Technical Report

A. Report Information

- **Course / Môn học:** IS353.Q11
- **Project / Mã và Tên đề tài:** A10. Author Collaborator Finder (ArXiv Co-auth)
- **Duration / Thời gian thực hiện:** 9/2025 – 12/2025
- **Group / Nhóm**: 13 - CBV
- **Members / Thành viên:**

1. Đặng Trần Chương, 22520163, 22520163@gm.uit.edu.vn
2. Nguyễn Tường Vĩnh, 22521679, 22521679@gm.uit.edu.vn
3. Hà Xuấn Bắc, 22520088, 22520088@gm.uit.edu.vn

- **Supervisor / Giảng viên hướng dẫn**: Trần Hưng Nghiệp, nghiepth@uit.edu.vn

B. Report Content

Abstract

Đồ án xây dựng hệ thống gợi ý đồng tác giả tiềm năng trên mạng hợp tác ca-HepPh (lĩnh vực High Energy Physics – Phenomenology), trong đó mỗi cạnh biểu diễn hai tác giả đã từng đồng tác giả. Hệ thống tạo ứng viên bằng 2-hop proximity (bạn của bạn nhưng chưa từng hợp tác), khai thác cấu trúc cộng đồng bằng community detection để đo mức độ gần gũi, và chấm điểm bằng link prediction dựa trên các đặc trưng lân cận. Dữ liệu được đánh giá theo thiết lập tách cạnh train/test, kết hợp các thước đo AUC/AP và các chỉ số top-k nhằm so sánh hiệu quả giữa các phương pháp. Đầu ra gồm danh sách gợi ý xếp hạng theo điểm phù hợp và phần giải thích nêu rõ bạn chung/cộng đồng chung hỗ trợ cho mỗi đề xuất, đồng thời được triển khai dưới dạng prototype tra cứu trực quan.

1. Introduction

1.1 Motivation

Trong cộng đồng nghiên cứu khoa học, việc tìm được đồng tác giả phù hợp không chỉ giúp tăng năng suất công bố mà còn quyết định chất lượng và hướng phát triển của một nhóm nghiên cứu. Tuy nhiên, trong thực tế, cơ hội hợp tác thường bị chi phối bởi mạng lưới quan hệ sẵn có: các tác giả hay hợp tác sẽ tiếp tục hợp tác trong "vòng quen biết", còn các tác giả ít kết nối thì khó tiếp cận các nhóm khác.. Vì vậy, nhu cầu đặt ra là một hệ thống gợi ý có thể tận dụng tín hiệu cấu trúc đồ thị để đề xuất các kết nối hợp lý, đồng thời ưu tiên những gợi ý có khả năng mở rộng hợp tác liên-cộng-đồng nhằm tăng tính đa dạng và cơ hội giao thoa tri thức. Quan trọng hơn, gợi ý trong bối cảnh học thuật cần giải thích được để người dùng tin tưởng và có cơ sở ra quyết định.

1.2 Problem Statement

Bài toán nhóm giải quyết là gợi ý đồng tác giả tiềm năng cho một tác giả u trong mạng đồng tác giả ca-HepPh, với đầu ra gồm: danh sách gợi ý top-k các tác giả v mà u chưa từng hợp tác nhưng có khả năng hợp tác trong tương lai và phần giải thích cho từng gợi ý để người dùng hiểu "vì sao" đề xuất đó hợp lý. Bài toán được mô hình hóa như link prediction trên đồ thị: dự đoán xác suất xuất hiện cạnh dựa trên cấu trúc lân cận, vai trò cộng đồng và các đặc trưng đồ thị, sau đó dùng điểm dự đoán để xếp hạng.

1.3 Objectives & Scope

Mục tiêu chung: Xây dựng prototype ứng dụng cho bài toán gợi ý đồng tác giả trên mạng ca-HepPh, phân tích cấu trúc mạng để hiểu bản chất dữ liệu, xây dựng và so sánh mô hình theo 3 hướng chính: 2-hop proximity, community overlap, link prediction.

Để đạt được mục tiêu trên, nghiên cứu tập trung giải quyết các câu hỏi sau đây:

- Mạng có cộng đồng mạnh đến mức nào?
- 2-hop, community overlap và link prediction cho chất lượng top-k ra sao?

- Giải thích dạng bạn chung + cộng đồng chung có đủ “thuyết phục/hiểu được” để hỗ trợ quyết định không?

Giới hạn phạm vi: Giới hạn trên phân tích cấu trúc liên kết của đồ thị tĩnh vì tập dữ liệu không có các mốc thời gian

1.4 Contributions

Xây dựng pipeline gợi ý đồng tác giả trên ca-HepPh: 2-hop tạo ứng viên + community detection (Louvain/LPA) + link prediction features và XGBoost để chấm điểm và xếp hạng top-k.

Điểm mạnh: kết hợp nhiều tín hiệu cấu trúc và có giải thích cho từng gợi ý (bạn chung/cộng đồng).

Deliverables: pipeline + đánh giá top-k metrics + app desktop hiển thị gợi ý và lời giải thích.

2. Related Work

2.1 Background Knowledge

Co-author graph (đồ thị đồng tác giả): Đồ thị $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ trong đó mỗi nút $\mathrm{v} \in \mathrm{V}$ là một tác giả, mỗi cạnh $(\mathrm{u}, \mathrm{v}) \in \mathrm{E}$ nghĩa là u và v đã từng hợp tác với nhau. Với ca-HepPh, dữ liệu là edge list, thường xử lý như đồ thị vô hướng.

Ký hiệu cơ bản:

- $N(u)$: tập hàng xóm (các tác giả đã hợp tác với u).
- Deg(u) = $N(u)$: bậc của nút u.
- Ứng viên gợi ý cho u: các $v \notin N(u)$nhưng thuộc 2-hop (hàng xóm của hàng xóm).

Link Prediction: Bài toán dự đoán một cạnh (u, v)có thể xuất hiện trong tương lai hay không, dựa trên cấu trúc đồ thị. Trong mô hình gợi ý, dùng điểm dự đoán để xếp hạng top-k các ứng viên.

Các chỉ số lân cận (heuristics) phổ biến:

- Common Neighbors (**CN**): $| N(u) \cap N(v) |$
- Jaccard: $\frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$
- Adamic–Adar (**AA**): $\sum_{w \in N(u) \cap N(v)} \frac{1}{\log \deg(w)}$
- Resource Allocation (**RA**): $\sum_{w \in N(u) \cap N(v)} \frac{1}{\deg(w)}$
- Preferential Attachment (**PA**): $\deg(u) \deg(v)$

Community detection: tách đồ thị thành các cụm nút liên kết chặt (ví dụ Louvain, Label Propagation – LPA). Feature thường dùng: $same_comm(u,v) \in \{0,1\}$cho biết hai nút có cùng cộng đồng hay không.

XGBoost cho link prediction: Mô hình học máy nhận đầu vào là feature của cặp (u,v) (CN/AA/RA/PA, degree, same-community, …) và học để phân biệt cạnh thật (positive) vs cặp không có cạnh (negative), sau đó xuất score/probability để xếp hạng.

2.2 Prior Work Comparison & Limitations

Mạng hợp tác arXiv HEP-PH mô tả lịch sử đồng tác giả gần như đầy đủ của một cộng đồng nghiên cứu trong giai đoạn 1993–2003, nơi mỗi bài báo nhiều tác giả tạo thành các "cụm" liên kết dày đặc và toàn mạng thường hình thành cấu trúc cộng đồng rõ rệt. Chính đặc điểm này khiến các phương pháp gợi ý truyền thống dựa thuần vào độ nổi tiếng (degree cao) hoặc các heuristic lân cận đơn giản (như 2-hop, common neighbors) dễ thiên về việc đề xuất những tác giả đã ở rất gần trong cùng cụm, dẫn đến gợi ý lặp lại, kém đa dạng và khó mở rộng sang các nhóm nghiên cứu khác. Mặt khác, các cách tiếp cận học máy/embedding có thể cải thiện độ chính xác nhưng thường thiếu khả năng giải thích, trong khi gợi ý hợp tác khoa học cần lý do rõ ràng để người dùng tin. Vì vậy, khoảng trống nghiên cứu mà nhóm hướng tới là xây dựng một khung gợi ý đồng tác giả vừa xếp hạng top-k hiệu quả trên mạng có cộng đồng mạnh, vừa cung cấp justification paths dễ hiểu, và đánh giá theo đúng mục tiêu triển khai của đề tài, thay vì chỉ tối ưu các chỉ số tổng quát.

2.3 Positioning of This Work

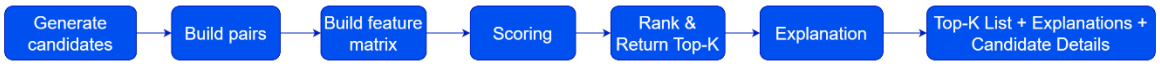
Đề tài thuộc hướng gợi ý trên đồ thị cho bài toán co-author. Khác với baseline heuristic chỉ dựa lân cận, đề tài kết hợp thêm tín hiệu cộng đồng và XGBoost để chấm điểm và xếp hạng top-k. Kết quả được đóng gói thành desktop app và đánh giá theo các ranking metrics phục vụ triển khai thực tế.

3. Methodology

3.1 Overview of Approach

Kiến trúc tổng thể của hệ gợi ý đồng tác giả gồm một pipeline theo tầng: Data ingestion & preprocessing: đọc dữ liệu bài báo/edge-list; sau đó tiến hàng phân chia dữ liệu: Tập dữ liệu cạnh được chia theo tỷ lệ 80% Train / 20% Test. Áp dụng Negative Sampling để xử lý mất cân bằng dữ liệu. Candidate generation: với mỗi tác giả truy vấn u, sinh tập ứng viên v bằng 2-hop; Scoring/Ranking layer: tính đặc trưng mạng (CN/Jaccard/AA/RA/PA, degree) và tín hiệu community (Louvain/LPA), rồi xếp hạng theo 4 phương pháp: 2-hop proximity, community overlap, hybrid (kết hợp RA + community), và XGBoost; Explanation & output: trả top-K kèm justification paths dạng $u \to w \to v$ (bạn chung/cùng cộng đồng) và đánh giá bằng các chỉ số top-K (P@K, R@K, Hit@K, NDCG@K) trước khi đưa vào prototype UI.

Pipeline tổng thể



Pipeline recoomend

3.2 Data

Dataset: ca-HepPh.txt (Collaboration Graphs) được lấy từ Kaggle. Có tổng cộng 12008 node và 237010 edge.

Mô tả tập dữ liệu: Mạng lưới cộng tác Arxiv HEP-PH (High Energy Physics – Phenomenology) được lấy từ kho e-print arXiv và phản ánh các hợp tác khoa học giữa các tác giả có bài nộp vào chuyên mục Vật lý năng lượng cao – Hiện tượng học. Nếu tác giả i đồng tác giả một bài báo với tác giả j, thì trong đồ thị sẽ có một cạnh vô hướng nối từ i đến j. Nếu một bài báo có k tác giả đồng viết, thì điều này tạo ra một (tiểu)đồ thị liên thông đầy đủ trên k nút (tức mọi cặp tác giả trong nhóm đều được nối với nhau).

Dữ liệu bao phủ các bài báo trong giai đoạn từ tháng 01/1993 đến tháng 04/2003 (tổng cộng 124 tháng). Dataset bắt đầu chỉ vài tháng sau khi arXiv ra đời, vì vậy nó gần như đại diện cho toàn bộ lịch sử của chuyên mục HEP-PH trên arXiv trong khoảng thời gian này.

3.3 System Design / ModelDesign

3.3.1. Xử lý dữ liệu

Data Ingestion chịu trách nhiệm nhận file dữ liệu đồng tác giả từ người dùng, đọc từng dòng cạnh u v, loại bỏ các dòng comment và dữ liệu lỗi, chuẩn hóa định dạng id tác giả, loại cạnh tự nối, sắp xếp cặp u v theo thứ tự cố định và khử trùng lặp để tạo danh sách cạnh sạch làm đầu vào ổn định cho các bước sau.

Graph Builder và Core Filtering xây dựng đồ thị vô hướng từ danh sách cạnh đã làm sạch, sau đó lọc bằng k core để loại bớt các node quá yếu và giữ lại phần lõi của mạng hợp tác nhằm giảm nhiễu và giảm kích thước tính toán, đồng thời tạo đồ thị làm nền cho việc học và suy luận gợi ý.

3.3.2. Trích xuất các đặc trưng

Community Detection tạo nhãn cộng đồng cho từng tác giả bằng cách chạy Louvain và Label Propagation trên đồ thị huấn luyện, kết quả là hai bảng ánh xạ node sang community id, giúp hệ thống có thêm tín hiệu cấu trúc trung mô để đánh giá mức độ liên quan giữa hai tác giả theo góc nhìn cùng cộng đồng.

Feature Engineering biến mỗi cặp tác giả u v thành một vector đặc trưng bằng cách đo độ gần gũi cục bộ và độ tương đồng cấu trúc, bao gồm số bạn chung và các heuristic kinh điển của link prediction như Jaccard, Adamic Adar, Resource Allocation, Preferential Attachment, kết hợp với degree của hai đầu mút và cờ cùng cộng đồng từ Louvain và LPA để mô hình học máy có thể phân biệt cặp có khả năng hợp tác cao.

Training và Negative Sampling tạo dữ liệu học cho bài toán phân loại cạnh bằng cách lấy các cạnh thật trong đồ thị làm nhãn dương, sau đó sinh các cặp không có cạnh làm nhãn

âm bằng cách chọn ngẫu nhiên và kiểm tra tránh trùng với cạnh thật, cuối cùng huấn luyện XGBoost để dự đoán xác suất tồn tại cạnh dựa trên vector đặc trưng của từng cặp.

3.3.3. Mô hình huấn luyện

Candidate Generation và Ranking xử lý truy vấn theo tác giả đầu vào bằng cách lấy tập ứng viên từ hàng xóm bậc hai của tác giả đó và loại bỏ các cộng tác viên hiện tại, sau đó tính đặc trưng cho từng ứng viên và dùng mô hình đã huấn luyện để chấm điểm xác suất, sắp xếp giảm dần và chọn ra danh sách top k gợi ý.

3.3.4. Suy luận và gợi ý

Explanation và Justification Paths giải thích vì sao một ứng viên được đề xuất bằng cách trích ra các node trung gian là bạn chung giữa hai tác giả tạo thành các đường đi u w v, đồng thời hiển thị tín hiệu cùng cộng đồng từ hai thuật toán community detection để người dùng thấy rõ bằng chứng cấu trúc đằng sau điểm xếp hạng.

3.4 Training & Implementation Details

Kiến trúc thực thi: Hệ thống được phát triển bằng Python, chia thành hai luồng: Offline Training (xử lý dữ liệu, huấn luyện mô hình trên Jupyter Notebook) và Online Inference (ứng dụng demo tương tác thời gian thực trên Streamlit).

Phân chia dữ liệu: Tập dữ liệu cạnh được chia theo tỷ lệ 80% Train / 20% Test. Áp dụng Negative Sampling với tỷ lệ 1:5 (1 cạnh dương : 5 cạnh âm) để xử lý mất cân bằng dữ liệu.

Công cụ & Thư viện:NetworkX: Xây dựng đồ thị, tính toán đặc trưng mạng (Jaccard, AA) và phân rã cộng đồng (Louvain/LPA), XGBoost & Scikit-learn: Huấn luyện mô hình và đánh giá hiệu năng, Streamlit: Xây dựng giao diện Web App và trực quan hóa kết quả.

3.5 Evaluation Protocol

Kế hoạch kiểm thử:

Data Partitioning (Phân chia dữ liệu), sử dụng phương pháp Edge Split ngẫu nhiên trên đồ thị gốc G=(V, E).

- Tập Train E train: Chiếm 80% số cạnh. Đồ thị huấn luyện G_train được xây dựng chỉ từ các cạnh này. Các đặc trưng (Features) và cấu trúc cộng đồng (Communities) đều được tính toán dựa trên G_train để đảm bảo tính công bằng.
- Tập Test E test: Chiếm 20% số cạnh còn lại. Đây là các "liên kết ẩn" (missing links) mà hệ thống cần dự đoán.

Đánh giá khả năng phân loại nhị phân (Có liên kết / Không liên kết) trên toàn bộ tập Test kết hợp với tập mẫu âm (Negative samples) có kích thước tương đương.

Personalized Ranking (Top-K): Với mỗi tác giả u trong tập kiểm thử, hệ thống thực hiện quy trình thực tế:

1. Tạo tập ứng viên từ các node cách 2 bước (2-hop neighbors) trong G_train.
2. Loại bỏ các node đã là hàng xóm trực tiếp của u.
3. Tính điểm và xếp hạng.
4. Kiểm tra xem các đồng tác giả thực sự (nằm trong E_test) có xuất hiện trong Top-K của danh sách gợi ý hay không.

Độ đo đánh giá:
Precision@K: Tỷ lệ gợi ý đúng trong Top-K

$$P@K(u) = \frac{|R_u \cap G_u|}{K}$$

Recall@K: Tỷ lệ ground-truth được "bắt" lại trong Top-K

$$R@K(u) = \frac{|R_u \cap G_u|}{|G_u|}$$

Hit@K: Chỉ cần có ít nhất 1 gợi ý đúng trong Top-K

$$Hit@K(u) = \begin{cases} 1 & \text{nếu } |R_u \cap G_u| > 0 \\ 0 & \text{ngược lại} \end{cases}$$

NDCG@K là chỉ số đánh giá chất lượng xếp hạng của danh sách K phần tử được gợi ý đầu tiên, được tính bằng tỷ lệ giữa điểm DCG thực tế và điểm DCG lý tưởng

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

4. Experiments and Results

4.1 Experimental Settings

Để đảm bảo tính khách quan và khả năng tái lập kết quả (reproducibility), nhóm đã thiết lập môi trường thực nghiệm với các thông số chi tiết như sau:

Thực nghiệm được tiến hành trên bộ dữ liệu CA-HepPhtừ hệ thống ArXiv.

- Đặc điểm: Mạng lưới cộng tác khoa học, nơi các node đại diện cho tác giả và các cạnh đại diện cho quan hệ đồng tác giả (co-authorship).
- Thống kê gốc: 12,008 node và 237,010 cạnh.

Tiền xử lý & Phân chia:

- Edge Split: Tập dữ liệu cạnh được chia ngẫu nhiên thành hai phần:
  - Training Set (80%): Dùng để xây dựng đồ thị G_train và huấn luyện mô hình.
  - Test Set (20%): Dùng để đánh giá.
- Negative Sampling: Để giải quyết vấn đề mất cân bằng dữ liệu, áp dụng chiến lược lấy mẫu âm ngẫu nhiên với tỷ lệ 1:5.
  - Với mỗi cạnh dương (liên kết thật) trong tập Train, hệ thống sinh ngẫu nhiên 5 cặp node không có liên kết để làm mẫu âm (nhãn 0).

Cấu hình mô hình:
Sử dụng thuật toán XGBoost Classifier với bộ siêu tham số (Hyperparameters) được tinh chỉnh để cân bằng giữa độ chính xác và tốc độ hội tụ.

4.2 Quantitative Results

Bảng này dùng để đánh giá và so sánh 4 mô hình gợi ý đồng tác giả theo các mức Top-K (5/10/20) trên 2000 tác giả: xem mô hình nào gợi ý đúng nhiều hơn và xếp hạng tốt hơn.

	method	k	P@k	R@k	Hit@k	NDCG@k
0	2hop	5	0.418800	0.535075	0.7900	0.719696
1	community	5	0.378400	0.472603	0.7275	0.659080
2	hybrid	5	0.434900	0.573707	0.8225	0.752436
3	xgb	5	0.425000	0.547683	0.7985	0.722698
4	2hop	10	0.310350	0.645685	0.8360	0.730563
5	community	10	0.278700	0.588446	0.7945	0.676510
6	hybrid	10	0.324550	0.699439	0.8745	0.763714
7	xgb	10	0.320000	0.677391	0.8615	0.740339
8	2hop	20	0.208525	0.733211	0.8755	0.734004
9	community	20	0.183550	0.672143	0.8470	0.684491
10	hybrid	20	0.216950	0.773542	0.9000	0.765142
11	xgb	20	0.217375	0.769779	0.8955	0.742237

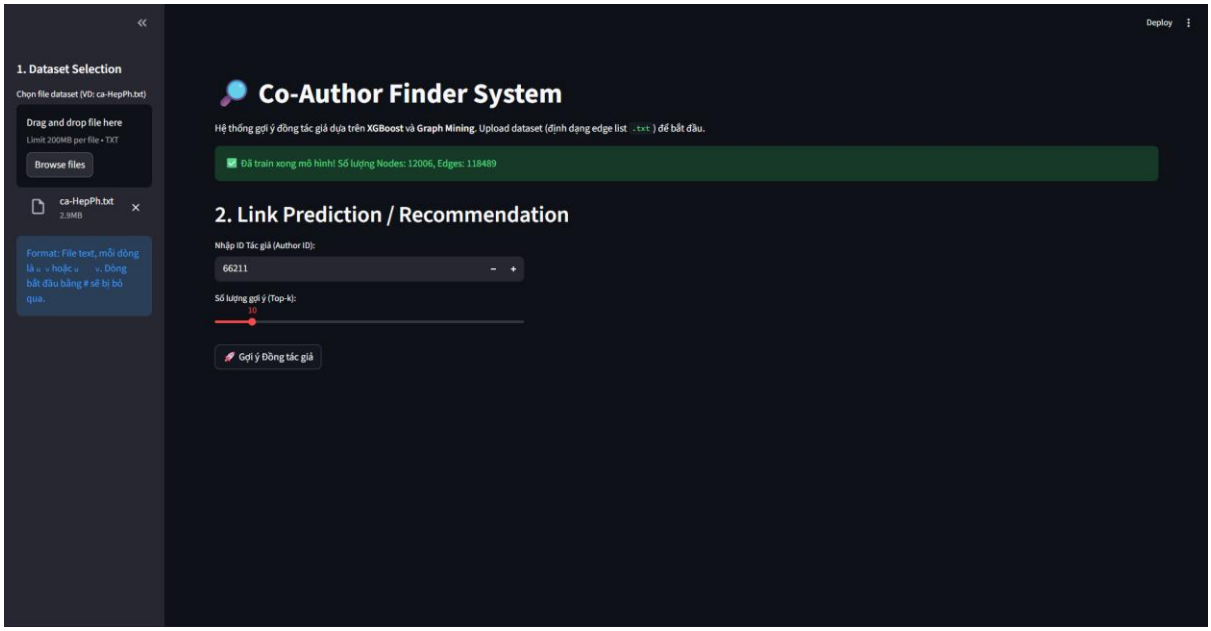
Nhìn vào số liệu, xu hướng chung là khi tăng k từ 5 lên 20 thì P@k giảm (ví dụ 2hop: 0.4188 → 0.2085) vì trả về nhiều kết quả hơn nên tỷ lệ đúng trong top-k bị loãng, trong khi R@k và Hit@k tăng (hybrid: R@k 0.5737 → 0.7735, Hit@k 0.8225 → 0.9000) vì cơ hội “bắt trúng” quan hệ đúng cao hơn.

Về so sánh phương pháp, hybrid đang tốt nhất và ổn định nhất trên hầu hết các k, đặc biệt là các chỉ số quan trọng cho hệ gợi ý là R@k, Hit@k và NDCG@k (ví dụ NDCG@20 = 0.765142 cao nhất, Hit@20 = 0.9000 cao nhất), nghĩa là vừa tìm được nhiều gợi ý đúng hơn vừa sắp xếp đúng người lên phía trên tốt hơn.

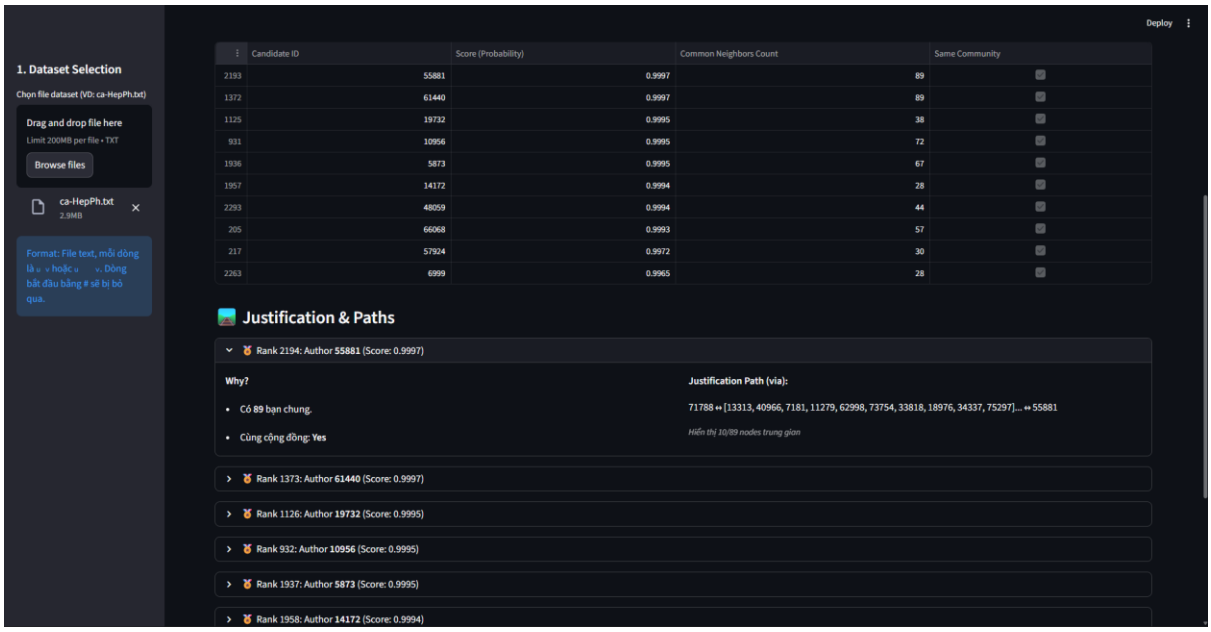
XGBoost bám khá sát hybrid và có P@20 nhỉnh hơn rất nhẹ (0.217375 so với 0.216950) nhưng thua về NDCG@k và Hit@k ở k=20, tức là tỷ lệ đúng tương đương nhưng thứ hạng và độ “trúng ít nhất 1” vẫn kém hơn một chút.

Community Detection là yếu nhất ở cả 4 chỉ số trong hầu hết trường hợp, cho thấy chỉ dựa vào cộng đồng có thể chưa đủ mạnh nếu không kết hợp thêm tín hiệu 2-hop và/hoặc học máy.

4.3 Qualitative Results



Giao diện tìm kiếm gợi ý cho 1 tác giả



Giao diện danh sách các ứng viên kèm theo lời giải thích

4.4 Empirical Analysis

Ablation Study:

Bảng này là bảng Ablation Study dùng để đo “đóng góp” của từng lựa chọn trong hệ gợi ý bằng cách lấy một mô hình chuẩn làm baseline rồi so sánh các biến thể khác trên cùng tập 2000 user

	variant	k	P@k	R@k	Hit@k	NDCG@k	P_base	R_base	Hit_base	NDCG_base	dP@k	dR@k	dHit@k	dNDCG@k	baseline
0	2hop	5	0.418800	0.535075	0.7900	0.719696	0.43490	0.573707	0.8225	0.752436	-0.016100	-0.038632	-0.0325	-0.032740	hybrid(full)
1	community	5	0.378400	0.472603	0.7275	0.659080	0.43490	0.573707	0.8225	0.752436	-0.056500	-0.101104	-0.0950	-0.093356	hybrid(full)
2	hybrid(full)	5	0.434900	0.573707	0.8225	0.752436	0.43490	0.573707	0.8225	0.752436	0.000000	0.000000	0.0000	0.000000	hybrid(full)
3	xgb(full)	5	0.425000	0.547683	0.7985	0.722698	0.43490	0.573707	0.8225	0.752436	-0.009900	-0.026024	-0.0240	-0.029738	hybrid(full)
4	2hop	10	0.310350	0.645685	0.8360	0.730563	0.32455	0.699439	0.8745	0.763714	-0.014200	-0.053753	-0.0385	-0.033151	hybrid(full)
5	community	10	0.278700	0.588446	0.7945	0.676510	0.32455	0.699439	0.8745	0.763714	-0.045850	-0.110993	-0.0800	-0.087204	hybrid(full)
6	hybrid(full)	10	0.324550	0.699439	0.8745	0.763714	0.32455	0.699439	0.8745	0.763714	0.000000	0.000000	0.0000	0.000000	hybrid(full)
7	xgb(full)	10	0.320000	0.677391	0.8615	0.740339	0.32455	0.699439	0.8745	0.763714	-0.004550	-0.022047	-0.0130	-0.023375	hybrid(full)
8	2hop	20	0.208525	0.733211	0.8755	0.734004	0.21695	0.773542	0.9000	0.765142	-0.008425	-0.040332	-0.0245	-0.031138	hybrid(full)
9	community	20	0.183550	0.672143	0.8470	0.684491	0.21695	0.773542	0.9000	0.765142	-0.033400	-0.101400	-0.0530	-0.080651	hybrid(full)
10	hybrid(full)	20	0.216950	0.773542	0.9000	0.765142	0.21695	0.773542	0.9000	0.765142	0.000000	0.000000	0.0000	0.000000	hybrid(full)
11	xgb(full)	20	0.217375	0.769779	0.8955	0.742237	0.21695	0.773542	0.9000	0.765142	0.000425	-0.003763	-0.0045	-0.022905	hybrid(full)

Trong bảng này nhóm đang lấy hybrid làm mô hình “đầy đủ” (baseline) và xem các biến thể còn lại như những phiên bản bị “tắt bớt thành phần” ở mức thô. Kết quả cho thấy khi bỏ phần kết hợp và chỉ giữ 2-hop thì chất lượng giảm vừa phải nhưng khá nhất trong các biến thể đơn giản: ở k=5 giảm khoảng 0.016 P@k, 0.039 R@k, 0.033 Hit@k và 0.033 NDCG@k so với baseline; sang k=10 và k=20 mức giảm vẫn tồn tại nhưng có xu hướng nhỏ hơn ở P@k và NDCG@k. Điều này nói rằng tín hiệu 2-hop tự thân đã mạnh, nhưng khi được kết hợp thêm thành phần cộng đồng trong hybrid thì hệ thống vừa bắt được nhiều “đúng” hơn vừa xếp hạng “đúng” lên cao hơn, nên NDCG@k và Hit@k cải thiện rõ.

Ngược lại, nếu chỉ dùng Community Detection thì tụt mạnh nhất ở mọi k, đặc biệt ở k=5 và k=10: ở k=5 giảm tới 0.0565 P@k, 0.1011 R@k, 0.095 Hit@k và 0.0934 NDCG@k; ở k=10 vẫn giảm khoảng 0.0459 P@k và 0.1110 R@k. Điều này cho thấy “chỉ dựa vào cùng cộng đồng” là chưa đủ để tách top ứng viên tốt, vì Community Detection có thể đúng về mặt “gần lĩnh vực” nhưng không đủ sắc để dự đoán liên kết hợp tác cụ thể, nhất là khi cần đẩy đúng người lên top đầu.

Với XGBoost, kết quả bám khá sát baseline nhưng nhìn chung không vượt được hybrid(full) theo các tiêu chí quan trọng về bao phủ và thứ hạng. Ở k=5 và k=10, xgb giảm nhẹ so với baseline cả P@k, R@k, Hit@k và NDCG@k; ở k=20 thì P@k nhỉnh hơn rất nhỏ (+0.000425) nhưng R@k, Hit@k và đặc biệt NDCG@k vẫn thấp hơn, nghĩa là mô hình học máy có thể “đúng tương đương về tỷ lệ” khi list dài, nhưng khả năng xếp đúng người lên cao chưa tốt bằng hybrid(full).

Failure Analysis:

- Cold-start author: tác giả degree thấp hoặc không nằm trong core → không có đủ 2-hop candidates → không gợi ý được hoặc gợi ý rất kém.
- Hub authors: tác giả/cộng đồng có degree cực cao → 2-hop tạo ra candidate quá nhiều và nhiều “bạn chung phổ biến” → mô hình dễ bị kéo về các ứng viên nổi tiếng nhưng không thật sự liên quan
- Cross-community collaborations: hợp tác tương lai xảy ra giữa 2 cộng đồng khác nhau → các đặc trưng như same_louvain/same_lpa có thể “phạt” sai, làm Community+RA giảm điểm.

Error Analysis:

False Negatives:

- Vấn đề Cold-start: Các tác giả mới hoặc ít bài báo có vector đặc trưng gần như bằng 0, khiến mô hình không đủ dữ kiện để dự đoán.
- Hạn chế của 2-hop: Hệ thống bỏ qua các hợp tác "tầm xa" do không có bạn chung hoặc hợp tác liên ngành (khác cộng đồng) do cơ chế lọc ứng viên ban đầu.

False Positives:
- Popularity Bias: Các node bậc cao (Hubs) tạo ra tín hiệu giả mạnh, khiến mô hình gợi ý kết nối bừa bãi với họ.
- Dense Triangles: Trong các nhóm nghiên cứu chặt chẽ, mô hình dễ lầm tưởng mọi thành viên đều kết nối trực tiếp với nhau.
- Tiềm năng thực tế: Một số lỗi FP thực chất là dự đoán đúng về mặt ngữ nghĩa nhưng liên kết đó chưa kịp xuất hiện trong dữ liệu quá khứ.

5 Discussion

5.1 Giải thích ý nghĩa của kết quả

Kết quả cho thấy mô hình hybrid(full) là phương án ổn định nhất khi xét đồng thời cả khả năng “đúng” trong top-k và chất lượng thứ hạng. Ở mọi mức k, hybrid(full) luôn đạt Recall, Hit và NDCG cao hơn các biến thể đơn giản, nghĩa là hệ thống không chỉ bắt được nhiều quan hệ hợp tác thật hơn mà còn đưa đúng người lên gần đầu danh sách hơn, đúng với nhu cầu của một ứng dụng gợi ý vì người dùng thường chỉ xem vài gợi ý đầu. Khi tách riêng từng thành phần, 2-hop vẫn cho kết quả khá tốt nhưng tụt nhẹ so với hybrid, chứng tỏ tín hiệu “bạn chung” là nền tảng mạnh. Ngược lại, Community Detection yếu nhất và giảm rõ rệt ở cả P, R, Hit, NDCG, cho thấy chỉ dựa vào việc “cùng cộng đồng” thì chưa đủ để dự đoán một liên kết hợp tác cụ thể; nó phản ánh sự gần nhau về chủ đề nhưng không đủ sắc để xếp đúng ứng viên lên top. XGBoost bám sát hybrid nhưng vẫn kém hơn ở đa số chỉ số, đặc biệt là NDCG và Hit, gợi ý rằng trong cấu hình hiện tại mô hình học máy chưa tận dụng được lợi thế để cải thiện thứ hạng so với cách kết hợp heuristic của hybrid, có thể do bộ feature chưa đủ phân biệt, cách lấy candidate bị giới hạn trong 2-hop, hoặc cách huấn luyện/hiệu chỉnh điểm chưa tối ưu cho mục tiêu top-k ranking.

5.2 Bài học rút ra

Bài học rút ra quan trọng là kết quả tốt nhất đến từ việc kết hợp nhiều tín hiệu bổ sung nhau. 2-hop cung cấp dấu hiệu trực tiếp và dễ giải thích, cộng đồng giúp lọc bối cảnh và giảm nhiễu nhưng không nên dùng độc lập, còn học máy chỉ thực sự phát huy khi feature giàu thông tin và quy trình huấn luyện phù hợp với bài toán xếp hạng top-k. Vì vậy, nếu mục tiêu là một prototype có thể giải thích rõ ràng, hybrid(full) là lựa chọn hợp lý vì vừa mạnh về chất lượng vừa giữ được tính minh bạch để tạo justification paths. Đồng thời, việc P@k giảm khi k tăng nhưng R@k và Hit@k tăng cũng cho thấy trade-off tự nhiên của hệ gợi ý: top-k nhỏ phục vụ “đúng và gọn”, top-k lớn phục vụ “bao phủ”, nên có thể chọn k=10 hoặc k=20.

5.3 So với mục tiêu ban đầu

Làm tốt: xây dựng được prototype gợi ý co-author theo đúng deliverables: danh sách top-K, có giải thích bằng bạn chung, có so sánh nhiều phương pháp và đánh giá định lượng theo K.

Hạn chế: ML chưa vượt baseline; cộng đồng Louvain/LPA chưa phát huy tác dụng; không tìm được dữ liệu có metadata phù hợp, nên giải thích chỉ dựa trên cấu trúc mạng.

6. Conclusion and Future Work

6.1 Conclusion

Đồ án đã xây dựng được một prototype gợi ý đồng tác giả, đồng thời cung cấp đầu ra gồm danh sách gợi ý top k và phần giải thích dựa trên các tín hiệu có thể truy vết như bạn chung và thông tin cộng đồng. Về kết quả định lượng, phương pháp hybrid(full) thể hiện tốt nhất và ổn định nhất trên các thước đo top k như Precision@k, Recall@k, Hit@k và NDCG@k, cho thấy hệ thống vừa có khả năng bắt được nhiều quan hệ hợp tác đúng hơn vừa xếp đúng ứng viên lên đầu danh sách tốt hơn so với các biến thể đơn lẻ. Phần ablation ở mức so sánh giữa các method lớn cũng chỉ ra rằng 2-hop là tín hiệu nền tảng mạnh, Community-overlay không đủ hiệu quả khi dùng độc lập, và XGBoost bám sát nhưng chưa vượt được hybrid trong cấu hình hiện tại. Nhìn chung, mục tiêu xây dựng một hệ gợi ý có đánh giá rõ ràng, hoạt động trên dữ liệu tương lai và có giải thích trực quan đã đạt được ở mức tốt, đặc biệt phù hợp với yêu cầu prototype ứng dụng.

6.2 Future Work

Tăng độ chính xác: dùng time-based split đúng theo thời gian; cải thiện negative sampling; đưa community vào ML như feature có trọng; bổ sung vài feature cấu trúc mạnh.

Mở rộng dữ liệu: Tìm được tập dữ liệu có metadata phù hợp hơn.

Thêm module mới: thử node embeddings/GNN (Node2Vec, GraphSAGE/GCN)

Đánh giá sâu hơn: ablation + error analysis theo nhóm tác giả và kiểm định thống kê khi so sánh model.

References

[1] https://www.kaggle.com/datasets/wolfram77/graphs-collaboration?select=ca-HepPh.txt

[2] https://arxiv.org/abs/0803.0476

[3] https://snap.stanford.edu/data/ca-HepPh.html

[4] https://www.sciencedirect.com/science/article/abs/pii/S0378873303000091

[5] https://web.cs.ucla.edu/~yzsun/classes/2014Spring_CS7280/Papers/Link%20Prediction/p556-liben-nowell.pdf

[6] https://dl.acm.org/doi/10.1145/582415.582418

Appendices

A. Technical Details

Link Github: https://github.com/chuongtran145/IS353_Co-Author_Finder_App

B. Project Planning

7.1 Timeline

Giai đoạn chuẩn bị (Tuần 1-2): Phân tích cấu trúc file ca-HepTh (node ID, edge list). Lựa chọn và cài đặt các thư viện cần thiết (NetworkX, python-louvain). Tiến hành thiết kế UI/UX cho úng dụng.

Giai đoạn phát triển(Tuần 3-8): Tiến hành xây dựng đồ thị. Triển khai thuật toán 2-Hop Proximity để tạo điểm dự đoán baseline. Triển khai Adamic-Adar Index (Link Prediction có trọng số). Triển khai Community Detection và tính Community Overlap (Jaccard Index).

Giai đoạn báo cáo(Tuần 9-10): Chuẩn bị báo cáo cuối kỳ và bản trình bày demo.

7.2 Team Responsibilities

Chương: Tiến hành thu thập data, triển khai các thuật toán cho phần backend và viết báo cáo

Vĩnh và Bắc: Thiết kế UI/UX và tiến hành triển khai các thuật toán cho phần backend ứng dụng