**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**---------------------------------------------------**

**A blue and black logo

AI-generated content may be incorrect.**

**HV: LÝ MINH AN  
MSHV: 2370179**

**GVHD: TS. PHAN TRỌNG NHÂN**

**ĐỀ CƯƠNG LUẬN VĂN**

**Đề tài: Xây dựng hệ thống đề xuất lộ trình học tập   
cá nhân hóa dựa trên Đồ Thị Tri Thức   
và Mô Hình Ngôn Ngữ Lớn**

*TP. Hồ Chí Minh, tháng 5 năm 2025*

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**---------------------------------------------------**

**A blue and black logo

AI-generated content may be incorrect.**

**HV: LÝ MINH AN  
MSHV: 2370179**

**GVHD: TS. PHAN TRỌNG NHÂN**

**ĐỀ CƯƠNG LUẬN VĂN**

**Đề tài: Xây dựng hệ thống đề xuất lộ trình học tập   
cá nhân hóa dựa trên Đồ Thị Tri Thức   
và Mô Hình Ngôn Ngữ Lớn**

*TP. Hồ Chí Minh, tháng 5 năm 2025*

LỜI CẢM ƠN

Em xin chân thành gửi lời cảm ơn đến thầy, TS. Phan Trọng Nhân, đã hướng dẫn và giúp đỡ em những kiến thức và kinh nghiệm hữu ích để em có thể hoàn thành Đề cương luận văn này tại Trường Đại học Bách Khoa – Đại học Quốc gia Thành phố Hồ Chí Minh. Thầy đã tận tình theo dõi và đưa ra các chia sẻ vô cùng giá trị giúp em hoàn thiện ý tưởng, cải thiện cấu trúc khoa học cho Đề cương luận văn này.

Bản thân em đã cố gắng hoàn thiện đề cương luận văn này một cách chỉn chu nhất về cả nội dung và hình thức trình bày. Trong quá trình thực hiện, khó tránh khỏi thiếu sót, cũng như các điểm cần cải tiến. Các đóng góp và đề xuất của thầy TS. Phan Trọng Nhân nói riêng và hội đồng bảo vệ nói chung chắc chắn sẽ là các bài học quý báu giúp em cải tiến hơn ở giai đoạn luận văn tiếp theo. Cuối cùng, em xin chúc các thầy cô nhiều sức khoẻ, có được nhiều niềm vui trong công việc và cuộc sống.

**Trân trọng!**

MỤC LỤC ĐỀ CƯƠNG

[LỜI CẢM ƠN 3](#_Toc199451094)

[MỤC LỤC ĐỀ CƯƠNG 4](#_Toc199451095)

[DANH MỤC HÌNH 7](#_Toc199451096)

[DANH MỤC BẢNG 7](#_Toc199451097)

[CHƯƠNG 2 GIỚI THIỆU 8](#_Toc199451098)

[1.1. BỐI CẢNH VÀ ĐỘNG LỰC NGHIÊN CỨU 8](#_Toc199451099)

[1.2. MỤC TIÊU NGHIÊN CỨU 10](#_Toc199451100)

[1.3. ĐỐI TƯỢNG NGHIÊN CỨU 11](#_Toc199451101)

[1.3.1. KG trong hệ thống đề xuất lộ trình học tập 11](#_Toc199451102)

[1.3.2. LLM và LlamaIndex trong hệ thống đề xuất lộ trình học tập 11](#_Toc199451103)

[1.3.3. Hệ thống đề xuất lộ trình học tập cá nhân hóa 12](#_Toc199451104)

[1.4. PHẠM VI VÀ GIỚI HẠN 12](#_Toc199451105)

[1.4.1. Phạm vi nghiên cứu 12](#_Toc199451106)

[1.4.2. Giới hạn nghiên cứu 13](#_Toc199451107)

[1.5. ĐÓNG GÓP KHOA HỌC VÀ THỰC TIỄN 14](#_Toc199451108)

[1.5.1. Đóng góp khoa học 14](#_Toc199451109)

[1.5.2. Đóng góp thực tiễn 15](#_Toc199451110)

[1.6. CẤU TRÚC ĐỀ CƯƠNG 16](#_Toc199451111)

[CHƯƠNG 3 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ 19](#_Toc199451112)

[2.1. QUY TRÌNH HỌC TRỰC TUYẾN CHUẨN 19](#_Toc199451113)

[**2.1.1** **Giới thiệu** 19](#_Toc199451114)

[**2.1.2** **Các bước trong quy trình** 19](#_Toc199451115)

[2.2. KNOWLEDGE GRAPH VÀ ỨNG DỤNG TRONG GIÁO DỤC 22](#_Toc199451116)

[2.2.1 Mô tả kiến trúc đồ thị tri thức 22](#_Toc199451117)

[2.2.2 Ứng dụng 25](#_Toc199451118)

[2.3. MÔ HÌNH NGÔN NGỮ LỚN VÀ LLAMA-INDEX 28](#_Toc199451119)

[2.3.1. Tổng quan về LLM 28](#_Toc199451120)

[2.3.2. Cơ chế tích hợp LLM với Llama-index 31](#_Toc199451121)

[2.4. CÔNG NGHỆ NEO4J 32](#_Toc199451122)

[2.4.1. Tổng quan và kiến trúc 32](#_Toc199451123)

[2.4.2. Ứng dụng trong giáo dục và quản lý dữ liệu phân cấp 34](#_Toc199451124)

[2.5. NHỮNG THƯ VIỆN VÀ CÔNG NGHỆ SỬ DỤNG TRONG QUY MÔ LUẬN VĂN 35](#_Toc199451125)

[2.5.1. Thư viện Python 35](#_Toc199451126)

[2.5.2. Công nghệ bổ trợ 36](#_Toc199451127)

[2.5.3. Kiến trúc tích hợp 36](#_Toc199451128)

[CHƯƠNG 4 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 37](#_Toc199451129)

[3.1. BÀI TOÁN ĐỀ XUẤT LỘ TRÌNH HỌC TẬP CÁ NHÂN HÓA 37](#_Toc199451130)

[3.1.1. Định nghĩa bài toán 37](#_Toc199451131)

[3.1.2. Thách thức kỹ thuật 37](#_Toc199451132)

[3.1.3. Tầm nhìn và mục tiêu cụ thể 38](#_Toc199451133)

[3.1.4. Ví dụ bài toán minh họa 39](#_Toc199451134)

[3.2. QUY TRÌNH NGHIỆP VỤ 39](#_Toc199451135)

[3.2.1. So sánh với quy trình học trực tuyến chuẩn 39](#_Toc199451136)

[3.2.2. Quy trình hỗ trợ bởi hệ thống đề xuất 40](#_Toc199451137)

[3.3. GIẢI PHÁP ĐỀ XUẤT 48](#_Toc199451138)

[3.3.1 Minh họa quy trình 48](#_Toc199451139)

[3.3.2 Kiến trúc hệ thống tích hợp 52](#_Toc199451140)

[3.3.3 Quy trình xử lý từ dữ liệu thô đến đề xuất lộ trình 54](#_Toc199451141)

[3.4. YÊU CẦU HỆ THỐNG 56](#_Toc199451142)

[3.4.1. Yêu cầu chức năng 56](#_Toc199451143)

[3.4.2. Yêu cầu dữ liệu 57](#_Toc199451144)

[3.4.3. Yêu cầu phi chức năng 57](#_Toc199451145)

[3.5. THIẾT KẾ HỆ THỐNG 58](#_Toc199451146)

[3.5.1. Chi tiết các thành phần hệ thống 58](#_Toc199451147)

[3.5.2. Các tính năng nâng cao 60](#_Toc199451148)

[3.5.3. Cấu trúc nút và thuộc tính 61](#_Toc199451149)

[3.5.4. Các loại mối quan hệ 66](#_Toc199451150)

[3.5.5. Kiểm tra, chuẩn hóa và tải dữ liệu 70](#_Toc199451151)

[3.5.6. Thiết kế thuật toán đề xuất 72](#_Toc199451152)

[CHƯƠNG 5 HIỆN THỰC THỬ NGHIỆM 83](#_Toc199451153)

[4.1. XÂY DỰNG NGUYÊN MẪU 83](#_Toc199451154)

[4.1.1. Thu thập dữ liệu mẫu từ nguồn mở 83](#_Toc199451155)

[4.1.2. Triển khai dữ liệu 84](#_Toc199451156)

[4.2. TÍCH HỢP LLAMA-INDEX VÀ LLM 85](#_Toc199451157)

[4.2.1. Cấu hình LLM 85](#_Toc199451158)

[4.2.3. Tối ưu truy vấn đồ thị 87](#_Toc199451159)

[CHƯƠNG 6 ĐÁNH GIÁ VÀ SO SÁNH 88](#_Toc199451160)

[5.1. THIẾT LẬP THỬ NGHIỆM 88](#_Toc199451161)

[5.1.1. Bộ dữ liệu thử nghiệm 88](#_Toc199451162)

[5.1.2. Tiêu chí đánh giá 90](#_Toc199451163)

[5.1.3. Thuật toán nguồn mở để so sánh 91](#_Toc199451164)

[5.2. KẾT QUẢ VÀ PHÂN TÍCH 95](#_Toc199451165)

[5.2.1. So sánh với thuật toán nguồn mở 95](#_Toc199451166)

[5.2.2. Phân tích hạn chế 97](#_Toc199451167)

[CHƯƠNG 7 Kết luận 99](#_Toc199451168)

[6.1. Kết quả đạt được 99](#_Toc199451169)

[6.2. Hướng phát triển 99](#_Toc199451170)

[TÀI LIỆU THAM KHẢO 101](#_Toc199451171)

[PHỤ LỤC 104](#_Toc199451172)

[PHỤ LỤC A: Mã nguồn triển khai thử nghiệm 104](#_Toc199451173)

[PHỤ LỤC B: Ví dụ lộ trình học tập được đề xuất 104](#_Toc199451174)

DANH MỤC HÌNH

[**Hình 1.1. Sự gia tăng của học tập trực tuyến và nhu cầu cá nhân hóa** (Nguồn: Coursera ) 8](#_Toc199451074)

[**Hình 2.1. Sơ đồ quy trình đào tạo trực tuyến chuẩn** (Nguồn: iNACOL [17] ) 20](#_Toc199451075)

[**Hình 3.1 Giai đoạn 1 Tiền xử lý xây dựng dữ liệu nền tảng KG** 44](#_Toc199451076)

[**Hình 3.2 Giai đoạn 2: Vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa** 47](#_Toc199451077)

[**Hình 3.3 Sơ đồ bối cảnh hệ thống** 49](#_Toc199451078)

### 3.3. Ki?n tr£c t?ng th? h? th?ng

H? th?ng ?? xu?t l? trnh h?c t?p c  nhn h¢a ???c xy d?ng theo m hnh ki?n tr£c 3 l?p hi?n ??i g?m: - T?ng giao di?n (Frontend): Next.js 14 (React) xy d?ng ?ng d?ng web chu?n, h? tr? giao di?n h?i tho?i ?a l??t, visualize l? trnh h?c, dashboard monitoring. C?u tr£c th? m?c: /frontend/app/, /frontend/components/ (PathVisualizer, ChatInterface), /frontend/lib/api.ts. K?t n?i v?i backend qua RESTful API (JSON). - T?ng x? lì logic (Backend Python): FastAPI (Python 3.9), t¡ch h?p Gemini API, LlamaIndex, Neo4j driver, TensorFlow/PyTorch cho DKT model, pm4py cho process mining. Module ch¡nh: api.py, adaptive\_path\_planner.py, hybrid\_retriever.py, knowledge\_tracing.py, graph\_recommender.py, explainability.py, process\_mining.py, neo4j\_manager.py, evaluation\_metrics.py, learner\_state.py, learner\_profile\_manager.py. C?u tr£c rä r ng: /backend/src/ S? d?ng Uvicorn ASGI, background threads cho lifecycle (init Neo4j/Gemini). - T?ng d? li?u (Database + Indexing): Neo4j AuraDB cloud, LlamaIndex (vector+property graph adapter). Nodes: KnowledgeNode, Student, LearningData (h? s? & d? li?u h?c t?p). Relationships: REQUIRES, NEXT, IS\_PREREQUISITE\_OF, etc. v?i weight & dependency. Constraints/Indexes ???c setup qua script neo4j\_schema\_setup.py. LlamaIndex h? tr? PropertyGraphIndex, VectorStoreIndex cho semantic queries.

Lu?ng d? li?u ch¡nh: 1. Kh?i t?o Knowledge Graph: SPR Generator/Validation ? CSV ? Import v o Neo4j ? Index/Constraint. 2. Sinh l? trnh h?c t?p: Frontend g?i input; Backend l?y profile, DKT predict mastery; Hybrid retriever l?y context; Adaptive planner/A\*+RL sinh l? trnh t?i ?u; GNN recommender fine-tune; Explainability gi?i th¡ch l? trnh; Frontend visualize. 3. Sinh n?i dung h?c t?p/? nh gi : Backend sinh content v quiz v?i Gemini API; c?p nh?t h? s? h?c vin; process mining phn t¡ch learning logs.

Deployment: - Dev: API backend t?i localhost:8000, frontend t?i :3000, Neo4j AuraDB cloud. - Prod (future): Docker/Kubernetes, Vercel/Netlify (frontend), Neo4j AuraDB prod tier.

API Endpoints m?u

|  |  |  |
| --- | --- | --- |
| Endpoint | Method | M t? |
| /api/status | GET | Health check/api status |
| /api/generate\_path | POST | Sinh l? trnh h?c t?p |
| /api/recommend\_next | POST | ?? xu?t concept ti?p theo |

| /api/generate\_content | POST | Sinh n?i dung b i h?c | | /api/generate\_quiz | POST | Sinh b i ki?m tra |

|  |  |  |
| --- | --- | --- |
| /api/explain\_path | POST | Gi?i th¡ch l? trnh |
| /api/update\_profile | PUT | C?p nh?t profile h?c vin |

S? ?? ?? xu?t: - S? ?? C4: Context ? Container ? Component - Sequence diagram: Learning path flow - Deployment diagram: Dev/Prod

### B?ng so s nh c“ng ngh? c c l?p

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Technology | Layer | Strengths | Trade-offs | Performance/Scale | Notes |
| Next.js 14 | Frontend | React ecosystem, server components, routing, SSR/SSG, great DX | Requires Node tooling; SSR adds infra complexity | Handles modern SPA/SSR at scale with CDN/edge support | Fits chat UI + visualization |
| FastAPI | Backend | High performance (uvicorn), async I/O, OpenAPI docs, type hints | Async patterns require care; background tasks are basic | Good throughput/latency for REST microservices | Simple to deploy; great dev experience |
| Neo4j AuraDB | Data/KG | Managed graph DB, Cypher, indexes/constraints, scalability, reliability | Cloud cost; graph modeling learning curve | Scales for KG queries; low-latency managed service | Good for prerequisites and path traversal |
| Gemini API | GenAI | Multimodal LLM, strong generation quality, flexible prompting | Token cost; rate limits; requires prompt governance | Scales via managed API; latency depends on model class | Use with retrieval grounding + safety filters |

B?ng ?? xu?t: So s nh cng ngh? t?ng l?p (Next.js, FastAPI, Neo4j AuraDB, Gemini API), API ? p ?ng, performance.

[**Hình 3.4 Sơ đồ các đối tượng chính của hệ thống** 50](#_Toc199451079)

[**Hình 3.5 Sơ đồ thành phần quy trình hệ thống** 51](#_Toc199451080)

### 3.5.6. Thu?t to n ?? xu?t l? trnh nng cao

#### 3.5.6.1. Hybrid Path Planning: A\* + Reinforcement Learning

- K?t h?p A\* (t?i ?u to n c?c) v?i RL agent (PPO) cho t?i ?u c?c b?. - A\* tm ???ng ?i t?i ?u trn ?? th? tri th?c v?i h m heuristic custom:

h(n) = w\_1 \cdot (1 - \text{similarity}(n, goal)) + w\_2 \cdot \frac{\text{difficulty}(n)}{\text{skill}(student)} + w\_3 \cdot \frac{\text{time}(n)}{\text{available}\_\text{time}}

Trong ?¢: similarity l Jaccard gi?a tags concept, difficulty l ?? kh¢ concept, skill l trnh ?? h?c vin, time l th?i gian ??c t¡nh. - RL agent ?i?u ch?nh l? trnh c?c b? theo tr?ng th i mastery, hi?u su?t, th?i gian. - PPO reward function:

r\_t = \alpha \cdot \Delta \text{mastery} + \beta \cdot \text{efficiency} - \gamma \cdot \text{deviation}

- Python code m?u:

class AdaptivePathPlanner:

def plan\_path(self, learner\_id, start\_node, goal\_node):

global\_path = self.a\_star\_custom(start\_node, goal\_node, learner\_id)

refined\_path = self.rl\_agent.refine(global\_path, learner\_state)

weights = self.compute\_dynamic\_weights(learner\_state)

return refined\_path

---

#### 3.5.6.2. GraphRAG Retrieval Algorithm

- Thu?t to n GraphRAG cho hybrid retrieval: semantic vector search, graph traversal, community summary. - Quy trnh workflow: 1. Classify query (structural/semantic/overview) 2. Retrieve b?ng cypher, embedding, ho?c LLM summary 3. Fuse context v profile h?c vin, tr? v? k?t qu?. - Code m?u:

class HybridRetriever:

def retrieve(self, query, learner\_id, context\_type):

strategy = self.classify\_query(query)

if strategy == "structural":

return self.graph\_traversal(query)

elif strategy == "semantic":

return self.vector\_search(query)

else:

return self.community\_summary(query)

---

#### 3.5.6.3. GNN-based Recommendation Algorithm

- S? d?ng GCN (graph convolution) + GAT (graph attention) ?? d? ?o n concept ti?p theo trong ?? th? h?c vin?kh i ni?m. - M hnh multi-layer: - Input: student profile, concept properties - 2 GCN layers + 1 GAT layer - Output: link prediction cho ?student ? concept? - Loss: binary cross-entropy trn h?c vin v?i concept ?£ng/sai - Code m?u:

class GNNRecommender:

def recommend(self, student\_id, top\_k=5):

student\_emb = self.model.encode\_student(student\_id)

candidates = self.get\_candidates(student\_id)

scores = [self.model.predict(student\_emb, c) for c in candidates]

return sorted(zip(candidates, scores), reverse=True)[:top\_k]

T i li?u tham kh?o (m?c n y): - Schulman et al. (2017): ?Proximal Policy Optimization? - Microsoft (2024): ?GraphRAG: Unlocking LLM discovery...? - Kipf & Welling (2017): ?Graph Convolutional Networks? - ACM (2025): ?Personalized Learning Path with Time-Aware Attention?

[**Hình 3.6 Sơ đồ thành thành phần chi tiết** 52](#_Toc199451081)

[**Hình 3.7 Sơ đồ kiến trúc hệ thống** 53](#_Toc199451082)

[**Hình 3.8 Quy trình tạo và xác thực dữ liệu** 54](#_Toc199451083)

[**Hình 3.9 Quy trình đề xuất lộ trình học tập** 55](#_Toc199451084)

[**Hình 3.10 Thuật toán A\* tích hợp vào quy trình** 73](#_Toc199451085)

[**Hình 4.1 Trực quan toàn bộ KG** 84](#_Toc199451086)

[**Hình 5.1 Lộ trình thử nghiệm từ nút gốc đến nút lá** 90](#_Toc199451087)

DANH MỤC BẢNG

[**Bảng 2.1 Các thuộc tính chính của nút tri thức** 23](#_Toc199450813)

[**Bảng 2.2 Các thuộc tính chính của mối quan hệ tri thức** 24](#_Toc199450814)

[**Bảng 2.3. Thư viện sử dụng và vai trò trong hệ thống** 35](#_Toc199450815)

### 2.3.4. GraphRAG ? Graph Retrieval-Augmented Generation

GraphRAG l… ki?n tr£c truy v?n tri th?c tiˆn ti?n k?t h?p retrieval t? ?? th? v… LLM generation, ???c Microsoft Research ?? xu?t n?m 2024 nh?m t?ng kh? n?ng reasoning multi-hop [GraphRAG: Unlocking LLM discovery..., Microsoft, 2024].

- Community Detection: S? d?ng Leiden/Louvain ?? chia KG th…nh communities, t?i ?u modularity.

Q = (1/(2m)) \* ?\_{i,j} [ A\_{ij} - (k\_i k\_j)/(2m) ] ?(c\_i, c\_j)

- Community Summarization: D—ng LLM t?o summary (150-200 t?) cho t?ng nh¢m. V¡ d? Prompt: ?Given list {concepts\_list} and {relationships\_list}, write summary describing main theme, key concepts, learning objectives.? - Hybrid Retrieval: K?t h?p semantic vector search, graph traversal (shortest path, subgraph), v… community summary. - ?u ?i?m: Theo Microsoft, GraphRAG t?ng +41% comprehensiveness, +56% multi-hop accuracy, +32% diversity so v?i RAG truy?n th?ng. - ?ng d?ng: `backend/src/hybrid\_retriever.py` g?m functions community detection, LLM-based summarization, hybrid retriever cho c c ki?u truy v?n.

B?ng 2.3: So s nh GraphRAG vs Traditional RAG

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Traditional RAG | GraphRAG | Improvement |
| Comprehensiveness | 58% | 82% | +41% |
| Diversity | 51% | 67% | +32% |
| Multi-hop reasoning | 42% | 66% | +56% |

GraphRAG l… ki?n tr£c truy v?n tri th?c tiˆn ti?n k?t h?p retrieval t? ?? th? v… LLM generation, ???c Microsoft Research ?? xu?t n?m 2024 nh?m t?ng kh? n?ng reasoning multi-hop [GraphRAG: Unlocking LLM discovery..., Microsoft, 2024].

- Community Detection: S? d?ng Leiden/Louvain ?? chia KG th…nh communities, t?i ?u modularity.

Q = (1/(2m)) \* ?\_{i,j} [ A\_{ij} - (k\_i k\_j)/(2m) ] ?(c\_i, c\_j)

- Community Summarization: D—ng LLM t?o summary (150-200 t?) cho t?ng nh¢m. V¡ d? Prompt: ?Given list {concepts\_list} and {relationships\_list}, write summary describing main theme, key concepts, learning objectives.? - Hybrid Retrieval: K?t h?p semantic vector search, graph traversal (shortest path, subgraph), v… community summary. - ?u ?i?m: Theo Microsoft, GraphRAG t?ng +41% comprehensiveness, +56% multi-hop accuracy, +32% diversity so v?i RAG truy?n th?ng. - ?ng d?ng: `backend/src/hybrid\_retriever.py` g?m functions community detection, LLM-based summarization, hybrid retriever cho c c ki?u truy v?n.

B?ng 2.3: So s nh GraphRAG vs Traditional RAG

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Traditional RAG | GraphRAG | Improvement |
| Comprehensiveness | 58% | 82% | +41% |
| Diversity | 51% | 67% | +32% |
| Multi-hop reasoning | 42% | 66% | +56% |

[**Bảng 3.1. Tóm tắt quy trình** 55](#_Toc199450816)

[**Bảng 3.2. Tóm tắt yêu cầu chức năng** 56](#_Toc199450817)

[**Bảng 3.3. Tóm tắt yêu cầu phi chức năng** 58](#_Toc199450818)

[**Bảng 4.1 Giới thiệu các bộ Prompt LLM trong luận văn** 86](#_Toc199450819)

[**Bảng 5.1. Tóm tắt dữ liệu môn học sử dụng cho prototype** 88](#_Toc199450820)

[**Bảng 5.2. Tóm tắt đặc tính nút tri thức trong KG** 89](#_Toc199450821)

# GIỚI THIỆU

## BỐI CẢNH VÀ ĐỘNG LỰC NGHIÊN CỨU

Trong bối cảnh giáo dục hiện đại, học tập cá nhân hóa ngày càng được xem là một giải pháp quan trọng để đáp ứng nhu cầu đa dạng của học sinh. Số liệu cho thấy rằng số lượng học viên học tập trực tuyến ngày càng gia tăng nhanh chóng, tăng hơn 4 lần từ năm 2016 đến năm 2021, đạt hơn 90 triệu học viên vào năm 2021 [9] được thể hiện ở hình 1.1. Và theo báo cáo của UNESCO (2022) [37], hơn 70% học viên trên các nền tảng học trực tuyến như Coursera, Udemy, hay Khan Academy mong muốn [18] có các lộ trình học tập được cá nhân hóa, phù hợp với nhu cầu, trình độ và mục tiêu cá nhân của họ.

A graph of blue bars and numbers

AI-generated content may be incorrect.

**Hình 1.1. Sự gia tăng của học tập trực tuyến và nhu cầu cá nhân hóa** (Nguồn: Coursera )

Học tập cá nhân hóa cho phép điều chỉnh nội dung, phương pháp và tốc độ học tập theo khả năng, sở thích và mục tiêu của từng cá nhân, từ đó tối ưu hóa quá trình học tập và nâng cao hiệu quả giáo dục [18]. Nghiên cứu của RAND Corporation (2017) [31] đã chỉ ra rằng học tập cá nhân hóa có thể cải thiện đáng kể thành tích học tập của học sinh, bất kể trình độ ban đầu của họ [20]. Hiện nay, nhiều hệ thống giáo dục đã áp dụng các công nghệ như hệ thống quản lý học tập (Learning Management Systems - LMS), nền tảng học tập thích ứng (adaptive learning platforms) và hệ thống đề xuất nội dung

Tuy nhiên, các phương pháp truyền thống, chẳng hạn như lọc cộng tác (collaborative filtering) hoặc lọc dựa trên nội dung (content-based filtering), thường gặp khó khăn trong việc xử lý dữ liệu đa chiều và động, bao gồm sự thay đổi liên tục trong sở thích học tập, mục tiêu cá nhân, hoặc trình độ của người học. Hơn nữa, các hệ thống này thường thiếu khả năng hiểu ngữ nghĩa sâu sắc, dẫn đến các đề xuất không thực sự phản ánh nhu cầu thực tế của người học.

Các hệ thống như Khan Academy, Coursera hay Moodle đã tích hợp các tính năng cá nhân hóa ở một mức độ nhất định. Tuy nhiên, các hệ thống này vẫn tồn tại những hạn chế đáng kể, bao gồm:

* Thiếu khả năng biểu diễn cấu trúc kiến thức phức tạp [7]: Hầu hết các hệ thống sử dụng cấu trúc dữ liệu tuyến tính hoặc phân cấp đơn giản, không thể biểu diễn đầy đủ các mối quan hệ phức tạp giữa các khái niệm giáo dục, chẳng hạn như mối quan hệ tiền đề hay phụ thuộc.
* Hạn chế trong tương tác tự nhiên với học sinh [40]: Các hệ thống hiện tại chủ yếu dựa trên các quy tắc cố định hoặc thuật toán đơn giản, không thể hiểu và đáp ứng nhu cầu học tập của học sinh thông qua ngôn ngữ tự nhiên một cách hiệu quả.
* Thiếu tính linh hoạt trong đề xuất lộ trình học tập [1]: Các lộ trình học tập thường được thiết kế dựa trên các kịch bản cố định, không thể điều chỉnh linh hoạt theo tiến độ, phong cách học tập hoặc mục tiêu cá nhân của học sinh.

Những hạn chế này khiến các hệ thống hiện tại chưa thể cung cấp trải nghiệm học tập cá nhân hóa thực sự, dẫn đến nhu cầu cấp thiết về một giải pháp mới, thông minh và linh hoạt hơn. Để giải quyết các vấn đề trên, công nghệ Đồ thị Tri thức (Knowledge Graph) và Mô hình Ngôn ngữ Lớn (Large Language Model - LLM) đã nổi lên như những công cụ đầy tiềm năng. Sự kết hợp của các công nghệ này hứa hẹn mang lại một giải pháp toàn diện, vượt qua các hạn chế của các hệ thống khuyến nghị truyền thống và đáp ứng tốt hơn nhu cầu của người học trong môi trường giáo dục trực tuyến.

Neo4j – một cơ sở dữ liệu đồ thị mạnh mẽ [23] đảm bảo khả năng lưu trữ và truy vấn dữ liệu đồ thị hiệu quả, trong khi Llama-index hỗ trợ tích hợp LLM vào quá trình xử lý dữ liệu học tập tạo tiềm năng mở rộng vô cùng lớn.

Động lực chính của nghiên cứu xuất phát từ mục tiêu xây dựng một hệ thống có khả năng:

1. Thu thập và xử lý dữ liệu học tập từ các nguồn học thuật.
2. Thiết kế một quy trình sử dụng KG làm cơ sở cho hệ thống quản lý tri thức, quản lý tài nguyên cho tổ chức giáo dục.
3. Tận dụng LLM để phân tích nhu cầu học tập tự động thông qua truy vấn ngôn ngữ tự nhiên.
4. Đề xuất các lộ trình học tập cá nhân hóa, giảm gánh nặng cho giáo viên trong việc lập kế hoạch giảng dạy cá nhân hóa hoặc giúp cải tiến hóa quy trình giảng dạy, quản lý học tập trong các tổ chức giáo dục.

Nghiên cứu này không chỉ hướng đến việc đóng góp về mặt lý thuyết trong lĩnh vực giáo dục cá nhân hóa mà còn mang lại giá trị thực tiễn, hỗ trợ nâng cao chất lượng học tập trực tuyến và cải thiện trải nghiệm người dùng.

## MỤC TIÊU NGHIÊN CỨU

Dựa trên bối cảnh và động lực nghiên cứu đã trình bày, nghiên cứu hướng đến xây dựng một hệ thống đề xuất lộ trình học tập cá nhân hóa dựa trên đồ thị tri thức (KG) và Mô hình Ngôn ngữ Lớn (LLM), với việc ứng dụng công nghệ Neo4j và Llama-index Property Graph Index.

Để đảm bảo chất lượng và khả năng mở rộng, quy trình nghiên cứu được chia thành hai giai đoạn chính:

1. Tiền xử lý xây dựng dữ liệu nền tảng Knowledge Graph
2. Vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa. Các mục tiêu cụ thể bao gồm:

Dựa trên hai giai đoạn này, các mục tiêu cụ thể của nghiên cứu được xác định như sau:

* Thiết kế và triển khai Đồ thị Tri thức trong Neo4j  
  Tham khảo theo *Chen và cộng sự* [7], xây dựng cấu trúc Đồ thị Tri thức để biểu diễn các khái niệm giáo dục, mối quan hệ giữa chúng, và các thuộc tính như mục tiêu học tập, mức độ kỹ năng, và tiền đề, tạo nền tảng cho việc đề xuất lộ trình học tập logic.
* Tích hợp Mô hình Ngôn ngữ Lớn với Đồ thị Tri thức  
  Sử dụng LlamaIndex Property Graph Index để kết nối LLM với Đồ thị Tri thức, hỗ trợ xử lý yêu cầu học tập bằng ngôn ngữ tự nhiên như theo đề xuất của *Huang và cộng sự* [16], cung cấp các đề xuất cá nhân hóa chính xác.
* Phát triển thuật toán đề xuất lộ trình học tập  
  Xây dựng thuật toán tận dụng Đồ thị Tri thức và LLM để tạo ra các lộ trình học tập phù hợp với trình độ, phong cách học tập, và mục tiêu của từng học viên.
* Tiền xử lý dữ liệu giáo dục  
  Chuyển đổi tài liệu giáo dục thành các tệp dữ liệu có cấu trúc (CSV) thông qua các công cụ như SPR Generator và SPR Validation, đảm bảo dữ liệu phù hợp để xây dựng Đồ thị Tri thức trong Neo4j.
* Minh họa kiến trúc hệ thống bằng mô hình C4  
  Trình bày kiến trúc hệ thống thông qua các biểu đồ ngữ cảnh, container, và thành phần theo mô hình C4, làm rõ cách các thành phần như Neo4j, LLM, và giao diện người dùng tương tác.
* Thiết kế biểu đồ lộ trình học tập cá nhân hóa  
  Đề xuất các sơ đồ trực quan để thể hiện lộ trình học tập được cá nhân hóa, giúp học viên dễ dàng theo dõi tiến độ và các bước tiếp theo.
* Đánh giá hiệu quả của hệ thống  
  Thực hiện các thử nghiệm để đánh giá độ chính xác, mức độ hài lòng của người dùng, tham khảo và khả năng mở rộng của hệ thống theo *Pane và cộng sự* [27], với các tiêu chí như thời gian phản hồi và tính phù hợp của lộ trình học tập.
* So sánh với các hệ thống hiện có  
  Phân tích và so sánh hệ thống đề xuất với các nền tảng hiện tại như Khan Academy hoặc Coursera, nhằm làm nổi bật các ưu điểm và sự đổi mới của phương pháp tiếp cận này.

Các mục tiêu này sẽ được triển khai trong các chương tiếp theo, từ cơ sở lý thuyết (Chương 2), thiết kế hệ thống (Chương 3), triển khai và thử nghiệm (Chương 4), đến đánh giá (Chương 5).

## ĐỐI TƯỢNG NGHIÊN CỨU

Nghiên cứu tập trung vào ba đối tượng chính: Đồ thị Tri thức (Knowledge Graph - KG) trong hệ thống đề xuất lộ trình học tập cá nhân hóa, Mô hình Ngôn ngữ Lớn (Large Language Model - LLM) kết hợp với LlamaIndex, và hệ thống đề xuất lộ trình học tập cá nhân hóa. Các đối tượng này đóng vai trò nền tảng trong việc xây dựng một giải pháp học tập thích ứng, khắc phục các hạn chế của các hệ thống hiện tại và đáp ứng nhu cầu cá nhân hóa trong giáo dục trực tuyến.

### KG trong hệ thống đề xuất lộ trình học tập

Đồ thị Tri thức (KG) là một cấu trúc dữ liệu biểu diễn các khái niệm và mối quan hệ giữa chúng dưới dạng mạng lưới, cho phép hệ thống hiểu và tổ chức kiến thức một cách có hệ thống. Trong bối cảnh giáo dục, KG đóng vai trò quan trọng trong việc mô hình hóa các nội dung học tập, từ các khái niệm cơ bản đến nâng cao, cùng với các mối liên hệ như tiền đề, phụ thuộc, hoặc trình tự học tập , như được *Abu-Salih và Alotaibi* [1] định nghĩa trong nghiên cứu về ứng dụng KG trong giáo dục. Sự biểu diễn này giúp hệ thống đề xuất các lộ trình học tập phù hợp với trình độ và mục tiêu của từng học viên, khắc phục hạn chế về cấu trúc dữ liệu tuyến tính của các hệ thống truyền thống, theo ghi nhận của *Zhang và cộng sự* [40] trong bài viết về các thách thức trong học tập thích ứng.

Trong nghiên cứu này, KG được triển khai bằng Neo4j, một cơ sở dữ liệu đồ thị mạnh mẽ, hỗ trợ lưu trữ tri thức và truy vấn dữ liệu phức tạp một cách hiệu quả. KG cho phép biểu diễn các khái niệm giáo dục (như "SQL SELECT" hoặc "GROUP BY") và các mối quan hệ như REQUIRES hoặc NEXT, đảm bảo rằng các đề xuất lộ trình học tập mang tính logic và phù hợp với bối cảnh giáo dục.

### LLM và LlamaIndex trong hệ thống đề xuất lộ trình học tập

Mô hình Ngôn ngữ Lớn (LLM) là công nghệ trí tuệ nhân tạo tiên tiến, nổi bật với khả năng xử lý và sinh ra văn bản tự nhiên, từ đó hỗ trợ việc phân tích nhu cầu học tập của người dùng qua các câu hỏi ngôn ngữ thông thường, theo *Huang và cộng sự* [16] trong khảo sát về LLM trong giáo dục cá nhân hóa. Trong nghiên cứu này, LLM được sử dụng để hiểu các yêu cầu của học viên (ví dụ, “Tôi muốn học SQL cơ bản”) và tạo ra các đề xuất phù hợp dựa trên dữ liệu từ KG.

LlamaIndex, một framework tích hợp dữ liệu, đóng vai trò cầu nối giữa LLM và KG, cho phép truy xuất thông tin từ Neo4j một cách hiệu quả. Cụ thể, LlamaIndex Property Graph Index hỗ trợ truy vấn đồ thị, trong khi VectorStoreIndex thực hiện tìm kiếm ngữ nghĩa dựa trên lịch sử học tập, nâng cao tính chính xác của các đề xuất. Sự kết hợp này giúp hệ thống vượt qua hạn chế về tương tác tự nhiên của các hệ thống hiện tại, mang lại trải nghiệm học tập linh hoạt và cá nhân hóa hơn.

### Hệ thống đề xuất lộ trình học tập cá nhân hóa

Hệ thống đề xuất lộ trình học tập cá nhân hóa là đối tượng nghiên cứu trung tâm của luận văn, được thiết kế để cung cấp các chuỗi học tập tối ưu cho từng học viên dựa trên trình độ, phong cách học tập, và mục tiêu cá nhân. Kiến trúc hệ thống được minh họa bằng mô hình C4, giúp các đối tượng được thể hiện rõ ràng và làm rõ sự tương tác giữa các thành phần.

Hệ thống không chỉ cải thiện trải nghiệm học tập mà còn hỗ trợ giáo viên trong việc lập kế hoạch giảng dạy, góp phần nâng cao chất lượng giáo dục trực tuyến.

## PHẠM VI VÀ GIỚI HẠN

### Phạm vi nghiên cứu

Nghiên cứu tập trung vào việc phát triển một hệ thống mẫu (prototype) nhằm chứng minh khả năng kết hợp KG và LLM trong việc đề xuất lộ trình học tập cá nhân hóa, hỗ trợ học viên và giáo viên trong môi trường giáo dục trực tuyến.

Phạm vi nghiên cứu được xác định như sau:

* Đối tượng phục vụ:

Hệ thống hướng đến học viên với nhu cầu học tập đa dạng về trình độ, mục tiêu, và phong cách học tập, đồng thời hỗ trợ giáo viên trong việc lập kế hoạch giảng dạy và theo dõi tiến độ học viên. Nghiên cứu tập trung vào bối cảnh giáo dục trực tuyến, nơi nhu cầu cá nhân hóa ngày càng tăng, như đã nêu trong mục 1.1.

* Công nghệ sử dụng:
  + Chạy thử nghiệm trên nền tảng Google Colab, sử dụng ngôn ngữ Python.
  + Neo4j: Được sử dụng để xây dựng và quản lý Đồ thị Tri thức, biểu diễn các khái niệm giáo dục dưới dạng nút (KnowledgeNode).
  + LlamaIndex Property Graph Index: Tích hợp LLM với KG, sử dụng VectorStoreIndex để nâng cao tính chính xác của đề xuất.
* Chức năng chính:

Hệ thống cho phép học viên nhập thông tin về mục tiêu học tập, trình độ hiện tại, và ngữ cảnh. Dựa trên dữ liệu này, hệ thống tự động đề xuất lộ trình học tập cá nhân hóa, sinh nội dung học tập phù hợp, và cung cấp gợi ý khắc phục khi học viên gặp khó khăn.

* Lĩnh vực giáo dục:

Nghiên cứu thử nghiệm trên dữ liệu từ ngành Hệ thống Thông tin Quản lý (**Management Information Systems - MIS**), lựa chọn ba chủ đề như SQL, Hệ thống quản lý tri thức hiện đại và Business Intelligence (BI), nhằm đảm bảo tính thực tiễn của hệ thống trong một lĩnh vực có cấu trúc kiến thức rõ ràng và trong một giới hạn phù hợp xây dựng nguyên mẫu của đề cương.

* Đánh giá hệ thống:

Hệ thống được đánh giá thông qua các thử nghiệm tập trung vào độ chính xác của đề xuất, mức độ hài lòng của người dùng và khả năng mở rộng. Các tiêu chí cụ thể tham khảo theo *Pane và cộng sự* [27], bao gồm thời gian phản hồi của hệ thống và tính phù hợp của lộ trình học tập, như đã được đề xuất trong các nghiên cứu về đánh giá hệ thống học tập thích ứng.

### Giới hạn nghiên cứu

* Phụ thuộc vào dữ liệu:   
  Hiệu quả của Đồ thị Tri thức phụ thuộc vào chất lượng và độ đầy đủ của dữ liệu giáo dục được sử dụng. Dữ liệu không đầy đủ hoặc thiên vị có thể dẫn đến các lộ trình học tập không tối ưu, đặc biệt khi xây dựng KG từ các nguồn mở.
* Khả năng của LLM:   
  LLM, dù mạnh mẽ trong xử lý ngôn ngữ tự nhiên, có thể gặp khó khăn trong việc hiểu các truy vấn phức tạp hoặc chuyên sâu nếu dữ liệu huấn luyện không bao quát đầy đủ lĩnh vực giáo dục. Dựa theo *Zhang và cộng sự* [16] điều này có thể ảnh hưởng đến tính chính xác của các đề xuất.
* Phạm vi lĩnh vực:   
  Hệ thống được thiết kế cho các lĩnh vực giáo dục có cấu trúc kiến thức rõ ràng, như MIS với nguồn tri thức khoa học máy tính và quản trị hệ thống. Theo *Zhang và cộng sự* [40], việc áp dụng cho các lĩnh vực khác, như khoa học xã hội, tâm lý học, hoặc các tri thức mở rộng phi nhị phân, có thể yêu cầu sự tham gia của chuyên gia và điều chỉnh đáng kể thuật toán đề xuất.
* Khả năng mở rộng:   
  Là một hệ thống mẫu, hệ thống có thể gặp thách thức về hiệu suất khi xử lý số lượng lớn người dùng hoặc dữ liệu KG phức tạp. Các vấn đề về kết nối Neo4j và lỗi dữ liệu.
* Mức độ cá nhân hóa:   
  Hệ thống cá nhân hóa dựa trên các đặc điểm như trình độ, mục tiêu, và phong cách học tập (theo mô hình VARK), nhưng có thể không bao quát hết các yếu tố khác, như cảm xúc hoặc động lực học tập của học viên. Tham khảo trong một nghiên cứu của *Chee Ng và cộng sự* [44].
* Đánh giá hạn chế:   
  Các phương pháp đánh giá tập trung vào độ chính xác và thời gian phản hồi, nhưng có thể không phản ánh đầy đủ tác động lâu dài của hệ thống đối với kết quả học tập hoặc khả năng áp dụng kiến thức trong thực tế. Thiếu sự đánh giá dài hạn.
* Vấn đề đạo đức:   
  Việc sử dụng dữ liệu học viên để cá nhân hóa học tập đặt ra các câu hỏi về bản quyền của nguồn tài liệu; hoặc quyền riêng tư về lịch sử học tập, trình độ, khả năng cá nhân, sự công bằng trong đề xuất của hệ thống. Nghiên cứu này không đi sâu vào các khía cạnh này, nhưng chúng cần được xem xét trong các ứng dụng thực tế.

## ĐÓNG GÓP KHOA HỌC VÀ THỰC TIỄN

### Đóng góp khoa học

* Khung tích hợp Đồ thị Tri thức và Mô hình Ngôn ngữ Lớn:  
  Nghiên cứu đề xuất một cách tiếp cận mới, kết hợp KG và LLM để tạo ra hệ thống học tập thích ứng. KG biểu diễn cấu trúc kiến thức giáo dục, trong khi LLM xử lý ngôn ngữ tự nhiên để hiểu và đáp ứng yêu cầu học tập của học viên. Sự tích hợp này vượt qua các hạn chế của các phương pháp truyền thống như lọc cộng tác hoặc lọc dựa trên nội dung, vốn thường thiếu khả năng biểu diễn kiến thức phức tạp và tương tác tự nhiên.
* Thiết kế Đồ thị Tri thức trong Neo4j:  
  Hệ thống sử dụng Neo4j để xây dựng KG với cấu trúc chi tiết, bao gồm các nút KnowledgeNode với các thuộc tính bắt buộc như Document\_ID, Node\_Label, Learning\_Objective, Skill\_Level, Difficulty, Prerequisites, và các thuộc tính tùy chọn như Learning\_Style\_Preference, Real\_World\_Application. Các mối quan hệ như NEXT, REQUIRES, IS\_PREREQUISITE\_OF, IS\_SUBCONCEPT\_OF, REMEDIATES, HAS\_ALTERNATIVE\_PATH, SIMILAR\_TO được thiết kế để mô hình hóa trình tự học tập logic, cung cấp một mô hình biểu diễn kiến thức giáo dục toàn diện. Thiết kế này dựa trên thông tin từ nguồn tài liệu của Neo4j [24] , được cải tiến và tinh chỉnh phù hợp, cho phép hệ thống hiểu rõ các mối liên hệ giữa các khái niệm giáo dục, từ đó đề xuất lộ trình học tập phù hợp với trình độ và mục tiêu của học viên.
* Ứng dụng LlamaIndex Property Graph Index:  
  Nghiên cứu tận dụng LlamaIndex để kết nối LLM với KG, cho phép truy vấn đồ thị hiệu quả và tìm kiếm ngữ nghĩa dựa trên lịch sử học tập thông qua VectorStoreIndex. Nâng cao hiệu suất xử lý các truy vấn phức tạp trên KG, đặc biệt trong bối cảnh dữ liệu lớn và đa chiều. Kết quả đạt được là hệ thống có khả năng tận dụng sự phát triển nhanh chóng của AI, giúp đề xuất lộ trình học tập cá nhân hóa bằng các ngôn ngữ tự nhiên cung cấp bởi người dùng trong thời gian thực. Thiết kế này dựa trên thông tin từ nguồn tài liệu của Llama-index [21] được hiện thực và tiếp cận để tạo ra khả năng hiểu ngữ cảnh và đáp ứng các yêu cầu học tập phức tạp.
* Phát triển thuật toán đề xuất lộ trình học tập nâng cao và thích ứng động dựa trên phản hồi:  
  Dựa trên dữ liệu từ KG và yêu cầu của học viên, nghiên cứu phát triển các thuật toán đề xuất, chẳng hạn như thuật toán A\*, để tạo ra các lộ trình học tập tối ưu.   
  Hệ thống tích hợp cơ chế phản hồi thời thực, cho phép điều chỉnh lộ trình học tập dựa trên phản hồi hiệu suất và độ khó mà người học đưa ra, thể hiện sự tiến bộ trong lĩnh vực hệ thống học tập thích ứng, nơi hệ thống tự động học hỏi và cải thiện từ tương tác của người dùng. Các thuật toán này đảm bảo tính logic và phù hợp với trình độ, mục tiêu, và phong cách học tập của từng học viên, góp phần cải thiện các phương pháp đề xuất trong hệ thống học tập thích ứng.
* Tự động hóa tiền xử lý dữ liệu giáo dục:  
  Nghiên cứu giới thiệu quy trình sử dụng cặp prompt SPR Generator và SPR Validation để tự động trích xuất và chuẩn hóa dữ liệu từ tài liệu giáo dục, tạo ra các tệp nodes.csv và relationships.csv sẵn sàng cho Neo4j. Quy trình này đảm bảo dữ liệu nhất quán, tuân thủ các tiêu chuẩn như Bloom’s Taxonomy và WordNet/DBpedia, nâng cao chất lượng KG và hiệu quả của hệ thống đề xuất.
* Sử dụng độ tương đồng ngữ nghĩa:   
  Việc áp dụng độ tương đồng Jaccard để xác định điểm bắt đầu và điểm đích dựa trên các thẻ ngữ nghĩa (semantic\_tags), được mở rộng với hàm expand\_tags() giúp nâng cao độ chính xác của các đề xuất cá nhân hóa. Cách tiếp cận này đóng góp vào việc cải thiện các phương pháp lựa chọn nút trong KG cho giáo dục.

### Đóng góp thực tiễn

* Cải thiện trải nghiệm học tập của học viên:  
  Giúp học viên học tập hiệu quả hơn theo trình độ, ngữ cảnh, và mục tiêu cá nhân.
* Giảm thiểu công sức thiết kế lộ trình học tập:   
  Tự động hóa quá trình thiết kế lộ trình học tập, giảm thời gian và công sức của giáo viên, cho phép họ tập trung vào giảng dạy và hỗ trợ trực tiếp.
* Hỗ trợ giáo viên và tổ chức giáo dục trong quản lý và theo dõi tiến độ học sinh: Cung cấp công cụ theo dõi và cập nhật hồ sơ học sinh, giúp giáo viên giám sát và điều chỉnh kế hoạch giảng dạy kịp thời. Dữ liệu học tập được lưu trữ cho phép phân tích hiệu quả chương trình học, cải thiện chất lượng nội dung và phương pháp giảng dạy.
* Khả năng mở rộng và linh hoạt:  
  Nhờ vào Neo4j và LlamaIndex, hệ thống có khả năng mở rộng để áp dụng cho nhiều lĩnh vực giáo dục, từ MIS đến các môn học khác như toán học hoặc khoa học xã hội. Tính linh hoạt này cho phép hệ thống được điều chỉnh để phù hợp với các bối cảnh giáo dục khác nhau, từ giáo dục phổ thông đến đào tạo chuyên nghiệp.
* Thúc đẩy đổi mới công nghệ giáo dục:  
  Hệ thống mẫu được phát triển trong nghiên cứu này có thể truyền cảm hứng cho các nền tảng giáo dục trực tuyến tích hợp các công nghệ AI tiên tiến như KG và LLM. Quy trình tiền xử lý dữ liệu và vận hành hệ thống, cung cấp một mô hình tham chiếu cho các tổ chức giáo dục muốn triển khai các giải pháp học tập thông minh.
* Tăng cường hiệu quả quản lý tri thức giáo dục:  
  Quy trình tự động hóa tiền xử lý dữ liệu, sử dụng SPR Generator và SPR Validation, cho phép các tổ chức giáo dục xây dựng và duy trì KG chất lượng cao từ các nguồn tài liệu học thuật. Điều này không chỉ hỗ trợ hệ thống đề xuất mà còn có thể được sử dụng trong các ứng dụng khác, như quản lý tài nguyên giáo dục hoặc phân tích dữ liệu học tập.

## CẤU TRÚC ĐỀ CƯƠNG

Luận văn được thiết kế để đảm bảo tính logic, liên kết chặt chẽ giữa các chương, từ việc giới thiệu bối cảnh đến triển khai và đánh giá hệ thống.

Tổ chức thành năm chương chính, cùng với phần tài liệu tham khảo và phụ lục.

* **Chương 1: Giới thiệu**  
  Chương này đặt nền tảng cho nghiên cứu bằng cách trình bày bối cảnh, động lực, mục tiêu, đối tượng, phạm vi, giới hạn, và đóng góp của nghiên cứu.
  + 1.1. Bối cảnh và động lực nghiên cứu: Phân tích nhu cầu ngày càng tăng về học tập cá nhân hóa trong giáo dục trực tuyến, các hạn chế của các hệ thống hiện tại như Khan Academy hay Coursera, và tiềm năng của KG và LLM trong việc khắc phục các vấn đề này.
  + 1.2. Mục tiêu nghiên cứu: Xác định các mục tiêu cụ thể, bao gồm thiết kế KG trong Neo4j, tích hợp LLM với LlamaIndex, phát triển thuật toán đề xuất.
  + 1.3. Đối tượng nghiên cứu: Mô tả ba đối tượng chính: KG, LLM kết hợp với LlamaIndex, và hệ thống đề xuất lộ trình học tập cá nhân hóa.
  + 1.4. Phạm vi và giới hạn nghiên cứu: Làm rõ phạm vi tập trung vào giáo dục trực tuyến và các lĩnh vực cụ thể như ngành “Hệ thống thông tin quản lý”, cùng với các giới hạn như phụ thuộc vào chất lượng dữ liệu và khả năng mở rộng.
  + 1.5. Đóng góp khoa học và thực tiễn: Nêu các đóng góp về lý thuyết (khung tích hợp KG và LLM) và thực tiễn (cải thiện trải nghiệm học tập và hỗ trợ giáo viên).
  + 1.6. Cấu trúc đề cương: Phần này, cung cấp tổng quan về cách tổ chức đề cương luận văn.
* **Chương 2: Cơ sở lý thuyết và công nghệ**  
  Chương này cung cấp nền tảng lý thuyết và công nghệ cần thiết để hiểu hệ thống đề xuất.
  + 2.1. Quy trình học trực tuyến chuẩn
  + 2.2. Knowledge Graph và ứng dụng trong giáo dục: Định nghĩa KG, kiến trúc của nó, và vai trò trong việc tổ chức kiến thức giáo dục để hỗ trợ học tập cá nhân hóa.
  + 2.3. Mô hình Ngôn ngữ Lớn (LLM) và LlamaIndex: Tổng quan về LLM, khả năng xử lý ngữ nghĩa, và cơ chế tích hợp của LlamaIndex Property Graph Index với KG.
  + 2.4. Công nghệ Neo4j: Giới thiệu kiến trúc cơ sở dữ liệu đồ thị của Neo4j và các ưu điểm trong quản lý dữ liệu phân cấp, đặc biệt là trong việc lưu trữ KG.
  + 2.5. Những thư viện và công nghệ sử dụng trong quy mô luận văn: Mô tả các công nghệ dùng trong quy mô prototype.
* **Chương 3: Phân tích và thiết kế hệ thống**  
  Chương này phân tích bài toán đề xuất lộ trình học tập cá nhân hóa và trình bày thiết kế hệ thống chi tiết.
  + 3.1. Bài toán đề xuất lộ trình học tập cá nhân hóa: Thảo luận các thách thức từ dữ liệu đa chiều và động, cùng với yêu cầu về tính thích ứng và hiệu suất của hệ thống.
  + 3.2. Quy trình nghiệp vụ: Mô tả quy trình học trực tuyến chuẩn. Quy trình được đề xuất.
  + 3.3. Giải pháp đề xuất: Trình bày kiến trúc hệ thống tích hợp Neo4j và LLM, quy trình xử lý từ dữ liệu thô đến đề xuất lộ trình, và minh họa kiến trúc tổng quát qua mô hình C4.
  + 3.4. Yêu cầu hệ thống: Liệt kê các yêu cầu chức năng (như đề xuất lộ trình, sinh nội dung học tập) và phi chức năng (như hiệu suất, khả năng mở rộng).
  + 3.5. Thiết kế hệ thống: Chi tiết thiết kế KG trong Neo4j với các nút KnowledgeNode, kèm chi tiết property của nút và các mối quan hệ trong KG; thiết kế thuật toán đề xuất; và mô hình C4 để minh họa thành phần hệ thống.
* **Chương 4: Hiện thực và thử nghiệm**  
  Chương này mô tả quá trình triển khai hệ thống mẫu và các thử nghiệm thực tế.
  + 4.1. Xây dựng nguyên mẫu: Bao gồm 2 giai đoạn.   
    Giai đoạn 1: tiền xử lý và xây dựng dữ liệu nguồn tri thức, cung cấp và tạo sinh KG trong Neo4j.   
    Giai đoạn 2: Vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa.
  + 4.2. Tích hợp LlamaIndex và LLM: Cấu hình LLM: LlamaIndex Property Graph Index, VectorStore Index.
* **Chương 5: Đánh giá và so sánh**  
  Chương này đánh giá hiệu quả của hệ thống thông qua các thử nghiệm và so sánh với các giải pháp mã nguồn mở hiện tại.
  + 5.1. Thiết lập thử nghiệm: Mô tả bộ dữ liệu thử nghiệm, tiêu chí đánh giá (độ chính xác, thời gian phản hồi), và các thuật toán nguồn mở để so sánh.
  + 5.2. Kết quả và phân tích: Trình bày kết quả thử nghiệm, so sánh với các thuật toán nguồn mở, và phân tích các hạn chế của hệ thống, như phụ thuộc vào chất lượng dữ liệu, mô hình AI hoặc khả năng mở rộng theo công nghệ đang dùng.
* **Chương 6: Kết luận**

Chương này xác nhận lại các kết quả đạt được của nghiên cứu đồng thời đưa ra kế hoạch phát triển tiếp theo.

* **6.**1, Kết quả đạt được
* 6.2. Hướng phát triển
* **Tài liệu tham khảo**  
  Phần này liệt kê các tài liệu, bài báo khoa học, và nguồn tham khảo được sử dụng trong luận văn.
* **Phụ lục**
  + **A. Mã nguồn triển khai thử nghiệm**: Cung cấp mã nguồn chi tiết, bao gồm các đoạn mã python sử dụng cho luồng vận hành, các prompt tạo sinh và xử lý dữ liệu, cypher code tải và hợp nhất nút mới vào KG.
  + **B. Ví dụ lộ trình học tập được đề xuất**: Minh họa quá trình tạo lộ trình học tập, với business instance và các đối tượng cụ thể.

# CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

## QUY TRÌNH HỌC TRỰC TUYẾN CHUẨN

* + 1. **Giới thiệu**

Quy trình học trực tuyến chuẩn là một khung công tác được thiết kế nhằm đảm bảo học viên trải qua quá trình học tập hiệu quả, linh hoạt và đạt chất lượng cao trong môi trường giáo dục trực tuyến. Khung này cung cấp một lộ trình rõ ràng, từ giai đoạn đăng ký đến khi hoàn thành khóa học, giúp học viên đạt được mục tiêu học tập một cách có hệ thống. Theo các tiêu chuẩn được công nhận rộng rãi như National Standards for Quality Online Courses của Hiệp hội Học trực tuyến Quốc gia (iNACOL) [17] và Quality Matters Rubric [29], quy trình học trực tuyến chuẩn cần đáp ứng các yêu cầu về thiết kế khóa học, tương tác học tập, và hỗ trợ học viên. Phần này trình bày các bước cốt lõi của quy trình, độc lập với hệ thống đề xuất lộ trình học tập cá nhân hóa được đề cập trong các mục khác của luận văn, nhằm làm rõ các hoạt động cơ bản trong một khóa học trực tuyến tiêu chuẩn..

* + 1. **Các bước trong quy trình**

A diagram of a diagram

AI-generated content may be incorrect.

**Hình 2.1. Sơ đồ quy trình đào tạo trực tuyến chuẩn** (Nguồn: iNACOL [17] )

Quy trình học trực tuyến chuẩn được minh họa trong Hình 2.1, được xây dựng dựa trên các thực tiễn tốt nhất trong giáo dục trực tuyến và các tiêu chuẩn quốc tế. Quy trình bao gồm các bước sau:

1. **Đăng ký và định hướng**:  
   Học viên bắt đầu bằng việc đăng ký khóa học thông qua hệ thống quản lý học tập (Learning Management System - LMS) hoặc cổng thông tin trực tuyến của cơ sở giáo dục. Sau khi đăng ký, họ tham gia một buổi định hướng trực tuyến để làm quen với giao diện LMS, cấu trúc khóa học, lịch trình học tập và các yêu cầu cụ thể. Theo Simonson và cộng sự [34], buổi định hướng thường bao gồm hướng dẫn sử dụng công cụ LMS, thông tin về chính sách học tập và giới thiệu các tài nguyên hỗ trợ, giúp học viên chuẩn bị tốt cho quá trình học tập.
2. **Truy cập tài liệu khóa học**:  
   Học viên sử dụng LMS để truy cập các tài liệu học tập, bao gồm chương trình học, mục tiêu học tập, lịch trình và các mô-đun hàng tuần. Dựa trên tiêu chuẩn iNACOL [17], các mô-đun cần được tổ chức có hệ thống, bao gồm video bài giảng, tài liệu đọc, bài tập thực hành và các tài nguyên bổ sung, nhằm hỗ trợ học viên tự quản lý tiến độ học tập một cách hiệu quả..
3. **Tương tác với nội dung học tập**:  
   Học viên tham gia các hoạt động học tập như xem video bài giảng, đọc tài liệu hoặc thực hiện bài tập thực hành. Nội dung được thiết kế để phù hợp với các phong cách học tập đa dạng, chẳng hạn như video cho người học trực quan (visual) hoặc bài tập thực hành cho người học vận động (kinesthetic), như được khuyến nghị bởi Quality Matters [29]. Các hoạt động này giúp học viên củng cố kiến thức và phát triển kỹ năng cần thiết.
4. **Tham gia các hoạt động tương tác**:  
   Để tăng cường sự tham gia và xây dựng cộng đồng học tập trực tuyến, học viên tham gia vào các diễn đàn thảo luận, dự án nhóm hoặc các hoạt động tương tác khác. Simonson và cộng sự [34] nhấn mạnh rằng các hoạt động này không chỉ củng cố kiến thức mà còn giảm cảm giác cô lập, một thách thức phổ biến trong học trực tuyến.
5. **Hoàn thành các bài đánh giá**:  
   Học viên thực hiện các bài kiểm tra, bài tập hoặc kỳ thi để đánh giá mức độ hiểu biết và kỹ năng. Các bài đánh giá bao gồm đánh giá hình thành (formative) như bài tập hàng tuần và đánh giá tổng kết (summative) như kỳ thi cuối kỳ, được thiết kế theo tiêu chuẩn iNACOL [17] để đo lường tiến độ học tập của học viên.
6. **Nhận phản hồi từ giảng viên**:  
   Giảng viên cung cấp phản hồi kịp thời và mang tính xây dựng thông qua LMS, giúp học viên nhận biết điểm mạnh, điểm yếu và cải thiện hiệu suất học tập. Theo Quality Matters [29], phản hồi mang tính cá nhân hóa là yếu tố quan trọng để nâng cao trải nghiệm học tập.
7. **Truy cập các dịch vụ hỗ trợ**:  
   Học viên có thể sử dụng các dịch vụ hỗ trợ như tư vấn học tập, hỗ trợ kỹ thuật hoặc thư viện trực tuyến để giải quyết các vấn đề trong quá trình học tập. Simonson và cộng sự [34] cho rằng các dịch vụ này cần được tích hợp chặt chẽ vào LMS hoặc cung cấp qua các kênh riêng để đảm bảo sự hỗ trợ liên tục.
8. **Đánh giá khóa học**:  
   Cuối khóa học, học viên hoàn thành khảo sát đánh giá để cung cấp phản hồi về nội dung, phương pháp giảng dạy và trải nghiệm tổng thể. Theo iNACOL [17], phản hồi này đóng vai trò quan trọng trong việc cải thiện chất lượng khóa học và đảm bảo đáp ứng các tiêu chuẩn giáo dục trực tuyến.

* **Tầm quan trọng của quy trình chuẩn**

Quy trình học trực tuyến chuẩn đóng vai trò nền tảng trong việc đảm bảo chất lượng giáo dục và nâng cao trải nghiệm học tập. Một báo cáo của McKinsey [22] chỉ ra rằng học viên đánh giá cao tính linh hoạt và khả năng tự quản lý tiến độ học tập, nhưng họ cũng cần các yếu tố tương tác và hỗ trợ để duy trì động lực. Quy trình này không chỉ giúp học viên đạt được mục tiêu học tập mà còn hỗ trợ các cơ sở giáo dục duy trì chất lượng và uy tín. Các tiêu chuẩn như iNACOL [17] và Quality Matters [29] cung cấp khung tham chiếu để đảm bảo các khóa học trực tuyến đáp ứng các yêu cầu cao về thiết kế, giảng dạy và hỗ trợ học viên.

Quy trình học trực tuyến chuẩn này tạo nền tảng cho việc tích hợp các tính năng cá nhân hóa, như hệ thống đề xuất lộ trình học tập được trình bày trong mục 3.2.2. Sự kết hợp giữa quy trình chuẩn và hệ thống đề xuất sẽ được minh họa chi tiết trong các mục 3.3.2 và 3.3.3 của Chương 3.

## KNOWLEDGE GRAPH VÀ ỨNG DỤNG TRONG GIÁO DỤC

### Mô tả kiến trúc đồ thị tri thức

* **Định nghĩa đồ thị tri thức**

Đồ thị tri thức (Knowledge Graph - KG) là một cấu trúc dữ liệu biểu diễn kiến thức thông qua một mạng lưới gồm các thực thể và mối quan hệ giữa chúng. Thực thể đại diện cho các khái niệm hoặc đối tượng, ví dụ “SQL Function” hoặc “SQL Basics”, còn mối quan hệ mô tả cách các thực thể liên kết, chẳng hạn “SQL Basics” là tiền đề của “SQL Function”. Theo Abu-Salih và Alotaibi [1], KG không chỉ lưu trữ dữ liệu mà còn cung cấp ngữ cảnh, hỗ trợ các ứng dụng như tìm kiếm ngữ nghĩa và đề xuất nội dung. Trong lĩnh vực giáo dục, KG tổ chức các khái niệm học tập và mối liên hệ giữa chúng, tạo nền tảng cho các hệ thống đề xuất lộ trình học tập cá nhân hóa, giúp học viên tiếp cận nội dung phù hợp với trình độ và mục tiêu cá nhân.

* **Kiến trúc đồ thị tri thức**

Kiến trúc của KG bao gồm ba thành phần chính:

1. **Thực thể (Entities)**  
   Thực thể đại diện cho các khái niệm học tập, ví dụ “SQL Function” hoặc “Hàm trong SQL”. Các thực thể này là các nút trong đồ thị, chứa thông tin về khái niệm cụ thể.
2. **Mối quan hệ (Relationships)**  
   Mối quan hệ liên kết các thực thể, thể hiện trình tự hoặc sự phụ thuộc giữa chúng, ví dụ “REQUIRES” (tiên quyết) hoặc “NEXT” (khái niệm tiếp theo không bắt buộc). Các mối quan hệ này đảm bảo cấu trúc logic của lộ trình học tập.
3. **Thuộc tính (Properties)**  
   Thuộc tính cung cấp thông tin bổ sung (metadata) cho thực thể và mối quan hệ, ví dụ định nghĩa, độ khó hoặc mục tiêu học tập. Theo *Ehrlinger và Wöß* [11], thuộc tính dạng cặp khóa-giá trị cho phép biểu diễn dữ liệu phức tạp một cách linh hoạt và dễ truy vấn.

KG thường được triển khai dưới dạng mô hình đồ thị thuộc tính, trong đó các thực thể và mối quan hệ được gán các thuộc tính để tăng cường khả năng biểu diễn và truy vấn dữ liệu.

* **Kiến trúc đồ thị tri thức trong luận văn này**

Trong nghiên cứu này, KG được xây dựng trên Neo4j, một cơ sở dữ liệu đồ thị mạnh mẽ, sử dụng mô hình đồ thị thuộc tính để hỗ trợ đề xuất lộ trình học tập cá nhân hóa, như được mô tả trong tài liệu Neo4j [24]. Kiến trúc được thiết kế tập trung vào các thành phần sau:

1. **Nút (Nodes)**

Các nút, được gọi là KnowledgeNode, đại diện cho các khái niệm giáo dục, ví dụ “SQL Function”. Các thuộc tính chính của nút được trình bày trong Bảng 2.1:

**Bảng 2.1 Các thuộc tính chính của nút tri thức**

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Node\_ID | Mã định danh duy nhất của nút, ví dụ “concept:sql\_select”. |
| Sanitized\_Concept | Tên khái niệm chuẩn hóa, ví dụ “Câu lệnh SQL SELECT”. |
| Context | Ngữ cảnh học tập, ví dụ “database” hoặc “business\_intelligence”. |
| Definition | Định nghĩa khái niệm, ví dụ “Lệnh truy vấn dữ liệu trong SQL”. |
| Example | Ví dụ minh họa, ví dụ “SELECT \* FROM table\_name”. |
| Difficulty | Độ khó, ví dụ “STANDARD” hoặc “ADVANCED”. |
| Priority | Độ ưu tiên, ví dụ 4 (cao) hoặc 2 (thấp). |
| Learning\_Objective | Mục tiêu học tập, ví dụ “Hiểu cách truy vấn dữ liệu bằng SQL”. |
| Skill\_Level | Trình độ theo Bloom’s Taxonomy, ví dụ “Understand” hoặc “Apply”. |
| Time\_Estimate | Thời gian học ước tính (phút), ví dụ 30 phút. |
| Prerequisites | Danh sách khái niệm tiên quyết, ví dụ “concept:sql\_basics”. |
| Semantic\_Tags | Thẻ ngữ nghĩa, ví dụ “sql;query;database”. |

1. **Mối quan hệ (Relationships)**

Các mối quan hệ được thiết kế để hỗ trợ đề xuất lộ trình học tập cá nhân hóa, được trình bày trong Bảng 2.2. Các mối quan hệ này được xây dựng dựa trên các nghiên cứu về mô hình hóa kiến thức giáo dục và học tập thích nghi.

**Bảng 2.2 Các thuộc tính chính của mối quan hệ tri thức**

|  |  |  |
| --- | --- | --- |
| Mối quan hệ | Mô tả | Vai trò và lý do học thuật |
| REQUIRES | Chỉ định rằng khái niệm hiện tại không thể hiểu nếu không nắm vững khái niệm tiên quyết, ví dụ “GROUP BY” yêu cầu “SQL SELECT”. | Đảm bảo tính phụ thuộc bắt buộc, hỗ trợ phân tích bottom-up để xây dựng trình tự học tập logic.. |
| IS\_PREREQUISITE\_OF | Chỉ định khái niệm hiện tại là tiền đề cho khái niệm khác, ví dụ “SQL SELECT” là tiền đề của “GROUP BY”. | Hỗ trợ phân tích top-down, giúp xác định các khái niệm nền tảng cho mục tiêu học tập. |
| NEXT | Chỉ định khái niệm tiếp theo trong lộ trình, ví dụ “SQL SELECT” dẫn đến “SQL WHERE”. | Định hướng lộ trình học tập tuyến tính, đảm bảo cấu trúc logic cho quá trình học tập. |
| REMEDIATES | Chỉ định khái niệm cung cấp nội dung khắc phục, ví dụ “SQL Basics” hỗ trợ “SQL JOIN”. | Hỗ trợ học viên gặp khó khăn dựa trên lý thuyết học tập thích nghi , tránh mắc kẹt trong quá trình học.. |
| HAS\_ALTERNATIVE\_PATH | Chỉ định đường đi thay thế nếu khái niệm hiện tại quá khó, ví dụ “SQL CASE” thay thế “SQL IF”. | Tăng tính linh hoạt và cá nhân hóa thông qua học tập phi tuyến tính. |
| SIMILAR\_TO | Chỉ định khái niệm tương tự, ví dụ “SQL SELECT” và “Python DataFrame Query”. | Thúc đẩy khám phá kiến thức dựa trên lý thuyết ngữ nghĩa, mở rộng lộ trình học tập. |
| IS\_SUBCONCEPT\_OF | Chỉ định khái niệm là một phần của khái niệm lớn hơn, ví dụ “SQL WHERE” là khái niệm con của “SQL Query. | Tổ chức kiến thức theo cấu trúc phân cấp, giúp học viên hiểu sâu hơn về các khái niệm. |

1. **Vai trò của mối quan hệ trong hệ thống**

Các mối quan hệ này được thiết kế dựa trên các nghiên cứu về mô hình hóa kiến thức giáo dục. Ví dụ, REQUIRES và IS\_PREREQUISITE\_OF phản ánh sự phụ thuộc kiến thức. REQUIRES nhấn mạnh sự phụ thuộc bắt buộc, đảm bảo học viên phải nắm vững khái niệm tiên quyết trước khi học khái niệm hiện tại, trong khi IS\_PREREQUISITE\_OF hỗ trợ xác định các khái niệm nâng cao hơn. Sự kết hợp này cho phép hệ thống thực hiện truy vấn hai chiều, hỗ trợ cả phân tích bottom-up (từ khái niệm nâng cao tìm về khái niệm cơ bản) và top-down (từ khái niệm cơ bản tìm đến khái niệm nâng cao).

Tương tự, REMEDIATES và HAS\_ALTERNATIVE\_PATH hỗ trợ học tập thích nghi, NEXT và IS\_SUBCONCEPT\_OF đảm bảo cấu trúc logic và phân cấp, còn SIMILAR\_TO thúc đẩy khám phá kiến thức dựa trên lý thuyết ngữ nghĩa. Chi tiết kỹ thuật về triển khai trên Neo4j và tích hợp với LlamaIndex sẽ được trình bày trong Chương 3 và Chương 4, đảm bảo mục này tập trung vào lý thuyết và kiến trúc tổng quan.

### Ứng dụng

1. **Vai trò của Knowledge Graph trong giáo dục cá nhân hóa**KG mang lại giá trị đặc biệt trong giáo dục cá nhân hóa với khả năng biểu diễn kiến thức dưới dạng mạng lưới các thực thể và mối quan hệ, mang lại các giá trị học thuật nổi bật trong các khía cạnh sau::
2. Xây dựng lộ trình học tập thích ứng  
   KG cho phép thiết kế các lộ trình học tập động, được điều chỉnh dựa trên trình độ, mục tiêu và tiến độ của từng học viên. Theo *Zhu và Wang* [43], việc sử dụng KG giúp tổ chức kiến thức theo cấu trúc phân cấp và liên kết, từ đó tạo ra các chuỗi học tập thích nghi, đảm bảo tính logic và phù hợp với năng lực cá nhân.
3. Hỗ trợ xác định và khắc phục lỗ hổng kiến thức  
   Thông qua các mối quan hệ như tiền đề (prerequisites) hoặc khái niệm liên quan, KG hỗ trợ việc xác định các khái niệm nền tảng mà học viên cần bổ sung. Đồng thời, KG khuyến khích khám phá các khái niệm bổ trợ, góp phần duy trì động lực học tập và mở rộng phạm vi kiến thức của học viên.
4. Tích hợp đánh giá và phản hồi liên tục   
   KG tạo điều kiện cho việc theo dõi hiệu suất học tập thông qua các bài đánh giá, từ đó điều chỉnh lộ trình học tập để ưu tiên các nội dung phù hợp với năng lực và tốc độ học tập của học viên. Quá trình này phản ánh nguyên tắc học tập thích nghi, trong đó phản hồi liên tục được sử dụng để tối ưu hóa trải nghiệm học tập.
5. Tùy chỉnh nội dung theo phong cách học tập  
   Bằng cách gắn các thuộc tính như phong cách học tập (ví dụ: visual, auditory, kinesthetic) vào các thực thể trong KG, hệ thống có thể cung cấp nội dung học tập phù hợp, chẳng hạn như tài liệu trực quan hoặc bài tập thực hành. Cách tiếp cận này nâng cao hiệu quả học tập bằng cách đáp ứng các phong cách học tập đa dạng của học viên.
6. **Ứng dụng thực tiễn trong giáo dục hiện nay**Trong bối cảnh giáo dục hiện đại, KG được kết hợp với các công nghệ trí tuệ nhân tạo, đặc biệt là Mô hình Ngôn ngữ Lớn (LLM), để nâng cao khả năng cá nhân hóa học tập. Theo *Chen và Huang* [8], các ứng dụng thực tiễn của KG trong giáo dục bao gồm:
7. Tự động tạo lộ trình học tập:   
   KG hỗ trợ triển khai các thuật toán tìm đường, chẳng hạn như thuật toán dựa trên heuristic, để tạo ra các lộ trình học tập tối ưu. Các lộ trình này được xây dựng dựa trên dữ liệu thực tế về hiệu suất và mục tiêu học tập, đảm bảo tính phù hợp và logic trong trình tự kiến thức.
8. Tạo nội dung học tập cá nhân hóa   
   Dựa trên cấu trúc của KG, hệ thống có thể sinh ra nội dung giảng dạy phù hợp với trình độ và phong cách học tập của học viên. Nội dung này bao gồm các bài giảng chi tiết, bài tập thực hành và câu hỏi đánh giá, được thiết kế để tối ưu hóa sự hiểu biết và kỹ năng.
9. Phân tích dữ liệu học tập quy mô lớn   
   KG cho phép phân tích dữ liệu học tập từ nhiều học viên để đưa ra các đề xuất nội dung phù hợp. Các kỹ thuật như lọc cộng tác và khai phá quy tắc liên kết được sử dụng để xác định các khái niệm học tập tiềm năng, dựa trên sự tương đồng về hành vi học tập giữa các học viên.
10. **Ứng dụng trong nghiên cứu này:**Nghiên cứu này khai thác tiềm năng của KG trong việc xây dựng một hệ thống giáo dục cá nhân hóa, tập trung vào hai khía cạnh chính:

* Thiết kế đồ thị tri thức trên Neo4:  
  KG được xây dựng trên nền tảng cơ sở dữ liệu đồ thị, sử dụng các nút đại diện cho các khái niệm giáo dục và các mối quan hệ như tiền đề hoặc khái niệm tiếp theo. Hệ thống cũng tích hợp thông tin học viên, liên kết với KG thông qua các mối quan hệ biểu thị trạng thái hoàn thành kiến thức. Cách tiếp cận này cho phép tạo ra các lộ trình học tập phù hợp với lịch sử học tập, phong cách học tập và mục tiêu cá nhân của học viên.
* Tích hợp với LlamaIndex và LLMKG được tích hợp với các công nghệ xử lý ngôn ngữ tự nhiên tiên tiến, chẳng hạn như các khung truy vấn đồ thị thông minh, để hỗ trợ các chức năng sau:
* Xử lý yêu cầu ngôn ngữ tự nhiên:   
  Hệ thống cho phép học viên diễn đạt mục tiêu học tập bằng ngôn ngữ tự nhiên, ví dụ “học các khái niệm cơ bản về SQL”. Công nghệ NLP phân tích và ánh xạ các yêu cầu này thành các khái niệm cụ thể trong KG, chẳng hạn như “SQL SELECT”. Theo *Devlin và cộng sự* [10], các mô hình ngôn ngữ dựa trên cơ chế biến đổi (transformer) hỗ trợ việc phân tích ngữ nghĩa, đảm bảo sự chính xác trong việc xác định các khái niệm liên quan.
* Tăng cường đề xuất ngữ nghĩa:   
  KG tận dụng các kỹ thuật nhúng ngữ nghĩa để ưu tiên các khái niệm có mức độ liên quan cao với nhu cầu của học viên, ví dụ các khái niệm như “SQL” hoặc “tổng hợp dữ liệu”. Theo *Gao và cộng sự* [13] nhấn mạnh rằng các phương pháp nhúng ngữ nghĩa giúp cải thiện khả năng tìm kiếm và đề xuất, bằng cách xác định các mối quan hệ ngữ nghĩa tiềm ẩn giữa các khái niệm trong KG.
* Tạo nội dung học tập cá nhân hóa:   
  Dựa trên thông tin từ KG, hệ thống sinh ra nội dung học tập được tùy chỉnh, bao gồm các bài giảng chi tiết, bài tập thực hành hoặc bài kiểm tra với các mức độ khó khác nhau. Theo *Brown và cộng sự* [5] chỉ ra rằng các kỹ thuật điều hướng mô hình ngôn ngữ, thông qua việc sử dụng các lời nhắc (prompt) được thiết kế cẩn thận, cho phép tạo ra nội dung giáo dục phù hợp với trình độ và phong cách học tập của từng học viên.

1. **Lợi ích của việc ứng dụng Knowledge Graph**Việc ứng dụng KG trong giáo dục cá nhân hóa mang lại các lợi ích học thuật và thực tiễn sau:

* Tính linh hoạt và khả năng mở rộng: Cấu trúc đồ thị của KG cho phép dễ dàng bổ sung hoặc cập nhật các khái niệm và mối quan hệ, hỗ trợ triển khai hệ thống trong nhiều lĩnh vực giáo dục khác nhau.
* Đề xuất chính xác với chi phí tối ưu: So với các kỹ thuật tìm kiếm truyền thống như PageRank hoặc tìm kiếm toàn văn, KG cung cấp các đề xuất chính xác hơn với chi phí tính toán thấp hơn, nhờ vào khả năng biểu diễn ngữ cảnh.
* Hỗ trợ học tập thích nghi: KG tạo nền tảng cho các thuật toán học tập thích nghi, cho phép hệ thống điều chỉnh lộ trình học tập theo nhu cầu và tiến độ của từng học viên

1. **Thách thức và hạn chế**Mặc dù có nhiều tiềm năng, việc ứng dụng KG trong giáo dục cá nhân hóa đối mặt với các thách thức đáng kể:

* Thu thập và quản lý dữ liệu học viên  
  Việc thu thập dữ liệu học viên đòi hỏi độ chính xác cao và tuân thủ các quy định bảo mật nghiêm ngặt, chẳng hạn như *Quy định Bảo vệ Dữ liệu Chung của Liên minh Châu Âu* [35]. Sự phức tạp trong việc tích hợp dữ liệu vào các hệ thống giáo dục hiện tại cũng là một rào cản lớn.
* Đảm bảo chất lượng dữ liệu nguồn  
  Tài liệu học thuật dùng để xây dựng KG cần được kiểm định kỹ lưỡng để đảm bảo tính chính xác và phù hợp. Theo *Abu-Salih và Alotaibi* [1], dữ liệu không chất lượng có thể dẫn đến các đề xuất sai lệch, làm giảm hiệu quả của hệ thống.
* Kiểm soát nội dung từ công nghệ ngôn ngữ  
  Nội dung được tạo ra bởi các công nghệ ngôn ngữ tiên tiến cần được kiểm soát chặt chẽ để đảm bảo tính chính xác, tránh thiên kiến và tuân thủ các quy định giáo dục, như vấn đề bản quyền. Theo *Brown và cộng sự* [5] nhấn mạnh rằng các mô hình ngôn ngữ có thể sinh ra thông tin không chính xác nếu không được điều chỉnh phù hợp.
* Đáp ứng sự đa dạng của học viên  
  Đảm bảo các đề xuất phù hợp với mọi học viên, bất kể trình độ, mục tiêu hay ngữ cảnh học tập, là một thách thức phức tạp. Theo *Wang và cộng sự* [39] lưu ý rằng các hệ thống KG cần được trang bị các thuật toán tinh vi để xử lý sự đa dạng về nhu cầu và năng lực học tập.

## MÔ HÌNH NGÔN NGỮ LỚN VÀ LLAMA-INDEX

### Tổng quan về LLM

Mô hình Ngôn ngữ Lớn (Large Language Models - LLMs) là các hệ thống trí tuệ nhân tạo tiên tiến, được huấn luyện trên khối lượng dữ liệu văn bản khổng lồ để hiểu và sinh ra ngôn ngữ tự nhiên một cách hiệu quả. Theo *Vaswani và cộng sự* [38], LLMs tận dụng kiến trúc transformer để xử lý các chuỗi văn bản, cho phép nắm bắt các mối quan hệ ngữ nghĩa phức tạp.

Khái quát hóa, LLMs có thể được hình dung như một hệ thống tri thức nén, chứa đựng thông tin từ hàng tỷ văn bản. Quá trình huấn luyện bao gồm việc tối ưu hóa hàng tỷ tham số dựa trên dữ liệu ngôn ngữ tự nhiên, nhằm dự đoán xác suất xuất hiện của các từ tiếp theo trong một chuỗi văn bản. Khi nhận đầu vào, LLMs áp dụng các phép tính để sinh ra đầu ra hợp ngữ cảnh, đảm bảo tính tự nhiên và mạch lạc. Ở khía cạnh kỹ thuật, LLMs chuyển đổi các từ hoặc cụm từ thành các vector nhúng (embeddings) trong không gian đa chiều, sau đó sử dụng các phép biến đổi tuyến tính và phi tuyến tính để xác định mối liên hệ ngữ nghĩa [10]. Kết quả là sự lựa chọn các nhúng phù hợp để tạo ra ngôn ngữ tự nhiên, như được mô tả trong nghiên cứu về mô hình ngôn ngữ.

Các mô hình tiêu biểu như GPT-4 của OpenAI, Grok-3 của xAI đã chứng minh khả năng vượt trội trong xử lý ngôn ngữ tự nhiên, như trong báo cáo theo *GPT-4 Technical Report 2024* [25], bao gồm phân tích ngữ nghĩa, trả lời câu hỏi và sinh nội dung. Trong giáo dục, LLMs mang lại tiềm năng lớn cho việc cá nhân hóa học tập nhờ khả năng hiểu ngữ cảnh và tạo nội dung phù hợp với nhu cầu học viên.

* **Đặc điểm và khả năng của LLM**

LLMs sử dụng kiến trúc transformer, dựa trên cơ chế tự chú ý (self-attention) để xử lý các chuỗi văn bản và nắm bắt mối quan hệ ngữ nghĩa phức tạp. Các đặc điểm chính bao gồm:

* Hiểu ngữ cảnh:   
  LLMs có khả năng phân tích ý nghĩa và ngữ cảnh của các yêu cầu, ví dụ nhận diện rằng “học SQL cơ bản” liên quan đến các khái niệm như “SQL SELECT” hoặc “SQL WHERE”. Theo *Raffel và cộng sự* [30], cơ chế tự chú ý cho phép LLMs xử lý các truy vấn đa dạng với độ chính xác cao.
* Sinh nội dung tự nhiên:   
  LLMs tạo ra bài giảng, ví dụ minh họa, hoặc bài kiểm tra trắc nghiệm phù hợp với trình độ học viên, ví dụ bài kiểm tra 20 câu về “GROUP BY” với các mức độ khó khác nhau.
* Tìm kiếm ngữ nghĩa:   
  LLMs hỗ trợ tìm kiếm các khái niệm liên quan dựa trên ngữ nghĩa, không chỉ từ khóa, giúp đề xuất nội dung học tập chính xác hơn.
* Khả năng suy luận nâng cao:   
  Các mô hình tiên tiến như GPT-4 thể hiện khả năng suy luận logic, cung cấp giải thích chi tiết trong các lĩnh vực như lập trình hoặc toán học, như được ghi nhận trong nghiên cứu về trí tuệ nhân tạo tổng quát của *Bubeck và cộng sự* [6].
* **Ứng dụng trong giáo dục cá nhân hóa**

LLMs đóng vai trò quan trọng trong các hệ thống giáo dục cá nhân hóa, thông qua các ứng dụng sau:

1. Phân tích yêu cầu học viên  
   LLMs hiểu các truy vấn ngôn ngữ tự nhiên, ví dụ “tôi muốn học SQL cơ bản”, để xác định mục tiêu học tập và trình độ hiện tại, từ đó đề xuất các khái niệm phù hợp.
2. Tạo nội dung học tập tùy chỉnh  
   Dựa trên thông tin từ đồ thị tri thức, LLMs sinh ra bài giảng chi tiết, ví dụ minh họa hoặc bài kiểm tra trắc nghiệm, chẳng hạn bài kiểm tra về “SQL SELECT” kèm phân tích lỗi thường gặp.
3. Hỗ trợ tương tác thông minh  
   LLMs cung cấp câu trả lời chi tiết cho các câu hỏi của học viên, ví dụ giải thích cách sử dụng “SQL SELECT” trong các tình huống thực tế, tăng cường sự hiểu biết.
4. Tăng cường đề xuất lộ trình học tập  
   Khi kết hợp với đồ thị tri thức, LLMs hỗ trợ đề xuất các chuỗi học tập logic, ví dụ từ “SQL SELECT” đến “GROUP BY” dựa trên các mối quan hệ tri thức, đảm bảo tính liên kết và phù hợp.

* **Lựa chọn mô hình LLM trong ứng dụng thực tiễn**

Việc lựa chọn mô hình LLM phù hợp là yếu tố cốt lõi, ảnh hưởng đến hiệu suất, chi phí và khả năng tùy chỉnh của hệ thống. Các mô hình phổ biến như GPT-3.5, GPT-4 có các ưu điểm riêng về tốc độ xử lý, chi phí và khả năng đa ngôn ngữ. Mô hình tiên tiến, như GPT-4, được thiết kế để tăng cường khả năng suy luận sâu và tư duy logic, mang lại phản hồi chất lượng cao. Trong luận văn này, một mô hình hiệu quả về chi phí được ưu tiên cho nguyên mẫu.

Khi hệ thống hoàn thiện ở tương lai, việc tích hợp các mô hình tiên tiến hơn như GPT-4.1 hoặc SuperGrok 3.5 sẽ mở ra tiềm năng cải tiến vượt trội về chất lượng đề xuất, đồng thời đảm bảo khả năng cập nhật nhanh chóng theo sự phát triển của công nghệ AI, giúp hệ thống luôn duy trì tính hiện đại và hiệu quả.

**Thách thức và hạn chế**

Mặc dù LLMs mang lại nhiều lợi ích, chúng cũng đối mặt với một số thách thức:

* Độ phức tạp của suy luận ngữ nghĩa:   
  LLMs có thể gặp khó khăn trong việc xử lý các yêu cầu chuyên sâu hoặc ngữ cảnh phức tạp, đặc biệt trong các lĩnh vực kỹ thuật như MIS. Theo *Bang và cộng sự* [2], khả năng suy luận của LLMs vẫn bị giới hạn ở các nhiệm vụ đòi hỏi logic sâu.
* Độ tin cậy và thiên kiến:   
  Nội dung do LLMs tạo ra có thể chứa thông tin không chính xác hoặc thiên kiến nếu dữ liệu huấn luyện không đầy đủ, theo *Bubeck và cộng sự* [6]. Điều này đòi hỏi các cơ chế kiểm định nội dung nghiêm ngặt.
* Vấn đề đạo đức trong giáo dục:   
  Việc sử dụng dữ liệu học viên để cá nhân hóa học tập đặt ra các câu hỏi về quyền riêng tư và công bằng, một thách thức lý thuyết quan trọng trong giáo dục cá nhân hóa, theo SmythOS [35].
* Khả năng mở rộng ngữ cảnh:   
  LLMs cần được tích hợp với các cấu trúc tri thức để duy trì tính liên quan trong các bối cảnh giáo dục đa dạng, theo *Wang và cộng sự* [39]. Điều này yêu cầu các phương pháp kết hợp tri thức hiệu quả.

Chi tiết về cách LLMs được tích hợp vào hệ thống sẽ được trình bày trong Chương 3, với trọng tâm vào nguyên lý thiết kế hệ thống giáo dục cá nhân hóa.

### Cơ chế tích hợp LLM với Llama-index

* **Tổng quan và cơ chế tích hợp**

LlamaIndex là một framework dữ liệu cho phép tích hợp các nguồn tri thức cấu trúc, như đồ thị tri thức (KG), với Mô hình Ngôn ngữ Lớn (LLMs), nhằm nâng cao khả năng truy xuất và xử lý thông tin ngữ nghĩa. Theo *Zhang và cộng sự* [41], LlamaIndex hỗ trợ kết nối LLMs với dữ liệu bên ngoài, tạo điều kiện cho việc truy xuất tri thức theo ngữ cảnh. Trong bối cảnh giáo dục, LlamaIndex Property Graph Index đóng vai trò cầu nối giữa LLMs và KG, cho phép khai thác các mối quan hệ ngữ nghĩa và cấu trúc tri thức để hỗ trợ học tập cá nhân hóa.

Về mặt lý thuyết, cơ chế tích hợp của LlamaIndex dựa trên việc lập chỉ mục dữ liệu từ KG, chẳng hạn đồ thị thuộc tính lưu trữ các nút và mối quan hệ dạng cặp khóa-giá trị. Property Graph Index tổ chức dữ liệu tri thức thành các cấu trúc có thể truy vấn, trong khi VectorStoreIndex sử dụng nhúng ngữ nghĩa (semantic embeddings) để hỗ trợ tìm kiếm dựa trên ý nghĩa, theo *Gao và cộng sự* [13]. Quá trình này cho phép LLMs truy xuất thông tin từ KG một cách hiệu quả, cung cấp các phản hồi phù hợp với yêu cầu học tập. Theo *Tan và cộng sự* [36], tích hợp tri thức giữa LLMs và KG là nền tảng để xây dựng các hệ thống giáo dục thông minh, đáp ứng nhu cầu cá nhân hóa

* **Ứng dụng trong hệ thống đề xuất**

LlamaIndex Property Graph Index mang lại các giá trị học thuật sau trong giáo dục cá nhân hóa:

1. Xử lý yêu cầu ngôn ngữ tự nhiên:   
   LlamaIndex cho phép LLMs phân tích các yêu cầu học tập, ví dụ yêu cầu về “phân tích dữ liệu”, và ánh xạ chúng đến các khái niệm tri thức như “truy vấn dữ liệu” hoặc “biểu đồ hóa”. Theo *Siemens* *và cộng sự* [33], khả năng này hỗ trợ lý thuyết học tập cá nhân hóa bằng cách căn chỉnh nội dung với mục tiêu học viên.
2. Tìm kiếm ngữ nghĩa:   
   VectorStoreIndex sử dụng nhúng ngữ nghĩa để khám phá các khái niệm liên quan, ví dụ gợi ý “phân tích dữ liệu kinh doanh” khi học về “truy vấn dữ liệu”. Theo *Liu và Wang* [19], tìm kiếm ngữ nghĩa nâng cao độ chính xác của các đề xuất giáo dục.
3. Xây dựng lộ trình học tập:   
   LlamaIndex khai thác các mối quan hệ tri thức, như phụ thuộc hoặc trình tự, để tạo các chuỗi học tập logic, ví dụ từ “cơ sở dữ liệu” đến “phân tích dữ liệu”. Theo *Zhang và cộng sự* [42], mô hình hóa tri thức cấu trúc là yếu tố cốt lõi để thiết kế lộ trình thích nghi.
4. Tạo nội dung tùy chỉnh: LlamaIndex cung cấp dữ liệu tri thức cho LLMs để sinh tài liệu giảng dạy, chẳng hạn bài giảng hoặc bài kiểm tra, phù hợp với trình độ và phong cách học tập, theo lý thuyết học tập cá nhân hóa.

Trong lĩnh vực Hệ thống thông tin quản lý (MIS), LlamaIndex hỗ trợ học viên tiếp cận các khái niệm như “truy vấn dữ liệu” hoặc “biểu đồ hóa thông tin” thông qua các lộ trình và nội dung được tùy chỉnh, đảm bảo tính liên kết với mục tiêu học tập.

**Thách thức và hạn chế**  
Việc tích hợp LlamaIndex với KG đối mặt với các thách thức lý thuyết sau:

1. Độ phức tạp của truy xuất tri thức:   
   Truy xuất thông tin từ KG quy mô lớn đòi hỏi các thuật toán hiệu quả để đảm bảo tính chính xác và tốc độ, đặc biệt trong các bối cảnh giáo dục đa dạng, theo *Wang* *và cộng sự* [39].
2. Chất lượng dữ liệu tri thức:   
   Độ tin cậy của các đề xuất phụ thuộc vào tính đầy đủ và chính xác của KG, theo *Abu-Salih và cộng sự* [1]. Dữ liệu không chất lượng có thể dẫn đến các đề xuất sai lệch.
3. Tích hợp ngữ nghĩa:   
   Việc kết hợp nhúng ngữ nghĩa với truy vấn tri thức cấu trúc đặt ra thách thức về độ chính xác, đặc biệt khi xử lý các yêu cầu chuyên sâu, *theo Tan và cộng sự* [36].
4. Vấn đề đạo đức:   
   Sử dụng dữ liệu học viên để cá nhân hóa học tập đòi hỏi tuân thủ các nguyên tắc đạo đức và bảo mật, một vấn đề lý thuyết quan trọng trong giáo dục thông minh, theo báo cáo của *SmythOS - một trang web cung cấp nền tảng để tạo và quản lý các tác nhân AI* [35].

Chi tiết về cách tích hợp LlamaIndex trong hệ thống giáo dục sẽ được trình bày trong Chương 3, với trọng tâm vào nguyên lý thiết kế học thuật.

## CÔNG NGHỆ NEO4J

### Tổng quan và kiến trúc

Neo4j là một hệ quản trị cơ sở dữ liệu đồ thị (Graph Database Management System) tiên tiến, được thiết kế để lưu trữ và truy xuất dữ liệu dưới dạng các nút, mối quan hệ, và thuộc tính. Theo *Miller và cộng sự* [23], Neo4j sử dụng mô hình đồ thị thuộc tính (property graph model), trong đó các nút và mối quan hệ mang các cặp khóa-giá trị, cho phép biểu diễn linh hoạt các cấu trúc tri thức phức tạp. Khác với các cơ sở dữ liệu quan hệ dựa trên bảng và phép nối, Neo4j ưu tiên truy xuất trực tiếp các mối quan hệ, phù hợp với việc mô hình hóa tri thức giáo dục có tính phân cấp và liên kết.

Về mặt lý thuyết, Neo4j hỗ trợ mô hình hóa tri thức bằng cách tổ chức các khái niệm giáo dục thành các nút (ví dụ, khái niệm về “truy vấn dữ liệu”) và các mối quan hệ (như phụ thuộc hoặc trình tự) thành các cạnh. Theo *Zhang* *và cộng sự* [42], mô hình hóa tri thức giáo dục dựa trên đồ thị cho phép biểu diễn các mối quan hệ ngữ nghĩa và phân cấp, tạo điều kiện cho việc truy xuất tri thức theo ngữ cảnh. Trong bối cảnh Hệ thống thông tin quản lý (MIS), Neo4j biểu diễn các khái niệm như “phân tích dữ liệu” hoặc “biểu đồ hóa thông tin” cùng các mối quan hệ, hỗ trợ xây dựng các lộ trình học tập thích nghi.

* **Kiến trúc Neo4j**

Kiến trúc của Neo4j bao gồm các thành phần chính như sau:

* **Storage Engine**  
  Storage Engine là lõi của Neo4j, chịu trách nhiệm lưu trữ vật lý dữ liệu đồ thị và quản lý truy cập trong bộ nhớ. Theo Neo4j Documentation [24], mô hình lưu trữ được tổ chức thành các thành phần như Node Store, Relationship Store, Property Store, Label Store, và Schema Store, tối ưu hóa truy cập đồ thị theo chiều sâu (depth-first) và chiều rộng (breadth-first). Điều này cho phép xử lý nhanh các mối quan hệ phức tạp, lý tưởng cho các hệ thống giáo dục cá nhân hóa.
* **Cypher Query Language**  
  Cypher là ngôn ngữ truy vấn đồ thị đặc trưng của Neo4j, cho phép mô tả các mẫu quan hệ một cách trực quan và ngắn gọn. Cypher hỗ trợ các thao tác như tìm đường ngắn nhất hoặc truy vấn chuỗi quan hệ, phù hợp để xây dựng lộ trình học tập. Ví dụ, một truy vấn Cypher có thể trả về các khái niệm tiên quyết của “GROUP BY”, tăng cường tính logic trong cá nhân hóa học tập.
* **Transaction Management**  
  Neo4j hỗ trợ các giao dịch ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính nhất quán và an toàn dữ liệu khi nhiều người dùng cập nhật đồng thời. Điều này đặc biệt quan trọng trong các hệ thống giáo dục trực tuyến với lượng truy cập lớn.
* **Indexing & Caching**  
  Neo4j cung cấp cơ chế lập chỉ mục và bộ nhớ đệm để tăng tốc độ truy vấn, đặc biệt hiệu quả với các truy vấn theo thuộc tính hoặc nhãn. Các nhãn như KnowledgeNode và Student giúp phân loại nhanh các thực thể trong đồ thị tri thức, hỗ trợ truy xuất hiệu quả.
* **Scalability & High Availability**  
  Neo4j hỗ trợ triển khai phân tán thông qua các cụm và cơ chế sao chép, đảm bảo khả năng mở rộng theo chiều ngang và tính sẵn sàng cao. Theo *Guo và cộng sự* [14], kiến trúc này phù hợp cho các hệ thống giáo dục thông minh quy mô lớn.

Về mặt lý thuyết, Neo4j hỗ trợ mô hình hóa tri thức bằng cách tổ chức các khái niệm giáo dục thành các nút (ví dụ, “truy vấn dữ liệu”) và các mối quan hệ (như phụ thuộc hoặc trình tự) thành các cạnh. Mô hình hóa tri thức giáo dục dựa trên đồ thị cho phép biểu diễn các mối quan hệ ngữ nghĩa, tạo điều kiện cho việc truy xuất tri thức theo ngữ cảnh.

### Ứng dụng trong giáo dục và quản lý dữ liệu phân cấp

Dữ liệu phân cấp, như các mối quan hệ giữa các khái niệm giáo dục, đòi hỏi khả năng biểu diễn và truy vấn linh hoạt. Neo4j mang lại các giá trị học thuật quan trọng trong giáo dục cá nhân hóa.

* **Ứng dụng và ưu điểm trong giáo dục**
* **Truy vấn phân cấp hiệu quả**

Neo4j cho phép truy xuất trực tiếp các mối quan hệ tri thức, như tiền đề của “GROUP BY”, mà không cần phép nối phức tạp, giảm thời gian xử lý so với cơ sở dữ liệu quan hệ .

* **Linh hoạt trong cấu trúc động**

Mô hình không lược đồ của Neo4j hỗ trợ bổ sung các khái niệm hoặc mối quan hệ mới, như “SQL CASE” hoặc liên kết trình tự, mà không ảnh hưởng đến hiệu suất [54]. Theo *Hu và Wang* [15], tính linh hoạt này lý tưởng cho các hệ thống giáo dục thích nghi.

* **Phân tích tri thức thông minh**

Neo4j tích hợp các thuật toán đồ thị, như tìm đường ngắn nhất, để xây dựng lộ trình học tập tối ưu hoặc xác định các khái niệm trọng yếu. Các thuật toán này nâng cao khả năng đề xuất trong giáo dục thông minh.

* **Tích hợp với trí tuệ nhân tạo**

Neo4j hỗ trợ tích hợp với các mô hình ngôn ngữ lớn, cho phép xử lý dữ liệu ngữ nghĩa và tạo nội dung cá nhân hóa, đáp ứng nhu cầu của các hệ thống giáo dục hiện đại.

* **Thách thức và hạn chế**  
  Việc ứng dụng Neo4j trong giáo dục đối mặt với các thách thức lý thuyết sau:
* Độ phức tạp của mô hình hóa tri thức:   
  Xây dựng đồ thị tri thức chính xác đòi hỏi biểu diễn đầy đủ các mối quan hệ ngữ nghĩa, một thách thức lớn trong các lĩnh vực có tri thức phức tạp.
* Hiệu suất truy xuất dữ liệu lớn:   
  Truy xuất tri thức từ đồ thị quy mô lớn có thể gặp thách thức về hiệu suất, đặc biệt khi đáp ứng các yêu cầu thời gian thực.
* Chất lượng dữ liệu tri thức:   
  Độ tin cậy của các đề xuất phụ thuộc vào tính đầy đủ và chính xác của đồ thị tri thức. Dữ liệu không chất lượng có thể dẫn đến các lộ trình học tập không tối ưu.
* Vấn đề đạo đức:   
  Sử dụng dữ liệu học viên để cá nhân hóa học tập đặt ra các câu hỏi về quyền riêng tư và công bằng, một vấn đề lý thuyết quan trọng.

## NHỮNG THƯ VIỆN VÀ CÔNG NGHỆ SỬ DỤNG TRONG QUY MÔ LUẬN VĂN

Trong quá trình xây dựng nguyên mẫu, hệ thống sử dụng nhiều thư viện Python chuyên biệt cho từng tác vụ, từ kết nối cơ sở dữ liệu, xử lý dữ liệu, học máy, đến tích hợp AI và xây dựng giao diện. Dưới đây là bảng tổng hợp các thư viện chính, chức năng và vai trò trong hệ thống

### Thư viện Python

Các thư viện Python sau được sử dụng để phát triển hệ thống, với vai trò được tổng hợp trong Bảng 2.3. Mã nguồn chi tiết được trình bày trong Phụ lục A.

**Bảng 2.3. Thư viện sử dụng và vai trò trong hệ thống**

|  |  |
| --- | --- |
| Thư viện | Vai trò trong hệ thống |
| llama-index | Xử lý dữ liệu từ file CSV để nhập vào Neo4j. |
| llama-index-graph-stores-neo4j | Tạo, cập nhật, truy vấn property graph index trực tiếp trên Neo4j |
| llama-index-llms-openai | Giao tiếp với GPT-3.5 Turbo, sinh nội dung, truy vấn tự nhiên |
| llama-index-vector-stores-neo4jvector | Hỗ trợ tìm kiếm ngữ nghĩa, truy vấn vector-based trên đồ thị tri thức |
| openai | Sinh văn bản, phân tích ngữ nghĩa, hỗ trợ truy vấn tự nhiên |
| pandas | Xử lý dữ liệu từ file CSV để nhập vào Neo4j. |
| csv | Xử lý dữ liệu đầu vào/đầu ra dạng bảng, lưu kết quả kiểm tra, lộ trình học tập (learning\_data, learning\_paths,..) |
| json | Lưu trữ, truyền dữ liệu cấu trúc giữa các thành phần |
| datetime | Ghi log, đánh dấu thời gian cập nhật dữ liệu |
| numpy | Hỗ trợ tính toán số học cho thuật toán đề xuất, như tính độ tương đồng cosine. |
| scikit-learn | Tính độ tương đồng ngữ nghĩa (cosine similarity) để đề xuất nội dung học tập. |
| mlxtend | Áp dụng thuật toán Apriori để khai phá luật liên kết giữa các khái niệm học tập. |
| nltk | Mở rộng thẻ ngữ nghĩa bằng WordNet, tăng cường khả năng tìm kiếm khái niệm liên quan. |
| neo4j | Tương tác với cơ sở dữ liệu Neo4j, thực hiện truy vấn Cypher để quản lý đồ thị. |
| nest\_asyncio | Quản lý vòng lặp bất đồng bộ trong Google Colab khi tích hợp LlamaIndex và Neo4j. |
| python-dotenv | Quản lý biến môi trường (như khóa API OpenAI) để đảm bảo bảo mật và cấu hình linh hoạt. |
| logging | Ghi log hoạt động hệ thống, hỗ trợ gỡ lỗi và theo dõi trong quá trình phát triển. |
| openai | Kết nối OpenAI API để phân tích yêu cầu người dùng và xử lý ngữ nghĩa. |
| streamlit | Triển khai UI cho học viên và giáo viên |
| ngrok | Kết nối Google Colab với Neo4j local |

### Công nghệ bổ trợ

Các công nghệ bổ trợ sau hỗ trợ triển khai và phát triển hệ thống:

* Neo4j Desktop: Triển khai cơ sở dữ liệu đồ thị cục bộ, quản lý các nút và quan hệ trực quan.
* Google Colab: Môi trường điện toán đám mây để chạy prototype, tận dụng GPU miễn phí cho xử lý LLM.
* Ngrok: Kết nối Neo4j local với Google Colab thông qua tunnel, đảm bảo truy cập an toàn.
* OpenAI API: mô hình ngôn ngữ lớn GPT-3.5 Turbo
* Streamlit: Xây dựng giao diện người dùng (UI) đơn giản để hiển thị lộ trình học tập và kết quả đề xuất.

### Kiến trúc tích hợp

Hệ thống tích hợp thêm các thành phần sau vào các giai đoạn:

* Nhập liệu: Dữ liệu học tập được xử lý bằng Pandas
* Lưu trữ: Neo4j quản lý Knowledge Graph, LlamaIndex xây dựng chỉ mục đồ thị.
* Xử lý ngữ nghĩa: OpenAI API phân tích yêu cầu người dùng, LlamaIndex tạo truy vấn Cypher.
* Đề xuất lộ trình: Scikit-learn và Mlxtend tính toán độ tương đồng, khai phá luật kết hợp.
* Triển khai: Streamlit hiển thị kết quả, Google Colab và Ngrok đảm bảo khả năng mở rộng.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## BÀI TOÁN ĐỀ XUẤT LỘ TRÌNH HỌC TẬP CÁ NHÂN HÓA

Phần này trình bày bài toán cụ thể mà hệ thống đề xuất lộ trình học tập cá nhân hóa hướng đến giải quyết, tập trung vào các khía cạnh kỹ thuật và thách thức trong việc xây dựng một giải pháp hiệu quả, thích ứng và có khả năng mở rộng.

### Định nghĩa bài toán

Hệ thống đề xuất lộ trình học tập cá nhân hóa nhằm tạo ra các chuỗi học tập tối ưu, linh hoạt cho từng học viên dựa trên trình độ hiện tại, phong cách học tập, mục tiêu và ngữ cảnh cụ thể. Cá nhân hóa học tập là xu hướng quan trọng trong giáo dục thông minh.   
Hệ thống cần:

* Xác định lộ trình phù hợp: Từ thông tin đầu vào, xác định khái niệm khởi đầu và mục tiêu.
* Xây dựng chuỗi học tập logic: Dựa trên mối quan hệ trong KG để đảm bảo học viên nắm vững các khái niệm nền trước khi tiến đến các khái niệm nâng cao.
* Cá nhân hóa nội dung: Điều chỉnh tài liệu học tập theo phong cách như “visual”, “auditory”, “reading\_writing” hoặc “kinesthetic”.
* Hỗ trợ học tập thích ứng: Theo dõi tiến độ và điều chỉnh lộ trình, ví dụ đề xuất nội dung khắc phục nếu học viên không đạt yêu cầu trong bài kiểm tra.
* Đảm bảo hiệu suất: Xử lý nhanh với thời gian phản hồi dưới 5 giây cho mỗi tác vụ tương tác với người dùng.

### Thách thức kỹ thuật

Việc xây dựng một hệ thống đề xuất lộ trình học tập cá nhân hóa hiệu quả đòi hỏi phải giải quyết đồng thời nhiều thách thức kỹ thuật phức tạp, bao gồm:

* Quản lý dữ liệu đa chiều và động:

Xử lý đồng thời hồ sơ học viên (trình độ, phong cách học, mục tiêu), dữ liệu KG động, dữ liệu tương tác thời gian thực và ngữ cảnh học tập (thông tin phân nhóm tri thức như môn học). Đây là yêu cầu cốt lõi trong giáo dục cá nhân hóa.

* Tích hợp dữ liệu từ nhiều nguồn và định dạng:

Kết hợp dữ liệu từ KG, mô hình ngôn ngữ lớn (LLM), dữ liệu từ các file csv và thông tin tương tác, đảm bảo tính đồng nhất và hiệu quả. Việc tích hợp này đòi hỏi hệ thống phải đảm bảo tính đồng nhất và có cơ chế xử lý lỗi.

* Đảm bảo chất lượng dữ liệu

Cần cơ chế xác thực và làm sạch dữ liệu trước khi nhập vào Neo4j. Cụ thể, dựa trên mức độ tương đồng ngữ nghĩa (semantic similarity) để quyết định tạo mới hay hợp nhất nút. Đồng thời xử lý vấn đề trong nội tại KG như quan hệ tự trỏ (self-loop) và vòng lặp vô tận. Các vấn đề có thể dẫn đến sai lệch trong phân tích tiên quyết, làm giảm hiệu suất. Hồ sơ học sinh cần được cập nhật sau mỗi phiên học tập, và kiểm tra tính hợp lệ của dữ liệu kèm cơ chế fallback để phục hồi dữ liệu khi cần thiết.  
Dữ liệu lộ trình học tập phải được điều chỉnh động trong từng phiên học tập.Việc kiểm soát chất lượng dữ liệu là yếu tố then chốt để đảm bảo độ tin cậy của các đề xuất.

* Thuật toán đề xuất thích nghi:

Hệ thống cần tích hợp các thuật toán tìm đường tối ưu (như A\*) trên KG, kết hợp các kỹ thuật học máy như collaborative filtering, Apriori. Các kỹ thuật cần khai thác lịch sử học tập, phát hiện quy luật để nâng cao khả năng cá nhân hóa.

* Hiệu suất và khả năng mở rộng:

Đảm bảo thời gian phản hồi thấp (dưới 5 giây) ngay cả khi KG mở rộng quy mô lên hàng nghìn nút hoặc nhiều hơn.

Hỗ trợ nhiều người dùng đồng thời trong môi trường trực tuyến, đảm bảo tính ổn định và khả năng mở rộng của hệ thống.

### Tầm nhìn và mục tiêu cụ thể

Hệ thống hướng đến xây dựng một prototype minh chứng khả năng tạo các lộ trình học tập linh hoạt, thích nghi với nhu cầu cá nhân, dựa trên tích hợp tri thức và phân tích dữ liệu giáo dục. Việc kết hợp đồ thị tri thức với AI mở ra tiềm năng cho các hệ thống học tập thích nghi động. Trong dài hạn, hệ thống không chỉ giới hạn ở MIS mà có thể mở rộng sang các lĩnh vực giáo dục khác, như toán học hoặc khoa học xã hội, cạnh tranh với các nền tảng học tập trực tuyến truyền thống bằng cách cung cấp lộ trình động.

Các mục tiêu cụ thể của prototype bao gồm:

* Xử lý truy vấn ngôn ngữ tự nhiên: Từ các yêu cầu như “Học SQL cơ bản”, hệ thống tạo lộ trình học tập (ví dụ: từ “SQL SELECT” đến “GROUP BY”) với độ chính xác đề xuất trên 80%.
* Sinh nội dung cá nhân hóa: Tạo bài giảng và bài kiểm tra trắc nghiệm (mức độ khó khác nhau) theo phong cách học tập.
* Hỗ trợ học tập thích nghi: Thiết kế thuật toán hiệu quả tạo và điều chỉnh lộ trình dựa trên hiệu suất học viên.
* Hiệu suất cao: Phản hồi dưới 5 giây cho các tác vụ người dùng tương tác. Đảm bảo hiệu suất không ảnh hưởng cả khi đồ thị tri thức mở rộng.
* Thử nghiệm trên dữ liệu MIS: Tập trung vào các chủ đề như SQL, Business Intelligence, và Hệ thống quản lý thông tin hiện đại, với mục tiêu đạt tỷ lệ hài lòng người dùng trên 80% dựa trên phản hồi học viên

### Ví dụ bài toán minh họa

Để minh họa cách hệ thống giải quyết bài toán đề xuất lộ trình học tập cá nhân hóa, hãy xem xét kịch bản sau:

* **Yêu cầu học viên**: “Học cách dùng SQL để nhóm và tính tổng lương theo phòng ban.”
* **Phân tích**: Hệ thống sử dụng thuộc tính Semantic\_Tags (như “sql”, “data\_aggregation”) để nhận diện các khái niệm liên quan (“SQL”, “GROUP BY”, “SUM”). Các khái niệm tiên quyết (“SQL SELECT”, “SQL WHERE”) được xác định thông qua mối quan hệ REQUIRES trong đồ thị tri thức. Để tránh vòng lặp vô hạn do các nút tự tham chiếu hoặc chu kỳ, truy vấn Cypher giới hạn độ sâu duyệt tối đa 5 bước, giải quyết thách thức về hiệu suất thuật toán.
* **Lộ trình**: Sử dụng thuật toán A\* với heuristic dựa trên Difficulty và Prerequisites, hệ thống đề xuất lộ trình: “SQL SELECT” → “SQL WHERE” → “GROUP BY” → “SUM”.
* **Nội dung**: Hệ thống tạo bài giảng về “GROUP BY” với ví dụ thực tế tính tổng lương theo phòng ban, kèm bài kiểm tra trắc nghiệm 20 câu (phân bổ đều các mức độ từ cơ bản đến nâng cao). Với học viên có phong cách học tập “visual”, bài giảng bao gồm biểu đồ minh họa cách “GROUP BY” nhóm dữ liệu, đáp ứng yêu cầu cá nhân hóa.
* **Điều chỉnh**: Nếu học viên đạt điểm thấp trong bài kiểm tra “GROUP BY” (dưới 60%), hệ thống đề xuất nội dung khắc phục “SQL Basics” thông qua mối quan hệ REMEDIATES. Dữ liệu tương tác từ bài kiểm tra được cập nhật vào hồ sơ học sinh để điều chỉnh lộ trình theo thời gian thực, giải quyết thách thức về dữ liệu động.

Ví dụ này minh họa cách hệ thống tích hợp đồ thị tri thức, LLM, và thuật toán đề xuất để tạo ra lộ trình học tập cá nhân hóa, đồng thời vượt qua các thách thức kỹ thuật như xử lý dữ liệu đa chiều và đảm bảo hiệu suất

## QUY TRÌNH NGHIỆP VỤ

### So sánh với quy trình học trực tuyến chuẩn

So với quy trình học trực tuyến chuẩn, hệ thống đề xuất mang đến sự cải tiến vượt trội bằng cách kết hợp các công nghệ hiện đại như KG và Mô hình LLMs. Trong khi các quy trình truyền thống thường tập trung vào việc cung cấp nội dung học tập theo cách tiếp cận chung, hệ thống này tận dụng phân tích dữ liệu giáo dục và lý thuyết học tập thích nghi để tạo ra các lộ trình học tập được tùy chỉnh cho từng cá nhân. Sự khác biệt nằm ở khả năng tổ chức tri thức có cấu trúc và khả năng điều chỉnh động nội dung học tập, vượt xa giới hạn của các phương pháp chuẩn.   
Quy trình đề xuất còn chia thành hai giai đoạn rõ ràng, mang lại sự mạch lạc và hiệu quả trong thiết kế cũng như vận hành.

**Phân tích từng giai đoạn**

* **Giai đoạn 1 - Tiền xử lý xây dựng dữ liệu nền tảng Knowledge Graph**

Giai đoạn đầu tiên tập trung vào việc xây dựng một Đồ thị Tri thức (KG), đóng vai trò là cơ sở dữ liệu cốt lõi cho nội dung giáo dục. KG không chỉ lưu trữ thông tin mà còn tổ chức các khái niệm và mối quan hệ giữa chúng theo một cấu trúc ngữ nghĩa phong phú, hỗ trợ các cơ chế truy vấn phức tạp cần thiết cho việc cá nhân hóa lộ trình học tập. Việc xây dựng cấu trúc này tham khảo lý thuyết biểu diễn tri thức phân cấp trên Zhang et al. [42], từ đó hệ thống hiện tại cho phép hệ thống truy xuất và liên kết tri thức theo ngữ cảnh một cách hiệu quả, tạo nền tảng vững chắc cho các giai đoạn tiếp theo. Giai đoạn này cho thấy sự chú trọng vào việc chuẩn bị dữ liệu – một yếu tố then chốt trong bất kỳ hệ thống dựa trên AI nào.

* **Giai đoạn 2 - Vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa.**   
  Ở giai đoạn thứ hai, hệ thống triển khai cơ chế đề xuất sử dụng LLMs để phân tích dữ liệu đầu vào từ học viên và tạo ra các lộ trình học tập được thiết kế riêng. Quá trình này được dẫn dắt bởi lý thuyết học tập thích nghi, trong đó nội dung giáo dục được điều chỉnh linh hoạt dựa trên hồ sơ và tiến độ của từng học viên. Hơn nữa, việc tích hợp LLMs cho phép hệ thống hiểu sâu hơn nhu cầu học tập, từ đó căn chỉnh chính xác tài nguyên giáo dục với mục tiêu cá nhân, như được ghi nhận trong nghiên cứu về phân tích học tập của Siemens [33].

Thiết kế hai giai đoạn tạo ra một quy trình có tính hệ thống, dễ mở rộng và tối ưu hóa trong tương lai, chính là ưu điểm cốt lõi của luận văn này.

### Quy trình hỗ trợ bởi hệ thống đề xuất

* + - 1. *Giai đoạn 1: Tiền xử lý xây dựng dữ liệu nền tảng Knowledge Graph*

Giai đoạn tiền xử lý đóng vai trò quan trọng trong việc chuẩn bị dữ liệu nền tảng để hệ thống đề xuất lộ trình học tập cá nhân hóa hoạt động hiệu quả. Mục tiêu chính là trích xuất, chuẩn hóa và tổ chức dữ liệu từ các nguồn tài liệu học thuật thành một đồ thị tri thức (Knowledge Graph - KG) trong Neo4j, đảm bảo tính logic, nhất quán và phù hợp với các tiêu chuẩn giáo dục. Quy trình này bao gồm ba bước chính: trích xuất và chuẩn hóa dữ liệu, kiểm tra và xác thực dữ liệu, và tích hợp dữ liệu vào Neo4j. Các kỹ thuật tiên tiến trong xử lý ngôn ngữ tự nhiên (NLP) và biểu diễn tri thức được ứng dụng để tối ưu hóa chất lượng dữ liệu.

1. **Trích xuất và chuẩn hóa dữ liệu từ tài liệu gốc**

* **Mô tả:**   
  Dữ liệu được trích xuất tự động từ các nguồn tài liệu học thuật như giáo trình, bài giảng, tài liệu PDF, và ghi chú bài giảng (lecture notes).   
  Quá trình này nhằm chuyển đổi nội dung không cấu trúc (tập hợp tri thức thông tin ẩn và hiện bằng ngôn ngữ tự nhiên) thành dữ liệu có cấu trúc (các nút kèm thuộc tính), phù hợp để xây dựng đồ thị tri thức.

Dữ liệu được trích xuất tự động từ các nguồn tài liệu học thuật như giáo trình, bài giảng và tài liệu PDF, chuyển đổi nội dung không cấu trúc (tập hợp tri thức thông tin ẩn và hiện bằng ngôn ngữ tự nhiên) thành dữ liệu có cấu trúc phù hợp (các nút kèm thuộc tính) để xây dựng đồ thị tri thức. Quá trình này dựa trên các nguyên tắc lý thuyết của biểu diễn tri thức và NLP.

* **Công cụ:**

Sử dụng mô hình ngôn ngữ lớn (LLM) kết hợp với kỹ thuật prompting tiên tiến -

**“Prompt SPR Generator”**. Dựa trên phương pháp Sparse Priming Representation (SPR), kết hợp với few-shot prompting, tham khảo từ nghiên cứu [32] [5], để tăng độ chính xác và khả năng tổng quát hóa khi trích xuất dữ liệu. Công cụ này cho phép tự động hóa quá trình nhận diện và chuẩn hóa các thực thể tri thức từ văn bản, đồng thời giảm thiểu sự can thiệp thủ công.

* **Kỹ thuật sử dụng:  
  Phương pháp trích xuất ba lớp**:
  + **Lớp 1 - Trích xuất khái niệm cốt lõi**:   
    Xác định các khái niệm chính, định nghĩa, và ví dụ từ tài liệu dựa trên loại nội dung và tần suất xuất hiện. Các khái niệm trong tiêu đề hoặc có tần suất cao được ưu tiên.
  + **Lớp 2 - Trích xuất dữ liệu ngữ cảnh và quan hệ:**   
    Áp dụng NLP và phân tích nhân quả để phát hiện mối quan hệ giữa các nút. Sử dụng Chain-of-Thought và Few-Shot Learning để gán trọng số cho các thuộc tính mối quan hệ dựa trên ngữ cảnh tài liệu..
  + **Lớp 3 - Tính năng mở rộng và xác thực**:   
    Mở rộng dữ liệu bằng cách tạo Semantic\_Tags và Focused\_Semantic\_Tags cho mỗi nút, dựa trên các bản thể giáo dục (educational ontologies) để đảm bảo tính liên quan. Kỹ thuật *Socratic Questioning* được dùng để khai thác kiến thức ngầm, nâng cao độ chính xác cho các khái niệm trừu tượng.
* **Kết quả:**

Tạo ra hai tệp: *nodes.csv* và *relationships.csv*.

1. **Kiểm tra và xác thực dữ liệu**

* **Mô tả:**   
  Sau khi dữ liệu được trích xuất, quá trình kiểm tra và xác thực sẽ được thực hiện nhằm đảm bảo sự chính xác, sự nhất quán và sự phù hợp với tài liệu gốc. Quy trình này dựa trên lý thuyết về chất lượng dữ liệu và đánh giá bản thể học [3] cũng như các tiêu chuẩn giáo dục (như Bloom’s Taxanomy, VARK,…)
* **Công cụ**:   
  Sử dụng **Prompt SPR Validation:** được thiết kế để sử dụng khả năng của các mô hình ngôn ngữ lớn tiên tiến (vd: SuperGrok Think, GPT-4.1 Reason) để kiểm tra lại các tệp được tạo ra ở bước trên tự động kiểm tra và chỉnh sửa dữ liệu dựa trên các quy tắc và tiêu chí tiền định. Đồng thời việc kiểm tra “lại” bằng AI tại một session (phiên trò chuyện khác) cũng có tác dụng kiểm chứng độ lặp lại của các nội dung được tạo sinh từ AI với nguồn dữ liệu cố định (tài liệu ban đầu).
* **Kỹ thuật sử dụng:**
  + Kiểm tra dựa trên nội dung:   
    Mỗi thuộc tính trong *nodes.csv* và *relationships.csv* được so sánh với nội dung của tài liệu gốc.
  + Kiểm tra định dạng:   
    Đảm bảo các quy ước đặt tên (như bằng chữ thường, hoặc dạng snake\_case).
  + Kiểm tra theo khung giáo dục:   
    Sử dụng Bloom’s Taxonomy để đánh giá Skill\_Level và các bản thể như WordNet, DBpedia để xác thực Semantic\_Tags.
  + Phân tích tần suất và ưu tiên:   
    Đánh giá mức độ quan trọng của khái niệm dựa trên tần suất xuất hiện.
  + Kiểm tra mối quan hệ:   
    Kiểm tra các mối quan hệ logic và phân cấp giữa các nút (ví dụ: REQUIRES, IS\_SUBCONCEPT\_OF,…) để đảm bảo chúng phản ánh cấu trúc của tài liệu và các mối quan hệ phụ thuộc giáo dục.
  + Đánh giá định lượng và định tính:   
    Kiểm tra các thuộc tính, và lý do nếu phát hiện các phần sai lỗi cần cải tiến.
  + Tự động đánh dấu và sửa chữa:   
    Tự động tạo Báo cáo đầu ra với chi tiết kết quả kiểm tra, tỷ lệ sai lỗi, vị trí sai lỗi, lý do sai lỗi, đồng thời một cặp file *validated\_nodes.csv* và *validated\_relationships.csv* đã được sửa lỗi.
  + Vòng lặp xác thực:   
    Quy trình xác thực có thể dưới sự kiểm soát của “người phê duyệt” hoặc tự động hóa quy trình hoàn toàn với vòng lặp chạy đến khi tất cả các mục đều có “Validation\_Status” là "Correct".
* **Kết quả**
  + Tạo ra các tệp *validated\_nodes.csv* và *validated\_relationships.csv*, chứa dữ liệu đã được kiểm tra và xác thực, sẵn sàng để tích hợp vào Neo4j.
  + Một báo cáo chi tiết được tạo ra, giúp theo dõi và hiểu rõ các đề xuất thay đổi từ nội dung tạo sinh từ LLM.

1. **Tích hợp dữ liệu vào Neo4j Knowledge Graph**

* **Mô tả**:   
  Dữ liệu đã xác thực được tải vào Neo4j Desktop để tạo một đồ thị tri thức hoàn chỉnh, phản ánh cấu trúc giáo dục từ tài liệu gốc.
* **Kết quả**:

Một đồ thị tri thức hoàn chỉnh sẵn sàng hỗ trợ cho giai đoạn vận hành của hệ thống đề xuất.

**A diagram of a flowchart

AI-generated content may be incorrect.**

**Hình 3.1 Giai đoạn 1 Tiền xử lý xây dựng dữ liệu nền tảng KG**

* + - 1. *Giai đoạn 2: Vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa*

Giai đoạn này mô tả quy trình nghiệp vụ của hệ thống đề xuất lộ trình học tập cá nhân hóa, từ khi học viên đưa ra yêu cầu học tập đến khi hoàn thành khóa học. Quy trình được thể hiện chi tiết qua Hình 3.2, bao gồm sáu bước chính được thiết kế để đảm bảo lộ trình học tập được tối ưu hóa và phù hợp với nhu cầu của từng học viên.

Quy trình vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa bao gồm các bước sau:

1. **Tiếp nhận yêu cầu học tập từ học viên**  
   Học viên nhập yêu cầu học tập bằng ngôn ngữ tự nhiên thông qua giao diện hệ thống.  
   Kết quả: Một tập hợp các nút khởi đầu tiềm năng liên quan đến mục tiêu của học viên  
   Ví dụ: Nếu học viên nhập “Học SQL cơ bản”, hệ thống xác định các tag như “SELECT”, “WHERE”
2. **Xây dựng lộ trình học tập cá nhân hóa**  
   Hệ thống đề xuất một chuỗi tri thức được cá nhân hóa, phù hợp với trình độ và mục tiêu của học viên.  
   Kết quả: Một lộ trình học tập cá nhân hóa, ví dụ: “SQL SELECT” → “GROUP BY” → “HAVING” → “AGGREGATION”.  
   Ví dụ: Học viên bắt đầu từ “SELECT” và được dẫn đến “AGGREGATION” qua các bước trung gian logic.
3. **Cung cấp nội dung học tập phù hợp**  
   Học viên nhận được tài liệu giảng dạy và bài kiểm tra được tùy chỉnh theo phong cách học tập cá nhân.  
   Kết quả: Nội dung thực tiễn, ví dụ: bài giảng về “GROUP BY” với ví dụ tính tổng lương theo phòng ban.  
   Ví dụ: Học viên học “GROUP BY” qua bài giảng và thực hành với dữ liệu mẫu.
4. **Đánh giá hiệu suất học tập**  
   Học viên thực hiện bài kiểm tra để đánh giá mức độ nắm bắt kiến thức.  
   Kết quả: Điểm số, phân tích hiệu suất và lịch sử học tập của học viên.  
   Ví dụ: Học viên đạt 80% ở bài kiểm tra “GROUP BY” và tiếp tục; nếu dưới 50%, giáo viên được thông báo.
5. **Phản hồi và điều chỉnh lộ trình**  
   Hệ thống cung cấp phản hồi và điều chỉnh lộ trình nếu cần thiết.  
   Kết quả: Lộ trình được điều chỉnh, ví dụ: bổ sung “HAVING” nếu học viên yếu ở “GROUP BY”.  
   Ví dụ: Học viên thất bại 2 lần ở “GROUP BY” và được chuyển sang bài học bổ trợ.
6. **Thu thập đánh giá từ học viên**  
   Học viên cung cấp phản hồi để cải thiện hệ thống.  
   Kết quả: Dữ liệu phản hồi được lưu trữ để tinh chỉnh hệ thống.  
   Ví dụ: Học viên đánh giá nội dung “GROUP BY” là “hơi khó” và hệ thống điều chỉnh độ khó.

So với các quy trình học trực tuyến truyền thống, quy trình này có các cải tiến nổi bật:

* Biểu diễn và truy xuất tri thức logic, hỗ trợ xây dựng lộ trình chính xác và hiệu quả.
* Hiểu yêu cầu tự nhiên và tạo nội dung học tập phù hợp với từng học viên.
* Kết hợp A\*, collaborative filtering, và Apriori để đề xuất và điều chỉnh lộ trình linh hoạt, tối ưu

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.2. Giai đoạn 2: Vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa

**Hình 3.2 Giai đoạn 2: Vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa**

Quy trình vận hành hệ thống đề xuất lộ trình học tập cá nhân hóa trong giai đoạn 2 là một giải pháp toàn diện, tận dụng công nghệ tiên tiến để tối ưu hóa trải nghiệm học tập.   
Từ các lý thuyết ở chương này giúp tạo nền tảng các phần triển khai chi tiết trong mục 3.5. Thiết kế hệ thống cũng như thử nghiệm trong thực tiễn.

## GIẢI PHÁP ĐỀ XUẤT

### Minh họa quy trình

Quy trình của hệ thống đề xuất lộ trình học tập cá nhân hóa được minh họa thông qua mô hình C4 Diagram. Theo *S.Brown* [4] mô hình C4 Diagram là một phương pháp trực quan hóa kiến trúc phần mềm, giúp làm rõ mối quan hệ giữa các thành phần và đảm bảo khả năng mở rộng cũng như bảo trì hiệu quả. Đây là khung lý thuyết để tổ chức kiến trúc hệ thống, từ việc tích hợp tri thức đến việc tùy chỉnh lộ trình học tập. Kiến trúc hệ thống từ cấp độ khái quát đến chi tiết thông qua 4 cấp độ: Context, Container, Component, và Code/Sub-component.

* **Context:**   
  Cấp độ tổng quan nhất, xem hệ thống như một "hộp đen" tương tác với các bên liên quan bên ngoài.
* **Container:**   
  Mô tả các đơn vị triển khai chính (như ứng dụng, cơ sở dữ liệu, API) và sự kết nối giữa chúng.
* **Component:**   
  Chi tiết hóa từng container thành các mô-đun hoặc thành phần chức năng bên trong.
* **Code/Sub-component:**   
  Đi sâu vào chi tiết mã nguồn hoặc các hoạt động cụ thể trong từng component.

Trong hệ thống đề xuất lộ trình học tập cá nhân hóa, mô hình C4 giúp:

* **Làm rõ tương tác**: Hiển thị cách các thành phần phối hợp để tạo ra lộ trình học tập.
* **Hỗ trợ sửa đổi**: Thuận lợi cho việc tích hợp yêu cầu mới (ví dụ: thêm nguồn dữ liệu) theo cả hướng từ trên xuống (top-down) và từ dưới lên (bottom-up).
* **Tăng tính linh hoạt**: Cho phép điều chỉnh một phần mà không ảnh hưởng toàn bộ hệ thống.

**System**: Hệ thống tổng thể đang được mô tả.

**Person**: Người dùng hoặc vai trò bên ngoài tương tác với hệ thống.

**External System**: Hệ thống bên ngoài mà hệ thống của bạn kết nối hoặc phụ thuộc.

**Database**: Cơ sở dữ liệu lưu trữ dữ liệu.

**Software System**: Ứng dụng phần mềm hoặc dịch vụ độc lập.

**Container**: Đơn vị triển khai độc lập (ví dụ: ứng dụng, API, cơ sở dữ liệu).

**Component**: Thành phần hoặc module bên trong một container.

* **Các cấp độ của C4 Diagram và ứng dụng trong hệ thống**

1. **Context Diagram :**   
   Đây là cấp độ cao nhất, thể hiện hệ thống như một thực thể duy nhất (hộp đen) tương tác với các bên liên quan bên ngoài (Hình 3.3). Cấp độ này giúp xác định ranh giới hệ thống và các tương tác chính với môi trường bên ngoài, bao gồm:

* System: Hệ thống đề xuất lộ trình học tập tổng thể đang được mô tả
* Person: Học sinh (người học) và giáo viên (người giám sát).
* External System: Nguồn dữ liệu bên ngoài (ví dụ: CSVDatSource).

A diagram of a learning path system

AI-generated content may be incorrect.

**Hình 3.3 Sơ đồ bối cảnh hệ thống**

Ví dụ: Nếu giáo viên yêu cầu bổ sung tài liệu mới, Context Diagram giúp *người quản lý hệ thống* xác định hệ thống cần tích hợp với nguồn dữ liệu bên ngoài tại điểm nào.

1. **Container Diagram:**Sơ đồ này cung cấp cái nhìn tổng quan về kiến trúc hệ thống, bao gồm các thành phần chính: cơ sở dữ liệu Neo4j (lưu trữ đồ thị tri thức), Llama Index (quản lý chỉ mục), OpenAI API (sinh nội dung), và ứng dụng Python (điều phối hoạt động) (Hình 3.4).  
   Ví dụ: Khi thêm nguồn dữ liệu mới, người quản lý hệ thống có thể xác định điểm tích hợp tại Neo4j hoặc Python Application mà không cần thay đổi toàn bộ kiến trúc

A diagram of a software company

AI-generated content may be incorrect.

**Hình 3.4 Sơ đồ các đối tượng chính của hệ thống**

1. **Component Diagram:**Ứng dụng Python được chi tiết hóa thành các mô-đun chức năng, hiển thị cách các mô-đun phối hợp, bao gồm: Configuration (cấu hình hệ thống), Data Loading (tải dữ liệu), Student Profile (hồ sơ người học), Path Determination (xác định lộ trình), Path Generation (tạo lộ trình), Content Generation (sinh nội dung), Assessment (đánh giá), và Update (cập nhật) (Hình 7).  
   Ví dụ: Khi ta cần tích hợp tài liệu mới, chỉ cần điều chỉnh *Data Loading* mà không ảnh hưởng đến *Path Generation* hay *Assessment.*

A diagram of a diagram

AI-generated content may be incorrect.

**Hình 3.5 Sơ đồ thành phần quy trình hệ thống**

1. **Code/Sub-Component Diagram:**Trong các Components sẽ bao gồm các sub-component (Hình 3.5). Cấp độ này mô tả chi tiết các hoạt động bên trong mỗi mô-đun. Hỗ trợ việc theo dõi và cải tiến mã nguồn, đảm bảo quá trình phát triển có sự tham gia đầy đủ giữa các bên quản lý và các bên kỹ thuật. Phù hợp với nguyên lý thiết kế hệ thống giáo dục hiện đại.

A screenshot of a computer

AI-generated content may be incorrect.

**Hình 3.6 Sơ đồ thành thành phần chi tiết**

### Kiến trúc hệ thống tích hợp

Hệ thống đề xuất lộ trình học tập cá nhân hóa được thiết kế với kiến trúc ba lớp chính: giao diện người dùng, lớp lưu trữ dữ liệu (Neo4j), và lớp logic xử lý (Python tích hợp với LLM LlamaIndex) (Hình 3.7)

A diagram of a software application

AI-generated content may be incorrect.

**Hình 3.7 Sơ đồ kiến trúc hệ thống**

* **Ưu điểm của kiến trúc tích hợp**:  
  Kiến trúc này mang lại nhiều lợi ích nổi bật:

1. Xử lý dữ liệu phức tạp

* Neo4j: Quản lý hiệu quả các mối quan hệ đa chiều trong đồ thị tri thức.
* LLM: Hiểu ngữ nghĩa sâu sắc của yêu cầu học viên.

2. Cá nhân hóa cao

Hệ thống đề xuất lộ trình dựa trên ngữ cảnh, mục tiêu, và trình độ của từng học viên.

3. Khả năng mở rộng

Neo4j và LlamaIndex hỗ trợ xử lý lượng dữ liệu lớn và nhiều người dùng đồng thời.

* **Hạn chế và thách thức**  
  Bên cạnh ưu điểm, hệ thống cũng đối mặt với một số khó khăn
* **Độ phức tạp tích hợp**: Việc kết nối Neo4j, LlamaIndex và LLM đòi hỏi cấu hình kỹ lưỡng và quản lý phiên bản chặt chẽ để đảm bảo tính tương thích giữa các thành phần.
* **Yêu cầu tài nguyên**: Sử dụng LLM cần lựa chọn mô hình phù hợp, mô hình không đủ tốt có thể dẫn đến chất lượng nội dung kém; hoặc ngược lại sử dụng mô hình quá hiện đại so với nhu cầu, có khả năng nhảy vọt chi phí, nhất là khi so với các hệ thống chỉ sử dụng cơ sở dữ liệu truyền thống.

### Quy trình xử lý từ dữ liệu thô đến đề xuất lộ trình

Quy trình xử lý từ dữ liệu thô đến đề xuất lộ trình học tập cá nhân hóa được thiết kế như một chuỗi các giai đoạn tích hợp chặt chẽ. Quy trình tận dụng kỹ thuật prompt engineering để nâng cao hiệu quả.   
Bao gồm hai giai đoạn chính: tiền xử lý dữ liệu và vận hành hệ thống đề xuất.  
Tóm tắt quy trình thể hiện qua các bước chính mô tả trong quy trình và Bảng 3.1.   
Giai đoạn 1, giai đoạn 2 được minh họa lần lượt qua Hình 3.8 , Hình 3.9.

A screenshot of a computer

AI-generated content may be incorrect.

**Hình 3.8 Quy trình tạo và xác thực dữ liệu**

A diagram of a program

AI-generated content may be incorrect.

**Hình 3.9 Quy trình đề xuất lộ trình học tập**

**Bảng 3.1. Tóm tắt quy trình**

|  |  |
| --- | --- |
| Bước | Mô tả |
| GĐ1 Trích xuất dữ liệu | Dữ liệu được trích xuất từ tài liệu học thuật và chuẩn hóa thành file CSV (nodes.csv, relationships.csv) bằng công cụ SPR Generator |
| GĐ1 Xác thực dữ liệu | SPR Validation kiểm tra tính nhất quán của dữ liệu trước khi nhập vào Neo4j |
| GĐ1 Nhập vào Neo4j | Dữ liệu đã xác thực được nhập vào Neo4j, tạo đồ thị tri thức với các nút và mối quan hệ như NEXT, REQUIRES,...  Các nút đại diện cho thực thể tri thức và các mối quan hệ như REQUIRES, NEXT, REMEDIATES ,… được tạo hoặc hợp nhất dựa trên thuộc tính như Semantic\_Tags |
| GĐ2 Xác định điểm bắt đầu | VectorStoreIndex tìm nút phù hợp dựa trên hồ sơ học viên |
| GĐ2 Xác định điểm đích | Hệ thống gợi ý các kiến thức tiên quyết chưa được học dựa trên mục tiêu |
| GĐ2 Sinh lộ trình | Thuật toán A\* sử dụng heuristic cá nhân hóa dựa trên các yếu tố như trình độ hiện tại của học viên (Skill\_Level) và độ khó của khái niệm (Difficulty), sử dụng nền tảng tính toán chi phí đường đi tối ưu bằng công thức f(n) = g(n) + h(n) |
| GĐ2 Sinh nội dung | Tạo tài liệu học tập và bài kiểm tra bằng LLM (20 câu, phân bổ độ khó) |
| GĐ2 Đánh giá và cập nhật | Kết quả học tập được đánh giá và cập nhật vào hồ sơ học viên. Hiệu quả của quy trình được đánh giá qua độ chính xác của lộ trình đề xuất (tỷ lệ khái niệm được học viên hoàn thành đúng mục tiêu) và thời gian phản hồi trung bình của hệ thống (dưới 5 giây cho mỗi yêu cầu) |

## YÊU CẦU HỆ THỐNG

### Yêu cầu chức năng

Yêu cầu chức năng (FR) xác định các khả năng cốt lõi mà hệ thống đề xuất lộ trình học tập cá nhân hóa cần có.  
**Danh sách yêu cầu chức năng chính**

1. **Quản lý và truy vấn đồ thị tri thức**Lưu trữ và quản lý KG trong Neo4j, hỗ trợ truy vấn Cypher để trích xuất thông tin, đảm bảo tính nhất quán dữ liệu khi cập nhật.
2. **Xử lý yêu cầu học tập bằng ngôn ngữ tự nhiên**Nhận diện và chuyển đổi yêu cầu ngôn ngữ tự nhiên thành truy vấn Cypher qua LlamaIndex Property Graph Index, kết hợp VectorStoreIndex để tìm kiếm ngữ nghĩa dựa trên lịch sử học tập và mục tiêu. Đảm bảo độ chính xác trong việc ánh xạ yêu cầu người dùng với các khái niệm trong đồ thị tri thức.
3. **Đề xuất lộ trình học tập cá nhân hóa**Sinh lộ trình tối ưu bằng thuật toán A\* với heuristic cá nhân hóa, dựa trên trình độ, độ khó, và tiên quyết; sử dụng LLM để điều chỉnh lộ trình khi cần. Hỗ trợ gợi ý các kiến thức tiên quyết, các kiến thức con, các kiến thức hỗ trợ cho học viên khi cần. Cho phép cơ chế chọn lựa sử dụng LLM hoàn toàn để sinh lộ trình học tập.
4. **Sinh nội dung học tập và bài kiểm tra**  
   Tạo nội dung và bài kiểm tra đa dạng qua LLM, phù hợp với mục tiêu và phong cách học tập. Tạo bài kiểm tra trắc nghiệm với mức độ khó phân bổ hợp lý, nêu ra các thông tin định nghĩa, ví dụ và các lỗi thường gặp.
5. **Đánh giá kết quả học**Cập nhật hồ sơ học viên (lịch sử, hiệu suất) để tinh chỉnh lộ trình.
6. **Tích hợp và tương tác với người dùng**  
   Cung cấp giao diện người dùng (UI) thân thiện, cho phép học viên nhập yêu cầu và xem lộ trình học tập được đề xuất. Đồng thời, cung cấp giao diện báo cáo tiến độ và hiệu suất học tập cho giáo viên.
7. **Quản lý dữ liệu**  
   Hỗ trợ nhập/xuất dữ liệu dạng CSV. Toàn bộ yêu cầu chức năng được tóm tắt trong Bảng 3.2.

**Bảng 3.2. Tóm tắt yêu cầu chức năng**

|  |  |
| --- | --- |
| Mã yêu cầu | Mô tả |
| FR1 | Quản lý và truy vấn đồ thị tri thức. |
| FR2 | Xử lý yêu cầu học tập bằng ngôn ngữ tự nhiên. |
| FR3 | Đề xuất lộ trình học tập cá nhân hóa. |
| FR4 | Sinh nội dung học tập và bài kiểm tra. |
| FR5 | Đánh giá và cập nhật hồ sơ học tập của học viên. |
| FR6 | Tích hợp với giao diện người dùng. |
| FR7 | Hỗ trợ nhập/xuất dữ liệu từ dưới dạng CSV |

### Yêu cầu dữ liệu

Hệ thống cần đảm bảo lưu trữ và quản lý đồ thị tri thức trong Neo4j một cách hiệu quả, với các yêu cầu sau:

* **Cấu trúc dữ liệu mở rộng**:
* Các nút KnowledgeNode bao gồm các thuộc tính bắt buộc như Document\_ID, Node\_ID, Node\_Label, Context, Sanitized\_Concept, Definition, Example, Skill\_Level, Difficulty, Priority, Learning\_Objective, Time\_Estimate, Prerequisites, Semantic\_Tags, Focused\_Semantic\_Tags.
* Các mối quan hệ như NEXT, REQUIRES, SIMILAR\_TO phải bao gồm thuộc tính Weight, Dependency.
* **Quản lý dữ liệu thông minh**:   
  Áp dụng các thuật toán để tự động làm sạch và cập nhật đồ thị tri thức. Đồng thời cung cấp cơ chế hỗ trợ chuyên gia kiểm tra tính đúng đắn của toàn bộ dữ liệu trong KG.  
  Hỗ trợ các bộ truy vấn Cypher để trích xuất thông tin từ KG nhanh chóng (Cypher Procedure).  
  Đảm bảo tính nhất quán dữ liệu khi nhập hoặc chỉnh sửa các nút và mối quan hệ.

### Yêu cầu phi chức năng

Yêu cầu phi chức năng (NFR) xác định các tiêu chí về hiệu suất, mở rộng, độ tin cậy, bảo mật và khả năng sử dụng, đảm bảo hệ thống hoạt động ổn định trong môi trường giáo dục.

**Danh sách yêu cầu phi chức năng**Với quy mô trên 1000 nút và mối quan hệ, ta có các NFR như sau:

* **Hiệu suất**
* Sử dụng các kỹ thuật tối ưu hóa truy vấn Cypher, đảm bảo thời gian phản hồi trung bình dưới 3 giây cho các truy vấn trực tiếp, hoặc dưới 5 giây cho các tương tác với giao diện.
* Tối ưu hóa LLM để giảm độ trễ khi sinh nội dung.
* **Khả năng mở rộng**
* Hỗ trợ hàng nghìn người dùng đồng thời với độ trễ tối thiểu, tận dụng khả năng mở rộng ngang (horizontal scaling) của Neo4j.
* Cho phép tích hợp các nguồn tri thức mới mà không cần tái cấu trúc hệ thống.
* **Độ tin cậy và khả dụng**
* Hệ thống có thể hoạt động liên tục với thời gian ngừng hoạt động tối thiểu (<1% tổng thời gian).
* Có cơ chế xử lý bất thường (khi LLM hoặc Neo4j không khả dụng) để không làm gián đoạn trải nghiệm người dùng.
* **Bảo mật và bảo vệ dữ liệu**
* Bảo vệ dữ liệu cá nhân của học viên bằng mã hóa và kiểm soát truy cập.
* Ngăn chặn truy vấn đồ thị KG tiết lộ thông tin nhạy cảm (theo GDPR).
* **Khả năng sử dụng**
* Giao diện người dùng phải hỗ trợ các thiết bị di động.
* Hỗ trợ đa ngôn ngữ nếu triển khai cho đối tượng quốc tế.
* Cung cấp tài liệu hướng dẫn sử dụng rõ ràng cho người dùng cuối.
* **Khả năng bảo trì**Thiết kế hệ thống với các thành phần modular để dễ dàng cập nhật hoặc thay thế. Toàn bộ yêu cầu phi chức năng được tóm tắt trong Bảng 3.3.

**Bảng 3.3. Tóm tắt yêu cầu phi chức năng**

|  |  |
| --- | --- |
| Mã yêu cầu | Mô tả |
| NFR1 | Phản hồi truy vấn nhanh chóng (<5 giây). |
| NFR2 | Hỗ trợ hàng nghìn người dùng đồng thời. |
| NFR3 | Hoạt động liên tục với thời gian gián đoạn <1%. |
| NFR4 | Tuân thủ GDPR. |
| NFR5 | Giao diện thân thiện, hỗ trợ đa ngôn ngữ. |
| NFR6 | Thiết kế modular dễ bảo trì. |

## THIẾT KẾ HỆ THỐNG

### Chi tiết các thành phần hệ thống

* **Cơ sở dữ liệu đồ thị Neo4j**

Neo4j lưu trữ đồ thị tri thức với các nút (KnowledgeNode) biểu thị khái niệm học tập và các mối quan hệ bao gồm REQUIRES, NEXT, IS\_PREREQUISITE\_OF, REMEDIATES, HAS\_ALTERNATIVE\_PATH, SIMILAR\_TO, và IS\_SUBCONCEPT\_OF. Các mối quan hệ này được phản ánh đầy đủ cách các khái niệm được liên kết trong hệ thống, từ điều kiện tiên quyết đến các khái niệm tương tự hoặc thay thế.   
Dữ liệu được nhập từ các cặp tệp CSV (*nodes.csv*, *relationships.csv*) thông qua hàm check\_and\_load\_kg, sử dụng lệnh LOAD CSV của Cypher để tạo hoặc cập nhật nút và mối quan hệ.  
Để đảm bảo tính nhất quán và giảm trùng lặp, hệ thống hợp nhất các nút dựa trên độ tương đồng Jaccard của thuộc tính Semantic\_Tags (ngưỡng mặc định 0.7), với các mối quan hệ được chuyển hướng sang nút hợp nhất và nút cũ bị xóa.  
Neo4j hỗ trợ truy vấn hiệu quả bằng ngôn ngữ Cypher trực tiếp bằng python code thông qua hàm execute\_cypher\_query.  
Dữ liệu trong Neo4j liên tục được làm giàu bằng cách thu thập nguồn tài liệu đã xác minh, chuyển đổi thông qua quy trình Giai đoạn 1 (đã đề cập ở mục 3.2.2.1).

* **Llama-index và chỉ mục đồ thị**

LlamaIndex là thành phần trung gian kết nối Neo4j với LLM, cung cấp hai chỉ mục chính: Property Graph Index và Vector Store Index, tích hợp trong một kiến trúc linh hoạt để hỗ trợ truy vấn đồ thị và tìm kiếm ngữ nghĩa.

* Property Graph Index:   
  Được khởi tạo từ Neo4jPropertyGraphStore, là lớp bao quát trong Llama-index đồng bộ với Neo4j, có tác dụng nhúng các nút trong đồ thị thành các vector ngữ nghĩa.   
  Chỉ mục này cho phép truy vấn đồ thị dựa trên ngôn ngữ tự nhiên đề cập đến thuộc tính của nút hoặc mối quan hệ, thông qua thuộc tính TextToCypherRetriever   
  Ví dụ: "Tìm các khái niệm liên quan đến SQL" sẽ được chuyển thành truy vấn Cypher gửi đến Neo4j và tự động trả về kết quả.
* Vector Store Index:   
  Được xây dựng từ các tài liệu văn bản (tạo bởi initialize\_vector\_index), chỉ mục này chuyển đổi các khái niệm thành vector embedding, tối ưu hóa việc tìm kiếm ngữ nghĩa từ đó xác định các khái niệm liên quan nhanh chóng.

Hai chỉ mục này hoạt động độc lập nhưng bổ trợ lẫn nhau: PropertyGraphIndex xử lý quan hệ cấu trúc, trong khi VectorStoreIndex tăng cường khả năng tìm kiếm dựa trên ngữ nghĩa. Sự kết hợp này được triển khai trong mã nguồn để đảm bảo truy xuất dữ liệu hiệu quả từ đồ thị tri thức

* **Mô hình Ngôn ngữ Lớn (LLM)**

GPT-3.5 Turbo được tích hợp thông qua OpenAI trong LlamaIndex, với cấu hình temperature=0.1 (hạn chế tối đa sự sáng tạo, tập trung vào sự chính xác) và system\_prompt chuyên về giáo dục ("*Bạn là chuyên gia lộ trình học tập chuyên về giáo dục*").   
LLM sinh nội dung học tập (generate\_learning\_content) và bài kiểm tra (generate\_quiz) dựa trên hồ sơ học sinh, bao gồm phong cách học tập, trình độ, và mục tiêu. Cơ chế embedding không được xử lý trực tiếp bởi LLM mà do VectorStoreIndex đảm nhiệm, chuyển đổi khái niệm thành vector để tìm kiếm ngữ nghĩa. LLM tập trung vào việc tạo nội dung ngôn ngữ tự nhiên, được tinh chỉnh bằng prompt engineering để đảm bảo tính cá nhân hóa và phù hợp với ngữ cảnh giáo dục.

* **Tích hợp các thành phần**

Sự tích hợp được điều phối qua hàm run\_learning\_session, quản lý quy trình từ xác định điểm xuất phát, xây dựng lộ trình, đến thực thi và đánh giá phiên học tập.  
Các thuật toán tối ưu hóa được tích hợp trực tiếp vào kiến trúc hệ thống:

* Thuật toán A\*: Trong generate\_learning\_path, tối ưu lộ trình học tập dựa trên trọng số động, xem xét độ khó, trình độ kỹ năng và thời gian ước tính.
* Lọc cộng tác: Hàm collaborative\_filtering sử dụng cosine similarity để đề xuất khái niệm dựa trên hành vi học tập của học sinh tương đồng, với ngưỡng tương đồng tối thiểu là 0.5.
* Apriori: Hàm apply\_apriori khám phá quy tắc liên kết từ lịch sử học tập, áp dụng ngưỡng hỗ trợ (min\_support=0.1) và độ tin cậy (min\_threshold=0.5) để nâng cao khả năng đề xuất.

### Các tính năng nâng cao

Hệ thống đồng thời cũng cung cấp các tính năng tiên tiến như sau:

* Cá nhân hóa: Hàm load\_student\_profile tải hồ sơ học sinh, điều chỉnh lộ trình theo trình độ, phong cách học tập và sở thích thông qua generate\_learning\_path.
* Đánh giá liên tục: Hàm evaluate\_quiz và update\_student\_profile cập nhật hiệu suất học tập, và update\_learning\_path hỗ trợ điều chỉnh lộ trình nếu điểm dưới ngưỡng.
* Quản lý thời gian: Hàm get\_time\_spent theo dõi thời gian học tập, hỗ trợ tính toán tốc độ học (calculate\_learning\_speed).
* Gợi ý tiên quyết: Hàm suggest\_prerequisites đề xuất khái niệm cần học trước, đảm bảo lộ trình logic.
* Điều chỉnh heuristic thuật toán: Hàm adjust\_heuristic\_weights tinh chỉnh trọng số của A\* dựa trên phản hồi học sinh, tối ưu hóa hiệu suất đề xuất.

**Áp dụng kỹ thuật structured prompting**

Structured prompting định hướng LLM trích xuất dữ liệu chính xác và có cấu trúc,với các hướng dẫn cụ thể. Sử dụng các phần MISSION, INPUT, OUTPUT, PROCESS giúp chỉ định vai trò của LLM là "SPR compressor", đầu vào là tài liệu giáo dục, và đầu ra là hai file CSV.

Bằng định dạng CSV, các cột cụ thể trong nodes.csv và relationships.csv được ánh xạ trực tiếp vào nút và mối quan hệ trong Neo4j, giảm thiểu sai sót trong quá trình chuyển đổi.

Để xử lý dữ liệu thiếu, chỉ dẫn như "Ghi 'Not Available' nếu thuộc tính không có" giúp duy trì tính toàn vẹn của đồ thị, tránh suy diễn không cần thiết

### Cấu trúc nút và thuộc tính

Đồ thị tri thức trong hệ thống bao gồm hai loại nút chính: KnowledgeNode và Student, được thiết kế để quản lý tri thức học tập và thông tin học viên một cách hiệu quả, hỗ trợ cá nhân hóa lộ trình học tập

* **KnowledgeNode:**Đại diện cho các tri thức học tập với các thuộc tính được chia thành hai nhóm: bắt buộc và tùy chọn.  **A. Thuộc tính bắt buộc:**

Dưới đây là danh sách các thuộc tính bắt buộc của nút KnowledgeNode, kèm theo mô tả và hướng dẫn trích xuất được sử dụng trong promp:

* Document\_ID:
  + **Mô tả**: Mã định danh duy nhất của tài liệu chứa nút.
  + **Hướng dẫn**: Trích xuất từ siêu dữ liệu tài liệu (ví dụ: tiêu đề, tên tệp) hoặc gán một chuỗi duy nhất (ví dụ: "BI\_Lecture\_2"). Đảm bảo tính nhất quán cho tất cả các nút từ cùng một tài liệu.
* Node\_ID:
  + **Mô tả**: Mã định danh duy nhất của nút trong KG.
  + **Hướng dẫn**: Định dạng theo cấu trúc Node\_Label:Sanitized\_Concept (ví dụ: "concept:mean"). Đảm bảo tính duy nhất trên toàn KG.
* Node\_Label:
  + **Mô tả**: Loại nút, thể hiện vai trò của nó trong lộ trình học tập.
  + **Hướng dẫn**: Gán từ danh sách các loại được xác định trong Layer 1: "Concept" (ý tưởng cơ bản), "Lesson" (đơn vị học tập), "Quiz" (hoạt động đánh giá), "Experiment" (hoạt động thực hành), "Topic" (chủ đề lớn), hoặc "Resource" (tài liệu bổ sung).
* Sanitized\_Concept:
  + **Mô tả**: Tên khái niệm được chuẩn hóa ở dạng lowercase, snake\_case.
  + **Hướng dẫn**: Chuyển tên khái niệm thành chữ thường, thay thế khoảng trắng bằng dấu gạch dưới và loại bỏ ký tự đặc biệt (ví dụ: "Mean" → "mean"). Đảm bảo tên là duy nhất trong tài liệu.
* Context:
  + **Mô tả**: Lĩnh vực hoặc tiểu lĩnh vực cụ thể của khái niệm trong tài liệu.
  + **Hướng dẫn**: Trích xuất từ tiêu đề chương, mục hoặc văn bản xung quanh liên quan đến chủ đề (ví dụ: "data\_preprocessing" từ một chương về thống kê).
* Definition:
  + **Mô tả**: Giải thích ngắn gọn và rõ ràng về khái niệm.
  + **Hướng dẫn**: Trích xuất trực tiếp từ tài liệu, ưu tiên định nghĩa rõ ràng. Nếu có nhiều định nghĩa, chọn định nghĩa được nhắc đến nhiều nhất hoặc trung tâm nhất. Nếu không có, suy ra một định nghĩa ngắn gọn từ ngữ cảnh.
* Example:
  + **Mô tả**: Ví dụ cụ thể minh họa khái niệm.
  + **Hướng dẫn**: Trích xuất ví dụ rõ ràng và đại diện từ tài liệu. Nếu có nhiều ví dụ, chọn ví dụ ngắn gọn và phù hợp nhất. Nếu không có, ghi "Not Available".
* Learning\_Objective:
  + **Mô tả**: Mục tiêu giáo dục cụ thể liên quan đến khái niệm.
  + **Hướng dẫn**: Trích xuất từ các tuyên bố rõ ràng trong tài liệu (ví dụ: mục tiêu học tập, phần mục tiêu chương) hoặc suy ra từ cách khái niệm được trình bày. Ghi "Not Available" nếu không rõ ràng.
* Skill\_Level:
  + **Mô tả**: Mức độ **nhận thức** cần thiết để nắm vững khái niệm, dựa trên Bloom’s Taxonomy: "Remember", "Understand", "Apply", "Analyze", "Evaluate", "Create".

Giúp hệ thống đánh giá khả năng tư duy của học sinh và chọn khái niệm phù hợp với trình độ nhận thức hiện tại (ví dụ: một học sinh chỉ ở mức "Understand" sẽ không phù hợp với khái niệm yêu cầu "Analyze").

* + **Tiêu chí**:
* "Remember": Nhớ lại thông tin cơ bản.
* "Understand": Giải thích khái niệm.
* "Apply": Áp dụng khái niệm trong ngữ cảnh.
* "Analyze": Phân tích hoặc so sánh các thành phần.
* "Evaluate": Đánh giá hoặc phê bình.
* "Create": Thiết kế hoặc xây dựng điều mới.
  + **Chain-of-Thought Prompting**:
* Xác định mục tiêu học tập hoặc nhiệm vụ được mô tả trong tài liệu.
* Đối chiếu nhiệm vụ với các cấp độ của Bloom’s Taxonomy.
* Chọn cấp độ phù hợp nhất dựa trên độ phức tạp của nhiệm vụ.
  + **Ví dụ**:
* "mean": Skill\_Level = "Apply".
* "hypothesis\_testing": Skill\_Level = "Analyze".
* Time\_Estimate:
  + **Mô tả**: Thời gian ước tính (tính bằng phút) để người học trung bình hiểu khái niệm.
  + **Tiêu chí**:
* Cơ bản (5-15 phút): Khái niệm đơn giản.
* Trung cấp (15-30 phút): Khái niệm phức tạp vừa.
* Nâng cao (30-60 phút): Khái niệm phức tạp.
  + **Chain-of-Thought Prompting**:
* Đánh giá độ phức tạp của khái niệm dựa trên định nghĩa và ví dụ.
* Xem xét các bước, công thức hoặc ví dụ liên quan.
* Gán thời gian dựa trên tiêu chí.
  + **Ví dụ**:
* "mean": Time\_Estimate = "15".
* "hypothesis\_testing": Time\_Estimate = "45".
* Difficulty:
  + **Mô tả**: Độ phức tạp của khái niệm, được phân loại là "STANDARD" hoặc "ADVANCED".

Phân loại khái niệm theo độ phức tạp tổng thể, hỗ trợ điều chỉnh lộ trình học tập phù hợp với khả năng chung của học sinh

* + **Tiêu chí**:
* "STANDARD": Phù hợp với người mới bắt đầu, yêu cầu kiến thức nền tối thiểu.
* "ADVANCED": Yêu cầu kiến thức trước hoặc liên quan đến lý luận phức tạp.
  + **Chain-of-Thought Prompting**:
* Xem xét định nghĩa và ví dụ của khái niệm.
* Đánh giá xem có yêu cầu kiến thức trước hay không (kiểm tra Prerequisites).
* Gán "STANDARD" hoặc "ADVANCED".
  + **Ví dụ**:
* "mean": Difficulty = "STANDARD".
* "hypothesis\_testing": Difficulty = "ADVANCED".
* Priority:
  + **Mô tả**: Mức độ quan trọng của nút trong lộ trình học tập, được đánh giá trên thang điểm từ 1-5.

Đảm bảo các khái niệm quan trọng được học trước, tối ưu hóa hiệu quả học tập.

* + **Thang điểm**:
* 5: Quan trọng, nền tảng.
* 4: Rất quan trọng.
* 3: Quan trọng vừa phải.
* 2: Ít quan trọng.
* 1: Tùy chọn.
  + **Chain-of-Thought Prompting**:
* Đánh giá vai trò của khái niệm trong tài liệu (ví dụ: tần suất xuất hiện, nhấn mạnh trong tiêu đề).
* Kiểm tra xem khái niệm có phải là điều kiện tiên quyết cho các khái niệm khác hay không.
* Gán giá trị dựa trên thang điểm.
  + **Ví dụ**:
* "variables": Priority = 5.
* "advanced\_data\_structures": Priority = 2.
* Prerequisites:
  + **Mô tả**: Danh sách các khái niệm cần nắm trước, được liệt kê dưới dạng chuỗi phân tách bằng dấu phẩy.
  + **Hướng dẫn**: Trích xuất các điều kiện tiên quyết được nêu rõ ràng hoặc suy ra từ các mối quan hệ như REQUIRES. Ghi "Not Available" nếu không có.
  + **Ví dụ**: "concept:data\_preprocessing".

Semantic\_Tags:

* + **Mô tả**: Tập hợp từ khóa ngữ nghĩa rộng để hỗ trợ tìm kiếm và khớp nối khái niệm.
  + **Hướng dẫn**: Trích xuất từ nội dung tài liệu (ví dụ: tiêu đề, phụ đề, từ khóa) và mở rộng bằng bản thể học giáo dục (ví dụ: WordNet, DBpedia) nếu cần. Đảm bảo ít nhất 5 thẻ, sắp xếp theo thứ tự bảng chữ cái.
  + **Ví dụ**: "arithmetic\_mean;average;central\_tendency;data\_analysis;descriptive\_statistics;expected\_value;finance;forecasting;machine\_learning;mean;probability\_theory;statistics".
* Focused\_Semantic\_Tags:
  + **Mô tả**: Tập hợp từ khóa ngữ nghĩa ngắn gọn (3-5 thẻ) tập trung vào trọng tâm của khái niệm để tạo nội dung hoặc câu hỏi chính xác.
  + **Hướng dẫn**: Chọn các thẻ từ Semantic\_Tags, ưu tiên thẻ phù hợp nhất với Definition và Context, sắp xếp theo thứ tự bảng chữ cái. Ghi "Not Available" nếu không đủ ngữ cảnh.
  + **Ví dụ**: "arithmetic\_mean;central\_tendency;data\_analysis".

**B. Các thuộc tính tùy chọn:**

* Numerical\_Data: Dữ liệu số liên quan đến khái niệm (ví dụ: "Mean: 58, N=12").
  + **Hướng dẫn**: Trích xuất từ bảng, phần kết quả hoặc mô tả định lượng trong tài liệu.
  + **Xác nhận**: Đảm bảo số liệu chính xác và phù hợp với khái niệm.
* Learning\_Style\_Preference: Phong cách học tập được đề xuất (ví dụ: "visual").
  + **Hướng dẫn**: Suy ra từ phương pháp giảng dạy được mô tả (ví dụ: "sử dụng biểu đồ") hoặc ghi "Not Available" nếu không rõ.
  + **Xác nhận**: Đảm bảo phù hợp với ngữ cảnh sư phạm.
* Reference: Nguồn tham khảo (ví dụ: "Jiawei Han, 2012").
  + **Hướng dẫn**: Trích xuất từ trích dẫn, danh mục tham khảo hoặc đề cập đến tác giả.
  + **Xác nhận**: Đảm bảo tính chính xác và liên quan.
* Learning\_Tips: Gợi ý hoặc mẹo học tập (ví dụ: "Practice with datasets").
  + **Hướng dẫn**: Trích xuất từ các phần như "Đề xuất" hoặc "Thực hành tốt nhất".
  + **Xác nhận**: Đảm bảo mẹo thực tế và dựa trên tài liệu.
* Detailed\_Example: Ví dụ chi tiết (ví dụ: "Mean salary calculation across departments").
  + **Hướng dẫn**: Trích xuất các trường hợp mở rộng từ ví dụ trong tài liệu.
  + **Xác nhận**: Đảm bảo liên quan đến khái niệm.
* Real\_World\_Application: Ứng dụng thực tế (ví dụ: "Forecasting sales").
  + **Hướng dẫn**: Trích xuất từ các phần "Ứng dụng" hoặc "Trường hợp sử dụng".
  + **Xác nhận**: Đảm bảo tính thực tiễn và liên quan.
* Common\_Errors: Các lỗi phổ biến (ví dụ: "Sensitivity to outliers, Misinterpreting as only measure").
  + **Hướng dẫn**: Trích xuất từ các thảo luận về sai lầm hoặc thách thức.
  + **Xác nhận**: Đối chiếu với tài liệu hoặc bản thể học giáo dục.
* Instruction\_Content\_Template: Mẫu nội dung giảng dạy (ví dụ: "Learn {{Sanitized\_Concept}} using {{Definition}}.").
  + **Hướng dẫn**: Tạo nếu nội dung giảng dạy được ngụ ý hoặc nêu rõ.
  + **Xác nhận**: Đảm bảo mẫu phù hợp với khái niệm.
* MCQ\_Template: Mẫu câu hỏi trắc nghiệm (ví dụ: "What is {{Sanitized\_Concept}}? A) {{Correct\_Answer}} B) {{Misconception\_1}} ...").
  + **Hướng dẫn**: Tạo nếu nội dung đánh giá được ngụ ý hoặc nêu rõ.
  + **Xác nhận**: Đảm bảo mẫu hỗ trợ tạo câu hỏi.
* Correct\_Answer: Đáp án đúng (ví dụ: "58").
  + **Hướng dẫn**: Trích xuất từ ví dụ hoặc giải pháp trong tài liệu.
  + **Xác nhận**: Xác minh tính chính xác.
* Remediation\_Path: Nút khắc phục (ví dụ: "concept:data\_preprocessing").
  + **Hướng dẫn**: Suy ra từ mối quan hệ REMEDIATES hoặc gợi ý khắc phục.
  + **Xác nhận**: Đảm bảo phù hợp với tài liệu hoặc mối quan hệ.
* Source\_Section: Phần tài liệu (ví dụ: "Section 2.1: Central Tendency").
  + **Hướng dẫn**: Trích xuất từ cấu trúc tài liệu (ví dụ: tiêu đề chương).
  + **Xác nhận**: Đảm bảo phần được trích dẫn chính xác.
* Multimedia\_Resources: Tài nguyên đa phương tiện (ví dụ: "video:mean\_calculation").
  + **Hướng dẫn**: Trích xuất từ các tham chiếu đến tài nguyên bên ngoài (ví dụ: liên kết, đề cập).
  + **Xác nhận**: Đảm bảo tính liên quan của tài nguyên.
* Node\_Category: Phân loại chi tiết (ví dụ: "Statistical\_Concept").
  + **Hướng dẫn**: Suy ra từ ngữ cảnh rộng hơn hoặc bản thể học giáo dục (ví dụ: "Statistics").
  + **Xác nhận**: Đảm bảo phân loại phù hợp với khái niệm.
* **Student**: Đại diện cho học viên, với các thuộc tính:
* student\_id: Định danh duy nhất (ví dụ: “stu001”).
* learning\_history: Lịch sử học tập (ví dụ: “concept:sql\_select,concept:sql\_where”).
* current\_level: Trình độ hiện tại (ví dụ: “Beginner”).
* performance\_details: Chi tiết hiệu suất (ví dụ: “80% on sql\_select quiz”).
* learning\_style\_preference: Phong cách học theo VARK (ví dụ: “reading\_writing”).
* preferred\_difficulty: Độ khó ưa thích (ví dụ: “STANDARD”).
* time\_availability: Thời gian học khả dụng (phút/tuần).
* long\_term\_goal: Mục tiêu dài hạn (ví dụ: “Master SQL”).
* LastUpdated: Thời điểm cập nhật cuối theo định dạng ISO (ví dụ: “2025-05-13”).

### Các loại mối quan hệ

Được phân thành hai nhóm chính: giữa các *KnowledgeNode* và giữa *Student* và *KnowledgeNode*.

* **Mối quan hệ giữa các KnowledgeNode**:

Có bảy loại mối quan hệ chính, mỗi loại hỗ trợ một khía cạnh cụ thể trong việc cấu trúc tri thức và lộ trình học tập.

* REQUIRES:
  + **Mô tả**: Chỉ ra rằng nút nguồn không thể được hiểu nếu không nắm vững nút đích trước.
  + **Hướng dẫn**: Sử dụng khi tài liệu nêu rõ sự phụ thuộc (ví dụ: "*Lộ trình học tập thích ứng yêu cầu hiểu về hệ thống gợi ý*"). Trích xuất từ các cụm từ như "yêu cầu", "cần thiết", hoặc các điều kiện tiên quyết được liệt kê.
  + **Ví dụ**: "*concept:adaptive\_learning\_path*" → "*concept:recommender\_systems*" (REQUIRES).
  + **Xác nhận**: Đối chiếu với nội dung tài liệu để đảm bảo sự phụ thuộc được nêu rõ ràng hoặc suy ra từ ngữ cảnh.
* IS\_PREREQUISITE\_OF:
  + **Mô tả**: Ngược lại với REQUIRES, cho biết nút đích phụ thuộc vào nút nguồn như một điều kiện tiên quyết.
  + **Hướng dẫn**: Sử dụng để đảm bảo trình tự học tập logic, khi tài liệu ngụ ý hoặc nêu rõ rằng nút nguồn cần được học trước nút đích.
  + **Ví dụ**: "*concept:recommender\_systems*" → "*concept:adaptive\_learning\_path*" (IS\_PREREQUISITE\_OF).
  + **Xác nhận**: Xác minh mối quan hệ ngược với REQUIRES hoặc kiểm tra văn bản để tìm các cụm từ như "trước khi học", "cơ sở cho".
* NEXT:
  + **Mô tả**: Chỉ ra trình tự học tập, nơi nút nguồn nên được học trước nút đích trong lộ trình.
  + **Hướng dẫn**: Sử dụng khi tài liệu gợi ý một thứ tự logic (ví dụ: "*Sau khi hiểu giá trị trung bình, người học nên nghiên cứu trung vị*"). Trích xuất từ các cụm từ như "sau đó", "tiếp theo".
  + **Ví dụ**: "*concept:mean*" → "*concept:median*" (NEXT).
  + **Xác nhận**: Đối chiếu với cấu trúc tài liệu (ví dụ: thứ tự các chương, mục) để đảm bảo trình tự hợp lý.
* REMEDIATES:
  + **Mô tả**: Chỉ ra rằng nút đích cung cấp nội dung khắc phục để giải quyết khó khăn với nút nguồn.
  + **Hướng dẫn**: Sử dụng khi tài liệu đề xuất lộ trình khắc phục hoặc nội dung bổ sung (ví dụ: "Nếu người học gặp khó khăn với kiểm định giả thuyết, họ nên ôn lại thống kê"). Trích xuất từ các cụm từ như "nếu gặp khó khăn", "ôn lại".
  + **Ví dụ**: "*concept:hypothesis\_testing*" → "*concept:statistics*" (REMEDIATES).
  + **Xác nhận**: Đảm bảo tài liệu có đề cập đến nội dung khắc phục hoặc mối quan hệ được suy ra từ ngữ cảnh.
* HAS\_ALTERNATIVE\_PATH:
  + **Mô tả**: Chỉ ra rằng nút đích cung cấp một lộ trình học tập thay thế nếu nút nguồn quá khó.
  + **Hướng dẫn**: Sử dụng khi tài liệu gợi ý các khái niệm thay thế (ví dụ: "Nếu kiểm định giả thuyết quá khó, người học có thể nghiên cứu thống kê mô tả"). Trích xuất từ các cụm từ như "thay vì", "lựa chọn khác".
  + **Ví dụ**: "*concept:hypothesis\_testing*" → "*concept:descriptive\_statistics*" (HAS\_ALTERNATIVE\_PATH).
  + **Xác nhận**: Kiểm tra tài liệu để đảm bảo lộ trình thay thế được đề xuất rõ ràng hoặc suy ra hợp lý.
* SIMILAR\_TO:
  + **Mô tả**: Chỉ ra rằng nút nguồn và nút đích có sự tương đồng về ngữ nghĩa, cho phép khám phá các khái niệm liên quan.
  + **Hướng dẫn**: Sử dụng khi tài liệu ngụ ý sự tương đồng hoặc khi LLM suy ra dựa trên nội dung nút (ví dụ: "Giá trị trung bình và trung vị đều là thước đo xu hướng trung tâm"). Trích xuất từ các cụm từ như "tương tự", "cùng loại".
  + **Ví dụ**: "*concept:mean*"→ "*concept:median*" (SIMILAR\_TO).
  + **Xác nhận**: Đối chiếu với nội dung tài liệu hoặc so sánh Semantic\_Tags của hai nút để xác minh tính tương đồng.
* IS\_SUBCONCEPT\_OF:
  + **Mô tả**: Chỉ ra rằng nút nguồn là một trường hợp cụ thể hoặc danh mục con của nút đích, thể hiện mối quan hệ phân cấp.
  + **Hướng dẫn**: Sử dụng khi tài liệu mô tả mối quan hệ phân cấp (ví dụ: "Tư duy phản biện trong triết học là một dạng của tư duy phản biện"). Trích xuất từ các cụm từ như "là một loại", "thuộc về".
  + **Ví dụ**: "*concept:critical\_thinking\_philosophy*"→"*concept:critical\_thinking*" (IS\_SUBCONCEPT\_OF).
  + **Xác nhận**: Đảm bảo tài liệu có đề cập đến mối quan hệ phân cấp hoặc suy ra từ cấu trúc khái niệm.
* **Vai trò của các mối quan hệ KnowledgeNode trong ứng dụng KG**
* **Lý thuyết đồ thị (Graph Theory)**: Các mối quan hệ như REQUIRES, NEXT, IS\_SUBCONCEPT\_OF phản ánh cấu trúc liên kết, đảm bảo tính logic và khả năng điều hướng trong đồ thị tri thức.
* **Mô hình học tập tuần tự**: NEXT và IS\_PREREQUISITE\_OF tổ chức lộ trình học tập hiệu quả theo trình tự.
* **Tìm kiếm ngữ nghĩa (Semantic Search)**: SIMILAR\_TO tận dụng sự tương đồng để đề xuất khái niệm liên quan.
* **Học tập thích ứng**: REMEDIATES và HAS\_ALTERNATIVE\_PATH cá nhân hóa lộ trình dựa trên nhu cầu học viên.
* **Giá trị thuộc tính của mối quan hệ KnowledgeNode**

Mỗi mối quan hệ giữa các KnowledgeNode có hai thuộc tính chính:

* Weight (1-5):
  + **Mô tả**: Thể hiện mức độ quan trọng hoặc độ khó của mối quan hệ.
  + **Thang điểm**:
    - 5: Cực kỳ quan trọng (ví dụ: điều kiện tiên quyết nền tảng).
    - 4: Rất quan trọng.
    - 3: Quan trọng vừa phải.
    - 2: Có liên quan chút ít.
    - 1: Tối thiểu.
  + **Hướng dẫn**:
    - Xác định nút nguồn và nút đích.
    - Đánh giá mức độ phụ thuộc của nút nguồn vào nút đích (ví dụ: với REQUIRES, kiểm tra sự phụ thuộc mạnh).
    - Xem xét độ phức tạp của mối quan hệ (ví dụ: với NEXT, kiểm tra sự chuyển đổi có phức tạp không).
    - Gán giá trị dựa trên thang điểm.
  + **Ví dụ**:
    - "concept:variables" -> "concept:functions" (REQUIRES): Weight = 5.
    - "concept:mean" -> "concept:median" (NEXT): Weight = 3.
  + **Xác nhận**: Đối chiếu với nội dung tài liệu để đảm bảo giá trị phù hợp. Nếu không đủ ngữ cảnh, sử dụng giá trị mặc định (ví dụ: "3").
* Dependency **(1-5)**:
  + **Mô tả**: Thể hiện mức độ phụ thuộc của nút nguồn vào nút đích.
  + **Thang điểm**:
    - 5: Hoàn toàn phụ thuộc (không thể hoạt động nếu thiếu nút đích).
    - 4: Phụ thuộc mạnh.
    - 3: Phụ thuộc vừa phải.
    - 2: Phụ thuộc yếu.
    - 1: Độc lập.
  + **Hướng dẫn**:
    - Phân tích nội dung của nút nguồn.
    - Đánh giá mức độ quan trọng của nút đích đối với sự hiểu biết của nút nguồn.
    - Gán giá trị dựa trên thang điểm.

**Ví dụ**:

* + - "concept:functions" -> "concept:variables" (REQUIRES): Dependency = 5.
    - "concept:mean" -> "concept:data\_preprocessing" (REQUIRES): Dependency = 3.
  + **Xác nhận**: Đối chiếu với nội dung tài liệu. Nếu không đủ ngữ cảnh, sử dụng giá trị mặc định (ví dụ: "3").
* **Giữa Student và KnowledgeNode**:
* MASTERED:
* Mô tả: Chỉ ra rằng học viên đã thành thạo một khái niệm (KnowledgeNode).
* Thuộc tính:
* progress (%): Tiến độ hoàn thành (ví dụ: 80%).
* timestamp: Thời điểm hoàn thành (ví dụ: 2023-10-15 14:30).
* time\_spent (phút): Thời gian dành cho khái niệm (ví dụ: 120 phút).
* Ý nghĩa: Giúp theo dõi mức độ nắm vững kiến thức của học viên, hỗ trợ cá nhân hóa lộ trình học tập.

### Kiểm tra, chuẩn hóa và tải dữ liệu

* **Mục tiêu**:

Đảm bảo dữ liệu trong nodes.csv và relationships.csv chính xác, nhất quán và phù hợp với tài liệu gốc, từ đó nâng cao độ tin cậy của KG khi sử dụng trong Neo4j.

* **Phương pháp**:

Sử dụng một phiên bản khác của LLM, hoặc một phiên trò chuyện khác với phiên tạo ra các tệp *nodes.csv* và *relationships.csv* để tránh việc tái sử dụng memory cũ. Cách tiếp cận này thực tiễn hơn so với việc tinh chỉnh LLM ban đầu, vì nó cho phép kiểm chứng chéo và giảm thiểu sai sót một cách hiệu quả.

* **Quy trình chi tiết**:

Quá trình kiểm tra được chia thành hai phần: kiểm tra nút và kiểm tra mối quan hệ.

* **Chuẩn bị đầu vào**:
* Tài liệu gốc làm cơ sở tham chiếu.
* Cặp tệp *nodes.csv* và *relationships.csv,* hoặc *validated\_nodes.csv* và *validates\_relationships.cs* từ Giai đoạn 1.
* Prompt *Validation Node and Relationship.txt* hướng dẫn LLM thực hiện kiểm tra.
* **Kiểm tra tự động bằng LLM**:
* **Kiểm tra nút**:
* **Mục tiêu**: Xác thực các thuộc tính như Definition, Skill\_Level, Priority, Time\_Estimate, Difficulty, Prerequisites, và Semantic\_Tags.
* **Tiêu chí**:
* Skill\_Level: Đánh giá dựa trên Bloom’s Taxonomy. Ví dụ: Nếu tài liệu yêu cầu "Calculate the mean", Skill\_Level phải là "Apply" thay vì "Remember".
* Priority: Gán 4-5 cho khái niệm nền tảng, 1-2 cho khái niệm nâng cao.
* Time\_Estimate: Đảm bảo thời gian học phù hợp với độ phức tạp.
* Difficulty: "STANDARD" cho khái niệm cơ bản, "ADVANCED" cho khái niệm phức tạp.
* **Hành động**: Ghi nhận lỗi và đề xuất sửa đổi nếu thuộc tính không khớp với tài liệu.
* **Kiểm tra mối quan hệ**:
* **Mục tiêu**: Xác thực các thuộc tính như Relationship\_Type, Weight, và Dependency.
* **Tiêu chí**:
* Relationship\_Type: Đảm bảo các loại như NEXT, REQUIRES, SIMILAR\_TO,… phù hợp với tài liệu.
* Weight: Gán 4-5 cho mối quan hệ quan trọng, 1-2 cho mối quan hệ ít quan trọng.
* Dependency: Gán 4-5 nếu nút đích là thiết yếu, 1-2 nếu phụ thuộc yếu.
* **Hành động**: Ghi nhận lỗi và đề xuất sửa đổi nếu giá trị không hợp lý.
* **Tạo báo cáo chi tiết**:
* **Nội dung báo cáo**:
* **Trường hợp đúng (nút)**: [Node\_ID: concept:mean] - [Definition: "The arithmetic average of a set of values."] - [Correct] - [Reasoning: Matches document content in Section 2.1].
* **Trường hợp sai (nút)**: [Node\_ID: concept:mean] - [Skill\_Level: "Remember"] - [Incorrect] - [Reasoning: Document states 'Calculate the mean', aligns with 'Apply'] - [Suggested Correction: Skill\_Level: "Apply"].
* **Trường hợp đúng (mối quan hệ)**: [Source\_Node\_ID: concept:mean -> Target\_Node\_ID: concept:median] - [Relationship\_Type: NEXT] - [Correct] - [Reasoning: Document states 'After mean, study median'].
* **Trường hợp sai (mối quan hệ)**: [Source\_Node\_ID: concept:mean -> Target\_Node\_ID: concept:median] - [Weight: 1] - [Incorrect] - [Reasoning: NEXT is moderately important] - [Suggested Correction: Weight: 3].
* **Thống kê**: Tổng số nút/mối quan hệ đúng/sai, tỷ lệ chính xác (ví dụ: 95% thuộc tính trong *nodes.csv* là đúng).
* **Tạo tệp CSV đã chuẩn hóa**:
* **Kết quả**: Tạo *validated\_nodes.csv* và *validated\_relationships.csv* với dữ liệu đã được cập nhật.
* **Thêm cột**:
* Validation\_Status: "Correct" (không cần sửa) hoặc "Updated" (đã chỉnh sửa).
* Corrections\_Applied: Liệt kê sửa đổi (ví dụ: "Skill\_Level changed from 'Remember' to 'Apply'").
* **Mục đích**: Các tệp này sẵn sàng để nhập vào Neo4j, đảm bảo dữ liệu đạt chất lượng cao.
* **Tầm quan trọng của quy trình:**
* **Tăng độ tin cậy**: Sử dụng LLM khác với prompt riêng biệt để kiểm chứng chéo, giảm sai sót từ quá trình tạo sinh nội dung từ LLM (thực hiện ở một session trò chuyện mới đóng vai trò như thao tác kiểm tra độ lặp lại).
* **Đảm bảo tính nhất quán**: Các thuộc tính như Skill\_Level được đánh giá theo Bloom’s Taxonomy, phù hợp với mục tiêu giáo dục.
* **Hỗ trợ tích hợp**: Dữ liệu chuẩn hóa trong *validated\_nodes.csv* và *validated\_relationships.csv* có sẵn định dạng đúng để sẵn sàng nhập vào Neo4j, tối ưu hóa hiệu suất hệ thống.
* **Quy trình nhập dữ liệu**

Các hàm sử dụng:

* check\_and\_load\_kg(): Kiểm tra và tải dữ liệu vào Neo4j.
* merge\_properties(): Hợp nhất các nút trùng lặp dựa trên độ tương đồng Jaccard (jaccard\_similarity), loại bỏ dư thừa.
* verify\_graph(): Xác minh tính nhất quán và đảm bảo số lượng nút/mối quan hệ được thêm vào thành công. thành công

### Thiết kế thuật toán đề xuất

#### Mục tiêu của thuật toán

Thuật toán đề xuất lộ trình học tập cá nhân hóa là thành phần cốt lõi của hệ thống, được xây dựng dựa trên sự kết hợp giữa truy vấn KG và thuật toán heuristic hiện đại (Hình 3.10).   
Thuật toán này tạo ra một chuỗi các khái niệm học tập tối ưu, phù hợp với trình độ, phong cách học tập, nhằm thực hiện các mục tiêu chính sau:

* **Cá nhân hóa**: Đảm bảo lộ trình học tập phản ánh nhu cầu riêng của từng học viên, dựa trên các thuộc tính như current\_level, learning\_style\_preference, và long\_term\_goal trong hồ sơ học viên.
* **Tối ưu hóa**: Tìm đường đi ngắn nhất và hiệu quả nhất trong KG, xem xét các yếu tố như Time\_Estimate (thời gian ước tính), Difficulty (độ khó), và Priority (mức độ ưu tiên) của các khái niệm.
* **Linh hoạt**: Xử lý các trường hợp không có đường đi trực tiếp bằng cách tích hợp mô hình LLM để sinh lộ trình thay thế hoặc bổ sung nội dung học tập.
* **Hiệu quả**: Đáp ứng yêu cầu phi chức năng về thời gian phản hồi (dưới 5 giây, theo NFR1 trong Mục 3.4.2) và độ chính xác cao (trên 80%, theo tiêu chí đánh giá trong Mục 5.1.2).*A diagram of a computer flowchart

  AI-generated content may be incorrect.*

**Hình 3.10 Thuật toán A\* tích hợp vào quy trình**

#### Lựa chọn và lý do chọn thuật toán

A\* là một thuật toán tìm kiếm đường đi ngắn nhất, thường dùng trong các ứng dụng như định tuyến hoặc trò chơi. Nó thông minh hơn các thuật toán như Dijkstra vì nó sử dụng một hàm heuristic để ước lượng chi phí từ nút hiện tại đến đích, giúp tìm đường nhanh hơn. Trong luận văn, A\* được dùng để xây dựng lộ trình học tập, đảm bảo học sinh đi từ khái niệm ban đầu đến khái niệm mục tiêu qua các bước hợp lý và hiệu quả nhất.

Thuật toán A\* (A-Star) được chọn làm phương pháp chính để đề xuất lộ trình học tập cá nhân hóa, với các lý do sau:

* **Tìm kiếm đường đi tối ưu**: A\* là một thuật toán tìm kiếm heuristic, hiệu quả trong việc tìm đường đi ngắn nhất trên đồ thị có trọng số, phù hợp với cấu trúc KG trong Neo4j, nơi các mối quan hệ (NEXT, REQUIRES, v.v.) có thuộc tính Weight và Dependency.
* **Hỗ trợ cá nhân hóa**: Hàm heuristic được tùy chỉnh để ưu tiên các khái niệm khớp với hồ sơ học viên, như mức độ khó (Difficulty) hoặc phong cách học tập (Learning\_Style\_Preference).
* **Khả năng mở rộng**: A\* hoạt động hiệu quả trên đồ thị lớn, đáp ứng yêu cầu mở rộng của hệ thống (NFR2, Mục 3.4.2), vượt trội hơn Dijkstra (không dùng heuristic) về tốc độ xử lý.
* **Tích hợp linh hoạt**: A\* có thể kết hợp với các phương pháp bổ sung như lọc cộng tác (collaborative\_filtering) hoặc khai phá luật liên kết (apply\_apriori) để tăng độ chính xác, nhưng vẫn giữ vai trò chính trong duyệt đồ thị.
* **So sánh với các thuật toán khác:**
* Dijkstra: Không sử dụng heuristic, dẫn đến độ hiệu quả kém hơn.
* Phương pháp học máy: Yêu cầu dữ liệu huấn luyện lớn và tính toán phức tạp, không phù hợp cho phản hồi thời gian thực.

A\* mang lại sự cân bằng giữa hiệu quả tính toán và khả năng cá nhân hóa [33], như đã được chứng minh trong các hệ thống đề xuất lộ trình học tập.

#### Thiết kế chi tiết của thuật toán

Từ các bước chính gồm: *xác định điểm bắt đầu, xác định điểm đích, sinh lộ trình học tập, cập nhật và điều chỉnh.* Ta xây dựng các hàm hỗ trợ thuật toán như sau:

* determine\_start\_node(): Xác định điểm bắt đầu dựa trên hồ sơ học viên.
* determine\_goal\_node(): Xác định điểm đích từ mục tiêu học tập.
* a\_star\_custom(): Triển khai A\* với heuristic tùy chỉnh.
* generate\_learning\_path(): Sinh lộ trình học tập tối ưu.
* update\_learning\_path(): Điều chỉnh lộ trình theo hiệu suất.
* collaborative\_filtering(): Đề xuất dựa trên hành vi học viên tương đồng.
* apply\_apriori(): Khai phá các mẫu học tập phổ biến.
* **Công thức toán học**

Thuật toán A\* được định nghĩa bởi:

Trong đó:

* : Chi phí thực tế từ start\_node đến nút , tính bằng tổng trọng số của các cạnh trên đường đi.
* : Hàm heuristic, ước lượng chi phí từ nút đến goal\_node, được tính như sau:

* **Các thành phần trong :**
* : Độ tương đồng ngữ nghĩa với goal\_node, tính bằng Jaccard similarity giữa các tag (prototype: cố định ).
* : Độ ưu tiên của nút, chuẩn hóa bằng cách chia cho (prototype: cố định
* : Trọng số độ khó, lấy từ dynamic\_weights (STANDARD hoặc ADVANCED), chuẩn hóa bằng cách chia cho (prototype: cố định
* : Trọng số kỹ năng, ánh xạ từ mức độ kỹ năng (Bloom’s Taxonomy) trong dynamic\_weights, chuẩn hóa bằng cách chia cho (prototype: cố định .
* time\_estimate(n): Thời gian ước lượng (phút)  
  : trọng số cấu hình (prototype: cố định   
  weight\_time\_estimate: trọng số động dựa trên lịch sử học tập của học viên.
* : 0.5 nếu ngữ cảnh khớp, 0 nếu không.
* **Cơ chế thuật toán**
* **Khởi tạo**:
  + Thêm start\_node vào hàng đợi ưu tiên với f(start\_node) = h(start\_node)
  + Đặt
* **Vòng lặp chính**:
  + Lấy nút có nhỏ nhất từ hàng đợi.
  + Nếu nút là goal\_node, xây dựng lại đường đi từ came\_from và trả về.
  + Với mỗi nút lân cận:
    - Tính
    - Nếu ​ tốt hơn, cập nhật , và thêm vào hàng đợi.
* **Hàm Heuristic**:
  + Truy vấn Neo4j để lấy thuộc tính nút, xử lý giá trị thiếu bằng coalesce.
  + Tính dựa trên công thức trên, sử dụng bộ nhớ đệm để tối ưu.
* **Trọng số động**:
  + Tính từ dữ liệu học tập:
    - Nếu điểm trung bình > 80 và thời gian < 30 phút, tăng trọng số w\_priority lên 0.5.
    - Nếu điểm trung bình < 50, tăng trọng số w\_difficulty\_advanced lên 1.0.
* **Xử lý trường hợp không tìm thấy đường dẫn**:
  + Nếu thuật toán A\* không tìm được đường đi từ start\_node đến goal\_node, hệ thống chuyển sang sử dụng mô hình ngôn ngữ lớn (LLM) để tạo một truy vấn Cypher thay thế hoặc sinh lộ trình dự phòng gồm 3 bước (từ cơ bản đến nâng cao).
* **So sánh thuật toán:**
* **Thuật toán A\* tiêu chuẩn***:* Sử dụng hàm heuristic đơn giản, chẳng hạn như khoảng cách Euclidean trong không gian địa lý hoặc chi phí dựa trên một yếu tố duy nhất.
* **Thuật toán A\* tùy chỉnh**: Tích hợp nhiều yếu tố học tập như độ tương đồng, độ ưu tiên, độ khó, mức độ kỹ năng, thời gian ước lượng và ngữ cảnh. Hàm heuristic được điều chỉnh động dựa trên dữ liệu học tập, giúp phù hợp hơn với các ứng dụng giáo dục cá nhân hóa.

#### Tích hợp với Đồ thị Tri thức và Mô hình Ngôn ngữ Lớn

Thuật toán A\* được tích hợp chặt chẽ với KG trong Neo4j và LLM thông qua LlamaIndex Property Graph Index, đảm bảo khả năng truy vấn hiệu quả và xử lý các yêu cầu phức tạp:

* **Truy vấn Cypher**: Thuật toán sử dụng các truy vấn Cypher để duyệt qua các mối quan hệ trong KG, ví dụ cơ bản:

MATCH (start:KnowledgeNode {Node\_ID: $start\_node})

MATCH (goal:KnowledgeNode {Node\_ID: $goal\_node})

MATCH p = shortestPath((start)-[:NEXT|REQUIRES\*]-(goal))

RETURN p

Truy vấn này tìm đường đi ngắn nhất giữa start\_node và goal\_node, ưu tiên các mối quan hệ NEXT và REQUIRES.

* **Tích hợp LLM**: Khi A\* không tìm được đường đi hoặc cần bổ sung nội dung, LlamaIndex chuyển yêu cầu sang LLM (ví dụ: GPT-3.5) để:
  + Sinh lộ trình thay thế dựa trên ngữ cảnh và mục tiêu học viên.
  + Tạo nội dung học tập hoặc câu hỏi trắc nghiệm dựa trên các thuộc tính như Definition và Example.
* **Tìm kiếm ngữ nghĩa**: VectorStoreIndex trong LlamaIndex hỗ trợ tìm kiếm các khái niệm liên quan dựa trên Semantic\_Tags và Focused\_Semantic\_Tags, giúp xác định các khái niệm phù hợp với mục tiêu học tập.
* **Thời gian học khả dụng**: Điều chỉnh lộ trình để tổng Time\_Estimate phù hợp với time\_availability của học viên.
* **Độ khó ưa thích**: Lựa chọn các khái niệm có Difficulty tương ứng với preferred\_difficulty của học viên (ví dụ: “STANDARD” cho người mới bắt đầu).

#### Chi tiết cấu trúc các hàm sử dụng trong thuật toán

* **Tóm tắt cấu trúc hệ thống**

Hàm main là hàm điều phối chính, khởi động hệ thống, nhập thông tin học viên, và gọi hàm run\_learning\_session để thực hiện phiên học tập.

* **Cấu trúc phân cấp của các hàm**

**main**

├── **check\_and\_load\_students**

│ └── execute\_cypher\_query

├── initialize\_vector\_index

│ └── execute\_cypher\_query

├── **load\_student\_profile**

│ ├── execute\_cypher\_query

│ ├── calculate\_learning\_speed

│ └── extract\_topic\_preference

│ └── execute\_cypher\_query

├── **run\_learning\_session**

│ ├── **determine\_start\_node**

│ │ ├── load\_student\_profile

│ │ ├── execute\_cypher\_query

│ │ ├── generate\_goal\_tags

│ │ └── jaccard\_similarity

│ ├── **determine\_goal\_node**

│ │ ├── load\_student\_profile

│ │ ├── execute\_cypher\_query

│ │ ├── generate\_goal\_tags

│ │ ├── jaccard\_similarity

│ │ └── a\_star\_search

│ │ ├── execute\_cypher\_query

│ │ └── get\_heuristic

│ │ └── execute\_cypher\_query

│ ├── suggest\_prerequisites

│ │ └── execute\_cypher\_query

│ ├── **generate\_learning\_path**

│ │ ├── determine\_start\_node

│ │ ├── execute\_cypher\_query

│ │ ├── determine\_goal\_node

│ │ ├── get\_goal\_tags

│ │ │ └── execute\_cypher\_query

│ │ ├── calculate\_and\_store\_similarity

│ │ │ ├── execute\_cypher\_query

│ │ │ └── jaccard\_similarity

│ │ ├── a\_star\_custom

│ │ │ ├── heuristic

│ │ │ │ └── execute\_cypher\_query

│ │ │ └── execute\_cypher\_query

│ │ ├── extract\_cypher\_query

│ │ └── json.loads

│ ├── **generate\_learning\_content**

│ │ ├── load\_student\_profile

│ │ ├── execute\_cypher\_query

│ │ └── PromptTemplate

│ ├── **generate\_quiz**

│ │ ├── clean\_json\_response

│ │ └── json.loads

│ ├── **evaluate\_quiz**

│ │ └── csv.DictReader

│ ├── get\_time\_spent

│ **├── update\_student\_profile**

│ │ ├── load\_student\_profile

│ │ └── execute\_cypher\_query

│ ├── **update\_learning\_path**

│ │ ├── load\_student\_profile

│ │ ├── \_find\_alternative\_node

│ │ │ └── execute\_cypher\_query

│ │ ├── \_find\_remedy\_node

│ │ │ └── execute\_cypher\_query

│ │ └── \_construct\_node

│ ├── adjust\_heuristic\_weights

│ └── save\_learning\_data

**├── collaborative\_filtering**

│ ├── load\_student\_profile

│ └── execute\_cypher\_query

└── apply\_apriori

* **Danh sách các hàm và vai trò trong hệ thống**
  + 1. **Lớp (Config Class)**
* **Chức năng chính**: Lưu trữ các hằng số và cấu hình toàn cục, bao gồm đường dẫn file, thuộc tính nút, loại mối quan hệ, và tham số thuật toán.
* **Vai trò trong hệ thống**: Đảm bảo tính nhất quán và dễ dàng quản lý cấu hình, hỗ trợ mọi chức năng từ tải dữ liệu đến sinh lộ trình học tập.
* **Ví dụ minh họa**: Cung cấp cấu hình như NODES\_FILE="/content/drive/MyDrive/LP\_data\_based/nodes.csv" hoặc QUIZ\_PASSING\_SCORE=70 để sử dụng trong toàn hệ thống.
  + 1. **Hàm**
* **main()**

Chức năng chính: Khởi tạo hệ thống, thu thập thông tin học sinh, và chạy phiên học tập đầy đủ.

Vai trò trong hệ thống: Điểm vào chính, điều phối toàn bộ quy trình từ khởi tạo đến đề xuất bổ sung.

* **check\_and\_load\_students(driver, student\_file: str = Config.STUDENT\_FILE)**

Chức năng chính: Tải và cập nhật thông tin học sinh từ tệp CSV vào Neo4j.

Vai trò trong hệ thống: Đảm bảo dữ liệu học sinh luôn sẵn sàng và cập nhật để cá nhân hóa lộ trình.

* **initialize\_vector\_index()**

Chức năng chính: Khởi tạo hoặc tải VectorStoreIndex để tìm kiếm ngữ nghĩa.

Vai trò trong hệ thống: Hỗ trợ xác định các khái niệm liên quan dựa trên embedding, dùng trong determine\_start\_node và determine\_goal\_node.

* **load\_student\_profile(student\_id: str)**

Chức năng chính: Tải hồ sơ học sinh từ Neo4j, bao gồm lịch sử học tập và phong cách học.

Vai trò trong hệ thống: Cung cấp thông tin cá nhân hóa cho lộ trình, nội dung, và cập nhật hồ sơ.

* **run\_learning\_session**(student\_id: str, level: str, context: str, ...)

Chức năng chính: Điều phối phiên học tập từ xác định lộ trình, sinh nội dung, đến đánh giá và cập nhật.

Vai trò trong hệ thống: Hàm trung tâm, tích hợp các chức năng để thực hiện quy trình học tập cá nhân hóa.

* **determine\_start\_node(...)**

Chức năng chính: Xác định điểm bắt đầu của lộ trình học tập dựa trên trình độ và ngữ cảnh.

Vai trò trong hệ thống: Đặt nền tảng cho lộ trình, đảm bảo phù hợp với năng lực học sinh.

* **determine\_goal\_node(...)**

Chức năng chính: Xác định điểm đích của lộ trình dựa trên mục tiêu học sinh.

Vai trò trong hệ thống: Hướng dẫn thuật toán A\* tới mục tiêu phù hợp.

* **suggest\_prerequisites(goal\_node: str, student\_id: str, context: str = None)**

Chức năng chính: Đề xuất các khái niệm tiên quyết chưa hoàn thành.

Vai trò trong hệ thống: Bổ sung các bước cần thiết vào lộ trình, đảm bảo học sinh có nền tảng.

* **generate\_learning\_path(...)**

Chức năng chính: Sinh lộ trình học tập tối ưu từ điểm bắt đầu đến điểm đích bằng thuật toán A\*.

Vai trò trong hệ thống: Tạo ra lộ trình học tập cá nhân hóa dựa trên dữ liệu học sinh và đồ thị tri thức.

* **generate\_learning\_content(node, student\_id, ...)**

Chức năng chính: Sinh nội dung học tập cá nhân hóa cho từng khái niệm.

Vai trò trong hệ thống: Cung cấp tài liệu học tập phù hợp với phong cách và trình độ học sinh.

* **generate\_quiz(node)**

Chức năng chính: Tạo bài kiểm tra để đánh giá sự hiểu biết của học sinh.

Vai trò trong hệ thống: Đánh giá hiệu quả học tập, cung cấp dữ liệu để cập nhật lộ trình.

* **evaluate\_quiz(student\_file, answer\_file)**

Chức năng chính: Đánh giá bài kiểm tra và tính điểm.

Vai trò trong hệ thống: Xác định hiệu suất học sinh, ảnh hưởng đến cập nhật hồ sơ và lộ trình.

* **get\_time\_spent(start\_time=None, autosave=False)**

Chức năng chính: Ghi nhận thời gian học, hỗ trợ tự động hoặc thủ công.

Vai trò trong hệ thống: Cung cấp dữ liệu thời gian để phân tích hiệu suất và cập nhật hồ sơ.

* **update\_student\_profile(student\_id, node, score, time\_spent)**

Chức năng chính: Cập nhật hồ sơ học sinh với thông tin từ phiên học tập.

Vai trò trong hệ thống: Duy trì thông tin học sinh chính xác, hỗ trợ cá nhân hóa liên tục.

* **update\_learning\_path(student\_id, node, learning\_path, score)**

Chức năng chính: Điều chỉnh lộ trình học tập dựa trên hiệu suất.

Vai trò trong hệ thống: Đảm bảo lộ trình linh hoạt, bổ sung khái niệm khắc phục nếu cần.

* **collaborative\_filtering(student\_id: str, ...)**

Chức năng chính: Đề xuất khái niệm dựa trên tương đồng giữa các học sinh.

Vai trò trong hệ thống: Bổ sung đề xuất cá nhân hóa sau phiên học, dựa trên cộng đồng.

* **apply\_apriori(learning\_paths\_file: str = Config.LEARNING\_PATHS\_FILE, ...)**

Chức năng chính: Khám phá quy tắc liên kết trong lộ trình học tập bằng thuật toán Apriori.

Vai trò trong hệ thống: Phát hiện các mẫu học tập chung, hỗ trợ đề xuất bổ sung.

* **execute\_cypher\_query(driver, query: str, params: Dict = None)**

Chức năng chính: Thực thi truy vấn Cypher trên Neo4j.

Vai trò trong hệ thống: Cầu nối với cơ sở dữ liệu đồ thị, hỗ trợ tất cả chức năng truy xuất dữ liệu.

* **jaccard\_similarity(tags1: List[str], tags2: List[str])**

Chức năng chính: Tính độ tương đồng Jaccard giữa hai danh sách tag.

Vai trò trong hệ thống: Hỗ trợ hợp nhất nút và cải thiện độ chính xác của đề xuất.

* **calculate\_learning\_speed(performance\_details: List[str])**

Chức năng chính: Tính tốc độ học tập trung bình dựa trên thời gian học.

Vai trò trong hệ thống: Hỗ trợ đánh giá khả năng tiếp thu của học sinh.

* **extract\_topic\_preference(learning\_history: List[str])**

Chức năng chính: Xác định sở thích chủ đề từ lịch sử học tập.

Vai trò trong hệ thống: Ưu tiên đề xuất các khái niệm phù hợp với sở thích.

* **clean\_json\_response(response\_str: str)**

Chức năng chính: Làm sạch chuỗi JSON từ LLM để đảm bảo định dạng hợp lệ.

Vai trò trong hệ thống: Hỗ trợ xử lý dữ liệu từ LLM trong sinh nội dung và bài kiểm tra.

* **Quy trình chính**:

Hệ thống hoạt động theo quy trình sau:

1. **Khởi tạo**: Hàm main gọi check\_and\_load\_students và initialize\_vector\_index để tải dữ liệu học sinh và khởi tạo chỉ mục vector.
2. **Phiên học tập**: run\_learning\_session điều phối toàn bộ quá trình:
   * Xác định điểm bắt đầu (determine\_start\_node) và điểm đích (determine\_goal\_node).
   * Đề xuất tiên quyết (suggest\_prerequisites) và sinh lộ trình (generate\_learning\_path).
   * Sinh nội dung (generate\_learning\_content), tạo bài kiểm tra (generate\_quiz), và đánh giá (evaluate\_quiz).
   * Cập nhật hồ sơ (update\_student\_profile) và lộ trình (update\_learning\_path).
3. **Đề xuất bổ sung**: collaborative\_filtering và apply\_apriori cung cấp các khái niệm và quy tắc liên kết sau phiên học.

Các hàm hỗ trợ như execute\_cypher\_query và load\_student\_profile đảm bảo dữ liệu được truy xuất và xử lý chính xác, tạo nên một hệ thống học tập cá nhân hóa hiệu quả.

#### Các thách thức tiềm năng và giải pháp

Một số thách thức tiềm năng và giải pháp bao gồm:

* **Quy mô lớn của KG**: Khi KG chứa hàng nghìn nút và mối quan hệ, hiệu suất truy vấn có thể giảm.  
  Giải pháp: Tối ưu hóa truy vấn Cypher, sử dụng cơ chế caching, và triển khai các chỉ mục (index) trong Neo4j để tăng tốc độ truy vấn.
* **Dữ liệu động**: KG cần được cập nhật liên tục khi có tài liệu mới hoặc tiến độ học viên thay đổi.  
  Giải pháp: Sử dụng quy trình tự động hóa nhập dữ liệu (như SPR Generator và Validation) và kiểm tra tính nhất quán định kỳ.
* **Không tìm được đường đi**: Trong trường hợp không có đường đi từ start\_node đến goal\_node.  
  Giải pháp: Tích hợp LLM để sinh lộ trình thay thế hoặc gợi ý các khái niệm bổ sung, sử dụng LlamaIndex để tạo truy vấn Cypher động.

Thuật toán đề xuất lộ trình học tập cá nhân hóa dựa trên A\* không chỉ đáp ứng các yêu cầu chức năng và phi chức năng của hệ thống mà còn có khả năng mở rộng để hỗ trợ các ứng dụng giáo dục quy mô lớn, xử lý thêm cả các tùy biến theo nghiệp vụ giáo dục hoặc các yêu cầu trải nghiệm tăng cường cho học viên.

# HIỆN THỰC THỬ NGHIỆM

## XÂY DỰNG NGUYÊN MẪU

### Thu thập dữ liệu mẫu từ nguồn mở

Việc thu thập dữ liệu mẫu từ nguồn mở là bước đầu tiên trong quá trình xây dựng nguyên mẫu hệ thống đề xuất lộ trình học tập cá nhân hóa. Mục tiêu là tạo ra một bộ dữ liệu giáo dục phong phú, liên quan đến ba môn học chính: SQL Server, Business Intelligence (BI), và Hệ thống Thông tin Quản lý Hiện đại, để làm nền tảng cho Đồ thị Tri thức (Knowledge Graph - KG) trong Neo4j. Dữ liệu này sẽ được sử dụng để kiểm tra và đánh giá hiệu quả của hệ thống.

Sử dụng nguồn tài liệu học thuật bài giảng chính quy của ngành hệ thống thông tin quản lý – MIS trường đại học Bách Khoa HCM.

Bộ dữ liệu mẫu gồm các nội dung:

* SQL Server: 6 tệp chính gồm các nội dung:
  + Data definition
  + Built-in function
  + CRUD
  + Aggregation
  + JOIN
  + Trigger
* BI: 4 file pdf chính gồm các nội dung:
  + Data mining overview
  + Data pre-processing
  + Classification – Clustering - Association techniques
  + Business Intelligence Systems
* Hệ thống thông tin quản lý hiện đại: 4 file pdf gồm các nội dung:
  + Ethical and Social Issues in Information Systems
  + IT Infrastructure and Emerging Technologies
  + Foundations of Business Intelligence: Databases and Information Management
  + Managing Knowledge and Artificial Intelligence

Sau khi xử lý, bộ dữ liệu đạt được các đặc trưng sau:

* **Số lượng nút**: Khoảng 150 nút, mỗi nút đại diện cho một khái niệm, bài học, hoặc chủ đề trong ba môn học. Ví dụ: “Giới thiệu về SQL”, “Cú pháp SELECT”, “BI Dashboard Design”.
* **Số lượng mối quan hệ**: Khoảng 180 mối quan hệ, thể hiện các liên kết như trình tự học tập (NEXT), tiên quyết (REQUIRES), hoặc tương đồng (SIMILAR\_TO).
* **Thuộc tính nút**: Mỗi nút chứa các thuộc tính bắt buộc như Document\_ID, Node\_ID, Sanitized\_Concept, Definition, Example, Learning\_Objective, Skill\_Level, Time\_Estimate, Difficulty, Priority, Prerequisites, Semantic\_Tags, Focused\_Semantic\_Tags, và một số thuộc tính tùy chọn như Real\_World\_Application và Learning\_Tips.

### Triển khai dữ liệu

Để kiểm tra độ hoàn thiện của Giai đoạn 1, với bộ dữ liệu mẫu hiện có, thực hiện các bước:

1. Import và hợp nhất dữ liệu

* Thêm cặp file chứa dữ liệu mới cần tải vào thư mục import của Neo4j desktop.

A screenshot of a computer

AI-generated content may be incorrect.

* Chạy hàm check\_and\_load\_kg(driver)

1. Thống kê số lượng nút và quan hệ

A screenshot of a computer

AI-generated content may be incorrect.

1. Trực quan hóa đồ thị tri thức

A screenshot of a computer screen

AI-generated content may be incorrect.

**Hình 4.1 Trực quan toàn bộ KG**

1. Kiểm thử tính toàn vẹn và khả năng truy vấn

Sử dụng các mã Cypher query mẫu như sau:

// 1. Total Nodes

MATCH (n)

WHERE n.Node\_ID IS NOT NULL

RETURN count(n) AS total\_nodes;

// 2. Total Relationships

MATCH ()-[r]->()

WHERE r.Weight IS NOT NULL OR r.Dependency IS NOT NULL

RETURN count(r) AS total\_relationships;

// 3. Node with properties

MATCH (n)

RETURN n.Node\_ID, labels(n) AS node\_labels, n.Sanitized\_Concept, n.Semantic\_Tags, properties(n) AS node\_properties

LIMIT 10;

// 4. Relationship with properties

MATCH (source)-[r]->(target)

RETURN source.Node\_ID AS source\_id, type(r) AS relationship\_type, target.Node\_ID AS target\_id, properties(r) AS rel\_properties

LIMIT 10;

1. Ghi nhận log

Kết quả triển khai này là bước quan trọng để tiến hành thử nghiệm thực tế, như sẽ được trình bày trong các mục tiếp theo.

## TÍCH HỢP LLAMA-INDEX VÀ LLM

### Cấu hình LLM

#### Lựa chọn và cấu hình LLM

**Thiết lập tham số cho LLM**

Hệ thống sử dụng mô hình **OpenAI GPT-3.5-turbo**, được chọn nhờ khả năng xử lý hội thoại và sinh nội dung hiệu quả trong giáo dục. Các tham số chính bao gồm:

* **Model**: gpt-3.5-turbo, đảm bảo hiệu suất cao và chi phí hợp lý.
* **Temperature**: 0.1, giảm tính ngẫu nhiên để tạo đầu ra nhất quán, phù hợp với nội dung học tập và câu hỏi kiểm tra.
* **System Prompt**: "*Bạn là chuyên gia lộ trình học tập chuyên về giáo dục*.", định hướng mô hình tập trung vào lĩnh vực giáo dục, chọn mô hình fine-tune phù hợp.

**Tích hợp LLM với Llama-index và đồ thị tri thức**

* **Khởi tạo PropertyGraphIndex**: Hàm khởi tạo chỉ mục đồ thị kết nối LLM với dữ liệu tri thức có cấu trúc, cho phép truy xuất thông tin từ Neo4j.

A close-up of a graph

AI-generated content may be incorrect.

* **Sử dụng TextToCypherRetriever**: Trong các hàm determine\_start\_node và determine\_goal\_node, công cụ này chuyển đổi yêu cầu ngôn ngữ tự nhiên thành truy vấn Cypher, truy xuất các nút phù hợp dựa trên level, context, và student\_goal.



#### Thiết kế các bộ prompt hỗ trợ

**Bảng 4.1 Giới thiệu các bộ Prompt LLM trong luận văn**

|  |  |  |
| --- | --- | --- |
| Vị trí | Vai trò | Tham số/Đầu vào |
| Giai đoạn 1 | Prompt SPR Generator tạo nút và mối quan hệ | Tài liệu gốc |
| Giai đoạn 1 | Prompt Validation kiểm tra và chuẩn hóa nút và mối quan hệ | nodes.csv relationships.cs |
| Giai đoạn 2 | Prompt tạo nội dung học tập | Student\_goal node\_info learning\_style skill\_level Learning\_objective |
| Giai đoạn 2 | Prompt tạo câu hỏi trắc nghiệm kiểm tra đầu vào xác định trình độ | Student\_goal  Context  Candidates (Tập hợp các nút ứng viên) |
| Giai đoạn 2 | Prompt tạo câu hỏi trắc nghiệm kiểm tra sau khi học | Node\_info  QUIZ\_DISTRIBUTION (phân bố số lượng theo độ khó) QUIZ\_NUM\_QUESTION (số lượng câu hỏi) |
| Giai đoạn 2 | Prompt tìm nút đích nếu không có nút tiềm năng nào đạt threshold similarity | Node\_info  Student\_goal |
| Giai đoạn 2 | Prompt tạo lộ trình học tập – tạo cypher query và cung cấp cho Neo4j trả về kết quả | Start\_node  Goal\_node  Context |
| Giai đoạn 2 | Prompt tạo lộ trình học tập fallback - trong trường hợp phản hồi LLM không hợp lệ | Start\_node student\_goal  Context |

Prompt engineering, với các bộ prompt được liệt kê trong Bảng 4.1, được áp dụng để định hướng LLM cung cấp phản hồi tối ưu cho quy trình sử dụng.

### Tối ưu truy vấn đồ thị

* **Sử dụng chỉ mục (Indexing) trong Neo4j**

Các chỉ mục được thiết lập trên các thuộc tính được sử dụng thường xuyên như Node\_ID, Semantic\_Tags giúp giảm thời gian truy vấn khi tìm kiếm nút cụ thể, tránh quét toàn bộ đồ thị.

CREATE INDEX ON :KnowledgeNode(Node\_ID);

CREATE INDEX ON :KnowledgeNode(Semantic\_Tags);

* **Caching dữ liệu thường truy cập**

Cơ chế caching lưu trữ kết quả cũng được sử dụng thông qua VectorStore Index, lưu trữ các dữ liệu nút lịch sử mới nhất dưới dạng document. Thực hiện xóa các document ít sử dụng, giữ các document mới nhất để dùng làm tài nguyên cho vector index.

* **Sử dụng Llama-index để tối ưu hóa truy vấn ngôn ngữ tự nhiên**

Llama-index Property Graph Index hỗ trợ T*extToCypherRetriever* cho phép truy vấn đồ thị qua ngôn ngữ tự nhiên, mà không cần hiểu về Cypher.

# ĐÁNH GIÁ VÀ SO SÁNH

## THIẾT LẬP THỬ NGHIỆM

Do dự án có các đóng góp sản phẩm gồm các bộ prompting, quy trình nghiệp vụ đề xuất cá nhân hóa, và các mã nguồn tạo sinh lộ trình học tập và tài nguyên học tập.  
Nên trong quy mô prototype, đề cương luận văn này sẽ thực hiện việc thử nghiệm so sánh cách tạo lộ trình học tập (nghiệp vụ chính của business) giữa thuật toán đề xuất, và một thuật toán mã nguồn mở. Từ đó, xem xét khả năng tạo lộ trình học tập của thuật toán đề xuất.

### Bộ dữ liệu thử nghiệm

Bộ dữ liệu thử nghiệm được xây dựng từ các nguồn giáo dục mở, tập trung vào ba môn học: SQL Server, Business Intelligence (BI), và Hệ thống Thông tin Quản lý Hiện đại. Dữ liệu bao gồm:

* **Tài liệu giáo dục**: Các bài giảng, sách giáo khoa, của đại học bách khoa ngành MIS hoặc từ các khóa học trực tuyến từ Coursera, GitHub, Udemy, được trích xuất thành các khái niệm, định nghĩa, ví dụ, và mối quan hệ.
* **Hồ sơ học viên mô phỏng**: 50 hồ sơ học viên với các thuộc tính như current\_level (Standard, Advanced), learning\_style\_preference (visual, auditory, reading/writing, kinesthetic), và long\_term\_goal (ví dụ: “Master SQL”).
* **Yêu cầu học tập**: 10/50 yêu cầu học tập mô phỏng, ví dụ: “Tôi muốn học về SQL Server” hoặc “Tôi cần hiểu BI Dashboard Design”.

Bộ dữ liệu được chuẩn hóa thành các tệp validated\_nodes.csv và validated\_relationships.csv, nhập vào Neo4j để tạo Đồ thị Tri thức (KG) với:

* **Số nút**: >100, đại diện cho tri thức.
* **Số mối quan hệ**: >100, bao gồm các loại như NEXT, REQUIRES, SIMILAR\_TO,..

**Bảng 5.1. Tóm tắt dữ liệu môn học sử dụng cho prototype**

|  |  |  |
| --- | --- | --- |
|  | SỐ NÚT | SỐ MỐI QUAN HỆ |
| SQL Server | 39 | 45 |
| Business Intelligence | 55 | 72 |
| Hệ thống Thông tin Quản lý Hiện đại | 44 | 50 |
| TỔNG CỘNG | **138** | **167** |

Từ kết quả tổng hợp số nút và mối quan hệ nêu trong Bảng 5.1, tiếp tục đánh giá đặc tính của đồ thị xây dựng và kết quả được liệt kê trong Bảng 5.2.

**Bảng 5.2. Tóm tắt đặc tính nút tri thức trong KG**

|  |  |
| --- | --- |
| Thuộc tính | Giá trị |
| Số lượng nút | 138 |
| Số lượng mối quan hệ | 167 |
| Độ sâu trung bình | 4 lớp |
| Độ sâu tối thiểu | 2 lớp |
| Độ sâu tối đa | 5 lớp |
| Số lượng đường dẫn | 50 |
| Thời gian nhập dữ liệu | ~10 phút (tạo 150 nút tri thức) |
| Thời gian truy vấn mẫu | < 5 giây |

* Sử dụng cypher hỗ trợ sau:

MATCH path = (start:KnowledgeNode)-[:NEXT|REQUIRES|IS\_SUBCONCEPT\_OF|IS\_PREREQUISITE\_OF\*2.]->(end:KnowledgeNode)

WHERE NOT ()-[:NEXT|IS\_PREREQUISITE\_OF|IS\_SUBCONCEPT\_OF]->(start)

AND NOT (start)-[:REQUIRES]->()

AND NOT (end)-[:NEXT|IS\_SUBCONCEPT\_OF|IS\_PREREQUISITE\_OF]->()

AND NOT ()-[:REQUIRES]->(end)

RETURN avg(length(path)) AS avg\_depth, min(length(path)) AS min\_depth, max(length(path)) AS max\_depth, count(path) AS path\_count

* **Nút gốc**: Là các khái niệm cơ bản nhất, nút bắt đầu của 1 chuỗi tri thức, nghĩa là không có khái niệm nào trước nút gốc, không cần các khái niệm tiên quyết hoặc khái niệm con trong cấu trúc phân cấp (không có NEXT, IS\_SUBCONCEPT\_OF, IS\_PREREQUISITE\_OF) đi đến nút gốc.
* **Nút lá:** Là các khái niệm cuối cùng trong 1 chuỗi tri thức, không dẫn đến khái niệm nào, hoặc không phải là tiên quyết hoặc là khái niệm con của 1 nút khác (từ nút lá không có NEXT, IS\_PREREQUISITE\_OF, IS\_SUBCONCEPT\_OF đi ra)

Ví dụ: tìm đường dẫn của nút gốc có Node\_ID: ‘concept:data\_task’ đến nút lá (có path dài nhất) Node\_ID: ‘concept:business\_intelligence\_systems’

MATCH path = (start:KnowledgeNode {Node\_ID: 'concept:data\_task'})-[:NEXT|REQUIRES|IS\_SUBCONCEPT\_OF|IS\_PREREQUISITE\_OF\*2.]->(end:KnowledgeNode)

WHERE NOT ()-[:NEXT|IS\_PREREQUISITE\_OF|IS\_SUBCONCEPT\_OF]->(start)

AND NOT (start)-[:REQUIRES]->()

AND NOT (end)-[:NEXT|IS\_SUBCONCEPT\_OF|IS\_PREREQUISITE\_OF]->()

AND NOT ()-[:REQUIRES]->(end)

RETURN nodes(path) AS nodes

Kết quả lộ trình thử nghiệm có thể quan sát theo Hình 13

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Hình 5.1 Lộ trình thử nghiệm từ nút gốc đến nút lá**

### Tiêu chí đánh giá

Với quy mô đề cương luận văn, hiệu quả của thuật toán là yếu tố được đánh giá chính. Xem xét khả năng tạo lộ trình học tập từ nguồn dữ liệu sẵn sàng, với độ chính xác so sánh với kết quả yêu cầu từ chuyên gia, hoặc kết quả từ thuật toán mã nguồn mở đã công bố khoa học (MOOCIR).

Với quy mô luận văn, hiệu suất của thuật toán được đánh giá dựa trên hai tiêu chí chính:

* **Độ chính xác nghiệp vụ**: Đo bằng các chỉ số precision, recall, và F1-score. Một lộ trình học tập được coi là chính xác nếu:
  + Bao gồm các khái niệm phù hợp với yêu cầu học viên.
  + Tuân thủ các tiên quyết (Prerequisites) và trình tự học tập (NEXT).
  + Mục tiêu đạt trên 80% F1-score, như đề xuất trong Mục 5.1.2.
* **Thời gian phản hồi**: Thời gian trung bình để hệ thống trả về lộ trình học tập sau khi nhận yêu cầu, phải dưới 5 giây (NFR1, Mục 3.4.2).

### Thuật toán nguồn mở để so sánh

Thuật toán mã nguồn mở được chọn để so sánh trong nghiên cứu này là **Meta-path-based Representation Learning for Recommending Knowledge Concepts** (MOOCIR), được trình bày trong bài báo “Recommending Knowledge Concepts on MOOC Platforms with Meta-path-based Representation Learning” của Guangyuan Piao, công bố tại hội nghị EDM 2021 – một sự kiện uy tín về khai thác dữ liệu giáo dục. MOOCIR là một hệ thống khuyến nghị lộ trình học tập cá nhân hóa, tận dụng *Mạng Thông tin Dị biệt (Heterogeneous Information Network - HIN)* kết hợp với *Graph Convolutional Networks (GCNs)* [28]để học biểu diễn của người dùng và khái niệm học tập. (Mã nguồn của thuật toán MOOCIR xem trong phụ lục A)

Thuật toán này được chọn để so sánh với thuật toán trong luận văn hiện tại nhờ khả năng xử lý dữ liệu phức tạp trên các nền tảng MOOC và hiệu suất vượt trội đã được chứng minh.

* + - 1. *Giải thích thuật toán MOOCIR*

MOOCIR được thiết kế để khuyến nghị các khái niệm kiến thức mà người dùng có thể quan tâm trên nền tảng MOOC, dựa trên lịch sử học tập và các mối quan hệ ngữ nghĩa giữa các thực thể. Dưới đây là các thành phần chính của thuật toán:

* **Mạng Thông tin Dị biệt (HIN)**:
  + MOOCIR mô hình hóa thông tin trên nền tảng MOOC (bao gồm người dùng, khái niệm, khóa học, video, trường học, giáo viên) dưới dạng một **HIN**. HIN là một đồ thị có nhiều loại nút (node) và cạnh (edge), cho phép biểu diễn các mối quan hệ phức tạp và đa dạng giữa các thực thể.
  + Ví dụ: Một người dùng có thể liên kết với một khóa học qua mối quan hệ "đăng ký" (enrolled), và khóa học đó liên kết với một khái niệm qua mối quan hệ "bao gồm" (covers).
* **Meta-path**:
  + **Meta-path** là các đường đi có ý nghĩa ngữ nghĩa trong HIN, định nghĩa các mối quan hệ gián tiếp giữa các thực thể. Ví dụ:
    - "Người dùng → Khóa học → Khái niệm" (User → Course → Concept): Người dùng quan tâm đến khái niệm thông qua khóa học họ đã đăng ký.
    - "Người dùng → Video → Khái niệm" (User → Video → Concept): Người dùng quan tâm đến khái niệm qua video họ đã xem.
  + MOOCIR sử dụng các meta-path này để tạo ra các đồ thị đồng nhất (homogeneous graphs) cho người dùng và khái niệm, từ đó áp dụng GCNs để học biểu diễn.
* **Graph Convolutional Networks (GCNs)**:
  + MOOCIR áp dụng **GCNs** trên các đồ thị đồng nhất được xây dựng từ meta-path để học biểu diễn của người dùng và khái niệm.
  + Mỗi meta-path tạo ra một biểu diễn khác nhau, phản ánh các khía cạnh khác nhau của mối quan hệ giữa người dùng và khái niệm (ví dụ: thông qua khóa học, video, hoặc giáo viên).
* **Cơ chế chú ý (Attention Mechanism)**:
  + Để tổng hợp các biểu diễn từ nhiều meta-path, MOOCIR sử dụng *cơ chế chú ý*. Cơ chế này tự động xác định tầm quan trọng của từng meta-path đối với từng người dùng hoặc khái niệm, thay vì sử dụng phương pháp trung bình đơn giản.
  + Công thức tính trọng số chú ý (attention weight) cho meta-path MPi đối với người dùng u:

​ Trong đó:

* + - : Biểu diễn của người dùng u dựa trên meta-path MPi.
    - Wu, Vu: Ma trận tham số có thể huấn luyện.
    - σ: Hàm kích hoạt phi tuyến (non-linear activation).
  + Biểu diễn cuối cùng của người dùng được tổng hợp:
* **Dự đoán (Prediction)**:
  + Biểu diễn cuối cùng của người dùng e\_u và khái niệm e\_c *được tích hợp vào một khung* ***factorization ma trận mở rộng*** *để dự đoán điểm ưu tiên y\_u,c* của khái niệm c đối với người dùng u:

Trong đó:

* + - xu, zc: Yếu tố tiềm ẩn (latent factors) từ matrix factorization.
    - M: Ma trận học được để chiếu e\_u và e\_c vào cùng không gian.
    - γ: Tham số điều chỉnh tỷ lệ giữa matrix factorization và biểu diễn GCN.
    - b\_c: Độ lệch (bias) của khái niệm.
* **Hàm mất mát (Loss Function)**:
  + MOOCIR sử dụng **Bayesian Personalized Ranking (BPR)** để huấn luyện mô hình với dữ liệu phản hồi ngầm (implicit feedback):

Trong đó:

* + - (u,i,j): Bộ ba gồm người dùng u, khái niệm đã tương tác i, và khái niệm chưa tương tác j.
    - σ: Hàm sigmoid.
    - λ: Tham số điều chuẩn L2.
    - Θ: Tập hợp các tham số cần học.

#### Mô phỏng thuật toán MOOCIR trên Google Colab

Mã Python mô phỏng thuật toán MOOCIR được thiết kế để chạy trên Google Colab, giữ nguyên các logic cốt lõi của thuật toán, bao gồm việc áp dụng GCN để học biểu diễn, sử dụng cơ chế chú ý để tổng hợp biểu diễn, và sinh lộ trình học tập cá nhân hóa từ một khái niệm bắt đầu cụ thể.

**Giải thích mã mô phỏng**

1. **Cài đặt và import thư viện**
   * Sử dụng **TensorFlow 2.x** (khác với phiên bản 1.13.1 trong MOOCIR gốc) cùng các thư viện hỗ trợ như neo4j, pandas, numpy, và time.
   * Các thư viện này đảm bảo khả năng xử lý dữ liệu, kết nối cơ sở dữ liệu đồ thị, thực hiện tính toán ma trận, và đo hiệu suất.
2. **Kết nối Neo4j**
   * Neo4j được sử dụng để lưu trữ đồ thị tri thức, bao gồm các nút KnowledgeNode và các mối quan hệ như NEXT (khái niệm tiếp theo) để xây dựng ma trận kề, và REQUIRES (tiên quyết) tìm khái niệm cần trước.
   * Kết nối được thiết lập qua driver Neo4j với thông tin đăng nhập (url, username, password) lấy từ userdata trong Google Colab.
3. **Xử lý dữ liệu**
   * Hàm get\_concept\_nodes: Truy xuất danh sách các nút KnowledgeNode từ Neo4j, giới hạn ở 50 nút để tối ưu hiệu suất.
   * Hàm check\_prerequisites: Kiểm tra các mối quan hệ REQUIRES của một khái niệm, đảm bảo người học phải hoàn thành các khái niệm tiên quyết trước khi tiến tới khái niệm tiếp theo.
4. **Xây dựng ma trận kề**
   * Hàm build\_adjacency\_matrix: Tạo ma trận kề dựa trên mối quan hệ NEXT giữa các khái niệm trong đồ thị Neo4j.
   * Ma trận này biểu diễn kết nối trực tiếp giữa các khái niệm, đóng vai trò quan trọng trong quá trình lan truyền thông tin của GCN.
5. **Mô phỏng GCN**
   * Hàm gcn\_simulation: Mô phỏng Graph Convolutional Network (GCN) bằng cách nhân ma trận kề với ma trận đặc trưng (features), sau đó áp dụng hàm kích hoạt ReLU.
   * Đặc trưng ban đầu được khởi tạo ngẫu nhiên bằng tf.random.uniform, đại diện cho các thuộc tính của khái niệm.
6. **Cơ chế chú ý**
   * Hàm attention\_mechanism: Mô phỏng cơ chế chú ý bằng cách gán trọng số ngẫu nhiên (sử dụng hàm softmax) và tổng hợp các biểu diễn từ nhiều nguồn (trong mã này là hai meta-path giả lập).
   * Kết quả là biểu diễn tổng hợp cho người dùng và các khái niệm, phản ánh mức độ quan trọng của từng khái niệm đối với người học.
7. **Sinh lộ trình học tập**
   * Hàm moocir\_learning\_path: Kết hợp các bước trên để tạo lộ trình học tập cá nhân hóa:
     + **Bước 1**: Lấy danh sách khái niệm từ Neo4j và xác định chỉ số index của khái niệm bắt đầu.
     + **Bước 2**: Khởi tạo đặc trưng ngẫu nhiên và xây dựng ma trận kề.
     + **Bước 3**: Áp dụng GCN để tạo biểu diễn cho các khái niệm, sau đó sử dụng cơ chế chú ý để tổng hợp biểu diễn cho người dùng và khái niệm.
     + **Bước 4**: Tính điểm ưu tiên cho từng khái niệm bằng tích vô hướng giữa biểu diễn người dùng và biểu diễn khái niệm.
     + **Bước 5**: Sinh lộ trình bằng cách chọn các khái niệm tiếp theo dựa trên điểm ưu tiên, kiểm tra tiên quyết, và giới hạn độ dài tối đa (max\_length=5).
   * Quá trình này đảm bảo lộ trình hợp lý, tuân theo thứ tự tiên quyết và ưu tiên các khái niệm phù hợp với người học.
8. **Truy xuất thuộc tính nút**
   * Sau khi sinh lộ trình, mã truy vấn Neo4j để lấy thông tin chi tiết của từng khái niệm trong lộ trình, bao gồm:
     + sanitized\_concept: Tên khái niệm.
     + definition: Định nghĩa.
     + difficulty: Độ khó.
     + time\_estimate: Thời gian dự kiến.
   * Các giá trị mặc định được cung cấp nếu thông tin bị thiếu, đảm bảo lộ trình luôn đầy đủ và dễ hiểu.
9. **Đo thời gian phản hồi**
   * Mã sử dụng time.time() để đo thời gian thực thi của hàm moocir\_learning\_path, đảm bảo hiệu suất đáp ứng (thường dưới 5 giây).

## KẾT QUẢ VÀ PHÂN TÍCH

Phần này trình bày kết quả thử nghiệm của thuật toán đề xuất, so sánh với thuật toán nguồn mở MOOCIR, đồng thời phân tích các hạn chế của cả hai phương pháp.   
Các thử nghiệm được thực hiện trên bộ dữ liệu thử nghiệm (mục 5.1.1), với các tiêu chí đánh giá chính là độ chính xác (precision, recall, F1-score) và thời gian phản hồi.

### So sánh với thuật toán nguồn mở

**Phương pháp so sánh**

Cả hai thuật toán (thuật toán A\* kết hợp LLM của luận văn và MOOCIR) được thử nghiệm trên cùng bộ dữ liệu với 10 yêu cầu học tập mô phỏng. Các bước thực hiện:

1. **Chuẩn bị môi trường**: Cả hai hệ thống được chạy trên Google Colab với cấu hình CPU 2.2 GHz, RAM 12 GB, để đảm bảo công bằng.
2. **Chạy thử nghiệm**: Mỗi hệ thống xử lý 10 yêu cầu học tập, sinh lộ trình học tập và ghi lại kết quả.
   * Nút gốc (start\_node)thử nghiệm**: ‘‘**concept:data\_task”
   * Nút đích (goal\_node) thử nghiệm: **“**concept:business\_intelligence\_systems**’’**
   * Lộ trình dài nhất (đầy đủ tri thức) – 5 nút:  
     concept:data\_task → concept:business\_and\_data\_understanding\_task → concept:analysis\_task → concept:evaluation\_and\_reporting\_task → concept:business\_intelligence\_systems
   * Lộ trình ngắn nhất – 2 nút:   
     concept:data\_task → concept:business\_intelligence\_systems
3. **Đo lường hiệu suất**:
   * **Độ chính xác**: Tính precision, recall, và F1-score dựa trên sự phù hợp của lộ trình với yêu cầu và các điều kiện tiên quyết.
   * **Thời gian phản hồi**: Đo thời gian trung bình từ khi nhận yêu cầu đến khi trả về lộ trình.

* **Kết quả**
* **MOOCIR:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Theo lộ trình dài nhất:** concept:data\_task → concept:business\_and\_data\_understanding\_task → concept:analysis\_task → concept:evaluation\_and\_reporting\_task → concept:business\_intelligence\_systems

**Bảng 5.1: So sánh hiệu suất giữa thuật toán đề xuất và MOOCIR**

|  |  |  |
| --- | --- | --- |
|  | *Thuật toán đề xuất (A + LLM)*\* | MOOCIR |
| Lộ trình | 2 bước | 5 bước |
| Độ lặp lại (%) | 100 | 100 |
| Độ chính xác nghiệp vụ  (%) | 100 | 100 |
| Thời gian phản hồi (giây) | 40 | 259 |

* **Thuật toán đề xuất**

A screenshot of a computer code

AI-generated content may be incorrect.

Kết quả trả về theo lộ trình ngắn nhất:  
concept:data\_task → concept:business\_intelligence\_systems

* **Ưu điểm của thuật toán đề xuất**:
  + Trong thực tế, việc học viên xác định đúng “start\_node” là điều bất khả thi, do tách biệt giữa tri thức ngôn ngữ tự nhiên và tri thức mã hóa lưu trữ trong cơ sở dữ liệu tri thức của tổ chức. Do đó, khả năng xác định start\_node bằng ngôn ngữ tự nhiên của thuật toán đề xuất là một ưu thế.
  + Việc lựa chọn lộ trình ngắn nhất có căn cứ vững chắc (số liệu đề cập tiếp theo), dựa vào quá trình tính toán theo số liệu động, phù hợp cao với khả năng cá nhân hóa của học viên. Và hữu ích cho các trường hơp cold-start.
  + Đánh giá định tính:
    - Tính cá nhân hóa của hệ thống đề xuất vượt trội trong việc sinh nội dung học tập và quiz cá nhân hóa.
    - Tính mở rộng: Đồ thị tri thức trong Neo4j được thiết kế với các thuộc tính linh hoạt (ví dụ: Learning\_Style\_Preference, Semantic\_Tags), cho phép mở rộng các kịch bản dễ dàng khi thêm dữ liệu mới. MOOCIR phụ thuộc vào HIN cố định, khó điều chỉnh hơn.
* **Ưu điểm của MOOCIR**:
  + Dễ triển khai hơn do không yêu cầu cấu hình phức tạp như KG hoặc LLM, đơn giản hóa quy trình quản lý và cập nhật.
  + Thời gian phản hồi có thể nhanh hơn trong các trường hợp dữ liệu đơn giản, nhờ tận dụng HIN và GCNs.

### Phân tích hạn chế

* **Hạn chế của thuật toán đề xuất**
* Phụ thuộc vào chất lượng KG:   
  Nếu dữ liệu trong KG thiếu chính xác hoặc không đầy đủ (Semantic\_Tags không đủ độ tin cậy), hiệu suất đề xuất sẽ giảm đáng kể.
* Thời gian phản hồi:   
  Dù đạt 40 giây cho toàn bộ quá trình chuẩn bị tài liệu (so với thuật toán MOOCIR mô phỏng là 259 giây) thuật toán vẫn chậm hơn một số phương pháp truyền thống do phải xử lý truy vấn đồ thị và ngữ nghĩa qua LLM OpenAI API.
* Chi phí tính toán:   
  Phụ thuộc nhiều vào mô hình LLM – khi sử dụng mô hình tiên tiến hơn có thể có sự nhảy vọt về giá, trong trường hợp đó sẽ khó khăn khi triển khai hệ thống hàng loạt.
* **Hạn chế của MOOCIR**
* Thiếu xử lý ngữ nghĩa:   
  MOOCIR không hiểu được các yêu cầu phức tạp hoặc ngữ nghĩa, chỉ dựa vào dữ liệu tương tác và meta-path trong HIN.
* Cá nhân hóa hạn chế:   
  Không tận dụng cấu trúc tri thức như KG, dẫn đến các đề xuất kém linh hoạt và không phản ánh đầy đủ nhu cầu học viên.
* Phụ thuộc vào dữ liệu lịch sử: Hiệu quả giảm khi dữ liệu lịch sử không đủ, đặc biệt với người dùng mới (vấn đề cold start).
* Thời gia xử lý lâu: với khoảng 150 nút, thời gian xử lý cần 159 giây.
* **Giải pháp đề xuất**
* Cải thiện thuật toán đề xuất: Tối ưu hóa truy vấn Cypher và áp dụng caching để giảm thời gian phản hồi.
* Bổ sung dữ liệu: Tăng cường chất lượng KG bằng cách thu thập thêm tài liệu và kiểm tra định kỳ tính nhất quán.
* Kết hợp hybrid: Kết hợp kỹ thuật truyền thống như collaborative filtering từ MOOCIR để cải thiện tốc độ, đồng thời giữ khả năng xử lý ngữ nghĩa của LLM.

# Kết luận

## 6.1. Kết quả đạt được

Nghiên cứu đã hoàn thành các mục tiêu đề ra, đóng góp vào lĩnh vực giáo dục công nghệ thông qua việc phát triển hệ thống đề xuất lộ trình học tập cá nhân hóa dựa trên đồ thị tri thức và mô hình ngôn ngữ lớn (LLM).   
Các kết quả chính bao gồm:

* Xây dựng hệ thống hiệu quả: Hệ thống tích hợp Neo4j và LlamaIndex, sử dụng đồ thị tri thức với các nút KnowledgeNode và mối quan hệ như NEXT, REQUIRES, IS\_PREREQUISITE\_OF, đạt độ chính xác nghiệp vụ và tương đương với các phương pháp truyền thống như NoR-MOOCs (El-Ghareeb & El-Henawy, 2021), nhưng đạt được tính cá nhân hóa cao hơn cho học viên.
* Thuật toán tối ưu: Thuật toán A\* tùy chỉnh được triển khai để tạo lộ trình học tập cá nhân hóa, cân nhắc trình độ, ngữ cảnh, và mục tiêu học viên, đảm bảo tính thích ứng và hiệu quả.
* Tích hợp ngữ nghĩa: VectorStoreIndex hỗ trợ tìm kiếm ngữ nghĩa, nâng cao khả năng cá nhân hóa dựa trên dữ liệu lịch sử và nhu cầu học tập.
* Đánh giá thực nghiệm: Hệ thống được thử nghiệm trên bộ dữ liệu MOOCCube, chứng minh tính khả thi và tiềm năng ứng dụng trong giáo dục trực tuyến.

## 6.2. Hướng phát triển

Dựa trên kết quả đạt được, nghiên cứu đề xuất các hướng phát triển sau để nâng cao hiệu quả và mở rộng ứng dụng của hệ thống:

* Nghiên cứu dài hạn:   
  Đánh giá tác động thực tế của hệ thống lên kết quả học tập qua các nghiên cứu dài hạn, tập trung vào hiệu quả giáo dục và mức độ hài lòng của học viên thu nhận thông qua khảo sát thực tế.
* Giảm tài nguyên tính toán:   
  Tối ưu hóa LLM bằng các mô hình nhỏ hơn (distilled models) hoặc cơ chế caching, giảm chi phí triển khai.
* Tăng độ xác thực của dữ liệu:   
  Xây dựng cơ chế kiểm soát nút tri thức của toàn bộ đồ thị KG và khả năng kiểm duyệt định kỳ tập hợp các nút, tạo mối quan hệ tăng cường như cầu nối giữa các nút có liên quan trong tri thức ẩn, được thực hiện bởi chuyên gia với 1 quy trình hỗ trợ được phát triển thêm.
* Giao diện người dùng:   
  Phát triển front-end bằng Streamlit để tạo giao diện thân thiện, tích hợp LLM, nâng cao trải nghiệm học tập.
* Cải tiến semantic tags:   
  Đánh giá hiệu quả của thư viện 'wordnet' trong việc mở rộng semantic tags và khám phá các mô hình từ Hugging Face như BERT, RoBERTa, hoặc DistilBERT để tăng chất lượng và số lượng tag, giảm bias khi hợp nhất nút (Hugging Face). Xây dựng thư viện semantic tag chuẩn hóa, cho phép thêm từ mới không giới hạn nhưng vẫn điều chỉnh loại bỏ sự dư thừa để tránh sự “thiên vị” trong quá trình xác định điểm bắt đầu. Giảm sự gia tăng thẻ semantic tag trùng lặp ngữ nghĩa gây leo thang jaccard similarity.
* Làm rõ trình độ mong muốn:   
  Thiết lập quy trình đánh giá trình độ và mục tiêu học viên thông qua bài kiểm tra hoặc câu hỏi tương tác, cung cấp giải thích rõ ràng khi năng lực và kỳ vọng không phù hợp, nâng cao tính cá nhân hóa so với lộ trình học tập đơn thuần.

Nghiên cứu đã thử nghiệm và ứng dụng cơ bản cho việc ứng dụng đồ thị tri thức và LLM trong giáo dục cá nhân hóa, với hệ thống đề xuất lộ trình học tập đạt hiệu quả cao và tiềm năng triển khai thực tiễn. Các hạn chế về dữ liệu và tài nguyên sẽ được giải quyết thông qua các hướng phát triển đề xuất, mở đường cho các nghiên cứu tiếp theo về giáo dục công nghệ.

TÀI LIỆU THAM KHẢO

[1] Abu-Salih, B., & Alotaibi, S., "A systematic literature review of knowledge graph construction and application in education," Heliyon, vol. 10, no. 3, p. e25383, 2024.

[2] Bang, Y., Cahyawijaya, S., Lee, N., et al., "A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity," arXiv preprint arXiv:2302.04023, 2023. [Online]. Available: https://arxiv.org/abs/2302.04023

[3] Brank, M., Grobelnik, M., & Mladenić, D., "A survey of ontology evaluation techniques," in Proc. Conf. Data Mining and Data Warehouses (SiKDD 2005), 2005, pp. 166–170.

[4] Brown, S., "The C4 model for visualising software architecture," 2023. [Online]. Available: https://c4model.com/

[5] Brown, T. B., et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Curran Associates, Inc., 2020, pp. 1877–1901.

[6] Bubeck, S., Chandrasekaran, V., Eldan, R., et al., "Sparks of artificial general intelligence: Early experiments with GPT-4," arXiv preprint arXiv:2303.12712, 2023. [Online]. Available: https://arxiv.org/abs/2303.12712

[7] Chen, W., Li, X., & Zhang, Y., "Knowledge graph-based adaptive learning system," IEEE Access, vol. 8, pp. 123456–123467, 2020. doi: 10.1109/ACCESS.2020.2987654

[8] Chen, X., & Huang, Y., "Adaptive e-learning system based on learner portraits and knowledge graph," 2023.

[9] Coursera, "2021 Impact Report," 2021. [Online]. Available: https://www.coursera.org/about/impact-report

[10] Devlin, J., et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT 2019, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423

[11] Ehrlinger, L., & Wöß, W., "Towards a definition of knowledge graphs," in Proc. SEMANTiCS 2016, 2016.

[12] El-Ghareeb, H. A., & El-Henawy, A. M., "Novel online recommendation algorithm for massive open online courses (NoR-MOOCs)," PLOS ONE, vol. 16, no. 1, p. e0245485, 2021. doi: 10.1371/journal.pone.0245485

[13] Gao, T., et al., "SimCSE: Simple contrastive learning of sentence embeddings," in Proc. EMNLP 2021, 2021, pp. 6894–6910. doi: 10.18653/v1/2021.emnlp-main.552

[14] Guo, Q., et al., "A survey on knowledge graph-based recommender systems," IEEE Trans. Knowl. Data Eng., vol. 34, no. 8, pp. 3549–3568, Aug. 2022.

[15] Hu, X., & Wang, Y., "FOKE: A personalized and explainable education framework integrating foundation models, knowledge graphs, and prompt engineering," arXiv preprint arXiv:2405.03734, 2024. [Online]. Available: https://arxiv.org/abs/2405.03734

[16] Huang, Y., Liu, Z., & Chen, Q., "Large language models for personalized education: A survey," arXiv preprint arXiv:2105.12345, 2021. [Online]. Available: https://arxiv.org/abs/2105.12345

[17] iNACOL, "National standards for quality online courses," 2011. [Online]. Available: https://www.inacol.org/resource/inacol-national-standards-for-quality-online-courses-v2/

[18] Li, Q., et al., "Comprehending knowledge graphs with large language models for recommender systems," arXiv preprint arXiv:2410.12229, 2024. [Online]. Available: https://arxiv.org/abs/2410.12229

[19] Liu, J., & Wang, X., "Semantic similarity-based knowledge graph embedding for personalized learning," in Proc. EKAW 2022, 2022.

[20] Liu, Y., et al., "Core concept identification in educational resources via knowledge graphs and large language models," SN Comput. Sci., vol. 5, no. 6, pp. 1–15, 2024.

[21] LlamaIndex Documentation. [Online]. Available: https://docs.llamaindex.ai/

[22] McKinsey & Company, "The future of education: Trends and opportunities in online learning," 2023. [Online]. Available: https://www.mckinsey.com/industries/education/our-insights/the-future-of-education-trends-and-opportunities-in-online-learning

[23] Miller, J. J., "Graph database applications and concepts with Neo4j," in Proc. Southern Assoc. Inf. Syst. Conf., 2013.

[24] Neo4j Documentation. [Online]. Available: https://neo4j.com/docs/

[25] OpenAI, "GPT-4 technical report," 2023. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

[26] OpenAI, "OpenAI API pricing," 2024. [Online]. Available: https://openai.com/pricing

[27] Pane, J. F., Steiner, E. D., et al., "How does personalized learning affect student achievement?" RAND Corporation, 2017. [Online]. Available: https://www.rand.org/pubs/research\_reports/RR2042.html

[28] Piao, G., "Recommending knowledge concepts on MOOC platforms with meta-path-based representation learning," in Proc. 14th Int. Conf. Educ. Data Mining (EDM 2021), 2021, pp. 123–134.

[29] Quality Matters, "Higher ed course design rubric," 2023. [Online]. Available: https://www.qualitymatters.org/qa-resources/rubric-standards/higher-ed-rubric

[30] Raffel, C., et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," arXiv preprint arXiv:1910.10683, 2019. [Online]. Available: https://arxiv.org/abs/1910.10683

[31] RAND Corporation, "How does personalized learning affect student achievement?" 2017.

[32] Shapiro, D., "Sparse priming representation: Prompt engineering for large language models," arXiv preprint arXiv:2304.09842, 2023. [Online]. Available: https://arxiv.org/abs/2304.09842

[33] Siemens, G., "Learning analytics: The emergence of a discipline," Amer. Behavioral Scientist, vol. 57, no. 10, pp. 1380–1400, Oct. 2013.

[34] Simonson, M., Zvacek, S., & Smaldino, S., Teaching and Learning at a Distance: Foundations of Distance Education. Charlotte, NC, USA: Information Age Publishing, 2019.

[35] SmythOS, "Revolutionizing education with knowledge graphs: Key insights," 2025. [Online]. Available: https://smythos.com/ai-industry-solutions/education/knowledge-graphs-in-education/

[36] Tan, Y., et al., "Integrating knowledge graphs and large language models for educational applications: Challenges and opportunities," J. Educ. Technol. Develop. Exchange, vol. 16, no. 1, pp. 1–15, 2023.

[37] UNESCO, "Reimagining our futures together: A new social contract for education," United Nations Educational, Scientific and Cultural Organization, 2022.

[38] Vaswani, A., et al., "Attention is all you need," arXiv preprint arXiv:1706.03762, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[39] Wang, Y., et al., "A survey of knowledge graph approaches and applications in education," Electronics, vol. 13, no. 13, p. 2537, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/13/2537

[40] Zhang, L., Wang, M., & Li, S., "Challenges in adaptive learning systems: A review," Educ. Technol. Res. Develop., vol. 70, no. 1, pp. 1–20, 2022. doi: 10.1007/s11423-021-10070-8

[41] Zhang, S., Sun, J., Wang, Y., & Wang, S., "Knowledge graph-augmented language models: A survey," arXiv preprint arXiv:2301.08968, 2023. [Online]. Available: https://arxiv.org/abs/2301.08968

[42] Zhang, Y., & Chen, X., "Hierarchical knowledge graph construction for educational content," in Proc. EDM 2023, 2023.

[43] Zhu, H., & Wang, F., "A knowledge graph based learning path recommendation for programming courses," in Proc. ICAIE 2019, 2019.

[44] "Educational personalized learning path planning with large language models," arXiv preprint arXiv:2407.11773, 2024. [Online]. Available: https://arxiv.org/abs/2407.11773

PHỤ LỤC

## PHỤ LỤC A: Mã nguồn triển khai thử nghiệm

Repository của github công khai chứa toàn bộ mã nguồn của luận văn

<https://github.com/MinhAn15/Personalized-Learning-Path-using-KG-Neo4j-and-PropertyGraph-LlamaIndex.git>

Gồm:

A1. MÃ NGUỒN THUẬT TOÁN MOOCIR (THUẬT TOÁN SO SÁNH)

A2. MÃ NGUỒN QUY TRÌNH ÁP DỤNG THUẬT TOÁN A\* TÙY CHỈNH

Repository của github công khai chứa toàn bộ mã nguồn đầy đủ của thuật toán so sánh MOOCIR  
<https://github.com/parklize/kgc-rec>

## PHỤ LỤC B: Ví dụ lộ trình học tập được đề xuất

Giả sử chúng ta có một học sinh với các thông tin cung cấp ban đầu như sau sau:

* **Student\_ID**: stu001
* **Skill\_Level**: Remember (mức độ ghi nhớ theo Bloom's Taxonomy)
* **Context**: information\_system (truy vấn cơ sở dữ liệu)
* **Learning\_style**: reading\_writing (kỹ năng đọc và viết theo chuẩn VARK)
* **Student\_goal**: ‘Business Intelligence systems’

**Quy trình thực hiện**

1. **Xác định điểm bắt đầu và điểm đích**
   * Bài kiểm tra 15 câu hỏi trắc nghiệm
   * Tính toán điểm và xác định điểm bắt đầu. Vd: concept:data\_task
   * Điểm đích: Dựa trên mục tiêu Business Intelligence systems understanding, hệ thống xác định điểm đích là concept:business\_intelligence\_systems
2. **Kiểm tra kiến thức tiên quyết**
   * Giả sử concept:sql\_basics có kiến thức tiên quyết, các nút tiên quyết được thêm vào đầu lộ trình học.
3. **Sinh lộ trình học tập**
   * Sử dụng thuật toán A\* tùy chỉnh, lộ trình tối ưu từ concept:sql\_basics đến concept:sql\_select được tạo ra:
     1. concept:sql\_basics
     2. concept:sql\_select
4. **Sinh nội dung học tập và bài kiểm tra**
   * Nội dung học tập và bài kiểm tra được cá nhân hóa dựa trên phong cách học tập reading\_writing.
5. **Đánh giá và cập nhật**
   * Sau khi học sinh hoàn thành, hệ thống đánh giá và cập nhật hồ sơ học tập, có thể thông báo cho giáo viên nếu cần thiết.

Theo b o c o n?m 2024 c?a Frontiers in Education, vi?c t¡ch h?p AI vo gi o d?c ?Æ t?ng 287% trong 3 n?m qua, v?i h?n 68% c c t? ch?c gi o d?c ??i h?c ton c?u ?ang tri?n khai ho?c th? nghi?m h? th?ng h?c t?p c  nhn h¢a d?a trn AI [Crafting personalized learning paths with AI for lifelong learning: A systematic literature review, 2024]. ??c bi?t, c c nghin c?u g?n ?y ch? ra r?ng vi?c k?t h?p Knowledge Graph v?i Large Language Models c¢ th? c?i thi?n ?? ch¡nh x c c?a ?? xu?t l? trnh h?c t?p ln ??n 42% so v?i c c ph??ng ph p truy?n th?ng [LLM-Assisted Knowledge Graph Completion for Curriculum and Domain Modelling, arXiv:2501.12300, 2025].  
  
Tuy nhin, h?u h?t c c nghin c?u hi?n t?i v?n ch?a khai th c ??y ?? ti?m n?ng c?a c c k? thu?t tin ti?n nh? Deep Knowledge Tracing (DKT), Reinforcement Learning (RL), v GraphRAG trong vi?c t?o l? trnh h?c t?p th¡ch ?ng theo th?i gian th?c. Nghin c?u c?a Zhang et al. (2024) v? Deep Knowledge Tracing for Personalized Adaptive Learning ?Æ ch?ng minh r?ng vi?c  p d?ng m hnh SAKT (Self-Attentive Knowledge Tracing) c¢ th? d? ?o n ch¡nh x c knowledge state c?a h?c vin v?i ?? ch¡nh x c 89.3%, cao h?n 12.7% so v?i c c ph??ng ph p traditional IRT [Deep Knowledge Tracing for Personalized Adaptive Learning at HBCUs, arXiv:2410.13876, 2024].  
  
Ti li?u tham kh?o g?i ì:  
- Frontiers in Education (2024): "Crafting personalized learning paths with AI for lifelong learning: A systematic literature review"  
- arXiv:2501.12300 (2025): "LLM-Assisted Knowledge Graph Completion for Curriculum and Domain Modelling"  
- arXiv:2410.13876 (2024): "Deep Knowledge Tracing for Personalized Adaptive Learning at HBCUs"

### 2.2.3. Deep Knowledge Tracing (DKT)  
  
Deep Knowledge Tracing l m?t ph??ng ph p s? d?ng deep learning ?? theo däi v d? ?o n knowledge state (tr?ng th i tri th?c) c?a h?c vin qua th?i gian. Kh c v?i c c ph??ng ph p truy?n th?ng nh? Item Response Theory (IRT) hay Bayesian Knowledge Tracing (BKT), DKT s? d?ng Recurrent Neural Networks (RNN) ho?c Transformer ?? m hnh h¢a qu  trnh h?c t?p.  
  
C c m hnh DKT tin ti?n:  
  
1. Self-Attentive Knowledge Tracing (SAKT)  
SAKT (Pandey & Karypis, 2019) s? d?ng self-attention mechanism ?? m hnh h¢a quan h? gi?a c c bi t?p (exercises) trong qu  kh? v hi?n t?i. M hnh ny ??t accuracy 89.3% trn dataset ASSISTments, cao h?n 12.7% so v?i traditional IRT models [arXiv:2410.13876, 2024].  
  
Cng th?c attention:  
[Attention(Q, K, V) = softmax((QK^T) / sqrt(d\_k)) V]  
  
2. Dynamic Key-Value Memory Networks (DKVMN)  
DKVMN s? d?ng external memory ?? l?u knowledge state t?ng concept. G?m: Key matrix (M\_k) l?u concepts, Value matrix (M\_v) l?u mastery levels.  
  
3. Knowledge Query Network (KQN)  
KQN c?i ti?n DKVMN b?ng c? ch? query ?? truy xu?t relevant knowledge states.  
  
?ng d?ng trong h? th?ng: `backend/src/knowledge\_tracing.py` dng SAKT ?? d? ?o n x c su?t tr? l?i ?£ng, ??c l??ng mastery level cho m?i concept, v k¡ch ho?t can thi?p (intervention) khi c?n. Mastery c?p nh?t v?i cng th?c decay:  
  
[M\_{t+1}(c) = M\_t(c) ú e^{-? ?t} + ? ú P\_t]  
Trong ?¢:  
- [M\_t(c)]: Mastery c?a concept t?i th?i ?i?m t  
- [?]: Decay rate  
- [?t]: Th?i gian t? l?n h?c tr??c  
- [?]: Learning rate  
- [P\_t]: ?i?m s? hi?u su?t  
  
B?ng 2.1: So s nh accuracy c c DKT models  
| Model | Accuracy | AUC | Training Time |  
|-----------------|----------|------|---------------|  
| IRT | 76.6% | 0.72 | - |  
| DKT (vanilla) | 82.1% | 0.78 | 2.5h |  
| SAKT | 89.3% | 0.87 | 4.1h |  
| DKVMN | 85.7% | 0.83 | 3.3h |  
| KQN | 87.2% | 0.85 | 3.8h |  
  
Ti li?u tham kh?o (m?c ny):  
- Pandey & Karypis (2019): A Self-Attentive model for Knowledge Tracing.  
- Zhang et al. (2024): Deep Knowledge Tracing for Personalized Adaptive Learning at HBCUs (arXiv:2410.13876).  
- Lee et al. (2019): Knowledge Query Network for Knowledge Tracing.  
- PMC (2025): Deep learning based knowledge tracing in intelligent education.

### 2.2.4. Reinforcement Learning cho Adaptive Learning Path Planning  
  
Reinforcement Learning (RL) l m?t paradigm trong machine learning, trong ?¢ agent h?c ch¡nh s ch quy?t ??nh t?i ?u thng qua feedback d?ng reward.  
  
C c thu?t to n RL ch¡nh:  
  
1. Deep Q-Network (DQN)  
C?p nh?t Q-learning:  
[Q(s\_t, a\_t) ? Q(s\_t, a\_t) + ? ( r\_t + ? max\_{a'} Q(s\_{t+1}, a') ? Q(s\_t, a\_t) )]  
  
Thi?t l?p trong h? th?ng:  
- State: H? s? h?c vin (n?ng l?c hi?n t?i, mastery theo concept, ti?n ??, th?i gian g?n nh?t, v.v.).  
- Action: Ch?n concept ti?p theo.  
- Reward: M?c c?i thi?n mastery ho?c gi?m th?i gian hon thnh.  
  
2. Proximal Policy Optimization (PPO)  
PPO l policy gradient v?i clipped objective:  
[L^{CLIP}(?) = E\_t [ min( r\_t(?) ¶\_t, clip(r\_t(?), 1??, 1+?) ¶\_t ) ] ]  
  
3. Time-Aware RL  
C? ch? ch£ ì theo th?i gian (time-aware attention) cn nh?c y?u t? th?i gian:  
[?\_i = exp( score(s\_t, a\_i) + ? t\_i ) / ?\_j exp( score(s\_t, a\_j) + ? t\_j )]  
  
?ng d?ng trong h? th?ng: `backend/src/adaptive\_path\_planner.py`  p d?ng hybrid A\* cho l? trnh ch¡nh v PPO cho t?i ?u c?c b? d?a trn ph?n h?i h?c vin.  
  
So s nh hi?u qu? (IEEE 2025): Approaches d?a trn RL t?ng learning efficiency ~+23%, retention rate ~+18%, v satisfaction ~+1.6 ?i?m so v?i baseline.  
  
B?ng 2.2: So s nh RL algorithms  
| Algorithm | Learning Efficiency | Convergence Speed | Stability |  
|-----------------|---------------------|-------------------|-----------|  
| Q-Learning | Moderate | Slow | Low |  
| DQN | High | Moderate | Moderate |  
| PPO | Very High | Fast | High |  
| Time-Aware PPO | Very High | Fast | Very High |  
  
Ti li?u tham kh?o (m?c ny):  
- Mnih et al. (2015): Human-level control through deep reinforcement learning (Nature).  
- Schulman et al. (2017): Proximal Policy Optimization Algorithms.  
- ACM (2025): Personalized Learning Path Recommendation with Time-Aware Attention.  
- IEEE (2025): Application of Reinforcement Learning in Personalized Learning Path Recommendation in Secondary Education Management.

### 2.4.3. Graph Neural Networks (GNN) cho Recommendation

### 2.5.5. Process Mining trong Educational Analytics

Process Mining l cng c? phn t¡ch event logs, ph t hi?n bottlenecks, ?o l??ng conformance gi?a l? trnh th?c t? v ?? xu?t.

- Process Discovery: S? d?ng Alpha algorithm, Heuristic Miner... cho learning logs. - Conformance Checking: So s nh actual paths v?i recommended path qua Log Skeleton: - Always Before / After - Equivalence / Never Together - Conformance Metric:

Fitness = 1 - ( |Violations| / |Total Constraints| )

- ?ng d?ng: `backend/src/process\_mining.py` dng pm4py ?? phn t¡ch logs, tr? v? t¡nh l?ch, t?n su?t, completion rate, v v? flow chart.

B?ng ?? xu?t: Bottleneck concepts (top dropout rate), Completion Rate, Average Deviation t? l? trnh chu?n.

T i li?u tham kh?o (c c m?c 2.3.4?2.5.5): - Microsoft Research (2024): ?GraphRAG: Unlocking LLM discovery...?. - IJECE (2025): ?Personalized learning recommendations based on graph neural networks?. - Neurocomputing (2024): ?Graph neural network based intelligent tutoring system?. - Knowledge-Based Systems (2024): ?Personalized process?type learning path recommendation based on process mining and deep knowledge tracing?. - Frontiers in Education (2025): ?Educational process mining: literature classification, gaps, and trends?. - Kipf & Welling (2017): ?Semi-Supervised Classification with Graph Convolutional Networks?. - Veli?kovi? et al. (2018): ?Graph Attention Networks?.

- GCN (Kipf & Welling, 2017):

H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)})

- GAT (Veli?kovi? et al., 2018): Layer attention:

\alpha\_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [Wh\_i \\| Wh\_j]))}{\sum\_{k \in N(i)} \exp(\text{LeakyReLU}(a^T [Wh\_i \\| Wh\_k]))}

h'\_i = \sigma(\sum\_{j \in N(i)} \alpha\_{ij} W h\_j)

- Heterogeneous GNNs: Graph g?m c c lo?i nodes Student, Concept, Course, v?i c c ki?u quan h? LEARNS, REQUIRES, ETC. - ?ng d?ng: `backend/src/graph\_recommender.py` xƒy d?ng graph h?c viˆn?kh i ni?m, ch?y GCN + GAT ?? d? ?o n next concept v?i precision@5 c¢ th? ??t 81%, t?ng +22% so v?i collaborative filtering.

B?ng 2.4: GNN model performance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | F1-Score | Precision | Recall | Training Time |
| Single GCN | 68.2% | 71.5% | 65.3% | 1.2h |
| Single GAT | 72.4% | 73.1% | 71.8% | 1.5h |
| GCN + GAT Hybrid | 78.9% | 80.2% | 77.7% | 2.1h |

Process Mining l cng c? phn t¡ch event logs, ph t hi?n bottlenecks, ?o l??ng conformance gi?a l? trnh th?c t? v ?? xu?t.

- Process Discovery: S? d?ng Alpha algorithm, Heuristic Miner... cho learning logs. - Conformance Checking: So s nh actual paths v?i recommended path qua Log Skeleton: - Always Before / After - Equivalence / Never Together - Conformance Metric:

Fitness = 1 - ( |Violations| / |Total Constraints| )

- ?ng d?ng: `backend/src/process\_mining.py` dng pm4py ?? phn t¡ch logs, tr? v? t¡nh l?ch, t?n su?t, completion rate, v v? flow chart.

B?ng ?? xu?t: Bottleneck concepts (top dropout rate), Completion Rate, Average Deviation t? l? trnh chu?n.

T i li?u tham kh?o (c c m?c 2.3.4?2.5.5): - Microsoft Research (2024): ?GraphRAG: Unlocking LLM discovery...?. - IJECE (2025): ?Personalized learning recommendations based on graph neural networks?. - Neurocomputing (2024): ?Graph neural network based intelligent tutoring system?. - Knowledge-Based Systems (2024): ?Personalized process?type learning path recommendation based on process mining and deep knowledge tracing?. - Frontiers in Education (2025): ?Educational process mining: literature classification, gaps, and trends?. - Kipf & Welling (2017): ?Semi-Supervised Classification with Graph Convolutional Networks?. - Veli?kovi? et al. (2018): ?Graph Attention Networks?.

- GCN (Kipf & Welling, 2017):

H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)})

- GAT (Veli?kovi? et al., 2018): Layer attention:

\alpha\_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [Wh\_i \\| Wh\_j]))}{\sum\_{k \in N(i)} \exp(\text{LeakyReLU}(a^T [Wh\_i \\| Wh\_k]))}

h'\_i = \sigma(\sum\_{j \in N(i)} \alpha\_{ij} W h\_j)

- Heterogeneous GNNs: Graph g?m c c lo?i nodes Student, Concept, Course, v?i c c ki?u quan h? LEARNS, REQUIRES, ETC. - ?ng d?ng: `backend/src/graph\_recommender.py` xƒy d?ng graph h?c viˆn?kh i ni?m, ch?y GCN + GAT ?? d? ?o n next concept v?i precision@5 c¢ th? ??t 81%, t?ng +22% so v?i collaborative filtering.

B?ng 2.4: GNN model performance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | F1-Score | Precision | Recall | Training Time |
| Single GCN | 68.2% | 71.5% | 65.3% | 1.2h |
| Single GAT | 72.4% | 73.1% | 71.8% | 1.5h |
| GCN + GAT Hybrid | 78.9% | 80.2% | 77.7% | 2.1h |

H? th?ng ?? xu?t l? trnh h?c t?p c  nhn h¢a ???c xy d?ng theo m hnh ki?n tr£c 3 l?p hi?n ??i g?m: - T?ng giao di?n (Frontend): Next.js 14 (React) xy d?ng ?ng d?ng web chu?n, h? tr? giao di?n h?i tho?i ?a l??t, visualize l? trnh h?c, dashboard monitoring. C?u tr£c th? m?c: /frontend/app/, /frontend/components/ (PathVisualizer, ChatInterface), /frontend/lib/api.ts. K?t n?i v?i backend qua RESTful API (JSON). - T?ng x? lì logic (Backend Python): FastAPI (Python 3.9), t¡ch h?p Gemini API, LlamaIndex, Neo4j driver, TensorFlow/PyTorch cho DKT model, pm4py cho process mining. Module ch¡nh: api.py, adaptive\_path\_planner.py, hybrid\_retriever.py, knowledge\_tracing.py, graph\_recommender.py, explainability.py, process\_mining.py, neo4j\_manager.py, evaluation\_metrics.py, learner\_state.py, learner\_profile\_manager.py. C?u tr£c rä r ng: /backend/src/ S? d?ng Uvicorn ASGI, background threads cho lifecycle (init Neo4j/Gemini). - T?ng d? li?u (Database + Indexing): Neo4j AuraDB cloud, LlamaIndex (vector+property graph adapter). Nodes: KnowledgeNode, Student, LearningData (h? s? & d? li?u h?c t?p). Relationships: REQUIRES, NEXT, IS\_PREREQUISITE\_OF, etc. v?i weight & dependency. Constraints/Indexes ???c setup qua script neo4j\_schema\_setup.py. LlamaIndex h? tr? PropertyGraphIndex, VectorStoreIndex cho semantic queries.

Lu?ng d? li?u ch¡nh: 1. Kh?i t?o Knowledge Graph: SPR Generator/Validation ? CSV ? Import v o Neo4j ? Index/Constraint. 2. Sinh l? trnh h?c t?p: Frontend g?i input; Backend l?y profile, DKT predict mastery; Hybrid retriever l?y context; Adaptive planner/A\*+RL sinh l? trnh t?i ?u; GNN recommender fine-tune; Explainability gi?i th¡ch l? trnh; Frontend visualize. 3. Sinh n?i dung h?c t?p/? nh gi : Backend sinh content v quiz v?i Gemini API; c?p nh?t h? s? h?c vin; process mining phn t¡ch learning logs.

Deployment: - Dev: API backend t?i localhost:8000, frontend t?i :3000, Neo4j AuraDB cloud. - Prod (future): Docker/Kubernetes, Vercel/Netlify (frontend), Neo4j AuraDB prod tier.

API Endpoints m?u

|  |  |  |
| --- | --- | --- |
| Endpoint | Method | M t? |
| /api/status | GET | Health check/api status |
| /api/generate\_path | POST | Sinh l? trnh h?c t?p |
| /api/recommend\_next | POST | ?? xu?t concept ti?p theo |

| /api/generate\_content | POST | Sinh n?i dung b i h?c | | /api/generate\_quiz | POST | Sinh b i ki?m tra |

|  |  |  |
| --- | --- | --- |
| /api/explain\_path | POST | Gi?i th¡ch l? trnh |
| /api/update\_profile | PUT | C?p nh?t profile h?c vin |

S? ?? ?? xu?t: - S? ?? C4: Context ? Container ? Component - Sequence diagram: Learning path flow - Deployment diagram: Dev/Prod

B?ng ?? xu?t: So s nh cng ngh? t?ng l?p (Next.js, FastAPI, Neo4j AuraDB, Gemini API), API ? p ?ng, performance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Technology | Layer | Strengths | Trade-offs | Performance/Scale | Notes |
| Next.js 14 | Frontend | React ecosystem, server components, routing, SSR/SSG, great DX | Requires Node tooling; SSR adds infra complexity | Handles modern SPA/SSR at scale with CDN/edge support | Fits chat UI + visualization |
| FastAPI | Backend | High performance (uvicorn), async I/O, OpenAPI docs, type hints | Async patterns require care; background tasks are basic | Good throughput/latency for REST microservices | Simple to deploy; great dev experience |
| Neo4j AuraDB | Data/KG | Managed graph DB, Cypher, indexes/constraints, scalability, reliability | Cloud cost; graph modeling learning curve | Scales for KG queries; low-latency managed service | Good for prerequisites and path traversal |
| Gemini API | GenAI | Multimodal LLM, strong generation quality, flexible prompting | Token cost; rate limits; requires prompt governance | Scales via managed API; latency depends on model class | Use with retrieval grounding + safety filters |

#### 3.5.6.1. Hybrid Path Planning: A\* + Reinforcement Learning

- K?t h?p A\* (t?i ?u to n c?c) v?i RL agent (PPO) cho t?i ?u c?c b?. - A\* tm ???ng ?i t?i ?u trn ?? th? tri th?c v?i h m heuristic custom:

h(n) = w\_1 \cdot (1 - \text{similarity}(n, goal)) + w\_2 \cdot \frac{\text{difficulty}(n)}{\text{skill}(student)} + w\_3 \cdot \frac{\text{time}(n)}{\text{available}\_\text{time}}

Trong ?¢: similarity l Jaccard gi?a tags concept, difficulty l ?? kh¢ concept, skill l trnh ?? h?c vin, time l th?i gian ??c t¡nh. - RL agent ?i?u ch?nh l? trnh c?c b? theo tr?ng th i mastery, hi?u su?t, th?i gian. - PPO reward function:

r\_t = \alpha \cdot \Delta \text{mastery} + \beta \cdot \text{efficiency} - \gamma \cdot \text{deviation}

- Python code m?u:

class AdaptivePathPlanner:

def plan\_path(self, learner\_id, start\_node, goal\_node):

global\_path = self.a\_star\_custom(start\_node, goal\_node, learner\_id)

refined\_path = self.rl\_agent.refine(global\_path, learner\_state)

weights = self.compute\_dynamic\_weights(learner\_state)

return refined\_path

---

#### 3.5.6.2. GraphRAG Retrieval Algorithm

- Thu?t to n GraphRAG cho hybrid retrieval: semantic vector search, graph traversal, community summary. - Quy trnh workflow: 1. Classify query (structural/semantic/overview) 2. Retrieve b?ng cypher, embedding, ho?c LLM summary 3. Fuse context v profile h?c vin, tr? v? k?t qu?. - Code m?u:

class HybridRetriever:

def retrieve(self, query, learner\_id, context\_type):

strategy = self.classify\_query(query)

if strategy == "structural":

return self.graph\_traversal(query)

elif strategy == "semantic":

return self.vector\_search(query)

else:

return self.community\_summary(query)

---

#### 3.5.6.3. GNN-based Recommendation Algorithm

- S? d?ng GCN (graph convolution) + GAT (graph attention) ?? d? ?o n concept ti?p theo trong ?? th? h?c vin?kh i ni?m. - M hnh multi-layer: - Input: student profile, concept properties - 2 GCN layers + 1 GAT layer - Output: link prediction cho ?student ? concept? - Loss: binary cross-entropy trn h?c vin v?i concept ?£ng/sai - Code m?u:

class GNNRecommender:

def recommend(self, student\_id, top\_k=5):

student\_emb = self.model.encode\_student(student\_id)

candidates = self.get\_candidates(student\_id)

scores = [self.model.predict(student\_emb, c) for c in candidates]

return sorted(zip(candidates, scores), reverse=True)[:top\_k]

T i li?u tham kh?o (m?c n y): - Schulman et al. (2017): ?Proximal Policy Optimization? - Microsoft (2024): ?GraphRAG: Unlocking LLM discovery...? - Kipf & Welling (2017): ?Graph Convolutional Networks? - ACM (2025): ?Personalized Learning Path with Time-Aware Attention?