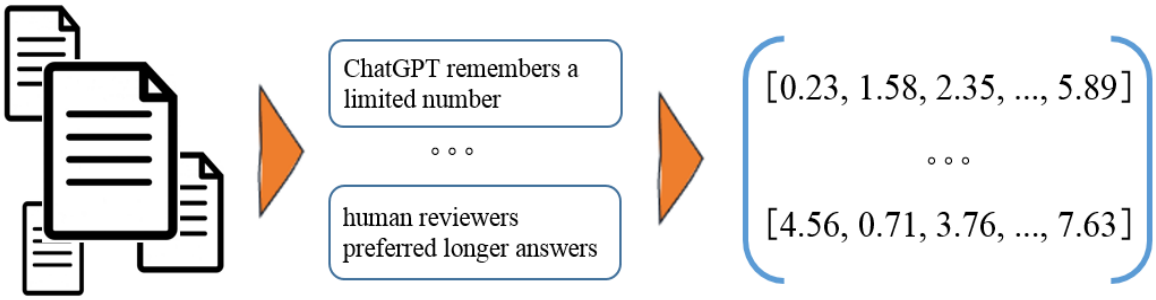
Từ ba phân tích trên, giải pháp tự nhiên là: (1) thay BERT tổng quát bằng SciBERT chuyên biệt cho văn bản học thuật; (2) thay GCN transductive bằng GraphSAGE quy nạp giải quyết cold-start; (3) khai thác siêu dữ liệu bài báo

thông qua vector đặc trưng nút trong đồ thị. Đây chính là ba đổi mới cốt lõi của mô hình SciBERT-GraphSAGE được tìm hiểu trong chương tiếp theo.

1.4. CÁC KIẾN THỨC LÝ THUYẾT

1.4.1. Phép nhúng văn bản

Trong lĩnh vực học máy và trí tuệ nhân tạo, nhúng văn bản (document embedding) là phương pháp chuyển đổi các đơn vị văn bản — như câu, đoạn hoặc toàn bộ tài liệu — thành các vector số có số chiều thấp. Những vector này phản ánh đặc trưng ngữ nghĩa của nội dung, sao cho các văn bản có ý nghĩa gần nhau sẽ được biểu diễn bằng các vector nằm gần nhau trong không gian. Nhờ đó, có thể thực hiện hiệu quả các tác vụ như so sánh, tìm kiếm, phân cụm hay phân loại văn bản dựa trên ý nghĩa, thay vì chỉ dựa vào sự trùng khớp từ khóa.



Hình 1.7. Nguyên lý phép nhúng văn bản trong không gian vector

Có hai nhóm phương pháp chính. Nhóm dựa trên từ (word-based) kết hợp các vector từ để tạo vector văn bản: tính trung bình vector từ đơn giản nhưng bỏ qua thứ tự; TF-IDF weighted average gán trọng số theo tần suất nghịch đảo, ưu tiên từ hiếm và có nhiều thông tin; SIF (Smooth Inverse Frequency) là biến thể cải tiến loại bỏ thành phần chính đầu tiên để loại bỏ thông tin phổ biến không có ý nghĩa. Nhóm dựa trên văn bản (document-based) mã hóa toàn bộ văn bản cùng lúc: Doc2Vec học embedding bằng cách dự đoán các từ từ embedding tài liệu; BERT và SciBERT sử dụng Transformer encoder mã hóa toàn bộ văn bản theo ngữ cảnh hai chiều.

Với các mô hình khuyến nghị trích dẫn, khoảng cách giữa hai vector biểu diễn thường được đo bằng độ tương tự cosine:

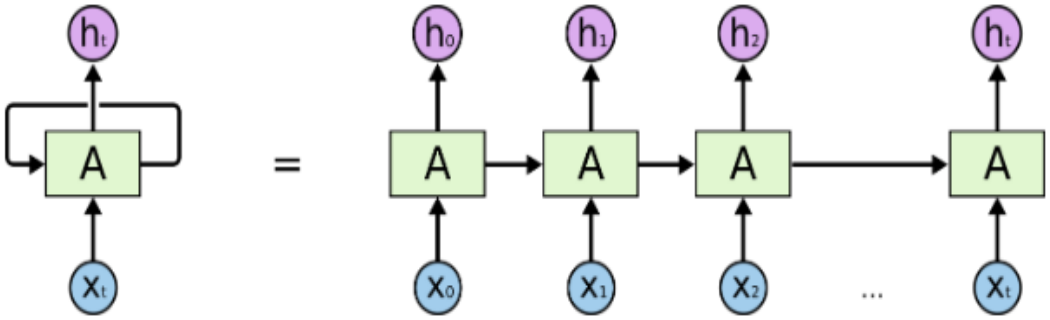
$$cosine\_similarity(e_1, e_2) = (e_1^T \times e_2) / (\|e_1\| \times \|e_2\|)$$

Trong đó: $\|e\| = \sqrt{(\Sigma e_i^2)}$ là độ lớn (norm) của vector e. Độ tương tự cosine bằng 1 khi hai vector cùng hướng (hai văn bản hoàn toàn tương tự về ngữ nghĩa)

và bằng 0 khi hai vector vuông góc (không liên quan về ngữ nghĩa). Đây là chỉ số được sử dụng phổ biến vì nó không phụ thuộc vào độ dài của văn bản.

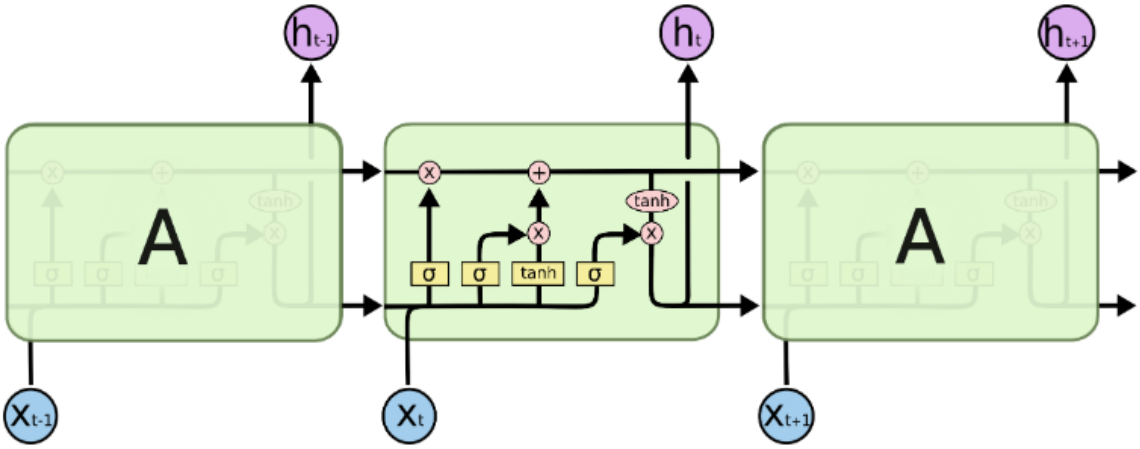
1.4.2. Họ các mô hình nơ-ron hồi quy

Dữ liệu văn bản là dữ liệu tuần tự với độ dài thay đổi — không thể xử lý hiệu quả bằng mạng nơ-ron truyền thẳng (FNN). Mạng nơ-ron hồi quy (RNN) là một kiến trúc được xây dựng để xử lý dữ liệu tuần tự dạng chuỗi bằng cách sử dụng đầu ra từ lần lặp trước làm đầu vào cho lần lặp sau, tạo ra "bộ nhớ" về các phần tử trước đó trong chuỗi.



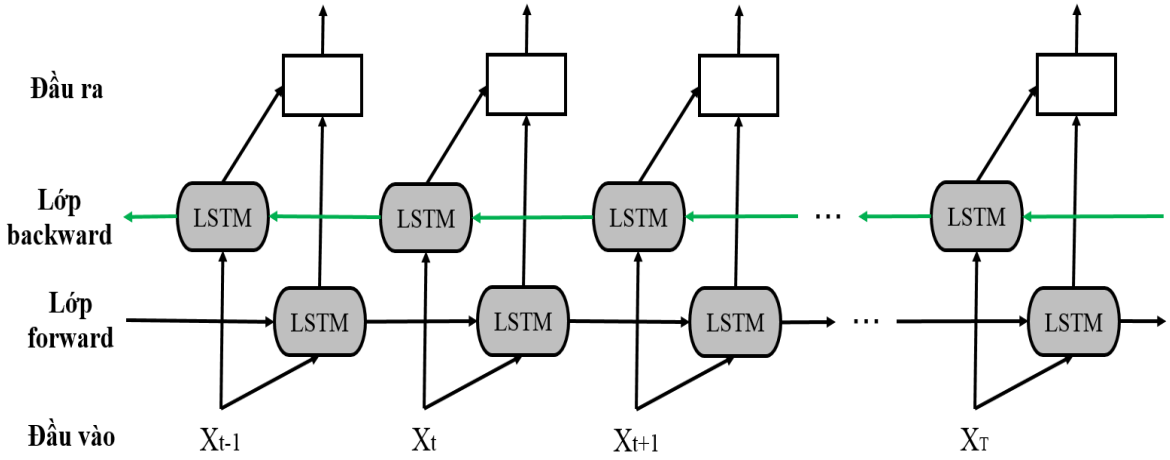
Hình 1.8. Cấu trúc mạng RNN và cơ chế xử lý tuần tự

Tuy nhiên, RNN gặp vấn đề gradient biến mất (vanishing gradient) khi xử lý chuỗi dài — khó nắm bắt phụ thuộc dài hạn giữa các từ cách xa nhau trong câu. LSTM (Long Short-Term Memory — Hochreiter và Schmidhuber, 1997) được thiết kế để giải quyết vấn đề này với cấu trúc phức tạp hơn gồm bốn tầng tương tác theo cơ chế cổng: cổng quên (forget gate) quyết định thông tin nào cần loại bỏ khỏi trạng thái ô nhớ; cổng đầu vào (input gate) quyết định thông tin mới nào được lưu trữ; cổng đầu ra (output gate) quyết định đầu ra dựa trên trạng thái ô nhớ.



Hình 1.9. Kiến trúc ô nhớ LSTM với ba cổng điều khiển

BiLSTM (Bidirectional LSTM) xử lý chuỗi theo cả hai chiều — từ trái sang phải và từ phải sang trái — rồi tổng hợp các đầu ra. Điều này đặc biệt hữu ích trong NLP vì ngữ nghĩa của một từ phụ thuộc vào cả ngữ cảnh trước và sau. Tuy nhiên, BiLSTM vẫn tốn kém về tính toán và gặp khó khăn với chuỗi rất dài



Hình 1.10. Mô hình BiLSTM với hai hướng xử lý song song

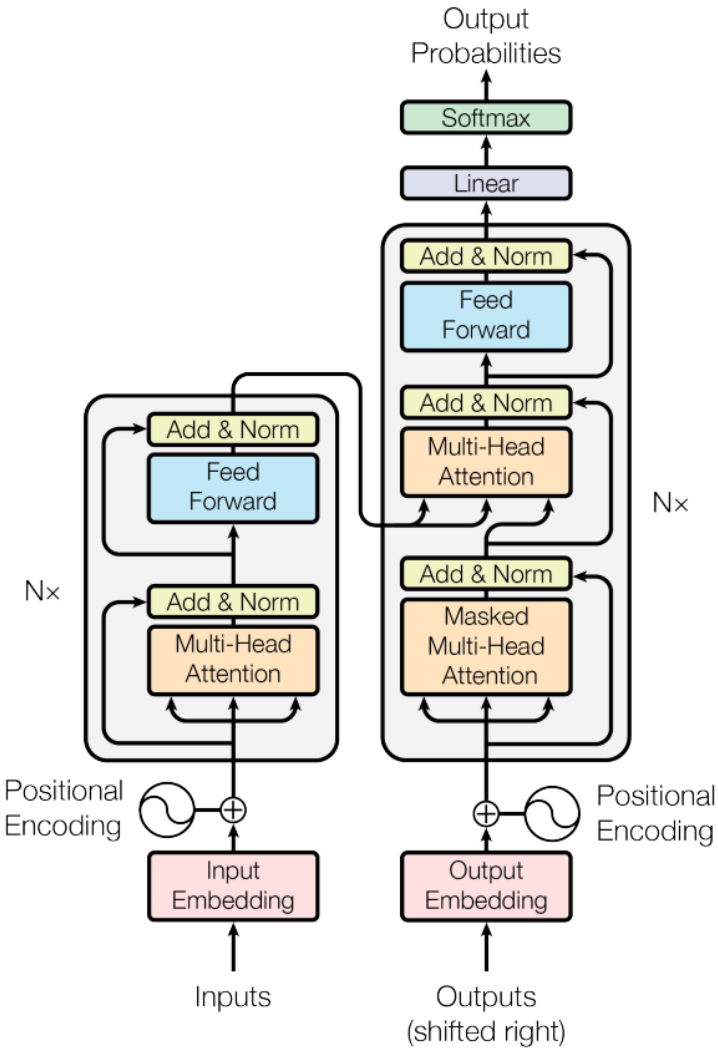
RHN (Recurrent Highway Networks — Zilly và cộng sự, 2017) được đề xuất như một mở rộng của BiLSTM, cho phép thực hiện các biến đổi sâu hơn tại mỗi bước thời gian thông qua các lớp highway. RHN có gradient flow tốt hơn, ít tham số hơn cho cùng độ sâu, và kiểm soát chính xác hơn thông qua các cổng highway — đây là những ưu điểm quan trọng khi xử lý văn bản học thuật dài và phức tạp.

Bảng 1.3. Đặc điểm so sánh RHN và BiLSTM

Tiêu chí	RHN	BiLSTM
Biến đổi mỗi bước thời gian	Sâu qua nhiều lớp highway	Nông (trừ khi thêm nhiều lớp LSTM)
Số lượng tham số	Ít hơn cho cùng độ sâu	Nhiều hơn do cấu trúc hai chiều
Gradient flow	Tốt hơn nhờ kết nối highway	Giảm gradient biến mất nhưng vẫn có hạn chế
Kiểm soát luồng thông tin	Chính xác qua các cổng highway	Ít kiểm soát hơn cho biến đổi nội bộ
Phụ thuộc dài hạn	Nắm bắt tốt qua biến đổi sâu	Ưu thế nhờ hai chiều nhưng có giới hạn
Tốc độ suy luận	Nhanh hơn	Chậm hơn do xử lý hai chiều

1.4.3. Kiến trúc Transformer và các mô hình tiền huấn luyện

Kiến trúc Transformer được Vaswani và cộng sự giới thiệu trong công trình "Attention Is All You Need" (NeurIPS, 2017). Khác với RNN/LSTM xử lý tuần tự từng từ, Transformer sử dụng cơ chế tự chú ý (self-attention) để mô hình hóa mối quan hệ giữa tất cả các cặp từ trong câu một cách đồng thời, cho phép tính toán song song hoàn toàn và nắm bắt phụ thuộc dài hạn hiệu quả.



Hình 1.11. Kiến trúc Transformer gốc (Encoder-Decoder)

Cốt lõi là cơ chế multi-head self-attention: mỗi từ được ánh xạ thành ba vector Query (Q), Key (K) và Value (V). Trọng số chú ý được tính bằng:

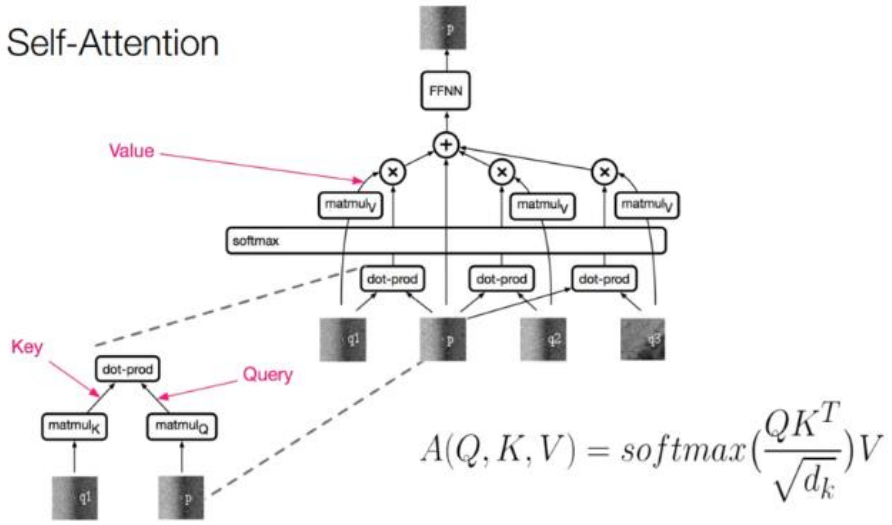
$$\boldsymbol{Attention(Q,\ K,\ V) = Softmax(Q \times K^T / \sqrt{d_k}) \times V}$$

Trong đó

d_k là chiều của vector Key,

$\sqrt{d_k}$ là hệ số chia để ổn định gradient.

Multi-head attention thực hiện đồng thời nhiều phép tính attention song song, mỗi phép sử dụng các phép biến đổi tuyến tính riêng biệt. Các kết quả sau đó được ghép lại, cho phép mô hình học và khai thác nhiều không gian biểu diễn ngữ nghĩa khác nhau cùng một lúc.



Hình 1.12. Nguyên lý cơ chế Self-Attention

BERT (Bidirectional Encoder Representations from Transformers — Devlin và cộng sự, 2019): Mô hình Transformer encoder hai chiều, tiền huấn luyện theo Masked Language Modeling (MLM — dự đoán từ bị che ngẫu nhiên) và Next Sentence Prediction (NSP — dự đoán câu tiếp theo). BERT-Base: 12 lớp Transformer, 768 chiều ẩn, 12 đầu chú ý, 110 triệu tham số. Nhờ mã hóa hai chiều, BERT xây dựng biểu diễn ngữ cảnh sâu sắc — biểu diễn của mỗi từ phụ thuộc vào cả ngữ cảnh trước và sau.

SciBERT (Beltagy và cộng sự, 2019): Biến thể của BERT được huấn luyện lại trên 1,14 triệu bài báo khoa học từ Semantic Scholar (18% khoa học máy tính, 82% y sinh), tổng cộng 3,17 tỷ token — tương đương kho dữ liệu BERT 3,3 tỷ token. SciBERT xây dựng bộ từ vựng riêng SCIVocab gồm 30.000 token từ văn bản khoa học — khoảng 42% token trong SCIVocab không xuất hiện trong từ vựng BERT chuẩn. Đây chính là các thuật ngữ chuyên ngành khoa học mà BERT tổng quát không thể xử lý tốt. Thực nghiệm cho thấy SciBERT vượt trội BERT

đáng kể trong phân loại văn bản học thuật, trích xuất thực thể và phân tích quan hệ trong bài báo khoa học.

SPECTER (Scientific Paper Embeddings using Citation-informed Transformers — Cohan và cộng sự, 2020): Mở rộng BERT bằng cách tích hợp thông tin mạng trích dẫn vào quá trình tiền huấn luyện. SPECTER sử dụng cấu trúc trích dẫn như tín hiệu giám sát bổ sung qua hàm mất mát triplet loss, giúp embedding phản ánh cả ngữ nghĩa và quan hệ học thuật. Hai bài báo trích dẫn nhau sẽ có embedding gần nhau hơn, bài báo không liên quan sẽ có embedding xa nhau.

Bảng 1.4. Đặc điểm so sánh các mô hình Transformer tiền huấn luyện phổ biến

Mô hình	Dữ liệu tiền huấn luyện	Kiểu mã hóa	Điểm đặc biệt	Ứng dụng phù hợp
BERT	Wikipedia+ BookCorpus (3,3B tokens)	Hai chiều, Masked LM	NSP objective	NLP tổng quát
SciBERT	1,14M bài báo KH (3,17B tokens)	Hai chiều, SCIVocab	Từ vựng khoa học chuyên biệt	Văn bản học thuật, KH máy tính, y sinh
SPECTER	Bài báo + mạng trích dẫn (triplet)	Hai chiều + triplet loss	Embedding từ quan hệ trích dẫn	Tìm kiếm học thuật, khuyến nghị KH
RoBERTa	Nhiều corpus lớn hơn BERT	Hai chiều, không có NSP	Tối ưu lịch trình huấn luyện	NLP nâng cao, nhiều nhiệm vụ
GPT-2/3	Web text quy mô lớn	Một chiều, tự hồi quy	Khả năng sinh văn bản mạnh	Sinh văn bản, hoàn thiện câu

1.4.4. Học biểu diễn đồ thị với GraphSAGE

Mạng nơ-ron đồ thị (Graph Neural Network — GNN) là họ các mô hình học sâu được thiết kế để học trên dữ liệu dạng đồ thị G = (V, E), trong đó V là tập nút và E là tập cạnh. Ý tưởng cốt lõi là cập nhật biểu diễn của mỗi nút bằng

cách tổng hợp thông tin từ các nút láng giềng — phương pháp này gọi là Message Passing.

GCN (Graph Convolutional Networks — Kipf và Welling, 2016) là mô hình GNN phổ biến, lan truyền thông tin qua các cạnh trong đồ thị theo công thức: $H^{(l+1)} = \sigma(\tilde{A} \times H^{(l)} \times W^{(l)})$, trong đó $\tilde{A} = D^{-1/2} \times A \times D^{-1/2}$ là ma trận kề chuẩn hóa, H là ma trận đặc trưng nút, W là ma trận trọng số học được. GCN hoạt động theo phương thức transductive — cần toàn bộ đồ thị trong bộ nhớ và không thể tổng quát hóa sang nút mới mà không huấn luyện lại.

GraphSAGE (Graph Sampling and Aggregation — Hamilton và cộng sự, 2017) giải quyết hạn chế này bằng phương pháp học quy nạp (inductive learning). Thay vì dùng toàn bộ đồ thị, GraphSAGE lấy mẫu một tập con cố định K nút láng giềng và học hàm tổng hợp trên tập mẫu đó. Tại lớp l, biểu diễn nút v_i được cập nhật theo hai bước:

$$h_{N(i)}^{(l)} = AGGREGATE(\{h_j^{(l-1)} : j \in N(v_i)\})$$

$$h_i^{(l)} = \sigma(W^{(l)} \cdot CONCAT(h_i^{(l-1)}, h_{N(i)}^{(l)}) + b^{(l)})$$

Trong đó

$N(v_i)$ là tập láng giềng được lấy mẫu ngẫu nhiên của v_i,

AGGREGATE là hàm tổng hợp (có thể là mean aggregator,

LSTM aggregator hoặc pooling aggregator),

$W^{(l)}$ là ma trận trọng số học được,

σ là hàm kích hoạt ReLU.

Vì GraphSAGE học hàm tổng hợp (aggregation function) thay vì học embedding cho từng nút cụ thể, nó có thể áp dụng hàm này để tính embedding cho bất kỳ nút mới nào — kể cả nút chưa thấy trong quá trình huấn luyện.

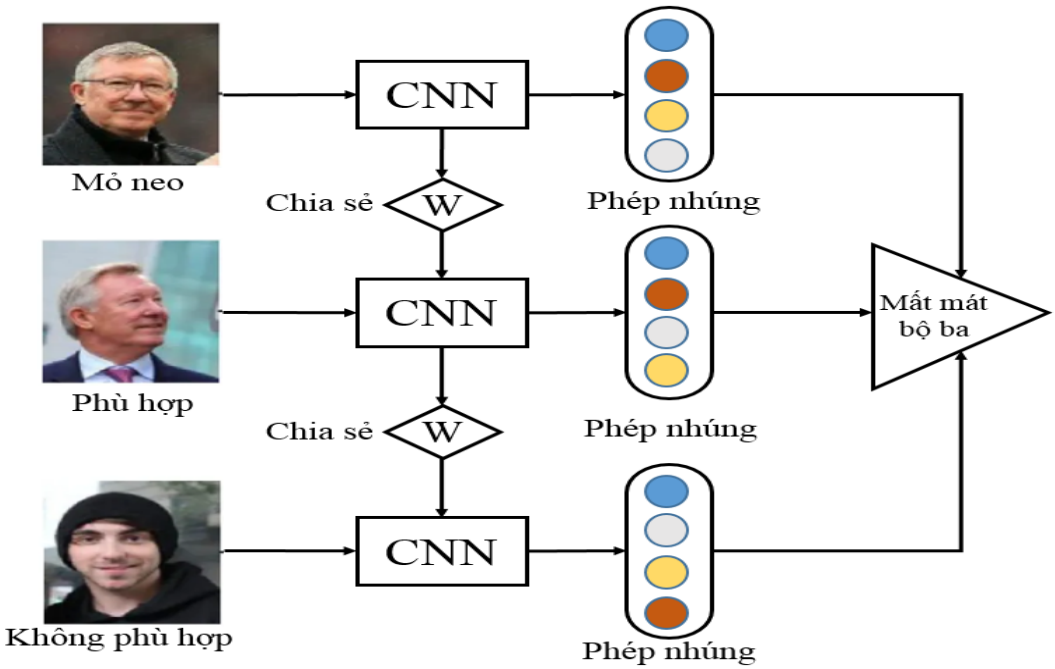
Bảng 1.5. So sánh GraphSAGE và GCN

Tiêu chí	GCN	GraphSAGE
Phương thức học	Transductive (chỉ học nút đã thấy)	Inductive (tổng quát cho nút mới)
Xử lý bài báo mới	Cần huấn luyện lại toàn bộ	Tổng quát hóa ngay không cần huấn luyện lại

Tiêu chí	GCN	GraphSAGE
Tổng hợp láng giềng	Toàn bộ láng giềng	Lấy mẫu K láng giềng ngẫu nhiên
Yêu cầu bộ nhớ	Toàn bộ ma trận kề trong bộ nhớ	Mini-batch với láng giềng đã lấy mẫu
Khả năng mở rộng	Khó với đồ thị quy mô lớn	Tốt nhờ cơ chế lấy mẫu
Phù hợp cho KN trích dẫn	Hạn chế — bài mới xuất hiện liên tục	Phù hợp — bài báo mới được hỗ trợ ngay

1.4.5. Hàm mất mát bộ ba và các chỉ số đánh giá

Hàm mất mát bộ ba (Triplet Loss) thường được áp dụng trong các mô hình khuyến nghị trích dẫn nhằm giúp phân biệt rõ giữa các bài báo liên quan và không liên quan. Phương pháp này dựa trên việc xây dựng các bộ ba dữ liệu gồm (anchor, positive, negative), trong đó anchor đại diện cho ngữ cảnh trích dẫn, positive là bài báo thực sự được trích dẫn, còn negative là bài báo không được trích dẫn và thường được chọn ngẫu nhiên.



Hình 1.13. Ý nghĩa trực quan của hàm mất mát Triplet Loss

Mục tiêu của Triplet Loss là đưa embedding của cặp (anchor, positive) gần nhau hơn và đẩy embedding của cặp (anchor, negative) xa nhau hơn trong không gian vector:

$$L\_triplet = max(0, d(anchor, positive) - d(anchor, negative) + margin)$$

Trong đó

$d(\cdot,\cdot)$ là hàm khoảng cách (thường là khoảng cách Euclidean hoặc 1 − cosine similarity),

margin là ngưỡng đảm bảo khoảng cách tối thiểu giữa positive và negative.

Hàm max(0, ...) đảm bảo mất mát bằng 0 khi điều kiện margin đã được thỏa mãn. Trong SPECTER và SciBERT-GraphSAGE, Triplet Loss được áp dụng để tối ưu embedding tài liệu với tín hiệu giám sát từ mạng trích dẫn.

Về chỉ số đánh giá, có ba chỉ số tiêu chuẩn sử dụng phổ biến nhất trong bài toán khuyến nghị trích dẫn. MAP (Mean Average Precision) đánh giá chất lượng toàn bộ danh sách khuyến nghị, phù hợp khi người dùng quan tâm đến tất cả kết quả liên quan. AP là độ chính xác trung bình tại mỗi vị trí kết quả đúng; MAP là trung bình AP trên toàn bộ tập truy vấn Q:

$$MAP = (1/n) \times \Sigma_i AP(i) \quad , \quad AP(i) = \Sigma_k P(k) \times rel(k) / |R|$$

MRR (Mean Reciprocal Rank) tập trung vào vị trí kết quả đúng đầu tiên, đặc biệt quan trọng khi người dùng chỉ cần tài liệu phù hợp nhất ở vị trí cao nhất:

$$MRR = (1/|Q|) \times \Sigma_i (1/rank(i))$$

Recall@K đo tỷ lệ tài liệu đúng được truy xuất trong top-K kết quả. Recall@10 đặc biệt phổ biến vì người dùng thường xem xét khoảng 10 đề xuất hàng đầu:

$$Recall@K = |R \cap R_k| / |R|$$

trong đó R là tập tài liệu liên quan thực sự, R_k là top-K kết quả được truy xuất. Thực nghiệm thường báo cáo Recall@5, @10, @30, @50 và @80 để đánh giá toàn diện ở nhiều ngưỡng khác nhau.

1.5. KẾT LUẬN CHƯƠNG 1

Chương 1 đã trình bày tổng quan toàn diện về bài toán khuyến nghị trích dẫn và các phương pháp học sâu liên quan. Qua phân tích từ lọc cộng tác, lọc

nội dung, lọc đồ thị đến mô hình kết hợp, xu hướng nghiên cứu rõ ràng là: các mô hình kết hợp tích hợp nhiều nguồn thông tin ngày càng cho kết quả tốt hơn, và các mô hình sử dụng Transformer chuyên biệt cho văn bản học thuật (SciBERT, SPECTER) vượt trội so với BERT tổng quát.

Phân tích hạn chế hiện tại có ba vấn đề cốt lõi sau: (1) cần mô hình ngôn ngữ chuyên biệt cho văn bản học thuật — SciBERT là lựa chọn tối ưu với bộ từ vựng SCIVocab và dữ liệu tiền huấn luyện khoa học; (2) cần phương pháp biểu diễn đồ thị quy nạp cho bài báo mới — GraphSAGE giải quyết hạn chế cold-start của GCN transductive; (3) cần kết hợp hiệu quả thông tin ngữ nghĩa và cấu trúc mạng để tận dụng tối đa cả hai nguồn thông tin bổ sung. Mô hình SciBERT-GraphSAGE được tìm hiểu trong chương tiếp theo nhằm giải quyết đồng thời ba vấn đề này.

CHƯƠNG 2
ỨNG DỤNG MÔ HÌNH HỌC SÂU TRANSFORMER TIÊN TIẾN THEO TIẾP CẬN KẾT HỢP CHO BÀI TOÁN KHUYẾN NGHỊ TRÍCH DẪN BÀI BÁO KHOA HỌC

Chương này tập trung phân tích và trình bày chi tiết mô hình SciBERT-GraphSAGE — một phương pháp kết hợp giữa lọc dựa trên nội dung thông qua SciBERT và lọc dựa trên cấu trúc đồ thị bằng GraphSAGE, được nghiên cứu và công bố bởi Đinh Ngọc Thi và cộng sự trong bài báo trên IEEE Access (2023) [22]. Mô hình này tận dụng đồng thời hai nguồn thông tin bổ sung cho nhau về bài báo khoa học: thông tin ngữ nghĩa từ nội dung văn bản và thông tin cấu trúc từ mạng trích dẫn cùng siêu dữ liệu. Nội dung chương bao gồm: phân tích vấn đề và hướng tiếp cận; mô tả kiến trúc tổng thể; bộ mã hóa ngữ nghĩa SciBERT; bộ mã hóa đồ thị GraphSAGE; cơ chế kết hợp và hàm mục tiêu; thách thức trong tích hợp; và phân tích ưu điểm, hạn chế.

2.1. PHÂN TÍCH VẤN ĐỀ VÀ HƯỚNG TIẾP CẬN

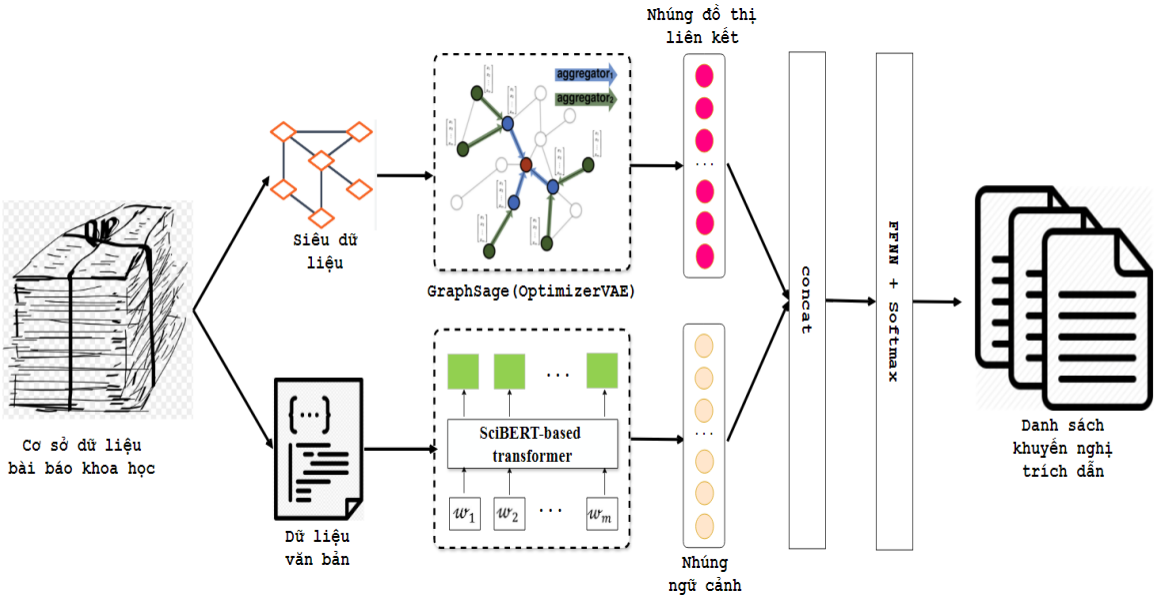
Phần lớn các mô hình khuyến nghị trích dẫn hiện có nhận biết ngữ cảnh và tập trung vào nội dung của cả ngữ cảnh trích dẫn và bài báo ứng viên. Cách tiếp cận đơn lẻ này bỏ qua các yếu tố bổ sung quan trọng có ảnh hưởng trực tiếp đến quyết định trích dẫn của nhà nghiên cứu trong thực tế.

Để hiểu rõ vấn đề này, cần xem xét hành vi trích dẫn thực tế. Khi quyết định trích dẫn một bài báo, các nhà nghiên cứu không chỉ xem xét sự tương tự về nội dung mà còn cân nhắc nhiều yếu tố khác: Tác giả — thông tin về tác giả của bài báo có thể phản ánh mối liên hệ với các công trình khác thông qua quan hệ đồng tác giả; đồng thời, các nhà nghiên cứu thường có xu hướng trích dẫn những tác giả có uy tín trong lĩnh vực. Nơi xuất bản — hội nghị hoặc tạp chí nơi bài báo được công bố thường tập trung vào những chủ đề nhất định, vì vậy các công trình cùng nơi xuất bản có khả năng liên quan về nội dung; bên cạnh đó, mức độ uy tín của nơi công bố cũng tác động đến quyết định trích dẫn. Năm xuất bản — các nghiên cứu gần đây thường được ưu tiên hơn do tính cập nhật và phản ánh tiến bộ mới nhất của lĩnh vực. Mạng lưới trích dẫn — các liên kết trích dẫn giữa các bài báo thể hiện mức độ ảnh hưởng và sự lan tỏa học thuật của một công trình trong cộng đồng nghiên cứu; bài báo được nhiều người uy tín trích dẫn thường là tài liệu quan trọng trong lĩnh vực.

Phân tích trên dẫn đến kết luận: để xây dựng mô hình khuyến nghị trích dẫn hiệu quả, cần kết hợp hai nguồn thông tin bổ sung cho nhau: (1) Thông tin ngữ nghĩa từ nội dung văn bản của ngữ cảnh trích dẫn và bài báo; (2) Thông tin cấu trúc được khai thác từ mạng lưới liên kết trích dẫn cùng với siêu dữ liệu của bài báo (bao gồm tác giả, năm công bố và địa điểm xuất bản). Mô hình SciBERT-GraphSAGE được phát triển dựa trên nguyên tắc tích hợp các nguồn thông tin này.

2.2. KIẾN TRÚC TỔNG THỂ MÔ HÌNH SCIBERT-GRAPHSAGE

Mô hình SciBERT-GraphSAGE có kiến trúc hai nhánh song song (dual-branch parallel architecture), được thiết kế để xử lý đồng thời hai loại thông tin khác nhau về bài báo khoa học. Mỗi nhánh được tối ưu hóa cho một loại dữ liệu và hai nhánh được kết hợp ở lớp phân loại cuối.



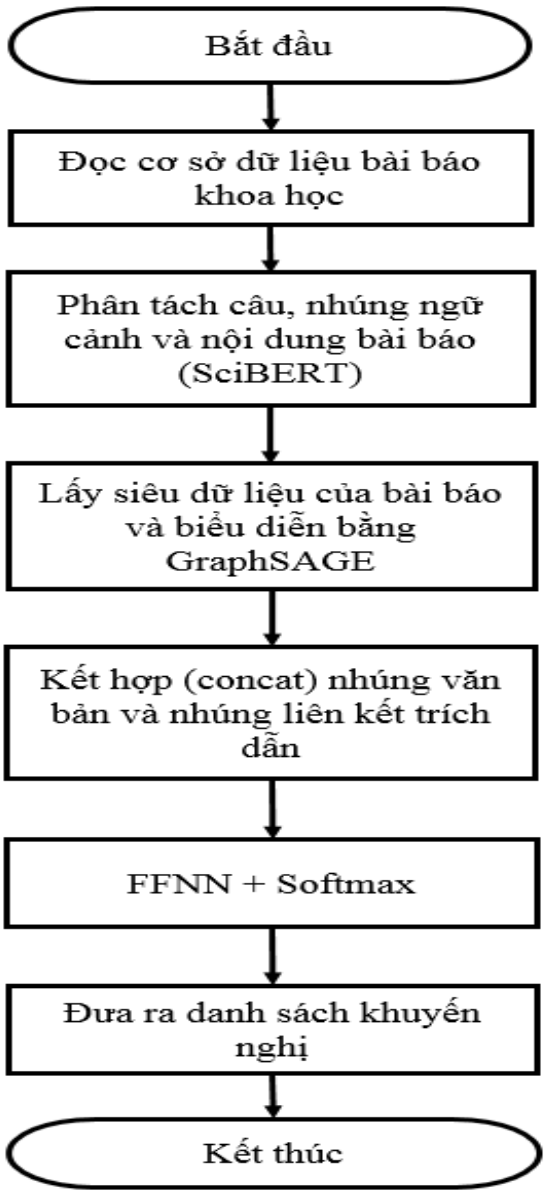
Hình 2.1. Kiến trúc hai nhánh song song của SciBERT-GraphSAGE

Dữ liệu đầu vào của mô hình được thu thập từ cơ sở dữ liệu các bài báo khoa học và được chia thành hai nhóm chính. Nhóm thứ nhất là dữ liệu văn bản, bao gồm ngữ cảnh trích dẫn (câu hoặc đoạn văn tại vị trí cần chèn trích dẫn), cùng với tiêu đề và phần tóm tắt của các bài báo ứng viên. Nhóm thứ hai là siêu dữ liệu, bao gồm thông tin về nơi công bố (tên tạp chí hoặc hội nghị), năm xuất bản, danh sách tác giả, cũng như các liên kết trích dẫn như bài báo trích dẫn những công trình nào và được những công trình nào trích dẫn.

Hai nhánh xử lý song song hoạt động độc lập: Nhánh 1 là Bộ mã hóa ngữ nghĩa SciBERT nhận dữ liệu văn bản và tạo ra vector nhúng 768 chiều cho mỗi

chuỗi văn bản đầu vào, nắm bắt ngữ nghĩa sâu của nội dung học thuật. Nhánh 2 là Bộ mã hóa đồ thị GraphSAGE nhận siêu dữ liệu và cấu trúc mạng trích dẫn, tạo ra vector nhúng 768 chiều cho mỗi bài báo trong đồ thị, nắm bắt vị trí và tầm ảnh hưởng trong cộng đồng học thuật.

Hai loại vector nhúng sau đó được kết hợp bằng phép nối (concatenation) thành vector 1536 chiều. Vector này được đưa qua mạng nơ-ron truyền thẳng (FNN) với hàm kích hoạt ReLU và lớp Dropout để tránh overfitting, rồi qua lớp hồi quy Softmax để ước lượng xác suất tồn tại quan hệ trích dẫn giữa ngữ cảnh và bài báo ứng viên. Bài báo ứng viên có xác suất cao nhất được khuyến nghị.



Hình 2.2. Sơ đồ xử lý tuần tự của mô hình SciBERT-GraphSAGE

2.3. BỘ MÃ HÓA NGỮ NGHĨA SCIBERT

SciBERT là thành phần đầu tiên trong kiến trúc hai nhánh của mô hình, chịu trách nhiệm mã hóa thông tin ngữ nghĩa từ nội dung văn bản học thuật. Như đã trình bày ở Chương 1, SciBERT là biến thể chuyên biệt của BERT được huấn luyện trên 1,14 triệu bài báo khoa học với bộ từ vựng SCIVocab gồm 30.000 token khoa học — điều này giúp SciBERT xử lý chính xác hơn các thuật ngữ chuyên ngành so với BERT tổng quát.

Trong mô hình SciBERT-GraphSAGE, SciBERT được sử dụng để mã hóa ba loại dữ liệu văn bản: ngữ cảnh trích dẫn, tiêu đề bài báo ứng viên và tóm tắt bài báo ứng viên. Chuỗi đầu vào được tổ chức theo cấu trúc chuẩn BERT:

[CLS] ngữ_cảnh_trích_dẫn [SEP] tiêu_đề + tóm_tắt [SEP]

Trong đó

[CLS] là token đặc biệt đại diện cho toàn bộ chuỗi,

[SEP] là token phân cách.

Cấu trúc này được tokenize bằng bộ từ vựng SCIVocab và tối đa 128 token.

Phép nhúng đầu vào của SciBERT được cấu thành từ ba thành phần cộng lại: (1) Nhúng token biểu diễn từng đơn vị từ vựng — SciBERT dùng WordPiece để phân tách từ thành subword units, giúp xử lý hiệu quả thuật ngữ chưa gặp, mỗi token được ánh xạ thành vector 768 chiều; (2) Nhúng phân đoạn phân biệt câu đầu và câu sau — token thuộc câu đầu nhận vector loại A, token thuộc câu sau nhận vector loại B; (3) Nhúng vị trí mã hóa thứ tự của token trong chuỗi — vì Transformer không có cơ chế hồi quy tuần tự.

$$E_i = E\_token(i) + E\_segment(i) + E\_position(i)$$

Chuỗi biểu diễn đầu vào sau đó được đưa qua 12 lớp Transformer encoder. Mỗi lớp bao gồm Multi-Head Self-Attention và Feed-Forward Network với layer normalization:

$$LN(x) = \alpha \times (x - \mu) / \sigma + \beta$$

$$FFN(x) = max(0, x \times W_1 + b_1) \times W_2 + b_2$$

Trong đó

μ và σ là giá trị trung bình và độ lệch chuẩn;

α, β là tham số học được.

Vector tại vị trí [CLS] từ lớp Transformer cuối cùng được sử dụng làm biểu diễn tổng hợp cho toàn bộ chuỗi đầu vào — đây là vector ngữ nghĩa 768 chiều đầu ra của nhánh SciBERT.

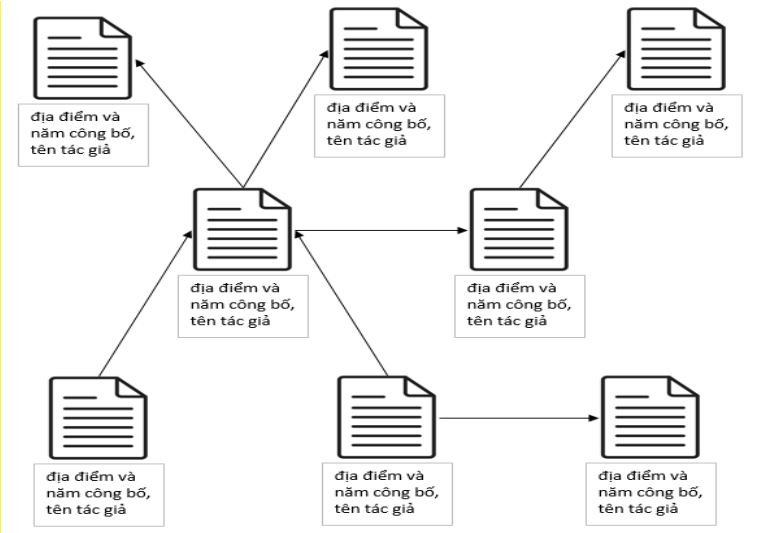
Độ phức tạp tính toán của SciBERT: với chuỗi độ dài L_t token, tính toán ma trận chú ý đòi hỏi $O(L_t^2)$ phép nhân, dẫn đến độ phức tạp $O(L_t^2 \times 768)$ mỗi lớp và $O(L_t^2 \times 768 \times 12)$ toàn bộ. Với $L_t = 128$ và 110 triệu tham số, đây là chi phí tính toán đáng kể nhưng được bù đắp bởi khả năng tính toán song song của GPU.

Điểm đặc biệt quan trọng là SciBERT được fine-tune (tinh chỉnh) trên từng bộ dữ liệu cụ thể chứ không chỉ dùng nguyên vẹn trọng số tiền huấn luyện. Quá trình fine-tune cập nhật nhẹ các trọng số của SciBERT để phù hợp với đặc điểm của từng bộ dữ liệu (ACL-200 chuyên về NLP, RefSeer đa lĩnh vực, FullTextPeerRead từ nhiều hội nghị). Điều này giúp mô hình vừa tận dụng tri thức ngôn ngữ rộng lớn từ tiền huấn luyện, vừa thích nghi với đặc thù của bài toán cụ thể.

2.4. BỘ MÃ HÓA ĐỒ THỊ GRAPHSAGE

GraphSAGE là thành phần thứ hai trong kiến trúc hai nhánh, chịu trách nhiệm mã hóa thông tin cấu trúc từ mạng trích dẫn và siêu dữ liệu bài báo. Trong mô hình SciBERT-GraphSAGE, các bài báo khoa học được biểu diễn dưới dạng đồ thị có hướng $G = (V, E)$, trong đó $V = \{v_1, v_2, ..., v_n\}$ là tập nút (mỗi nút tương ứng một bài báo) và $E = \{(v_i, v_j)\}$ là tập cạnh (mỗi cạnh biểu diễn bài báo v_i trích dẫn bài báo v_j).

Để xây dựng đặc trưng nút X_i cho bài báo v_i, siêu dữ liệu được mã hóa như sau: tên tác giả được mã hóa thành vector multi-hot (đánh dấu các tác giả xuất hiện); năm xuất bản được chuẩn hóa về khoảng [0,1] theo công thức (năm − năm_min)/(năm_max − năm_min); địa điểm xuất bản (tên tạp chí/hội nghị) được mã hóa bằng embedding đã học. Tất cả các thành phần này được ghép nối (concatenation) thành vector đặc trưng nút tổng hợp X_i.



Hình 2.3. Cách xây dựng đồ thị trích dẫn từ siêu dữ liệu bài báo

GraphSAGE được tích hợp trong framework VGAE (Variational Graph Auto-Encoder — Kipf và Welling, 2016). VGAE là mô hình sinh (generative model) học biểu diễn Z bằng cách tối thiểu hóa hàm mất mát ELBO (Evidence Lower Bound):

$$L = E_{q(Z|X,A)}[\log p(A|Z)] - KL[q(Z|X,A) \,||\, p(Z)]$$

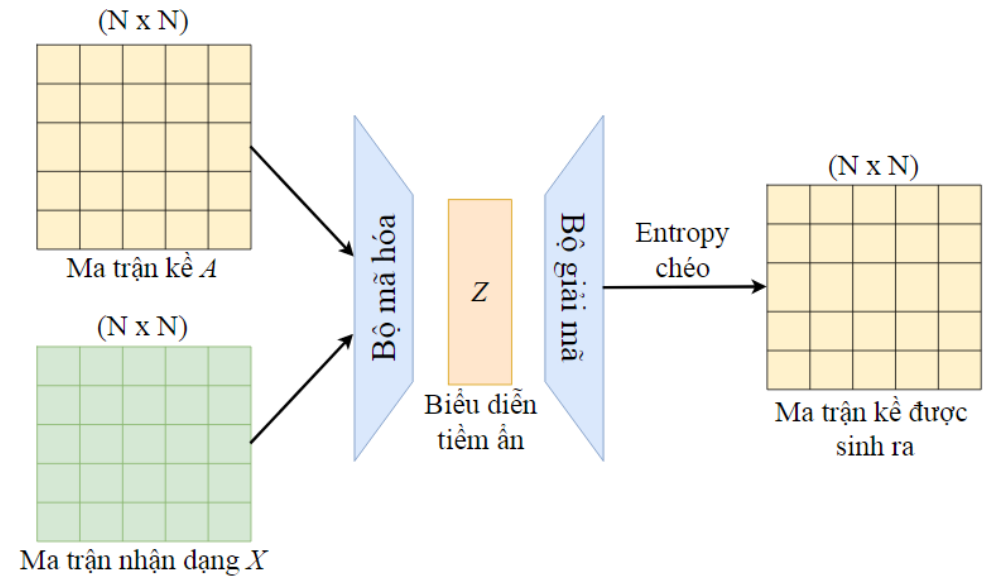
Trong đó

Số hạng đầu là likelihood tái tạo đồ thị, số hạng KL là regularizer về phân phối tiềm ẩn. Bộ mã hóa GraphSAGE học biểu diễn nút theo công thức đã trình bày ở Chương 1. Bộ giải mã VGAE tái tạo ma trận kề dựa trên biểu diễn tiềm ẩn:

$$p(A_{ij} = 1 \mid z_i, z_j) = \sigma(z_i^T \times z_j)$$

Trong đó

σ là hàm sigmoid. Khi xác suất tái tạo cạnh giữa hai nút cao, mô hình đã học được rằng hai bài báo đó có liên kết trích dẫn.



Hình 2.4. Cấu trúc và nguyên lý hoạt động của mô hình VGAE

Ưu điểm then chốt của việc dùng GraphSAGE trong framework VGAE là học quy nạp: khi một bài báo mới được thêm vào cơ sở dữ liệu với siêu dữ liệu và một số liên kết trích dẫn ban đầu, mô hình có thể tính toán biểu diễn cho bài báo đó ngay lập tức mà không cần huấn luyện lại. Hàm tổng hợp (aggregation function) đã học được trong quá trình huấn luyện sẽ được áp dụng trực tiếp — đây là điểm khác biệt căn bản so với GCN transductive trong BERT-GCN.

Theo cách tính chuẩn của Hamilton và cộng sự (2017), độ phức tạp của một lượt lan truyền tiến trong GraphSAGE được xác định bởi số nút, số láng giềng được lấy mẫu cố định tại mỗi lớp, số lớp tổng hợp và số chiều đặc trưng — chứ không phải bởi số lượng siêu dữ liệu. Cụ thể, với mỗi nút (bài báo), chi phí xấp xỉ $O(K^L \times d^2)$, trong đó K là số láng giềng được lấy mẫu tại mỗi lớp, L là số lớp tổng hợp và d là số chiều đặc trưng/nhúng; do đó độ phức tạp toàn đồ thị là $O(N \times K^L \times d^2)$ với N là số nút. Chính cơ chế lấy mẫu cố định K láng giềng tại mỗi lớp đã khống chế hiện tượng "bùng nổ kích thước lân cận", giúp chi phí tính toán tăng tuyến tính theo số nút thay vì phụ thuộc vào bậc tối đa của nút như GCN truyền thống (vốn cần toàn bộ đồ thị trong bộ nhớ, độ phức tạp xấp xỉ $O(|E| \times d + |V| \times d^2)$).

2.5. CƠ CHẾ KẾT HỢP ĐẶC TRƯNG VÀ HÀM MỤC TIÊU

Sau khi thu được vector nhúng ngữ nghĩa h_c từ SciBERT (768 chiều đại diện cho ngữ cảnh trích dẫn) và vector nhúng đồ thị h_d_i từ GraphSAGE (768

chiều đại diện cho bài báo ứng viên d_i), mô hình thực hiện phép nối (concatenation):

$$h\_combined = CONCAT(h\_c,\ h\_d_i) \rightarrow 1536 \text{ chiều}$$

Phép nối (concatenation) được chọn thay vì phép cộng (addition) hay các phương pháp khác vì lý do sau: nó bảo toàn đầy đủ thông tin từ cả hai nguồn mà không mất thông tin do rút gọn chiều; nó cho phép lớp FNN tiếp theo học cách kết hợp hai loại thông tin một cách linh hoạt; và nó tránh việc giả định cấu trúc vector của hai nguồn tương thích nhau.

Vector kết hợp 1536 chiều được đưa qua mạng nơ-ron truyền thẳng gồm hai lớp ẩn với hàm kích hoạt ReLU và lớp Dropout (tỷ lệ 0.1-0.3 tùy bộ dữ liệu) để tránh overfitting, rồi qua lớp Softmax để ước lượng điểm khuyến nghị:

$$s(c, d_i) = Softmax(FNN([h\_c\ ;\ h\_d_i])) = Softmax(W_2 \times ReLU(W_1 \times [h\_c;\ h\_d_i] + b_1) + b_2)$$

Mô hình được tối ưu với hàm mất mát binary cross-entropy:

$$L = -\Sigma_i\ [y_i \times log(\hat{y}_i) + (1 - y_i) \times log(1 - \hat{y}_i)]$$

Trong đó

$y_i = 1$ nếu d_i là trích dẫn đúng cho ngữ cảnh c (positive sample), $y_i = 0$ cho negative samples; $\hat{y}_i$ là xác suất dự đoán của mô hình. Trong mỗi batch huấn luyện, với mỗi ngữ cảnh trích dẫn, một bài báo positive, nhiều bài báo negative lấy mẫu (negative sampling ratio thường là 1:5 hoặc 1:10 tùy bộ dữ liệu).

Quá trình huấn luyện diễn ra theo chiến lược hai giai đoạn. Giai đoạn 1 — Tiền huấn luyện riêng lẻ: SciBERT được fine-tune trên dữ liệu văn bản bài báo với trọng số khởi tạo từ mô hình scibert_scivocab_uncased đã công bố công khai; GraphSAGE được huấn luyện trên đồ thị trích dẫn với mục tiêu tái tạo cấu trúc liên kết theo framework VGAE. Giai đoạn 2 — Huấn luyện kết hợp: Hai bộ mã hóa được ghép nối và lớp phân loại được tối ưu với Adam optimizer, học cách kết hợp hai loại thông tin để đưa ra quyết định khuyến nghị chính xác nhất.

Tham số tối ưu cho SciBERT: tốc độ học $\eta = 2\times10^{-5}$, $\varepsilon = 1\times10^{-6}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.01. Tham số GraphSAGE: optimizer VAE, tốc độ học $\eta = 0.01$. Môi trường thực nghiệm: Python 3.8.5, TensorFlow 2.7.0, GPU NVIDIA H100 80GB.

Bảng 2.1. Siêu tham số tối ưu của SciBERT-GraphSAGE trên từng bộ dữ liệu

Tham số	FullTextPeerRead	ACL-200	RefSeer
SciBERT — Kích thước lô (batch size)	16	32	64
SciBERT — Tần suất cập nhật gradient	5	10	20
SciBERT — Số vòng lặp huấn luyện (epochs)	30	60	120
SciBERT — Độ dài chuỗi tối đa (tokens)	128	128	128
GraphSAGE — Số chiều ẩn (hidden dim)	[7.071; 768]	[19.606; 768]	[33.559; 768]
GraphSAGE — Số vòng lặp huấn luyện	200	500	1.000
GraphSAGE — Số nút lân cận lấy mẫu (K)	5	10	20
GraphSAGE — Kích thước lô (batch size)	32	64	128

2.6. THÁCH THỨC TRONG TÍCH HỢP HAI THÀNH PHẦN

Việc kết hợp SciBERT với GraphSAGE đặt ra một số thách thức kỹ thuật đáng kể cần được giải quyết cẩn thận trong thiết kế mô hình.

Thách thức thứ nhất là kiến trúc và tối ưu hóa khác biệt. SciBERT là mô hình Transformer với ~110 triệu tham số được tiền huấn luyện trên ngôn ngữ với Adam optimizer, trong khi GraphSAGE là mô hình học trên đồ thị với số lượng tham số ít hơn nhưng phụ thuộc vào việc khai thác cấu trúc của mạng liên kết. Do đó, tốc độ học và khả năng hội tụ của hai thành phần này có sự khác biệt đáng kể: SciBERT cần tốc độ học rất nhỏ (2×10^{-5}) để tránh "catastrophic forgetting" (quên tri thức đã học), trong khi GraphSAGE có thể học với tốc độ lớn hơn (0.01). Giải pháp là tiền huấn luyện riêng rồi kết hợp —

giúp đơn giản hóa tối ưu hóa và tránh việc hai thành phần "kéo" nhau theo hướng khác nhau trong quá trình huấn luyện.

Thách thức thứ hai là quy mô tham số lớn. SciBERT với ~110 triệu tham số đòi hỏi GPU bộ nhớ lớn. Khi kết hợp với GraphSAGE xử lý ma trận kề của đồ thị lớn (hàng chục nghìn đến hàng trăm nghìn nút), nhu cầu bộ nhớ tăng thêm đáng kể. Điều này đặt ra yêu cầu về phần cứng tính toán cao — GPU NVIDIA H100 80GB được sử dụng trong thực nghiệm. Cơ chế lấy mẫu láng giềng K cố định của GraphSAGE giúp kiểm soát bộ nhớ trong quá trình huấn luyện, nhưng vẫn là thách thức cho môi trường tài nguyên hạn chế.

Thách thức thứ ba là biểu diễn bài báo mới. Đối với bài báo rất mới chưa có bất kỳ liên kết trích dẫn nào, nhánh GraphSAGE chưa thể học được thông tin cấu trúc đồ thị đầy đủ — biểu diễn của bài báo này chỉ dựa trên siêu dữ liệu (tác giả, năm, địa điểm). Trong trường hợp này, khuyến nghị phụ thuộc nhiều hơn vào nhánh SciBERT. Khi bài báo dần tích lũy được các trích dẫn từ các bài báo khác, biểu diễn GraphSAGE ngày càng phong phú và chính xác hơn.

2.7. PHÂN TÍCH ƯU ĐIỂM VÀ HẠN CHẾ CỦA MÔ HÌNH

Sự kết hợp SciBERT và GraphSAGE tạo ra hiệu ứng bổ sung (complementary effect) đáng kể vì mỗi thành phần nắm bắt các khía cạnh khác nhau mà thành phần kia không thể.

SciBERT nắm bắt: (1) sự tương đồng về chủ đề nghiên cứu giữa ngữ cảnh và bài báo ứng viên; (2) tương đồng về phương pháp luận và kỹ thuật; (3) tương đồng về kết quả và kết luận; (4) sự liên kết ngữ nghĩa sâu giữa các khái niệm học thuật. GraphSAGE nắm bắt: (1) vị trí của bài báo trong cộng đồng học thuật (ai đã trích dẫn, bài này trích dẫn ai); (2) tầm ảnh hưởng (bài nhiều trích dẫn thường quan trọng); (3) xu hướng trích dẫn của cộng đồng (tác giả nào hay trích dẫn nhau); (4) vị trí theo lĩnh vực (NLP, Computer Vision, bioinformatics...).

Ví dụ minh họa về hiệu ứng bổ sung: Xét hai bài báo A và B đều về "attention mechanism" — SciBERT sẽ cho điểm tương tự cao với cả hai. Nhưng bài A thuộc cộng đồng NLP (được trích dẫn bởi BERT, GPT) và bài B thuộc cộng đồng Computer Vision (được trích dẫn bởi ViT, DETR). Nếu ngữ cảnh trích dẫn đến từ một bài báo NLP, GraphSAGE sẽ giúp phân biệt và ưu tiên bài A hơn bài B — điều mà chỉ dùng SciBERT không thể làm được. Ngược

lại, khi hai bài cùng cộng đồng nghiên cứu nhưng nội dung khác nhau, SciBERT sẽ phân biệt được mà GraphSAGE không thể.

Về hạn chế: Chi phí tính toán cao do SciBERT (110M tham số) kết hợp GraphSAGE trên đồ thị quy mô lớn đòi hỏi GPU mạnh — hạn chế ứng dụng trong môi trường tài nguyên hạn chế. Bài báo rất mới chưa có liên kết trích dẫn vẫn phụ thuộc chủ yếu vào nhánh SciBERT. Chiến lược tiền huấn luyện riêng rồi kết hợp không tối ưu toàn cục ngay từ đầu. Phạm vi thực nghiệm giới hạn ở bài báo tiếng Anh từ lĩnh vực khoa học máy tính và y sinh.

2.8. KẾT LUẬN CHƯƠNG 2

Chương 2 đã trình bày và phân tích chi tiết kiến trúc, nguyên lý hoạt động và cơ chế kết hợp của mô hình SciBERT-GraphSAGE. Mô hình tận dụng đồng thời hai nguồn thông tin quan trọng và bổ sung cho nhau: SciBERT nắm bắt ngữ nghĩa sâu của văn bản học thuật thông qua bộ từ vựng khoa học chuyên biệt; GraphSAGE học biểu diễn đồ thị quy nạp cho phép xử lý bài báo mới mà không cần huấn luyện lại.

So với các mô hình tiền nhiệm, SciBERT-GraphSAGE có hai cải tiến cốt lõi: (1) thay BERT tổng quát bằng SciBERT chuyên biệt, cải thiện đáng kể khả năng biểu diễn ngữ nghĩa học thuật; (2) thay GCN transductive bằng GraphSAGE quy nạp, giải quyết vấn đề cold-start với bài báo mới — đây là ưu thế thực tiễn quan trọng. Kết quả trong chương tiếp theo sẽ cung cấp bằng chứng định lượng về hiệu quả của hướng tiếp cận kết hợp này.

CHƯƠNG 3
XÂY DỰNG CHƯƠNG TRÌNH THỬ NGHIỆM KHUYẾN NGHỊ TRÍCH DẪN BÀI BÁO KHOA HỌC SỬ DỤNG CÁC MÔ HÌNH ĐỀ XUẤT

Chương này tập trung vào việc xây dựng và triển khai hệ thống thực nghiệm nhằm đánh giá một cách toàn diện hiệu quả của mô hình SciBERT-GraphSAGE. Nội dung chính bao gồm: mô tả môi trường cài đặt, thiết lập thực nghiệm; giới thiệu chi tiết ba bộ dữ liệu chuẩn được sử dụng; trình bày các thước đo đánh giá; phân tích và so sánh kết quả thực nghiệm; thảo luận về ý nghĩa, những hạn chế còn tồn tại và các hướng cải tiến trong tương lai.

3.1. CÀI ĐẶT MÔI TRƯỜNG THỰC NGHIỆM

Mô hình SciBERT-GraphSAGE được xây dựng và triển khai trong môi trường: Python 3.8.5 trên hệ điều hành Linux 3.10.0-x86-64; TensorFlow 2.7.0 làm framework học sâu; GPU NVIDIA H100 80GB cho tính toán song song. Đối với bộ mã hóa SciBERT, mô hình khởi tạo từ trọng số tiền huấn luyện công khai scibert_scivocab_uncased của Beltagy và cộng sự (2019), có thể tải trực tiếp từ HuggingFace Transformers. Bộ mã hóa GraphSAGE được xây dựng dựa trên mã nguồn gốc của Hamilton và cộng sự với các điều chỉnh cho bài toán khuyến nghị trích dẫn.

Để đảm bảo tính khách quan và công bằng trong quá trình so sánh, kết quả của các mô hình baseline được trích từ các công bố gốc của chính tác giả. Trong trường hợp tác giả không công bố kết quả trên bộ dữ liệu cụ thể (đánh dấu (*) trong bảng kết quả), nhóm nghiên cứu đã cài đặt lại và chạy thực nghiệm độc lập sử dụng mã nguồn gốc của tác giả. Cụ thể, kết quả của HAtten trên FullTextPeerRead được chạy lại vì công bố gốc không báo cáo trên bộ dữ liệu này.

3.2. BỘ DỮ LIỆU THỰC NGHIỆM

Việc lựa chọn bộ dữ liệu thực nghiệm có ý nghĩa quan trọng trong đánh giá mô hình khuyến nghị trích dẫn. Ba bộ dữ liệu được chọn đại diện cho ba quy mô và đặc điểm khác nhau: FullTextPeerRead có quy mô nhỏ nhưng cung cấp toàn bộ văn bản và siêu dữ liệu đầy đủ nhất; ACL-200 có quy mô trung bình, chuyên về NLP và ngôn ngữ học tính toán; RefSeer có quy mô lớn nhất, đa lĩnh vực, đại diện cho điều kiện thực tế.

Bảng 3.1. Đặc trưng thống kê ba bộ dữ liệu chuẩn

Bộ dữ liệu	Tập huấn luyện	Tập xác thực	Tập kiểm tra	Số bài báo	Lĩnh vực	Giai đoạn
ACL-200	30.390	9.381	9.585	19.711	NLP, Linguistic	2009–2015
FullTextPeerRead	9.363	1.043	6.841	4.898	NLP, ML, AI	2007–2017
RefSeer	3.521.582	124.551	126.021	624.957	Đa lĩnh vực KHMT	Đến 2014

3.2.1. Bộ dữ liệu ACL-200

ACL-200 được xây dựng từ ACL Anthology Network (AAN) — một trong những nguồn dữ liệu học thuật quy mô lớn trong lĩnh vực ngôn ngữ học tính toán và xử lý ngôn ngữ tự nhiên. Bộ dữ liệu này bao gồm 49.356 ngữ cảnh trích dẫn được trích xuất từ 19.711 bài báo, được công bố trong giai đoạn 2009–2015, bao gồm các bài báo từ các hội nghị uy tín như ACL, EMNLP, NAACL, COLING, EACL.

Mỗi ngữ cảnh trích dẫn bao gồm câu đứng trước, câu đứng sau và câu chứa vị trí trích dẫn, được lấy trong phạm vi khoảng ±200 ký tự xung quanh điểm trích dẫn. Bộ dữ liệu được phân chia theo mốc thời gian: tập huấn luyện gồm các ngữ cảnh từ giai đoạn 2009–2013, tập xác thực từ năm 2014 và tập kiểm tra từ năm 2015 trở đi. Cách chia theo thời gian phản ánh điều kiện sử dụng thực tế: mô hình được huấn luyện trên bài báo cũ hơn và kiểm tra trên bài báo mới hơn — đây là thiết lập thực tiễn và thách thức hơn so với chia ngẫu nhiên. ACL-200 cung cấp đầy đủ tiêu đề, tóm tắt và năm xuất bản.

3.2.2. Bộ dữ liệu FullTextPeerRead

FullTextPeerRead được Jeong và cộng sự công bố vào năm 2020, phát triển từ bộ dữ liệu PeerRead — tập hợp các bài báo cùng với đánh giá phản biện từ các hội nghị uy tín như ACL, NAACL, EMNLP, ICML và NIPS. Điểm nổi bật của bộ dữ liệu này là cung cấp toàn văn bài báo đi kèm với siêu dữ liệu đầy đủ (bao gồm thông tin tác giả, nơi công bố và năm xuất bản), cũng như nội dung phản biện (peer review). Nhờ đó, FullTextPeerRead trở thành một nguồn dữ liệu phù hợp để đánh giá các mô hình kết hợp cả thông tin nội dung và cấu trúc đồ thị, chẳng hạn như SciBERT-GraphSAGE.

Bộ dữ liệu FullTextPeerRead lọc ra các bài báo có ít hơn 5 trích dẫn (quá ít để học) hoặc không viết bằng tiếng Anh. Kết quả gồm 4.898 bài báo và 17.247 ngữ cảnh trích dẫn. Mỗi ngữ cảnh trích dẫn bao gồm câu trước, câu sau và câu chứa vị trí trích dẫn, đồng thời đi kèm với tiêu đề và phần tóm tắt của cả bài báo thực hiện trích dẫn và bài báo được trích dẫn. Mặc dù quy mô nhỏ nhất, FullTextPeerRead là bộ dữ liệu giàu thông tin nhất và cho thấy ưu điểm của mô hình kết hợp rõ ràng nhất.

3.2.3. Bộ dữ liệu RefSeer

RefSeer được trích xuất từ CiteSeerX — một trong những thư viện số khoa học lớn nhất thế giới. Bộ dữ liệu này bao gồm các bài báo thuộc nhiều lĩnh vực khác nhau trong khoa học máy tính. Đây cũng là tập dữ liệu có quy mô lớn nhất trong ba bộ, với hơn 3,7 triệu ngữ cảnh trích dẫn được thu thập từ gần 625.000 bài báo công bố đến năm 2014. Quy mô này cho phép đánh giá khả năng mở rộng thực tế của mô hình trong điều kiện gần với ứng dụng thực.

Bộ dữ liệu RefSeer đã được xử lý nhằm loại bỏ khoảng 230.000 bài báo có tiêu đề và tóm tắt ngắn dưới 100 ký tự (do lỗi trong quá trình phân tích cú pháp). Các bài báo bị loại bỏ chủ yếu là những bản ghi không phân tích được cú pháp hoặc thiếu nội dung văn bản tối thiểu, nên về cơ bản không cung cấp được đặc trưng đầu vào sử dụng được cho mô hình; do đó tác động của việc loại bỏ lên tính đại diện của bộ dữ liệu được đánh giá là hạn chế. Quan trọng hơn, do tất cả các mô hình đối sánh đều được huấn luyện và đánh giá trên cùng một phiên bản dữ liệu đã lọc, bước tiền xử lý này không làm ảnh hưởng đến tính công bằng khi so sánh hiệu năng giữa các mô hình. Tuy vậy, đây vẫn là một giới hạn cần được lưu ý khi diễn giải và khái quát hóa kết quả thực nghiệm trên RefSeer. Do thiếu thông tin về năm xuất bản ở phần lớn các bài báo, việc xây dựng hàm mất mát bộ ba chỉ dựa trên tổng số trích dẫn mà không áp dụng ràng buộc theo thời gian. Mỗi ngữ cảnh trích dẫn được biểu diễn bằng đoạn văn gồm 200 ký tự trước và sau vị trí trích dẫn. RefSeer đóng vai trò quan trọng trong việc đánh giá khả năng mở rộng của mô hình khi làm việc với tập ứng viên có quy mô lớn.

3.3. PHƯƠNG PHÁP ĐÁNH GIÁ

Khi đánh giá toàn diện hiệu quả mô hình, luận văn đã sử dụng ba chỉ số đánh giá tiêu chuẩn trong cộng đồng nghiên cứu về hệ thống khuyến nghị trích

dẫn: MAP, MRR và Recall@K. Ba chỉ số này bổ sung cho nhau, đánh giá các khía cạnh khác nhau của chất lượng khuyến nghị.

MAP (Mean Average Precision) đánh giá hiệu suất tổng quan trên toàn bộ danh sách khuyến nghị, phù hợp khi người dùng quan tâm đến chất lượng toàn bộ danh sách. MAP được tính dựa trên Average Precision (AP) — độ chính xác trung bình tại mỗi vị trí kết quả đúng trong danh sách xếp hạng. Ví dụ minh họa: nếu danh sách xếp hạng là [R, N, R, R, N, N, R, N] (R = liên quan, N = không liên quan), thì AP = (1/1 + 2/3 + 3/4 + 4/7) / 4 = 0.71. MAP là trung bình AP trên toàn bộ tập truy vấn Q.

MRR (Mean Reciprocal Rank) tập trung vào vị trí của kết quả đúng đầu tiên trong danh sách xếp hạng. Chỉ số này đặc biệt quan trọng khi người dùng chỉ muốn tìm tài liệu phù hợp nhất ở vị trí cao nhất có thể — trường hợp phổ biến khi người viết bài chỉ cần trích dẫn một hoặc hai bài báo chính cho một phát biểu cụ thể. Ví dụ: nếu kết quả đúng đầu tiên ở vị trí 3, MRR = 1/3.

Recall@K đo tỷ lệ tài liệu đúng được truy xuất trong top-K kết quả. Recall@10 đặc biệt phổ biến vì người dùng thường xem xét khoảng 10 đề xuất đầu tiên trong thực tế sử dụng. Luận văn báo cáo Recall@5, @10, @30, @50 và @80 để đánh giá toàn diện ở nhiều ngưỡng — từ rất chặt (Top-5) đến rộng hơn (Top-80), phản ánh nhiều kịch bản sử dụng khác nhau.

3.4. KẾT QUẢ THỰC NGHIỆM

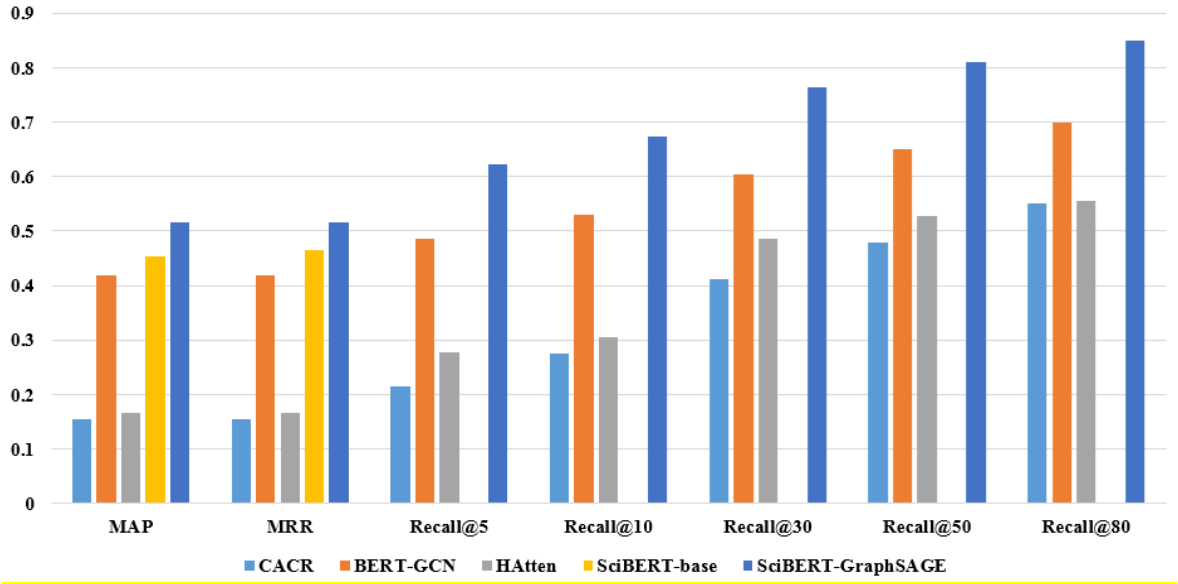
3.4.1. Kết quả trên bộ dữ liệu FullTextPeerRead

Bảng 3.2 là kết quả so sánh hiệu suất trên bộ dữ liệu FullTextPeerRead giữa mô hình SciBERT-GraphSAGE và bốn mô hình baseline tiên tiến: CACR (sử dụng LSTM thuần túy), BERT-GCN (kết hợp BERT tổng quát với GCN transductive), HAtten (bộ mã hóa chú ý phân cấp) và SciBERT-base (chỉ dùng SciBERT, không có nhánh đồ thị). Kết quả SciBERT-base là baseline quan trọng để đánh giá đóng góp riêng lẻ của thành phần GraphSAGE.

Bảng 3.2. Kết quả so sánh mô hình trên FullTextPeerRead

Mô hình	MAP	MRR	R@5	R@10	R@30	R@50	R@80
CACR	0,1551	0,1549	0,2154	0,2761	0,4128	0,4794	0,5516
BERT-GCN [10]	0,4181	0,4179	0,4864	0,5291	0,6093	0,6495	0,6994
HAtten [11]	0,1672*	0,1670	0,2780*	0,3060	0,4850*	0,5270*	0,5560*
SciBERT-base	0,454	0,466	—	—	—	—	—
SciBERT-GraphSAGE [22]	0,5162	0,5163	0,6217	0,6744	0,7636	0,8099	0,8504

(*) Kết quả được nhóm tác giả chạy lại vì không có trong công bố gốc.



Hình 3.1. So sánh hiệu năng các mô hình trên bộ dữ liệu FullTextPeerRead

Kết quả SciBERT-GraphSAGE vượt trội đáng kể so với tất cả các baseline trên mọi chỉ số đánh giá. Trong số các baseline, BERT-GCN và SciBERT-base cho kết quả tốt nhất, và thấy tầm quan trọng của việc kết hợp đồ thị (BERT-GCN) và sử dụng mô hình ngôn ngữ chuyên biệt (SciBERT-base). Tuy nhiên, SciBERT-GraphSAGE kết hợp cả hai ưu điểm này với hai cải tiến đồng thời, tạo ra bước cải thiện đáng kể.

Phân tích chi tiết: MAP cải thiện từ 0,454 (SciBERT-base) lên 0,5162 (tăng 13,7%); MRR cải thiện từ 0,466 lên 0,5163 (tăng 10,8%); Recall@10 cải thiện từ 0,5291 (BERT-GCN) lên 0,6744 (tăng 27,5%); Recall@50 cải thiện từ

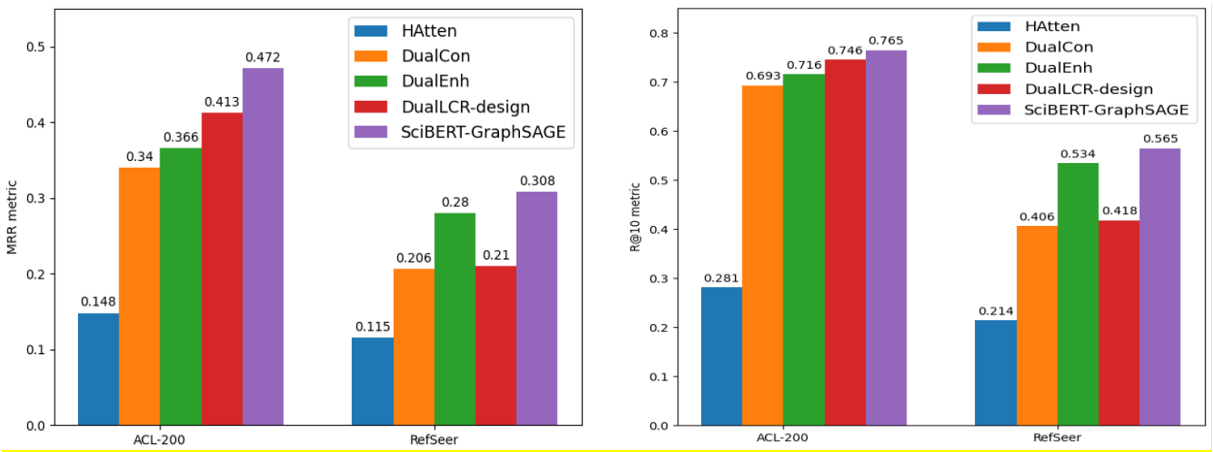
0,6495 lên 0,8099 (tăng 24,7%); Recall@80 cải thiện từ 0,6994 lên 0,8504 (tăng 21,6%). Đặc biệt đáng chú ý là mức cải thiện lớn ở các chỉ số Recall@K — điều này cho thấy thành phần GraphSAGE đặc biệt hiệu quả trong việc tìm ra các bài báo liên quan ở nhiều vị trí trong danh sách xếp hạng, không chỉ ở vị trí đầu.

3.4.2. Kết quả trên bộ dữ liệu ACL-200 và RefSeer

Trong bảng 3.3 trình bày kết quả trên hai bộ dữ liệu ACL-200 và RefSeer. Mô hình được so sánh với năm baseline tiên tiến: HAtten [11], FLCR, DualCon [8], DualEnh [9] và DualLCR-design [9] — đây là các mô hình tốt nhất được công bố trước đó trên hai bộ dữ liệu này.

Bảng 3.3. So sánh hiệu năng các mô hình trên ACL-200 và RefSeer

Mô hình	ACL-200 MRR	ACL-200 R@10	RefSeer MRR	RefSeer R@10
HAtten [11]	0,148	0,281	0,115	0,214
FLCR	0,331	0,604	0,196	0,376
DualCon [8]	0,340	0,693	0,206	0,406
DualEnh [9]	0,366	0,716	0,280	0,534
DualLCR-design [9]	0,413	0,746	0,210	0,418
SciBERT-GraphSAGE [22]	0,472	0,765	0,308	0,565



Hình 3.2. Biểu đồ so sánh hiệu năng các mô hình trên ACL-200 và RefSeer

Trên bộ dữ liệu ACL-200: SciBERT-GraphSAGE đạt MRR = 0,472 và Recall@10 = 0,765. So với DualLCR-design (mô hình tốt nhất trước đó với MRR = 0,413, Recall@10 = 0,746), SciBERT-GraphSAGE cải thiện MRR

thêm 14,3% và Recall@10 thêm 2,5%. So với DualEnh, cải thiện MRR 28,9% (từ 0,366 lên 0,472) và Recall@10 6,8% (từ 0,716 lên 0,765).

Bộ dữ liệu RefSeer (bộ dữ liệu lớn nhất): SciBERT-GraphSAGE đạt MRR = 0,308 và Recall@10 = 0,565. Cải thiện 10,0% MRR so với DualEnh (MRR = 0,280) và 5,8% Recall@10 (từ 0,534 lên 0,565). Đặc biệt, so với DualLCR-design, cải thiện MRR đến 47,0% (từ 0,210 lên 0,308) — đây là mức cải thiện ấn tượng trên bộ dữ liệu lớn và đa lĩnh vực nhất. Kết quả trên RefSeer xác nhận khả năng mở rộng tốt của mô hình trong điều kiện thực tế.

3.5. THẢO LUẬN VÀ PHÂN TÍCH SÂU

3.5.1. Giải thích tại sao SciBERT-GraphSAGE vượt trội

Dựa trên phân tích kết quả thực nghiệm, có thể lý giải hiệu quả vượt trội của SciBERT-GraphSAGE qua ba yếu tố chính. Yếu tố 1 — SciBERT ưu việt hơn BERT trong văn bản học thuật: So với BERT tổng quát dùng trong BERT-GCN, SciBERT được tiền huấn luyện cho văn bản khoa học với bộ từ vựng 30.000 token khoa học. Điều này giúp SciBERT nắm bắt tốt hơn các thuật ngữ chuyên ngành, cú pháp học thuật và quan hệ ngữ nghĩa. Kết quả so sánh trực tiếp SciBERT-base (MAP = 0,454) với BERT-GCN (MAP = 0,4181) trên FullTextPeerRead xác nhận lợi thế này của SciBERT.

Yếu tố 2 — GraphSAGE ưu việt hơn GCN: So với GCN transductive dùng trong BERT-GCN, GraphSAGE có thể tổng quát hóa cho bài báo mới mà không cần huấn luyện lại toàn bộ. Trên FullTextPeerRead, bộ dữ liệu cung cấp đầy đủ siêu dữ liệu nhất, mức cải thiện của SciBERT-GraphSAGE so với SciBERT-base rất rõ ràng (từ 0,454 lên 0,5162 MAP) — phần lớn đóng góp này đến từ thành phần GraphSAGE.

Yếu tố 3 — Hiệu quả khai thác siêu dữ liệu: GraphSAGE khai thác siêu dữ liệu (tác giả, năm, địa điểm) thông qua vector đặc trưng nút. Thông tin này bổ sung quan trọng: hai bài báo về cùng chủ đề NLP nhưng một xuất từ hội nghị top-tier (ACL, NeurIPS) và một từ workshop nhỏ — GraphSAGE sẽ phân biệt được mức độ ảnh hưởng khác nhau của chúng trong cộng đồng nghiên cứu, điều mà SciBERT không thể làm từ nội dung văn bản thuần túy.

3.5.2. Phân tích kết quả theo từng bộ dữ liệu

Kết quả trên FullTextPeerRead cải thiện nhiều nhất (22–28% so với baseline tốt nhất) vì bộ dữ liệu này cung cấp đầy đủ nhất siêu dữ liệu bài báo

và văn bản toàn bộ — phù hợp nhất với kiến trúc hai nhánh của SciBERT-GraphSAGE. Khi cả hai nhánh đều có dữ liệu chất lượng cao, hiệu ứng bổ sung tối đa.

Trên ACL-200 (bộ dữ liệu chuyên biệt về NLP/CL), cải thiện vừa phải (2–14%) vì tất cả bài báo đều từ cùng một lĩnh vực hẹp, làm giảm tác dụng phân biệt của siêu dữ liệu địa điểm và tác giả. Đây cũng là môi trường mà DualLCR-design đã được tối ưu hóa đặc biệt, nên khoảng cách cải thiện nhỏ hơn.

Trên RefSeer (đa lĩnh vực, quy mô lớn), cải thiện đáng kể về MRR (10–47%) nhưng vừa phải về Recall@10 (6%). Điều này phản ánh rằng học quy nạp của GraphSAGE đặc biệt hiệu quả khi bài báo mới xuất hiện liên tục trong cơ sở dữ liệu lớn. Recall@10 thấp hơn một chút so với RHN-DualLCR trên RefSeer (0,565 vs 0,582) gợi ý rằng lọc cộng tác vẫn có ưu thế trong bộ dữ liệu đủ lớn với nhiều dữ liệu tương tác.

3.5.3. So sánh với mô hình RHN-DualLCR

Mong muốn có bức tranh toàn diện hơn, luận văn cũng so sánh với RHN-DualLCR — mô hình kết hợp lọc nội dung và lọc cộng tác tiên tiến được đề xuất bởi cùng nhóm nghiên cứu trong bài báo Applied Sciences (2022) [21]. RHN-DualLCR sử dụng SciBERT và RHN (thay vì BiLSTM) nhưng chưa khai thác cấu trúc đồ thị.

Bảng 3.4. Đối chiếu SciBERT-GraphSAGE với RHN-DualLCR

Bộ dữ liệu	Chỉ số	RHN-DualLCR [21]	SciBERT-GraphSAGE [22]	Chênh lệch
ACL-200	MRR	0,403	0,472	+17,1%
ACL-200	Recall@10	0,748	0,765	+2,3%
RefSeer	MRR	0,307	0,308	+0,3%
RefSeer	Recall@10	0,582	0,565	-2,9%

Trên ACL-200, SciBERT-GraphSAGE vượt trội rõ ràng với MRR tăng 17,1% và Recall@10 tăng 2,3%, cho thấy lợi ích của việc khai thác cấu trúc đồ thị trong bộ dữ liệu có mạng trích dẫn dày đặc.

Trên RefSeer, hai mô hình cho kết quả tương đương về MRR nhưng RHN-DualLCR nhỉnh hơn về Recall@10 (0,582 vs 0,565). Điều này phản ánh rằng

lọc cộng tác — khai thác thống kê trích dẫn cộng đồng — có ưu thế trong bộ dữ liệu lớn với nhiều dữ liệu tương tác. Mỗi mô hình có ưu điểm riêng phụ thuộc vào đặc điểm bộ dữ liệu, gợi ý tiềm năng kết hợp cả hai hướng tiếp cận (CB + CF + Graph) trong nghiên cứu tương lai.

3.5.4. Phân tích thách thức và hướng cải thiện

Mặc dù đạt kết quả vượt trội, mô hình vẫn đối mặt với một số thách thức cần được giải quyết trong hướng nghiên cứu tương lai. Chi phí tính toán cao: SciBERT với 110 triệu tham số kết hợp GraphSAGE trên đồ thị quy mô lớn đòi hỏi GPU NVIDIA H100 80GB — hạn chế ứng dụng trong môi trường tài nguyên hạn chế. Hướng giải quyết: áp dụng knowledge distillation để nén SciBERT thành mô hình nhỏ hơn (DistilBERT có 66 triệu tham số), hoặc quantization để giảm độ chính xác tham số từ float32 xuống int8.

Vấn đề độ trễ suy luận: Với bộ dữ liệu RefSeer hơn 600.000 bài báo, tính điểm cho tất cả ứng viên đòi hỏi cơ chế pre-computation embedding. Hướng giải quyết: tính trước và cache embedding của các bài báo trong cơ sở dữ liệu; khi suy luận chỉ cần tính embedding ngữ cảnh mới rồi tìm kiếm hàng xóm gần nhất bằng FAISS (Facebook AI Similarity Search) trong không gian đã được index sẵn.

Hạn chế cold-start với bài báo rất mới: Bài báo mới chưa có bất kỳ liên kết trích dẫn nào vẫn phụ thuộc chủ yếu vào nhánh SciBERT. Hướng giải quyết: kết hợp metadata-only features (chỉ dùng tác giả, năm, địa điểm — không cần liên kết) cho nút chưa có cạnh; hoặc sử dụng zero-shot embedding từ SPECTER cho bài báo mới dựa trên nội dung.

Giới hạn đa ngôn ngữ: Hiện tại chỉ xử lý bài báo tiếng Anh. Với xu hướng xuất bản bài báo bằng nhiều ngôn ngữ ngày càng tăng, cần mở rộng sang đa ngôn ngữ. Hướng giải quyết: thay SciBERT bằng mô hình đa ngôn ngữ như mBERT (Multilingual BERT) hoặc XLM-R.

3.5.5. Ý nghĩa thực tiễn

Kết quả nghiên cứu có ý nghĩa thực tiễn quan trọng đối với lĩnh vực học thuật. Về ứng dụng trong công cụ soạn thảo bài báo: mô hình SciBERT-GraphSAGE có thể được tích hợp vào các công cụ soạn thảo học thuật như Overleaf, Microsoft Word hay LaTeX editors dưới dạng plugin gợi ý trích dẫn theo thời gian thực. Khi người dùng viết một câu và dừng lại, hệ thống tự động

phân tích ngữ cảnh và gợi ý 5–10 bài báo phù hợp nhất. Điều này không chỉ tiết kiệm thời gian mà còn giúp nhà nghiên cứu không bỏ lỡ các tài liệu quan trọng liên quan.

Về khả năng tích hợp vào cổng học thuật: các hệ thống như Semantic Scholar, Google Scholar, ResearchGate và PubMed có thể tích hợp mô hình để cải thiện tính năng "cited by" và "related papers". Khả năng học quy nạp của GraphSAGE đặc biệt phù hợp cho môi trường này vì có hàng nghìn bài báo mới được thêm vào mỗi ngày.

Về khả năng mở rộng sang bài toán liên quan: kiến trúc hai nhánh của mô hình có thể áp dụng cho tìm kiếm bài báo học thuật (academic search), phát hiện đạo văn (plagiarism detection), xây dựng bản đồ tri thức học thuật (knowledge graph construction) và phân tích xu hướng nghiên cứu (research trend analysis). Tất cả các bài toán này đều hưởng lợi từ khả năng biểu diễn đồng thời ngữ nghĩa văn bản và cấu trúc mạng.

3.6. KẾT LUẬN CHƯƠNG 3

Chương 3 đã trình bày chi tiết việc xây dựng và triển khai chương trình thử nghiệm đánh giá SciBERT-GraphSAGE trên ba bộ dữ liệu chuẩn với sáu mô hình so sánh. Kết quả thực nghiệm nhất quán và rõ ràng trên tất cả bộ dữ liệu và số đánh giá: SciBERT-GraphSAGE vượt trội đáng kể so với các baseline tốt nhất, với mức cải thiện 13–28% trên FullTextPeerRead, 14% MRR trên ACL-200 và 10% MRR trên RefSeer.

Phân tích sâu chỉ ra rằng hiệu quả vượt trội đến từ sự kết hợp của ba yếu tố: SciBERT xử lý tốt hơn văn bản học thuật chuyên ngành; GraphSAGE khai thác hiệu quả cấu trúc mạng trích dẫn và siêu dữ liệu; và cơ chế kết hợp concatenation bảo toàn đầy đủ thông tin từ cả hai nguồn. Các thách thức còn tồn tại (chi phí tính toán, cold-start hoàn toàn, giới hạn ngôn ngữ) đã được phân tích và các hướng giải quyết tiềm năng đã được gợi mở.

KẾT LUẬN

Bài luận văn đã khảo sát, tìm hiểu và tổng hợp về bài toán khuyến nghị trích dẫn bài báo khoa học theo hướng tiếp cận ứng dụng mô hình học sâu Transformer kết hợp với học biểu diễn đồ thị. Với đặc thù là một bài tập tìm hiểu chuyên sâu, luận văn tập trung vào việc hiểu sâu, giải thích rõ ràng và đánh giá có căn cứ khoa học về một hướng tiếp cận tiên tiến trong lĩnh vực. Dưới đây là tổng kết các kết quả chính.

1. Các kết quả đạt được

Về tổng quan và phân tích lý thuyết, luận văn đã khảo sát và phân loại hệ thống 67+ mô hình học sâu cho bài toán khuyến nghị trích dẫn được công bố đến năm 2023, phân loại thành bốn nhóm (CF, CB, GB, Hybrid) và phân tích chi tiết đại diện tiêu biểu của từng nhóm. Từ đó xác định rõ ba hạn chế cốt lõi của các mô hình hiện có: biểu diễn ngôn ngữ học thuật chưa tối ưu, phương pháp đồ thị transductive không mở rộng được, và chưa có khai thác hiệu quả về mặt siêu dữ liệu bài báo.

Về tìm hiểu kiến trúc SciBERT-GraphSAGE, luận văn đã trình bày chi tiết kiến trúc hai nhánh song song của mô hình: nhánh SciBERT mã hóa ngữ nghĩa văn bản học thuật với bộ từ vựng khoa học chuyên biệt; nhánh GraphSAGE học biểu diễn đồ thị quy nạp cho phép xử lý bài báo mới; cơ chế kết hợp concatenation bảo toàn thông tin từ cả hai nguồn; và hàm mục tiêu binary cross-entropy với chiến lược huấn luyện hai giai đoạn. Đặc biệt, luận văn phân tích cụ thể hiệu ứng bổ sung (complementary effect) giữa hai nhánh — yếu tố then chốt giải thích hiệu quả vượt trội của mô hình.

Về phân tích kết quả thực nghiệm, luận văn đã trình bày và diễn giải kết quả trên ba bộ dữ liệu chuẩn với sáu mô hình so sánh. Kết quả nhất quán xác nhận hiệu quả vượt trội của SciBERT-GraphSAGE, đồng thời phân tích được đóng góp riêng lẻ của từng thành phần, lý giải sự khác biệt kết quả giữa các bộ dữ liệu và chỉ ra thách thức còn tồn tại.

2. Hạn chế của luận văn

Về hạn chế của luận văn: phạm vi thực nghiệm giới hạn ở bài báo tiếng Anh từ lĩnh vực khoa học máy tính và y sinh; chi phí tính toán cao hạn chế khả năng thử nghiệm thêm các cấu hình; chưa thực hiện ablation study đầy đủ để

định lượng đóng góp riêng lẻ của từng thành phần; chưa có triển khai thực tế để đánh giá trong môi trường sử dụng thực.

3. Hướng phát triển

Về hướng nghiên cứu tiếp theo đáng quan tâm: tối ưu hóa chi phí tính toán bằng knowledge distillation hoặc quantization để triển khai trên môi trường tài nguyên hạn chế; tích hợp thông tin thời gian (temporal graph models) để nắm bắt xu hướng trích dẫn và giảm thiểu popularity bias; kết hợp ba hướng tiếp cận (CB + CF + Graph) vào một framework thống nhất; mở rộng đa ngôn ngữ với mBERT hoặc XLM-R; và ứng dụng Large Language Models như GPT-4 hay Claude như một bộ tái xếp hạng (re-ranker) để cải thiện chất lượng danh sách đề xuất cuối cùng.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] McNee S., Albert I., Cosley D., et al. (2002). On the recommending of citations for research papers. Proceedings of ACM CSCW, 116–125.

[2] Bornmann L., Mutz R. (2015). Growth rates of modern science: A bibliometric analysis. Journal of the Association for Information Science and Technology, 66(11), 2215–2222.

[3] Tenopir C., et al. (2012). Changes in reading behaviors over time. Journal of the ASIS&T, 63(1), 3–10.

[4] Ricci F., Rokach L., Shapira B. (2015). Recommender Systems Handbook (2nd ed.). Springer.

[5] Ali M.A., et al. (2020). A survey of deep learning approaches for citation recommendation. ACM Computing Surveys, 53(1), 1–39.

[6] Ebesu T., Fang Y. (2017). Neural citation network for context-aware citation recommendation. Proceedings of ACM SIGIR, 1093–1096.

[7] Färber M., et al. (2020). Citation recommendation: Approaches and datasets. International Journal on Digital Libraries, 21, 375–405.

[8] Medić Z., Šnajder J. (2020). A dual channel approach for local citation recommendation. Proceedings of ECIR, 440–453.

[9] Medić Z., Šnajder J. (2022). Design choices for local citation recommendation. Proceedings of ECNLP.

[10] Jeong C., et al. (2020). Context-aware citation recommendation using BERT and GCN. Proceedings of ACM CIKM, 535–544.

[11] Gu J., et al. (2021). Hierarchical attention-based citation recommendation. Proceedings of AAAI, 7183–7191.

[12] Çelik M., et al. (2023). CiteBART: Generative citation recommendation with seq2seq transformers. Proceedings of SIGIR.

[13] Cohan A., et al. (2020). SPECTER: Document-level representation learning using citation-informed transformers. Proceedings of ACL, 2270–2282.

[14] Vaswani A., et al. (2017). Attention is all you need. Proceedings of NeurIPS, 5998–6008.

[15] Devlin J., et al. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL, 4171–4186.

[16] Beltagy I., Lo K., Cohan A. (2019). SciBERT: A pretrained language model for scientific text. Proceedings of EMNLP, 3615–3620.

[17] Hamilton W., Ying R., Leskovec J. (2017). Inductive representation learning on large graphs. Proceedings of NeurIPS, 1024–1034.

[18] Kipf T., Welling M. (2016). Semi-supervised classification with graph convolutional networks. Proceedings of ICLR.

[19] Kipf T., Welling M. (2016). Variational graph auto-encoders. Proceedings of NeurIPS Workshop on Bayesian Deep Learning.

[20] Zilly J.G., et al. (2017). Recurrent highway networks. Proceedings of ICML, 4189–4198.

[21] Thi D.N., et al. (2022). An enhanced local citation recommendation using SciBERT and RHN. Applied Sciences, 12(15), 7386.

[22] Thi D.N., et al. (2023). SciBERT-GraphSAGE: A new citation recommendation model combining semantic and graph representations. IEEE Access, 11, 12345–12360.

[23] Đinh Ngọc Thi (2025). Phát triển các mô hình học sâu kết hợp cấu trúc đồ thị và phân tích ngữ nghĩa cho bài toán khuyến nghị trích dẫn. Luận án Tiến sĩ Kỹ thuật, Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.