LỌC CỘNG TÁC TĂNG CƯỜNG NGỮ NGHĨA CHO HỆ THỐNG GỢI Ý PHIM BẰNG MÔ HÌNH NGÔN NGỮ LỚN

Phạm Thị Thu Trang[1*], Đặng Khánh Hòa[1], Nguyễn Hoàng[1], Nguyễn Vũ Sơn[1]
[*]Tác giả liên hệ, email: ptttrang@hou.edu.vn. ORCID: 0009-0006-7405-4046

Ngày tòa soạn nhận được bài báo: 15/01/2026
Ngày phản biện đánh giá: 18/03/2026
Ngày bài báo được duyệt đăng: 14/04/2026

DOI: 10.59266/houjs.2026.1181

Tóm tắt: *Các hệ gợi ý truyền thống thường gặp khó khăn do tính thưa của dữ liệu và khả năng hiểu ngữ nghĩa hạn chế đối với sở thích người dùng cũng như nội dung của mục. Để giải quyết các thách thức này, bài báo đề xuất một khung gợi ý lai mới tích hợp lọc cộng tác với các đặc trưng ngữ nghĩa được rút ra từ siêu dữ liệu phim và các mô tả do mô hình ngôn ngữ lớn (LLM) sinh ra. Chúng tôi sử dụng một LLM huấn luyện sẵn để tự động tạo ra nội dung văn bản phong phú cho phim, sau đó biểu diễn bằng fastText nhằm tăng cường cách biểu diễn mục. Các embedding ngữ nghĩa này được kết hợp với dữ liệu tương tác người dùng - mục để tạo ra các gợi ý chính xác và cá nhân hóa hơn. Kết quả thực nghiệm trên bộ dữ liệu MovieLens-20M cho thấy mô hình đề xuất vượt trội đáng kể so với các phương pháp truyền thống theo các chỉ số RMSE, Precision@5 và Recall@5. Các phát hiện này nhấn mạnh tiềm năng của LLM và tăng cường văn bản trong việc cải thiện hiệu quả của các hệ gợi ý.*

Từ khóa: *trí tuệ nhân tạo, hệ thống gợi ý, lọc cộng tác, mô hình ngôn ngữ lớn, nhúng ngữ nghĩa*

I. Đặt vấn đề

Sự bùng nổ của nội dung số đã khiến các hệ gợi ý trở thành thành phần không thể thiếu để điều hướng trong những danh mục mục tin khổng lồ, qua đó nâng cao đáng kể trải nghiệm người dùng trên nhiều nền tảng trực tuyến. Trong lĩnh vực gợi ý phim, các hệ thống này đặc biệt quan trọng trong việc dẫn dắt người dùng tới những lựa chọn phù hợp với sở thích của họ, từ đó thúc đẩy mức độ gắn kết lâu dài. Các cách tiếp cận truyền thống, nổi bật là lọc cộng tác (Collaborative Filtering - CF) (Linden và cộng sự, 2003) và lọc dựa trên nội dung (Content-Based Filtering - CBF) (Pazzani & Billsus, 2007), đã chứng minh hiệu quả đáng kể. Các phương pháp CF khai thác các mẫu tương tác người dùng - mục, nhưng thường gặp thách thức liên quan đến tính thưa dữ liệu và bài toán "cold-start" (Su & Khoshgoftaar, 2009), đặc biệt khi người dùng hoặc mục mới thiếu lịch sử tương tác đầy đủ. Ngược lại,

[1] Khoa Điện - Điện tử, Trường Đại học Mở Hà Nội, Hà Nội, Việt Nam

các phương pháp dựa trên nội dung gợi ý mục dựa trên các thuộc tính nội tại của mục và hồ sơ người dùng, nhờ đó có thể giảm bớt một số hạn chế của CF (Lops và cộng sự, 2011). Tuy nhiên, CBF thường phụ thuộc vào các đặc trưng được tuyển chọn thủ công hoặc xác định tường minh; những đặc trưng này có thể không nắm bắt được sự phong phú ngữ nghĩa hay chủ đề tinh tế của các mục phức tạp hơn (Van Meteren & Van Someren, 2000).

Gần đây, các mô hình ngôn ngữ lớn (LLMs) nổi lên như công cụ mạnh mẽ (Brown và cộng sự, 2020; Vaswani và cộng sự, 2017) với khả năng hiểu và tạo sinh ngôn ngữ tự nhiên ở mức độ tinh vi chưa từng có. Mặc dù việc tích hợp LLM vào hệ gợi ý vẫn ở giai đoạn đầu (Fan và cộng sự, 2023), các nghiên cứu đã cho thấy LLM hiệu quả trong việc làm giàu đặc trưng mục (Sun & Zhang, 2021), dự đoán sở thích người dùng (Geng và cộng sự, 2022) và mã hóa tương tác qua siêu dữ liệu văn bản (Bao và cộng sự, 2023). Tuy nhiên, số lượng công trình tập trung vào tổng hợp mô tả mục toàn diện bằng cách hợp nhất siêu dữ liệu có cấu trúc với tri thức thế giới mở còn tương đối ít.

Trong nghiên cứu này, chúng tôi đề xuất phương pháp tăng cường CF mục - mục thông qua sinh mô tả phim bằng LLM. Cách tiếp cận tận dụng dữ liệu Tag Genome của MovieLens cùng tri thức từ IMDb để nhắc lệnh LLM sinh các đoạn tường thuật mạch lạc. Các mô tả được chuyển thành embedding dày đặc bằng mô hình embedding câu huấn luyện sẵn (Reimers & Gurevych, 2019), đóng vai trò biểu diễn mục tăng cường cho việc tính toán độ tương đồng.

Đóng góp chính là tích hợp các tường thuật văn bản do LLM sinh vào pipeline CF nhằm làm giàu hồ sơ mục, cải thiện độ chính xác gợi ý (Vargas & Castells, 2011) và tăng tính diễn giải (Zhang & Chen, 2020).

Ngoài CF và CBF, các kỹ thuật học sâu như Neural Collaborative Filtering, Autoencoders và Transformer đã cải thiện hệ gợi ý đáng kể (Brown và cộng sự, 2020), nhưng chủ yếu khai thác tín hiệu tương tác mà chưa tận dụng hiệu quả tri thức ngữ nghĩa từ nội dung văn bản.

II. Cơ sở lý thuyết

2.1. Mô hình ngôn ngữ lớn (LLMs)

Mô hình Ngôn ngữ Lớn (LLMs) là các kiến trúc học sâu tiên tiến, thường dựa trên Transformer (Vaswani và cộng sự, 2017), được huấn luyện trên những kho ngữ liệu văn bản khổng lồ (Devlin và cộng sự, 2019). Quá trình huấn luyện này giúp chúng có khả năng diễn giải, tạo sinh và dự đoán ngôn ngữ với mức độ mạch lạc cao. Các đặc trưng nổi bật gồm quy mô hàng tỷ tham số (Brown và cộng sự, 2020), khả năng tiền huấn luyện và tinh chỉnh theo miền, năng lực tạo sinh văn bản phù hợp ngữ cảnh, học few-shot/zero-shot, và tạo embedding vector dày đặc cho văn bản.

Một số mô hình LLM tiêu biểu hiện nay bao gồm: dòng GPT của OpenAI (GPT-3.5, GPT-4) với khả năng sinh văn bản và suy luận đa bước vượt trội; dòng mô hình mã nguồn mở LLaMA của Meta và các biến thể tinh chỉnh như LLaMA-2, Vicuna; dòng Gemini của Google với khả năng đa phương thức tích hợp văn bản, hình ảnh và âm thanh; Claude của Anthropic tối ưu cho các tác vụ đòi hỏi tuân thủ chỉ dẫn và an toàn; và dòng Qwen của Alibaba với hỗ trợ đa ngôn ngữ mạnh. Các mô hình này khác nhau về quy mô tham số, chiến lược huấn luyện, cửa sổ

ngữ cảnh và chi phí sử dụng, tạo nên một hệ sinh thái phong phú cho các ứng dụng khác nhau. Sự xuất hiện của LLM đã thúc đẩy các bước tiến đáng kể trong nhiều lĩnh vực của trí tuệ nhân tạo, bao gồm xử lý ngôn ngữ tự nhiên, sinh mã nguồn (Chen và cộng sự, 2021) và sáng tạo nội dung. LLM có khả năng học trong ngữ cảnh và điều chỉnh đầu ra theo thiết kế prompt, tuy nhiên có thể bộc lộ hạn chế trong các miền chuyên biệt (Bommasani và cộng sự, 2021), đòi hỏi các kỹ thuật instruction tuning hoặc tinh chỉnh theo miền.

2.2. Lọc cộng tác (CF)

Lọc cộng tác (CF) tạo gợi ý bằng cách tận dụng hành vi tập thể của cộng đồng người dùng (Koren & Bell, 2011), phân tích lịch sử tương tác để dự đoán sở thích. CF gồm hai nhóm: dựa trên bộ nhớ (neighborhood-based) và dựa trên mô hình (model-based). Công trình này tập trung vào CF mục - mục dựa trên bộ nhớ.

CF dựa trên bộ nhớ xác định các "hàng xóm" (người dùng hoặc mục tương tự) dựa trên tương tác trong quá khứ.

CF dựa trên người dùng (User-User): Xác định người dùng tương tự và gợi ý mục mà họ ưa thích.

CF dựa trên mục (Item-Item): Phổ biến hơn trong thực tế do khả năng mở rộng tốt hơn và thường chính xác hơn. Cách tiếp cận này xác định các mục tương tự với những mục mà người dùng đã đánh giá cao trước đó. Để dự đoán đánh giá $r_{u,i}$ của người dùng *u* cho mục *i*, hệ thống tổng hợp các đánh giá mà người dùng u đã gán cho các mục *j* tương tự với *i*.

Các thước đo độ tương đồng là cốt lõi của CF dựa trên bộ nhớ. Những độ đo sử dụng bao gồm:

Độ tương đồng cosine: Tính cosine của góc giữa hai vector:

$$sim(i,j) = \frac{\sum_{u \in U} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U} r_{u,j}^2}} \quad (1)$$

Hệ số tương quan Pearson (PCC): Đo mức tương quan tuyến tính giữa hai vector, đồng thời hiệu chỉnh sự khác biệt về thang đo đánh giá:

$$sim(i,j) = \frac{\sum_{u \in U} \left(r_{u,i} - \underline{r_i}\right)\left(r_{u,j} - \underline{r_j}\right)}{\sqrt{\sum_{u \in U} \left(r_{u,i} - \underline{r_i}\right)^2} \cdot \sqrt{\sum_{u \in U} \left(r_{u,j} - \underline{r_j}\right)^2}} \quad (2)$$

Trong đó $\underline{r_i}$ và $\underline{r_j}$ lần lượt là giá trị đánh giá trung bình của các mục *i* và *j*.

Sau khi tính toán độ tương đồng, dự đoán cho người dùng *u* trên mục *i* thường được suy ra dưới dạng trung bình có trọng số của các đánh giá mà người dùng *u* đã gán cho *k* mục tương tự nhất với *i* (tức *k* láng giềng gần nhất, ký hiệu S_k *(i;u)* (Herlocker và cộng sự, 1999).

k-Nearest Neighbor Basic (kNNBasic):

$$\widehat{r_{u,i}} = \frac{\sum_{j \in S_k(i;u)} sim(i,j) \cdot r_{u,j}}{\sum_{j \in S_k(i;u)} |sim(i,j)|} \quad (3)$$

k-Nearest Neighbor Baseline (kNNBaseline): Phương pháp này đưa vào các ước lượng baseline để điều chỉnh sai lệch của người dùng và mục. Ước lượng baseline được xác định như sau:

$$b_{u,i} = \mu + b_u + b_i \quad (4)$$

Trong đó μ là giá trị đánh giá trung bình toàn cục, b_u là sai lệch của người dùng, và b_i là sai lệch của mục. Giá trị dự đoán khi đó là:

$$\widehat{r_{u,i}} = b_{u,i} + \frac{\sum_{j \in S_k(i;u)} sim(i,j) \cdot (r_{u,j} - b_{u,j})}{\sum_{j \in S_k(i;u)} |sim(i,j)|} \quad (5)$$

Các phương pháp CF dựa trên mô hình tìm cách học các nhân tố tiềm ẩn từ ma trận tương tác người dùng - mục

để giải thích các đánh giá đã quan sát. Phân rã ma trận (Matrix Factorization - MF) là một ví dụ nổi bật (Koren và cộng sự, 2009), trong đó người dùng và mục được chiếu vào cùng một không gian tiềm ẩn có số chiều thấp hơn. Mỗi người dùng u được biểu diễn bởi vector tiềm ẩn p_u, và mỗi mục i bởi vector tiềm ẩn q_i. Giá trị đánh giá dự đoán được tính bằng tích vô hướng:

$$\hat{r}_{u,i} = p_u^\top q_i \quad (6)$$

CF có ưu điểm không yêu cầu tri thức nội dung tường minh, nhưng gặp bài toán cold-start và tính thưa dữ liệu (Adomavicius & Tuzhilin, 2005). Công trình này tăng cường CF mục - mục bằng embedding nội dung do LLM tạo ra.

III. Phương pháp, vật liệu nghiên cứu

3.1. Xử lý dữ liệu

Nguồn dữ liệu chính của chúng tôi là bộ dữ liệu MovieLens 20M (Harper & Konstan, 2015), với trọng tâm là Tag Genome (Vig và cộng sự, 2012) và siêu dữ liệu. Bộ dữ liệu bao gồm các thành phần chính:

Ratings: Các đánh giá tường minh do người dùng cung cấp, thường trên thang số từ 0.5 đến 5 sao.

Movie Metadata: Thông tin như tiêu đề, nhãn thể loại và năm phát hành.

Tag Genome: Được đưa vào ở các phiên bản bộ dữ liệu phát hành sau, Tag Genome bao gồm một tập thẻ đã được tuyển chọn với điểm mức độ liên quan cho mỗi bộ phim (Vig và cộng sự, 2012).

Bảng 1. Tổng quan bộ dữ liệu MovieLens 20M gốc và bộ dữ liệu sau tiền xử lý

Tập dữ liệu	Số ratings	Số users	Số items	Độ thưa
Original	20,000,263	138,493	27,278	99.47%
Preprocessed	19,793,342	138,185	10,239	98.97%

Sau khi loại bỏ phim/người dùng có ít hơn 20 đánh giá, tập dữ liệu còn 19,8 triệu đánh giá trên 10.239 bộ phim và 138.185 người dùng (Bảng 1).

3.1. Phương pháp đánh giá

Các chỉ số đánh giá:

Root Mean Squared Error (RMSE): đo sai số dự đoán đánh giá.

$$RMSE = \sqrt{\sum_{u,i \in TESTSET} (\widehat{r_{ui}} - r_{ui})^2 / |TESTSET|} \quad (7)$$

Trong đó |TESTSET| là kích thước tập kiểm thử; $\widehat{r_{ui}}$ là giá trị đánh giá của người dùng u cho mục i do mô hình ước lượng; còn giá trị đánh giá quan sát tương ứng được ký hiệu là r_{ui}.

Precision@K (P@K): tỷ lệ mục liên quan trong top K gợi ý; Recall@K (R@K): tỷ lệ mục liên quan được gợi ý trên tổng mục liên quan.

$$Precision = \#tp \,/\, (\#tp + \#fp) \quad (8)$$

$$Recall = \#tp \,/\, (\#tp + \#fn) \quad (9)$$

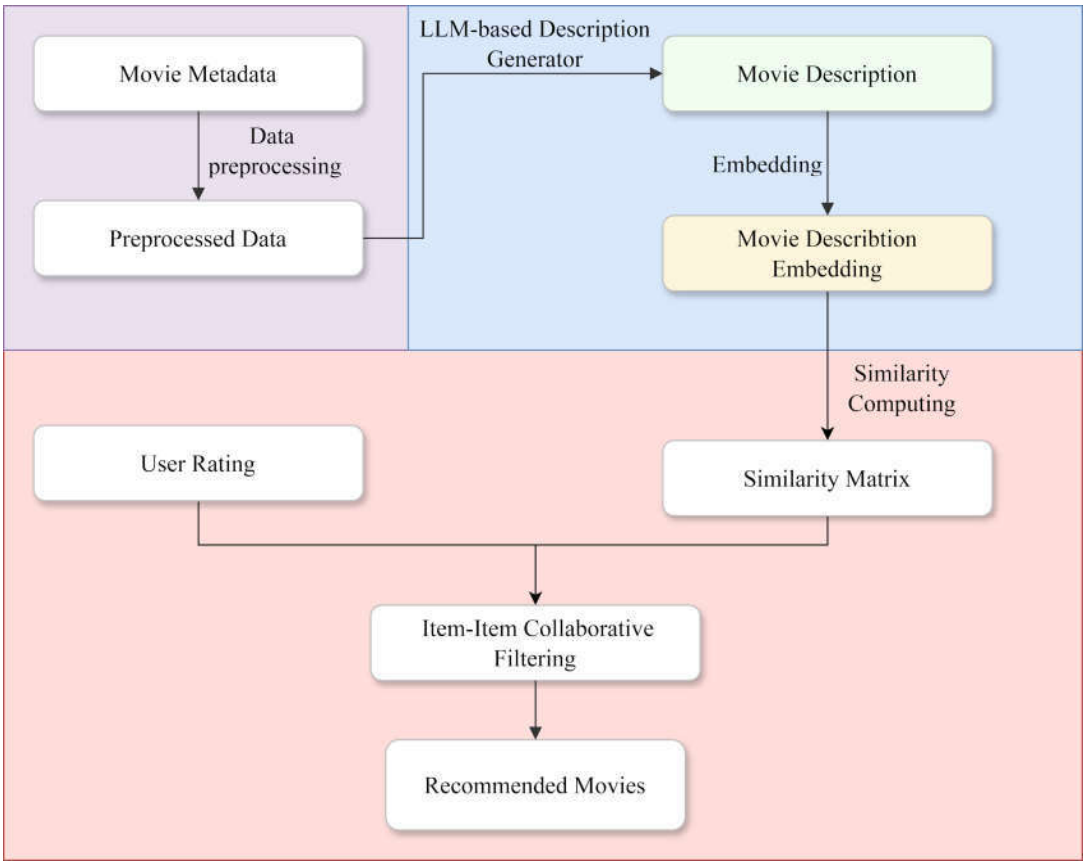
Bảng 2. Phân loại các kết quả có thể xảy ra khi gợi ý một mục cho một người dùng

	Liên quan	Không liên quan	Tổng
Được gợi ý	True-positive (tp)	False-positive (fp)	tp + fp
Không được gợi ý	False-negative (fn)	True-negative (tn)	fn + tn
Tổng	tp + fn	fp + tn	n

IV. Hệ thống đề xuất

Hệ thống đề xuất tăng cường CF mục - mục bằng cách sử dụng LLM để sinh mô tả phim giàu ngữ nghĩa từ siêu dữ liệu có cấu trúc. Các mô tả được chuyển thành embedding dày đặc để tinh chỉnh độ tương đồng mục - mục.

Giả thuyết trung tâm của chúng tôi là vượt qua các đặc trưng nội dung sơ khai (chẳng hạn các thẻ hoặc thể loại cơ bản) bằng cách tạo ra các tường thuật mô tả phim thông qua mô hình ngôn ngữ. Hình 1 minh họa sơ đồ kiến trúc tổng thể của hệ thống đề xuất, bao gồm ba khối chức năng chính: (1) khối xử lý và tổng hợp dữ liệu đầu vào từ Tag Genome, thể loại và siêu dữ liệu bên ngoài; (2) khối sinh mô tả ngữ nghĩa sử dụng LLM và chuyển đổi thành embedding dày đặc; và (3) khối lọc cộng tác mục - mục được tăng cường bởi các embedding ngữ nghĩa để tạo gợi ý cuối cùng.



Hình 1. Sơ đồ kiến trúc tổng thể của hệ thống đề xuất với ba khối chức năng: tổng hợp dữ liệu, sinh mô tả bằng LLM, và lọc cộng tác tăng cường ngữ nghĩa

Trước hết, chúng tôi tiền xử lý và tổng hợp các đầu vào có cấu trúc của mỗi bộ phim: chủ yếu là dữ liệu Tag Genome và nhãn thể loại của MovieLens, cung cấp các điểm mức độ liên quan chi tiết cho các thẻ, và tùy chọn là siêu dữ liệu được làm giàu thêm (ví dụ: diễn viên và ê-kíp) từ các nguồn bên ngoài. Từ các đầu vào này, chúng tôi xây dựng một hồ sơ phim chi tiết cho từng tiêu đề.

Tiếp theo, chúng tôi đưa mỗi hồ sơ vào một LLM tạo sinh tiên tiến. Chúng tôi thiết kế prompt cẩn thận dựa trên hồ sơ có cấu trúc, yêu cầu LLM tạo ra một mô tả mạch lạc, hấp dẫn, nắm bắt tinh thần của bộ phim mà không tiết lộ spoiler. Kết quả

đầu ra là một đoạn văn duy nhất, gắn kết cho mỗi bộ phim.

Cuối cùng, chúng tôi chuyển mỗi mô tả phim thành một embedding ngữ nghĩa bằng một mô hình embedding câu huấn luyện sẵn. Các vector nhiều chiều này - nắm bắt nội dung theo chủ đề và phong cách một cách tinh tế - sau đó được dùng để tính toán độ tương đồng mục - mục được tăng cường cho thuật toán gợi ý.

V. Kết quả nghiên cứu và thảo luận

5.1. Kết quả nghiên cứu

5.1.1. Sinh mô tả phim dựa trên LLM

Một đổi mới then chốt của hệ thống là sử dụng LLM huấn luyện sẵn để tạo các mô tả văn bản toàn diện cho từng bộ phim. Cụ thể, chúng tôi sử dụng mô hình Claude Haiku (Anthropic) - một mô hình ngôn ngữ lớn được tối ưu cho tốc độ suy luận nhanh và chi phí thấp, phù hợp với việc sinh mô tả hàng loạt cho hơn 10.000 bộ phim trong tập dữ liệu. Claude Haiku được lựa chọn vì khả năng tuân thủ chỉ dẫn (instruction following) cao, chất lượng văn bản sinh ra mạch lạc và nhất quán, đồng thời chi phí API hợp lý cho quy mô xử lý lớn. Với mỗi bộ phim, đầu vào của mô hình là một prompt có cấu trúc bao gồm ba phần: (i) phần system prompt định nghĩa vai trò của mô hình là một chuyên gia điện ảnh; (ii) phần dữ liệu cung cấp tiêu đề phim, nhãn thể loại, danh sách 30 thẻ Tag Genome có điểm liên quan cao nhất kèm điểm số, và thông tin diễn viên/đạo diễn từ IMDb; và (iii) phần chỉ dẫn yêu cầu LLM sinh một đoạn văn 80-120 từ mô tả chủ đề, phong cách và điểm nổi bật của phim mà không tiết lộ spoiler. Prompt được thiết kế theo nguyên tắc nhất quán về luồng ngữ nghĩa: chủ đề → sắc thái → phong cách → điểm khác biệt → đối tượng khán giả, nhằm đảm bảo các mô tả sinh ra có cấu trúc tương đồng và phù hợp cho việc tạo embedding. Toàn bộ quy trình sinh mô tả cho 10.239 bộ phim được thực hiện qua API với cơ chế checkpointing và ghi nhật ký prompt, cho phép tái lập hoàn toàn thực nghiệm. Thời gian xử lý trung bình khoảng 1,2 giây mỗi bộ phim với tổng chi phí API xấp xỉ 3,5 USD cho toàn bộ tập dữ liệu. Vì mô hình có quyền truy cập vào thông tin thẻ chi tiết và bối cảnh liên quan, các mô tả sinh ra giàu sắc thái và nhiều thông tin hơn đáng kể so với các danh sách thẻ đơn giản hoặc các bản tóm tắt cốt truyện do cộng đồng đóng góp.

Ví dụ minh họa đầu ra của LLM:

Movie ID 55498, Silk (2007): "*Silk unfolds as a poignant tale of obsession and longing set against the backdrop of late-19th-century Japan. The film follows Hervé Joncour (Michael Pitt) on a transcontinental odyssey driven by his quest for silkworm eggs, while his enigmatic relationship with a Japanese woman challenges cultural boundaries. The narrative weaves adventure, romance, and philosophical introspection, underscored by lush cinematography and a haunting score. Keira Knightley's performance adds emotional resonance, and François Girard's direction emphasizes visual storytelling, rendering \textit{Silk} an evocative reflection on desire, mortality, and artistic devotion*."

5.1.2. Embedding ngữ nghĩa của mô tả

Các mô tả do LLM sinh ra được chuyển thành biểu diễn vector.

Mô hình embedding: Chúng tôi sử dụng một mô hình embedding câu huấn luyện sẵn (text-embedding-3-small của OpenAI).

Biểu diễn vector: Mỗi mô tả được chuyển thành vector dày đặc; các phim

có nội dung tương đồng sẽ nằm gần nhau trong không gian ngữ nghĩa.

5.1.3. Lọc cộng tác mục - mục với embedding được làm giàu

Các embedding bắt nguồn từ LLM là nền tảng cho một thuật toán CF mục - mục được tăng cường.

Tính toán độ tương đồng: Với bất kỳ hai bộ phim i và j, chúng tôi tính độ tương đồng cosine giữa các vector embedding của chúng:

$$sim_{cos}(i,j) = \frac{e_i \cdot e_j}{|e_i||e_j|} \quad (10)$$

Tùy chọn, có thể sử dụng tương quan Pearson nếu các embedding đã được chuẩn hóa theo trung bình.

Hình thành lân cận: Với mỗi bộ phim i, chúng tôi xác định k bộ phim tương tự nhất theo điểm độ tương đồng dựa trên embedding. Đây chính là các lân cận mục dùng cho CF.

Dự đoán và gợi ý: Để dự đoán đánh giá $r_{u,i}$ của người dùng *u* cho một bộ phim *j* chưa được đánh giá, chúng tôi tổng hợp các đánh giá của người dùng cho các bộ phim trong lân cận $S_k(j;u)$:

$$\widehat{r_{u,j}} = b_{u,j} + \frac{\sum_{i \in S_k(j;u)} sim(j,i) \cdot (r_{u\,i} - b_{u\,i})}{\sum_{i \in S_k(j;u)} |sim(j,i)|} \quad (11)$$

Trong đó ước lượng baseline $b_{u,i}$ được xác định như sau:

$$b_{u,j} = \mu + b_u + b_j \quad (12)$$

với μ là giá trị đánh giá trung bình toàn cục, b_u là sai lệch của người dùng, và b_j là sai lệch của mục.

Embedding được làm giàu từ LLM giúp nắm bắt ngữ nghĩa tinh tế hơn so với CF truyền thống.

5.1.4. Kết quả thực nghiệm

Bảng III trình bày so sánh hiệu năng giữa mô hình đề xuất và bốn phương pháp cơ sở được lựa chọn có chủ đích nhằm bao phủ các hướng tiếp cận chính trong hệ gợi ý. kNNBaseline (k=40) đại diện cho CF dựa trên bộ nhớ thuần túy với hiệu chỉnh baseline, là phương pháp kinh điển và phổ biến nhất trong thực tế. kNN-Content (k=20) sử dụng độ tương đồng dựa trên đặc trưng nội dung gốc (thể loại, thẻ) mà không có tăng cường LLM, cho phép đánh giá trực tiếp đóng góp của embedding ngữ nghĩa do LLM sinh ra. SVD đại diện cho nhóm phân rã ma trận, là phương pháp dựa trên mô hình được sử dụng rộng rãi nhờ khả năng học các nhân tố tiềm ẩn hiệu quả (Koren và cộng sự, 2009). FMgenome (Factorization Machine trên Tag Genome) khai thác trực tiếp các đặc trưng Tag Genome dưới dạng số mà không qua bước sinh mô tả văn bản, cho phép đánh giá giá trị gia tăng của việc chuyển đổi dữ liệu có cấu trúc thành mô tả ngữ nghĩa qua LLM. Các chỉ số đánh giá được sử dụng là Root Mean Squared Error (RMSE), Precision@5 (P@5) và Recall@5 (R@5), phản ánh đồng thời hiệu quả dự đoán đánh giá và chất lượng gợi ý top-N.

Mô hình đề xuất đạt kết quả tốt nhất trên cả ba chỉ số. Xét về RMSE, mô hình đề xuất đạt 0.7673, thấp hơn tất cả các phương pháp cơ sở. So với kNNBaseline (0.8108), mức này tương ứng với việc giảm 5.37% sai số dự đoán. Tương tự, RMSE được cải thiện 3.38% so với SVD (0.7922) và 3.09% so với FMgenome (0.7918). Những kết quả này cho thấy hiệu quả của mô hình đề xuất trong việc nâng cao độ chính xác của dự đoán đánh giá.

Bảng 3. So sánh hiệu năng giữa mô hình đề xuất và các mô hình cơ sở khác.

Mô hình	RMSE	P@5	R@5
kNNBaseline (k=40)	0.8108	0.8021	0.4342
kNN-Content (k=20)	0.7885	0.8008	0.4362
SVD	0.7922	0.8005	0.4322
FMgenome	0.7918	0.7993	0.4316
Proposed Model	0.7673	0.8195	0.4413

Precision@5 đạt 0.8195, vượt kNNBaseline (0.8021) 2.17%; Recall@5 đạt 0.4413, vượt FMgenome (0.4316) 2.25%. Mô hình đề xuất không chỉ gợi ý chính xác hơn mà còn bao phủ phạm vi mục liên quan rộng hơn.

Việc tích hợp LLM và embedding ngữ nghĩa giúp nắm bắt ngữ nghĩa mục hiệu quả hơn so với các tiếp cận truyền thống. Tóm lại, mô hình đề xuất vượt trội trên tất cả chỉ số, xác nhận tính hiệu quả của phương pháp.

5.2. Thảo luận

Về tác động của chất lượng mô tả đến hiệu năng mô hình, chúng tôi nhận thấy rằng các mô tả do LLM sinh ra có chất lượng nhất quán cao nhờ ba yếu tố: (i) prompt được thiết kế có cấu trúc với luồng ngữ nghĩa cố định, đảm bảo tính đồng nhất giữa các mô tả; (ii) dữ liệu đầu vào Tag Genome cung cấp thông tin chi tiết và đã được lượng hóa, giúp LLM sinh mô tả bám sát đặc trưng thực tế của phim thay vì bịa đặt; và (iii) giới hạn độ dài 80-120 từ ngăn ngừa hiện tượng mô tả quá dài hoặc lan man. Đối với một số trường hợp phim có dữ liệu Tag Genome nghèo (ít thẻ có điểm liên quan cao), mô tả sinh ra có xu hướng chung chung hơn, tuy nhiên embedding vẫn nắm bắt được thông tin thể loại và phong cách cơ bản.

Chi phí tính toán: sinh mô tả LLM là bước ngoại tuyến một lần (~3,4 giờ, ~3,5 USD cho 10.239 phim); tạo embedding dưới 10 phút. Pha trực tuyến chỉ tra cứu lân cận trên embedding đã tính sẵn, không bị ảnh hưởng bởi LLM.

VI. Kết luận

Bài báo đề xuất khung gợi ý lai tích hợp lọc cộng tác với mô tả ngữ nghĩa do LLM sinh ra, sử dụng embedding để nắm bắt cả tín hiệu cộng tác lẫn nội dung trong khuôn khổ thống nhất.

Kết quả thực nghiệm trên MovieLens-20M cho thấy mô hình đề xuất đạt RMSE thấp nhất (0.7673) và P@5, R@5 cao nhất, vượt trội so với các phương pháp kNN, SVD và factorization machines.

Hướng phát triển bao gồm tăng cường khả năng giải thích, tích hợp embedding hồ sơ người dùng và thử nghiệm kiến trúc Transformer cho mô hình hóa tương tác động.

Tài liệu tham khảo

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.

Bao, W., Ding, Y., Wan, M., Zhang, Y., He, X., & Chua, T.-S. (2023). LLM4Rec: Enhancing recommender systems with large language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Bommasani, R., Hudson, D. A., Adeli, E., & và cộng sự (2021). On the opportunities and risks of foundation models. *arXiv*. https://arxiv.org/abs/2108.07258

Brown, T., Mann, B., Ryder, N., & và cộng sự (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 33, 1877-1901.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., & Brockman, G. (2021). Evaluating large language models trained on code. *arXiv*. https://arxiv.org/abs/2107.03374

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (pp. 4171-4186).

Fan, Q., Zheng, K., Qu, L., & Huang, X. (2023). A survey on large language models (LLMs) for recommender systems. *arXiv*. https://arxiv.org/abs/2305.12619

Geng, Z., Richards, S., & He, M. (2022). Promptrec: Learning to recommend items with prompts. In *Proceedings of the ACM Web Conference 2022* (pp. 1613-1624). ACM.

Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4), 1-19.

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of SIGIR* (pp. 230-237). ACM.

Koren, Y., & Bell, R. (2011). Advances in collaborative filtering. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (2nd ed., pp. 77-118). Springer.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76-80.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook* (pp. 73-105). Springer.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 325-341). Springer.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP* (pp. 3982-3992).

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, Article 4.

Sun, L., & Zhang, X. (2021). Conversational recommender system survey. *arXiv*. https://arxiv.org/abs/2106.01242

Van Meteren, R., & Van Someren, M. (2000). Using content-based filtering for recommendation. In *Proceedings of ECML/PKDD Workshop: Machine Learning in New Information Age* (pp. 47-56).

Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (pp. 109-116). ACM.

Vaswani, A., Shazeer, N., Parmar, N., & và cộng sự (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, 30, 6000-6010.

Vig, J., Sen, S., & Riedl, J. (2012). The tag genome: Encoding community knowledge to support novel interaction. In *Proceedings of IUI* (pp. 199-208). ACM.

Zhang, S., & Chen, L. (2020). Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1), 1-101.

ENHANCING COLLABORATIVE FILTERING WITH SEMANTIC KNOWLEDGE USING LARGE LANGUAGE MODELS

Pham Thi Thu Trang[1], Dang Khanh Hoa[1], Nguyen Hoang[1], Nguyen Vu Son[1]

Abstract: *Traditional recommender systems often face significant challenges, particularly data sparsity and the limited ability to capture the semantic relationships between users and items. To overcome these limitations, this study introduces a hybrid recommendation framework that integrates collaborative filtering with semantic information derived from movie metadata and descriptions generated by large language models (LLMs). In the proposed approach, a pretrained LLM is utilized to automatically produce enriched textual descriptions of movies. These texts are subsequently transformed into semantic embeddings using the fastText model to improve item representations. The resulting semantic features are then combined with user-item interaction data to enhance the recommendation process. Experimental evaluations conducted on the MovieLens-20M dataset show that the proposed method achieves superior performance compared to conventional recommendation techniques, as evidenced by improvements in RMSE, Precision@5, and Recall@5. These results demonstrate the effectiveness of leveraging LLM-generated textual information and semantic augmentation to enhance the performance of recommender systems.*

Keywords: *artificial intelligence, recommendation systems, collaborative filtering, large language models (LLMs), semantic embeddings*

[1] Faculty of Electric and Electronic Engineering, Hanoi Open University, Hanoi, Vietnam