FPT Education
FPT UNIVERSITY

Hướng dẫn

Cách thực hiện và viết báo cáo trong một dự án nghiên cứu

Research-Based Project

Đối tượng sinh viên CNTT: An toàn thông tin, Kỹ thuật phần mềm, Hệ thống thông tin, Trí tuệ nhân tạo, Khoa học máy tính

Ngô Tùng Sơn, PhD
Faculty of Information Assurance, FPT University

FPT Education
FPT UNIVERSITY

Bản chất của việc thực hiện một nghiên cứu

The nature of conducting research

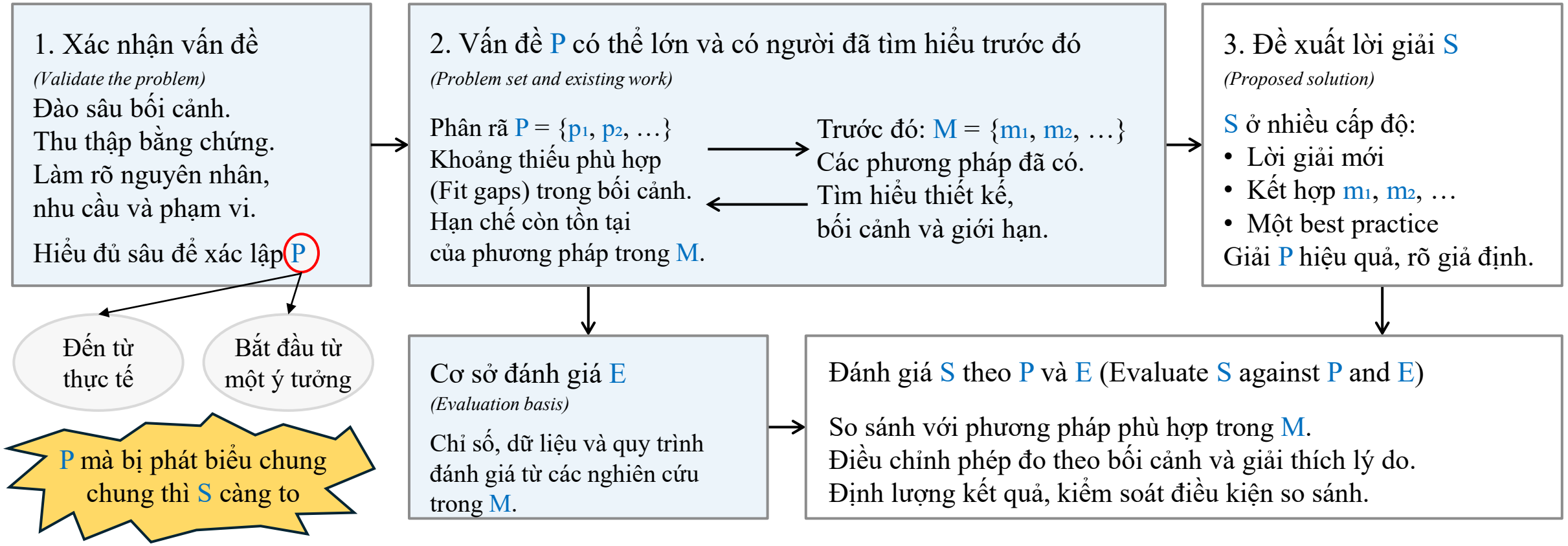
1. Xác lập bài toán P
2. Khảo sát phương pháp M
3. Đề xuất lời giải S
4. Đánh giá bằng cơ sở E

PHẦN

01

Nghiên cứu = Problem + Solution + Evaluation

Nghiên cứu để **làm rõ P**, **đề xuất S** và **chứng minh bằng E** rằng S giải được P tốt hơn M.
Gặp P khó mà nghĩ ngay ra S ở giây đầu tiên thông thường S đó là giải pháp tệ nhất (hời hợt, nông cạn, và không để làm gì cả).

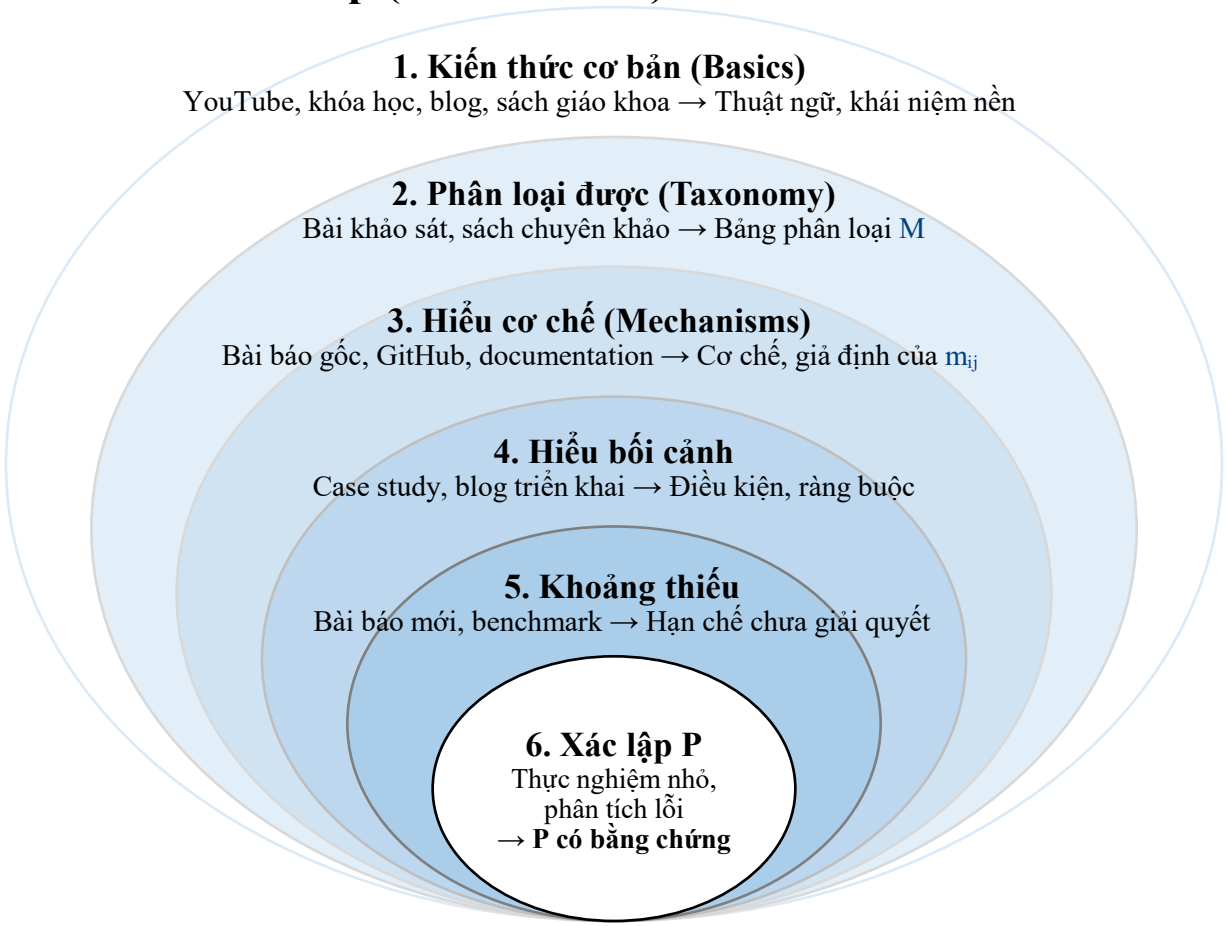


Mỗi p_i cần có cơ chế giải trong S và phép đo tương ứng trong E (Problem, mechanism and evidence).

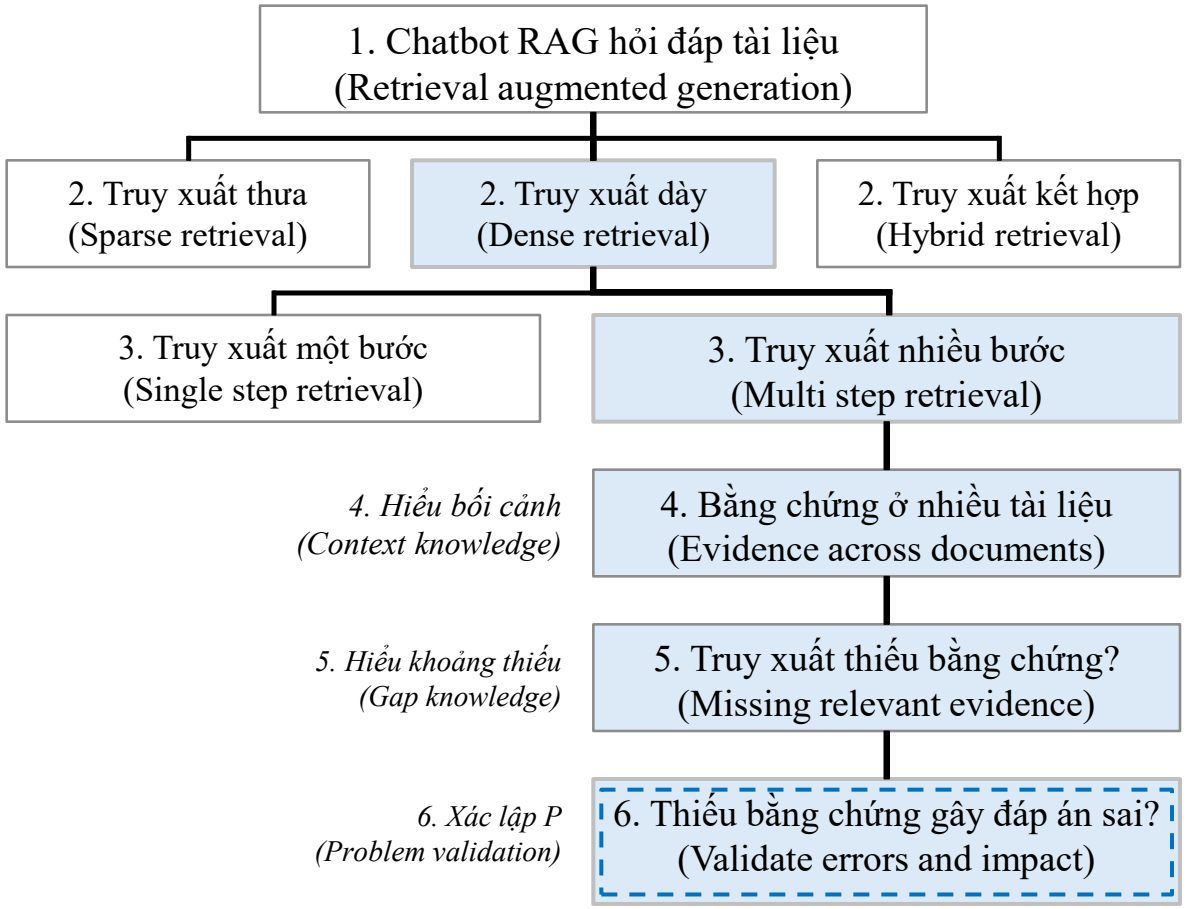
Xác định Problem P từ một ý tưởng

Đi từ hiểu biết sơ khởi đến thực sự am hiểu lĩnh vực, rồi mới xác lập P có căn cứ (From novice to domain expert).

Mô hình nhiều lớp (Onion model)



Cây kiến thức chatbot RAG (RAG knowledge hierarchy)



Điều kiện đưa vào P (Inclusion criteria): có bằng chứng, có ý nghĩa, M chưa đáp ứng và phạm vi khả thi.

Mỗi p_i gồm bối cảnh, hạn chế, bằng chứng, ảnh hưởng và phép đo (Context, limitation, evidence, impact, measure).

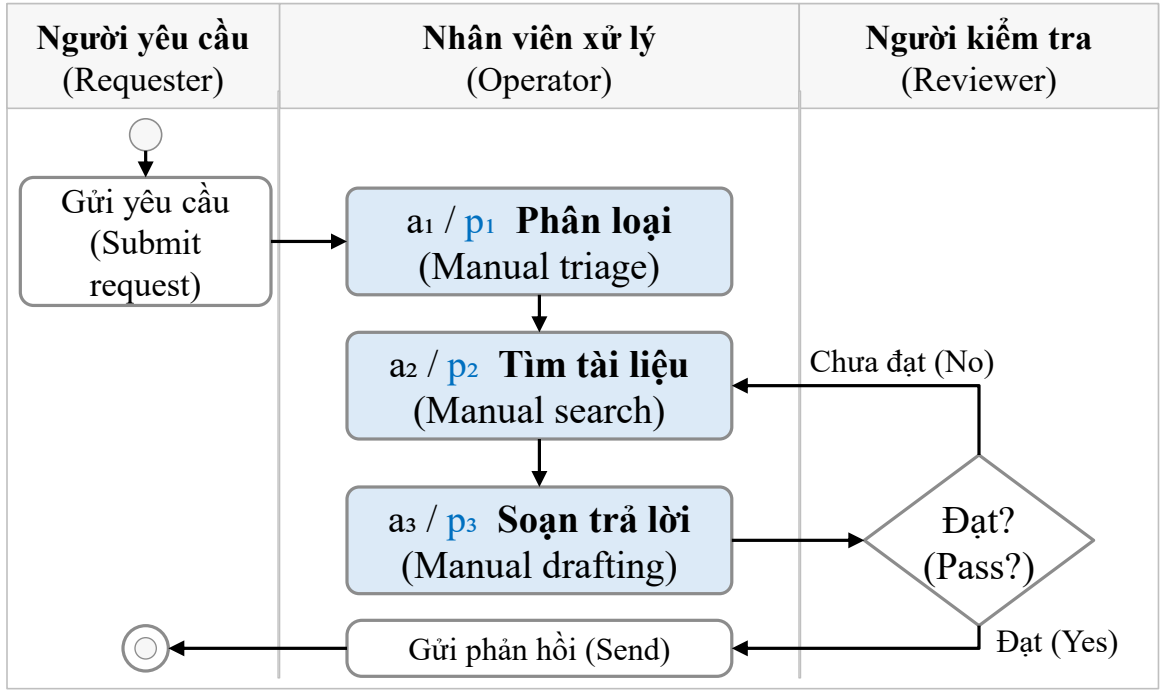
Xác định Problem P từ quy trình vận hành thực tế

Vấn đề thực tế của doanh nghiệp là bài toán nghiên cứu. Ví dụ: tra cứu quy định nội bộ (Internal policy requests)

Doanh nghiệp phát biểu bằng lời (Business need)
"Trả lời nhanh, đúng, đúng quyền, đừng tốn kém."

→ Researcher phát biểu kỹ thuật (Formulation)
max Acc(f) s.t. Time ≤ T, Cost ≤ C, Violations = 0

Quy trình hiện tại (As is activity diagram)



Tô xanh: xem xét cải tiến / tự động hóa (Automation candidates)

Mục tiêu và chỉ số công đoạn (Stage goals and metrics)

Vấn đề p_i (Candidate)	Mục tiêu (Stage goal)	Chỉ số (Stage metrics)
p_1: Phân loại sai *(Triage errors)*	**Đúng loại yêu cầu** *(Correct routing)*	Tỷ lệ phân loại đúng (%) Thời gian thao tác (min)
p_2: Tìm lâu, thiếu bằng chứng *(Search gaps)*	**Đủ tài liệu có hiệu lực** *(Valid evidence)*	Tỷ lệ đủ bằng chứng (%) Thời gian tìm (min)
p_3: Soạn lâu, sửa nhiều *(Drafting rework)*	**Trả lời đúng, có căn cứ** *(Grounded answer)*	Tỷ lệ trả lời đúng (%) Số vòng sửa / ca Thời gian soạn (min)

Xác nhận P bằng quan sát và dữ liệu (Evidence)
Ghi người làm, đầu vào, đầu ra, thời gian chờ và vòng sửa.
Đối chiếu ngưỡng chấp nhận để xác nhận từng p_i.

Đo hiện trạng B và chốt mục tiêu T trước khi chọn giải pháp (Baseline and target).
Sau cải tiến: cùng loại yêu cầu, cùng cách chấm, đo từng công đoạn và cả quy trình (Comparable evaluation).

Tìm kiếm các Lời giải M sẵn có cho từng bài toán con

$P = \{p_1, p_2, \ldots\}$ $M_i = \{m_{i1}, m_{i2}, \ldots\}$ $M = \cup_i M_i$

Tìm M từ tài liệu và hệ thống đã có
(Literature and existing systems)

P: chatbot RAG trả lời có căn cứ
(Grounded document question answering)

m_{ij} có cơ sở đánh giá E_{ij}
(Method specific evaluation)

p_1: lấy đủ bằng chứng (Retrieve sufficient evidence)	p_2: chọn ngữ cảnh phù hợp (Select useful context)	p_3: trả lời có căn cứ (Generate grounded answers)
$M_1 = \{m_{11}, m_{12}, m_{13}, \ldots\}$	$M_2 = \{m_{21}, m_{22}, m_{23}, \ldots\}$	$M_3 = \{m_{31}, m_{32}, \ldots\}$
m_{11} / E_{11} BM25 (Sparse) m_{12} / E_{12} Vector (Dense) m_{13} / E_{13} Kết hợp (Hybrid)	m_{21} / E_{21} Chọn top k (Top k) m_{22} / E_{22} Xếp hạng lại (Rerank) m_{23} / E_{23} Nén ngữ cảnh (Compress)	m_{31} / E_{31} Sinh kèm dẫn chứng (Generate with citations) m_{32} / E_{32} Sinh rồi kiểm chứng (Generate and verify)

Phân tích theo mục tiêu nghiên cứu (Analysis by research objective)

Bài toán ứng dụng (Applied problem)

Đối chiếu với bối cảnh C (Context fit gaps).
Yêu cầu nào đáp ứng, yêu cầu nào còn thiếu?
Chọn hoặc điều chỉnh phương pháp phù hợp.

Cải tiến phương pháp (Method improvement)

Phân tích ưu, nhược điểm (Strengths and limits).
Giới hạn nào có bằng chứng và còn cải thiện được?
Đề xuất cải tiến, kiểm chứng bằng phép đo E.

Khi xây S, các phương pháp được chọn cần phối hợp được và đáp ứng mục tiêu toàn hệ thống (System compatibility).

Software Engineering: Nghiên cứu cho các yêu cầu phi chức năng

Quyết định kỹ thuật cần khảo sát, so sánh và kiểm chứng theo bối cảnh → đây là dạng nghiên cứu có hình thức Applied research.

Hiệu năng (Performance) An toàn (Security) Khả năng mở rộng (Scalability)

Bài toán P *(Problem)*	**Phương pháp M** *(Methods)*	**Đánh giá E** *(Evaluation)*
Cô lập môi trường thực thi (Isolate runtime environment)	VM, container, SDN, phân bổ tài nguyên,… *(Virtualization, resource allocation)*	Chi phí tài nguyên, cách ly, tải đạt SLO *(Overhead, isolation, capacity at SLOs)*
Kiểm soát nghẽn mạng (Network congestion control)	Lập lịch hàng đợi, AQM, điều tiết tải,… *(Queue scheduling, AQM, traffic shaping)*	Thông lượng, p95 độ trễ, mất gói, công bằng *(Throughput, latency, packet loss, fairness)*
Kiểm soát truy cập (Access control)	R-BAC theo vai trò, A-BAC theo thuộc tính,… *(Role / Attribute based access control)*	Cấp sai quyền, từ chối nhầm, p95 độ trễ *(False permits, false denials, latency)*

Nguồn khảo sát và trích dẫn (Survey and citation sources)

Bài báo (Research papers)	**Blog kỹ thuật (Technical blogs)**	**Sách (Books)**	**Tài liệu chính thức (Documentation)**
Cơ chế, so sánh, thực nghiệm (Methods and evidence)	Triển khai, cấu hình, tình huống (Implementation experience)	Nền tảng và cách lập luận (Principles and reasoning)	API, hành vi, giới hạn phiên bản (Behavior and version limits)

Có thể trích dẫn cả bốn loại nguồn, theo độ tin cậy và nội dung được hỗ trợ (Source quality and claim support).

Ghi tác giả, thời điểm, phiên bản và điều kiện. Kiểm chứng lại với P (Traceability and contextual validation).

Ví dụ: Chức năng tìm kiếm bài viết – Sơ khởi đến nâng cao (hơn)

Khi đối mặt với bài toán tìm kiếm, Solution đầu tiên bạn có thể nghĩ tới:

SELECT * FROM Article WHERE title like '%'+query+'%' OR content like '%'+query+'%'

1. Quét chuỗi (Database fundamentals)

Query: mật khẩu

A1	cài máy in
A2	đổi mật khẩu
A3	quản lý VPN
A4	khôi phục tài khoản
A5	sao lưu dữ liệu
A6	mật khẩu an toàn

Tăng tốc chuỗi con (Trigram) [3]

Query → Trigram GIN → Recheck

Ưu: Có sẵn trong SQL, không cần hạ tầng thêm; đủ khi dữ liệu nhỏ, ít truy vấn.

Nhược: Quét toàn bảng nên chậm khi dữ liệu lớn; chỉ khớp đúng chuỗi, không xếp hạng.

2. Chỉ mục đảo (Full text search) [1]

Query: mật khẩu

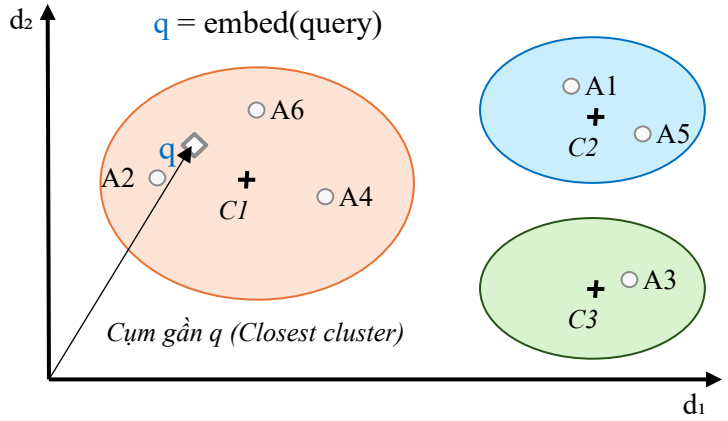
Tách từ, tra danh sách ID (Postings)

mật	A2, A6
khẩu	A2, A6
VPN	A3

Ưu: Truy vấn nhanh nhờ chỉ mục; xếp hạng theo độ liên quan [1].

Nhược: Phụ thuộc tách từ (khó với tiếng Việt); không hiểu từ đồng nghĩa; phải cập nhật chỉ mục.

3. Không gian vector (IVF / ANN) [2]



Vị trí 2D chỉ là minh họa (Schematic coordinates).

Ưu: Khớp theo nghĩa: A4 liên quan dù khác từ (Semantic match).

Nhược: Kết quả xấp xỉ, cần tuning nprobe và benchmark [2]; tốn RAM và chi phí embedding.

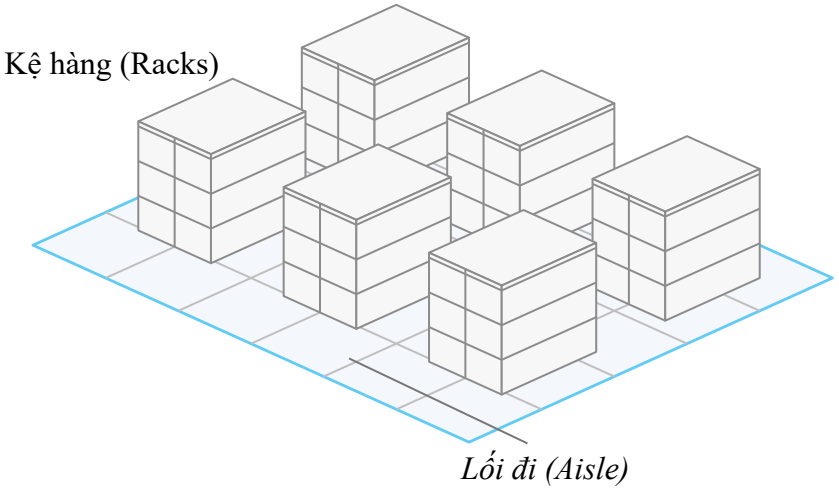
Tài liệu tham khảo: [1] Zobel & Moffat (2006), ACM CS 38(2), 10.1145/1132956.1132959 | [2] Douze et al. (2024), Faiss, arXiv:2401.08281 | [3] PostgreSQL Docs, pg_trgm.

Đánh giá theo context: p95 latency, Recall@k / nDCG@k, RAM, chi phí và cập nhật (Comparable workload).

Ví dụ: Tính năng hiển thị kho 3D

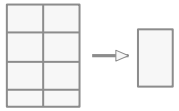
Dựng được kho là lời giải sơ khởi (Baseline). Quy mô và chuyển động đặt ra các bài toán cần khảo sát tài liệu và đánh giá theo bối cảnh (Technical literature and context-based evaluation).

1. Mô hình kho ban đầu (3D warehouse layout)



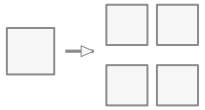
Mô hình, camera và ánh sáng (Scene setup).
Kho nhỏ, cảnh tĩnh: cấu hình đơn giản có thể đủ.
Nhiều kệ và xe: cần đo tải CPU / GPU.

2. Tối ưu hiển thị (Rendering optimization)



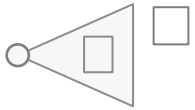
Chi tiết theo khoảng cách (LOD) [1, 3]

Kệ xa đơn giản, kệ gần rõ hơn. Đo chất lượng hình ảnh.



Dùng chung mesh và material (Instancing) [3]

Giảm draw calls cho nhiều kệ giống nhau.

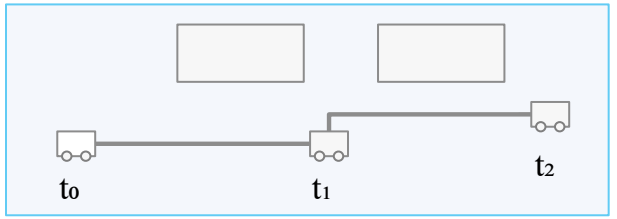


Loại ngoài vùng camera (Frustum culling) [3]

Xét thêm che khuất nếu cần (Occlusion culling).

3. Chuyển động mô phỏng (Animation)

Vị trí xe theo thời gian (Time samples)



Nội suy vị trí và góc quay [2] (Interpolation / quaternion).

Cập nhật theo Δt, quản lý clip [3] (AnimationMixer / time step).

Nếu mô phỏng vận hành: thêm quy tắc va chạm, chờ và phục vụ (Collision and event logic).

Lựa chọn theo bối cảnh và đo công bằng (Context and evaluation)

Cùng thiết bị, layout và kịch bản camera / xe: đo p95 frame time, FPS, draw calls, RAM / VRAM và thời gian tải.
Nếu mô phỏng vận hành, đo thêm thời gian lấy hàng và vi phạm va chạm. Mượt hiển thị và đúng mô phỏng là hai mục tiêu riêng.

[1] Luebke & Erikson (1997). View dependent simplification of arbitrary polygonal environments. SIGGRAPH. DOI: 10.1145/258734.258847

[2] Shoemake (1985). Animating rotation with quaternion curves. SIGGRAPH. DOI: 10.1145/325165.325242

[3] Three.js Documentation. InstancedMesh, LOD, Object3D, Animation System. threejs.org/docs và threejs.org/manual

Đánh giá từng phương pháp m_{ij} bằng E_{ij}

Các khía cạnh dưới đây chỉ là ví dụ; tùy p_i, E_{ij} có thể cần thêm nhiều khía cạnh khác.

E_{ij} = (dữ liệu, độ đo, quy trình, điều kiện)
(Data, metrics, test procedure, evaluation conditions)

Nhiều khía cạnh
(Multi-aspect)



Độ chính xác (Accuracy)
Độ đo: Tỷ lệ đúng (%), Recall@k, F1
Kiểm thử: *Bộ test có nhãn, đáp án chuẩn*

Độ trễ (Latency)
Độ đo: p50, p95 theo yêu cầu (ms)
Kiểm thử: *Cố định tải, phần cứng, cache*

Chi phí (Cost)
Độ đo: Chi phí / yêu cầu; token, CPU / GPU
Kiểm thử: *Cùng khối lượng việc và chất lượng*

Phương pháp m_{ij}
(Method under evaluation)

An toàn (Security)
Độ đo: Tỷ lệ tấn công thành công (ASR)
Kiểm thử: *Theo mô hình đe dọa*

Độ ổn định (Stability)
Độ đo: Độ biến thiên qua các lần chạy
Kiểm thử: *Lặp phép thử, nêu điều kiện lỗi*

… và khác (Others)
Mở rộng, công bằng, giải thích, năng lượng…
→ **Xem người khác đã đo p_i thế nào**



Cân nhắc đánh đổi
(Tradeoffs)

Muốn biết E_{ij} là gì: xem người khác đã đo bài toán tương tự như thế nào (Related work, benchmarks).

Cùng p_i: dữ liệu, độ đo và điều kiện tương đương; nêu đánh đổi chất lượng, chi phí, rủi ro (Comparable evaluation).

Diễn giải phương pháp E đánh giá S cũng vô cùng quan trọng

Ngay cả khi sử dụng chung một Metric, việc tính toán ra từng cấu phần cũng cần có sự diễn giải.

Prompt: điều kiện xem bảng điểm? (Question) → AI model → Response: đoạn văn tự do (Free text answer)

Đáp án chuẩn (Label) · *một đoạn văn, 3 câu*		**Câu trả lời (Response)** · *một đoạn văn, 3 câu*
L_1	⇔	R_1 Giảng viên được tra cứu điểm trong phạm vi lớp mình phụ trách.
L_2 Quyền truy cập áp dụng cho học kỳ hiện tại.		R_2 Việc truy cập phải dùng thiết bị đã đăng ký.
L_3 Thiết bị truy cập phải được đăng ký.		R_3 Hệ thống hiển thị thông tin dưới dạng bảng.

Cách 1: embedding cả đoạn
(Paragraph embedding similarity)

v_l = e(label) → Cosine ← v_r = e(response)

similarity(response, label) = $\cos(v_r, v_l)$

Cùng mô hình embedding cho cả hai đoạn.
(Same encoder for both paragraphs)

So toàn đoạn có thể che khuất ý thiếu hoặc ý sai.
(Whole paragraph similarity can hide errors)

Cách 2: trích và đối chiếu từng câu
(Sentence meaning matching)

Câu trong label (L_i)	Câu response khớp (R_j)
L_1: lớp được phân công	R_1: lớp mình phụ trách Khớp ý (Match)
L_2: học kỳ hiện tại	Không có câu tương ứng Ý thiếu (Missing)
L_3: thiết bị đã đăng ký	R_2: thiết bị đã đăng ký Khớp ý (Match)

R_3: ý bổ sung, chưa khớp label (Unmatched).
Response matching = 2/3. Label coverage = 2/3.

Similarity cần ngưỡng khớp phù hợp và kiểm tra mâu thuẫn trước khi suy ra Accuracy (Calibrated scoring).

FPT Education
FPT UNIVERSITY

Cách trình bày một nghiên cứu

Presenting a research

1. Nội dung và thứ tự hoàn thiện báo cáo
2. Tổng quan tài liệu
3. Phương pháp và lập luận thiết kế
4. Đánh giá: thiết lập, kết quả, lập luận
5. Kết luận
6. Định dạng báo cáo

PHẦN

02

Nội dung và thứ tự hoàn thiện báo cáo nghiên cứu

Báo cáo là **mỏ neo** chứ không phải sản phẩm cuối: được **chỉnh sửa liên tục** qua mọi giai đoạn để giữ context.
Khía cạnh nào bị **drift** khỏi P, M, S, E sẽ lộ ra ngay khi đối chiếu với báo cáo.

Thứ tự trong báo cáo (Report order) ↓

Thứ tự hoàn thiện (số = chương) →

1. Introduction

Bối cảnh, P, mục tiêu / RQ, phạm vi và đóng góp.
(Context, problem, objectives, scope, contributions)
P, mục tiêu, kết quả cuối (Problem, goals, outcomes)

2. Literature Review / Related Works

Kiến thức xác định **P** từ cơ bản đến chuyên sâu, cùng M và E_{ij}.
(Basics → taxonomy → mechanisms → context → gaps; methods, evaluation)
Ma trận P–M–E (Literature matrix, sources)

3. Methodology

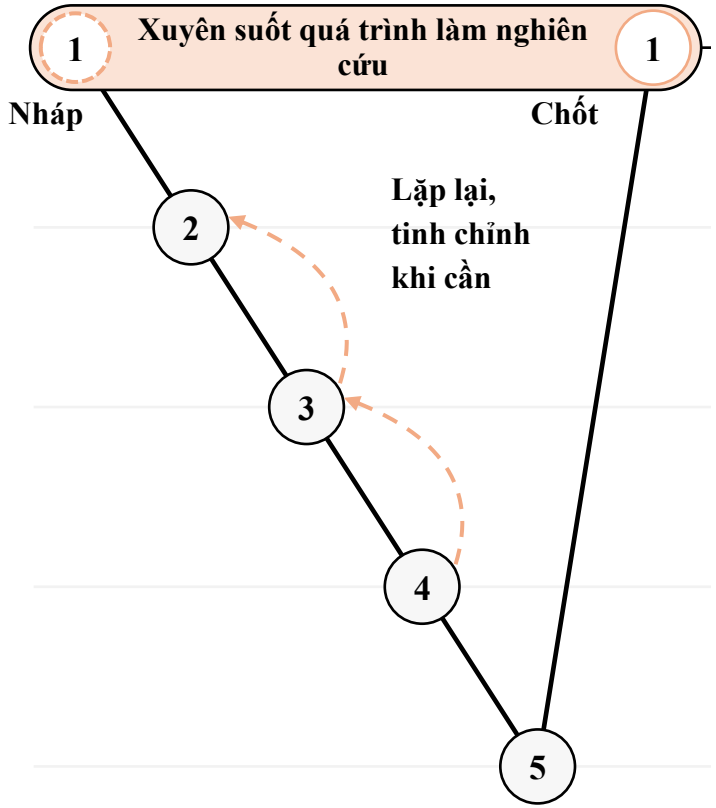
Thiết kế S giải từng p_i, lập luận và kế hoạch đánh giá E.
(Solution design, rationale and evaluation plan)
S={Design, code, data, config}

4. Evaluation & Discussion

Thiết lập thử nghiệm, kết quả, so sánh, đánh đổi, giới hạn.
(Experimental setup, results, trade offs, limitations)
Nhật ký, bảng, biểu đồ (Logs, tables, figures)

5. Conclusion

Mức đạt mục tiêu, đóng góp, hạn chế và hướng tiếp theo.
(Achievements, limitations and future work)
Kết quả, vấn đề còn lại (Results, remaining gaps)



Sau mỗi giai đoạn k *(after each phase)* → Cập nhật báo cáo → v_k *(report refinement)* → Đối chiếu P, M, S, E với *dữ liệu, kết quả, sản phẩm* → **Drift? Kéo về mỏ neo** → *sang giai đoạn k+1*

Viết Chương 2: Literature review / Related works

Tổng hợp theo bài toán và cơ chế (Thematic synthesis). Làm rõ phương pháp đã có và cách người khác đánh giá.
Đối chiếu bối cảnh để lập luận gap có bằng chứng (Evidence based gap analysis).

2.1 Kiến thức nền *(Basics)*	→	2.2 Phân loại *(Taxonomy)*	→	2.3 Cơ chế *(Mechanisms)*	→	2.4 Bối cảnh *(Context)*	→	2.5 Khoảng trống *(Gaps)*
RAG, đoạn tài liệu, embedding và độ đo. Định nghĩa phạm vi.		Sparse, dense, hybrid. Phân nhóm theo tiêu chí rõ ràng và nhất quán.		Đầu vào, các bước, đầu ra. Nêu giả định và giới hạn. Vì sao có thể hiệu quả?		Dữ liệu, ngôn ngữ, model. Tải và các ràng buộc. Điều kiện áp dụng.		Fit gaps trong ứng dụng. Hạn chế cần cải tiến. Gắn gaps với P và RQ.

Bảng 2.1. Ma trận P, M, E và nguồn (Literature matrix). Ví dụ RAG với kho tài liệu tiếng Việt.

P: bài toán *(Problem)*	**M: phương pháp và cơ chế** *(Method / mechanism)*	**E: đánh giá trong nguồn** *(Reported evaluation)*	**Đối chiếu context và gaps** *(Applicability analysis)*	**Nguồn** *(Source)*
p_1 **Truy hồi bằng chứng (Retrieval)**	m_{11} DPR: hai encoder, so vector câu hỏi và đoạn (Dual encoder).	Top k retrieval accuracy. NQ, TriviaQA. So với BM25.	Kho tiếng Việt, ít nhãn: kiểm tra chuyển miền, chi phí huấn luyện và truy hồi.	**[1]** Karpukhin et al. (2020)
p_1 **Truy hồi bằng chứng (Retrieval)**	m_{12} HyDE: sinh đoạn giả định, embed đoạn để tìm tài liệu thật (Hypothetical document).	BEIR: nDCG@10, Recall@100. DL19/20: nDCG@10, Recall@1k.	Ít nhãn là điều kiện phù hợp. Cần đo thêm độ trễ và chi phí của bước sinh (Latency / cost).	**[2]** Gao et al. (2023)
p_2 **Sinh câu trả lời có căn cứ (Generation)**	m_{21} RAG: truy hồi tài liệu, rồi điều kiện hóa bộ sinh (Retrieval and generation).	QA: Exact Match (EM). NQ, TriviaQA, WebQuestions. Đối chứng và ablation.	Câu trả lời dài cần đo thêm độ hỗ trợ từng ý (Faithfulness) và độ bao phủ (Coverage).	**[3]** Lewis et al. (2020)

Lập luận gap (Gap argument): benchmark gốc chưa đủ xác nhận hiệu quả trong kho tiếng Việt đã chọn.
Gap dự kiến cần xác nhận bằng baseline và dữ liệu của context. **E bổ sung:** faithfulness, coverage, p95 latency, cost.

[1] Karpukhin et al. (2020). Dense Passage Retrieval for Open Domain Question Answering. EMNLP, 6769–6781.

[2] Gao et al. (2023). Precise Zero Shot Dense Retrieval without Relevance Labels. ACL, 1762–1777.

[3] Lewis et al. (2020). Retrieval Augmented Generation for Knowledge Intensive NLP Tasks. NeurIPS. arXiv:2005.11401.

Mỗi hàng ghi nguồn, vị trí evidence và điều kiện đánh giá. E gốc và E bổ sung cần phân biệt (Source traceability).

Viết Chương 3: Trình bày phương pháp và lập luận thiết kế

Mỗi quyết định thiết kế cần nêu rõ: giải quyết gap nào, trả lời RQ nào, thực hiện RO nào và kiểm chứng bằng gì.
Kiến trúc, thành phần và thuật toán (Architecture, components, algorithms) phải có cơ chế tác động cụ thể.

Ví dụ chatbot RAG (Illustrative example)

Khoảng trống (Research gaps)	**Câu hỏi** (Research questions)	**Mục tiêu** (Research objectives)	**Thành phần và thuật toán** (Components / algorithms)	**Bằng chứng** (Evaluation)
G_1 Bỏ sót bằng chứng với câu hỏi diễn đạt khác tài liệu.	**RQ_1** Xếp hạng lại có cải thiện truy hồi với độ trễ cho phép?	**RO_1** Tăng Recall@10 so với đối chứng B. p95 latency ≤ 3 s.	**C_1 Bộ xếp hạng lại (Reranker)** A_1 Chấm điểm câu hỏi với đoạn, chọn 10 đoạn có điểm cao nhất. *Cơ chế: ưu tiên bằng chứng liên quan.*	**E_1** Recall@10 p95 latency So sánh B, S, S bỏ C_1 *(Ablation)*
G_2 Câu trả lời có ý không được tài liệu hỗ trợ.	**RQ_2** Kiểm chứng có giảm ý thiếu căn cứ và giữ độ bao phủ?	**RO_2** Giảm tỷ lệ ý thiếu căn cứ. Giữ coverage ≥ 80%.	**C_2 Bộ kiểm tra (Evidence verifier)** A_2 Ghép ý với nguồn, đo độ hỗ trợ. Thiếu hỗ trợ: loại ý hoặc từ chối. *Cơ chế: chặn ý thiếu căn cứ.*	**E_2** Ý thiếu căn cứ Coverage So sánh B, S, S bỏ C_2 *(Ablation)*

Đóng góp đề xuất: C_1, C_2

Kiến trúc đề xuất S (Proposed architecture)

Câu hỏi *(Query)* → Truy hồi, C_1 xếp hạng lại *(Retrieval and reranking)* → Sinh câu trả lời *(Generation)* → **C_2 Kiểm tra ý với nguồn** *(Evidence verification)* → Phản hồi *(Response)*

C_1 / A_1 gắn với G_1, RQ_1, RO_1

C_2 / A_2 gắn với G_2, RQ_2, RO_2

Đánh giá cùng dữ liệu và câu hỏi (Comparable evaluation). Loại từng thành phần để kiểm tra đóng góp (Ablation).

Ví dụ giả định (Hypothetical): gaps cần xác nhận trước. Ngưỡng 3 s và 80% là mục tiêu minh họa. B là RAG cơ bản, S là thiết kế đề xuất.

Viết Chương 3: Chi tiết từng thành phần

Với đồ án IT đại học, mô tả đủ để hiểu cơ chế, lập luận vì sao thiết kế phù hợp và tái lập cách triển khai.
Mỗi phần tiếp tục gắn với khoảng trống (Gaps), câu hỏi nghiên cứu (RQ) và mục tiêu nghiên cứu (RO).

Sáu nội dung cần làm rõ

1. Đặc tả bài toán (Problem formulation)
Đầu vào q, người dùng u, kho D. Đầu ra gồm câu trả lời và nguồn.
Nêu giả định, ràng buộc, điều kiện chấp nhận hoặc từ chối.

2. Dữ liệu và giao diện (Data and interfaces)
Mô tả cách tạo đoạn, metadata, phiên bản và quyền truy cập.
Có lược đồ dữ liệu, hợp đồng API và truy vết nguồn.

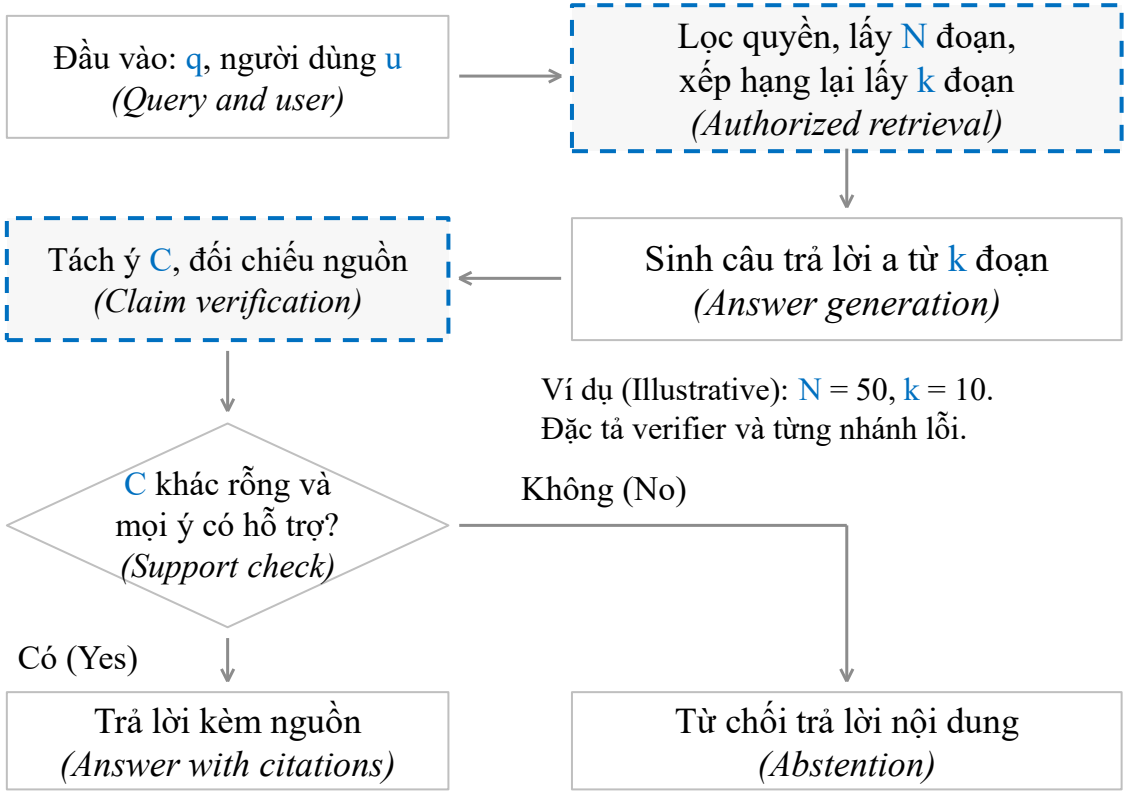
3. Thuật toán và cơ chế (Algorithms)
Viết giả mã, hàm chấm điểm, tham số và điều kiện dừng.
Chỉ rõ từng thành phần tác động tới bài toán con thế nào.

4. Lập luận lựa chọn (Design rationale)
Đối chiếu phương án thay thế với bối cảnh và ràng buộc.
Nêu chi phí xử lý, bộ nhớ và sự đánh đổi (Tradeoffs).

5. Lỗi và an toàn (Failures and security)
Nêu cách xử lý khi thiếu dữ liệu, hết thời gian hoặc sai quyền.
Mô tả nhánh dự phòng (Fallback) và kiểm soát truy cập.

6. Triển khai và tái lập (Implementation)
Ghi cấu hình, phiên bản model, prompt và cách ghi log.
Phân biệt phần kế thừa, phần điều chỉnh và phần đề xuất mới.

Ví dụ RAG: từ kiến trúc xuống luồng xử lý



Minh chứng trong chương: sơ đồ luồng (Activity diagram), mô hình dữ liệu, giả mã, cấu hình và lập luận thiết kế.
Chương đánh giá (Evaluation) sẽ kiểm chứng độ đúng, độ trễ và đóng góp của từng thành phần (Ablation).

Viết Chương 4 : Thiết lập đánh giá, kết quả và lập luận

Thiết lập thí nghiệm cần đủ để tái lập (Reproducibility). Mỗi kết quả cần dẫn tới kết luận về RO, RQ và gap, đồng thời giải thích đóng góp, sự đánh đổi và giới hạn (Contributions, tradeoffs and limitations).

4.1 Thiết lập thí nghiệm (Experimental setup)

Dữ liệu và nhãn chuẩn (Data and ground truth)
50 câu hỏi dev, 200 câu hỏi test.
Gán nhãn bằng chứng và ý đúng.
Khóa test, chọn ngưỡng trên dev.

Cấu hình và đối chứng (Configuration)
B, S, S bỏ C_1, S bỏ C_2 (Ablation).
Cùng corpus, model và phần cứng.
Ghi phiên bản, seed, prompt và tải.

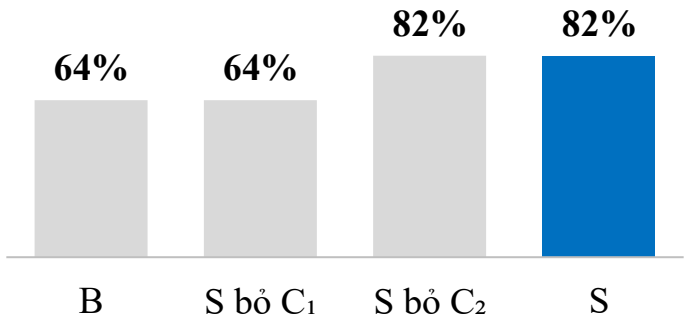
Độ đo và tiêu chí đạt (Metrics and criteria)
RO_1: Recall@10 tăng, p95 ≤ 3 s.
RO_2: ý thiếu căn cứ giảm,
coverage ≥ 80%. Nêu rõ mẫu số.

Quy trình và độ tin cậy (Protocol)
Lặp chạy, báo độ bất định (CI).
Phân tích lỗi và phạm vi áp dụng.

4.2 Kết quả và thảo luận (Results and discussion)

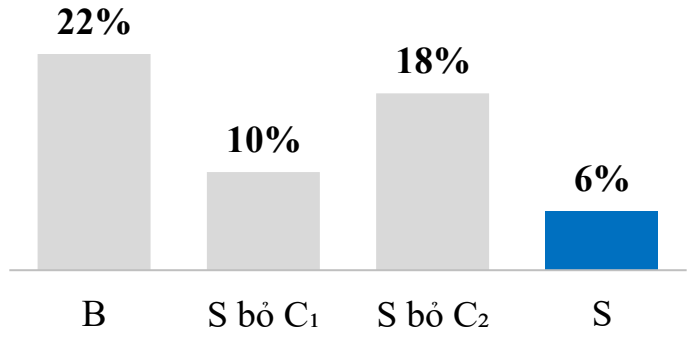
RAG: số liệu giả định (Hypothetical). C_1: reranker. C_2: verifier.

Recall@10, theo cấu hình



S: p95 latency = 2,7 s ≤ 3 s

Ý thiếu căn cứ, theo cấu hình



S: coverage = 84% ≥ 80%

RO_1 Đạt trong ví dụ. Recall@10: 64% lên 82%. Độ trễ đáp ứng ngưỡng.	→ **RQ_1 Có cải thiện truy hồi** với độ trễ cho phép trong cấu hình đã thử.	→ **G_1 Giảm bỏ sót bằng chứng.** Vẫn còn bằng chứng chưa truy hồi được.
RO_2 Đạt trong ví dụ. Ý thiếu căn cứ: 22% còn 6%. Coverage đáp ứng ngưỡng.	→ **RQ_2 Giảm ý thiếu căn cứ** và giữ độ bao phủ trên ngưỡng yêu cầu.	→ **G_2 Giảm ý thiếu căn cứ.** Còn 6%, chưa giải quyết hoàn toàn.

Phát hiện chính (Key insights): C_1 tăng Recall@10 thêm 18 điểm %. C_2 giảm ý thiếu căn cứ từ 18% xuống 6%, đồng thời coverage giảm từ 100% xuống 84%. Đối chiếu cả lợi ích và sự đánh đổi (Tradeoff).

Viết Chương 5: kết luận

Tổng hợp đóng góp có bằng chứng, nêu giới hạn của kết luận và chỉ ra việc cần làm tiếp.
Kết nối lại bài toán P, khoảng trống (Gaps), câu hỏi nghiên cứu (RQ) và mục tiêu nghiên cứu (RO).

5.1 Ưu điểm: đã làm được gì?
(Achievements)

Nêu đóng góp chính, RO đạt đến đâu, RQ được trả lời và gap giảm đến đâu.
Dẫn lại bằng chứng từ chương đánh giá.

5.2 Nhược điểm và điểm chưa rõ
(Limitations and open questions)

Nêu thiếu sót, sự đánh đổi (Tradeoffs), phạm vi áp dụng và điều chưa kiểm chứng.
Giải thích tác động tới kết luận.

5.3 Hướng nghiên cứu tiếp theo
(Future work)

Mỗi hướng xuất phát từ một giới hạn. Nêu sẽ làm gì, vì sao và đo bằng gì.
Ưu tiên phần còn cản trở mục tiêu.

Ví dụ chatbot RAG: *toàn bộ số liệu dưới đây là giả định minh họa (Hypothetical).*

Đóng góp đã được chứng minh trong phạm vi ví dụ

RO_1: Recall@10 từ 64% lên 82%. p95 latency 2,7 s ≤ 3 s.

RO_2: ý thiếu căn cứ từ 22% còn 6%. Coverage 84% ≥ 80%.

RQ_1, RQ_2 được trả lời trong cấu hình đã thử. **G_1, G_2** được cải thiện.

L_1 Vẫn còn 6% ý thiếu căn cứ. Coverage 84%: còn 16% câu hỏi không được trả lời nội dung.	**F_1** Cải tiến và hiệu chỉnh verifier. Đo ý thiếu căn cứ cùng coverage, độ đúng và độ đầy đủ câu trả lời.
L_2 Chưa rõ tác động của việc từ chối câu hỏi khó tới mức giảm lỗi. Không quy toàn bộ lợi ích cho verifier.	**F_2** So sánh trên tập câu hỏi chung, phân tích lỗi và sự đánh đổi giữa giảm lỗi với tăng từ chối.
L_3 Chưa rõ khả năng khái quát sang miền, model hoặc tải khác. Độ bất định chưa được lượng hóa.	**F_3** Mở rộng miền, model và tải thử. Lặp thí nghiệm, báo khoảng tin cậy và độ trễ, chi phí theo từng bối cảnh.

Cách kết luận: RO_1 và RO_2 đạt theo tiêu chí trong phạm vi đã thử; các gap được giảm nhưng chưa khép lại hoàn toàn.
Phần việc tiếp theo cần làm rõ độ tin cậy và khả năng áp dụng của phương pháp.

Giữ nhất quán với Introduction và Evaluation. Không thêm kết quả mới; không tuyên bố vượt quá bằng chứng.

Lưu ý định dạng khi viết báo cáo (Report formatting)

1. Phân cấp tiêu đề
(Heading hierarchy)

Thường đến cấp 4 là đủ: 1.1.1.1.
Dùng Heading Styles và đánh số tự động, giữ thứ bậc nhất quán.
(Styles and automatic numbering)

1. Tổng quan
Heading 1

1.1. Chatbot RAG
Heading 2

1.1.1. Truy hồi
Heading 3

1.1.1.1. Truy hồi vector
Heading 4

Mở đoạn	Nội dung chính	Kết đoạn

Mỗi mục viết thành **đoạn văn học thuật**, không gạch đầu dòng: báo cáo truyền đạt tri thức, không phải slide.
(Academic prose, not bullet points)
Tên mục rõ nội dung, không chia quá nhỏ.

2. Bảng và hình
(Tables and figures)

Mỗi bảng, hình có số và caption.
Dẫn số trong body text và giải thích nội dung liên quan đến lập luận.
(Captions and cross references)

Bảng 3.1. Phương pháp truy hồi
(Table 3.1. Retrieval methods)

Phương pháp (Method)	Độ đo (Metric)
BM25	Recall@k
Dense retrieval	Recall@k

Bảng 3.1 liệt kê các phương pháp và độ đo đánh giá truy hồi.
(Example body text)



Hình 3.1. Luồng xử lý RAG
(Figure 3.1. RAG pipeline)
Hình 3.1 cho thấy bước truy hồi đứng trước bước sinh câu trả lời.

3. Tài liệu tham khảo
(References)

Zotero / Mendeley
Quản lý thông tin nguồn và PDF, chèn trích dẫn, tạo danh mục.
Dùng một chuẩn trích dẫn nhất quán.
(Reference management)

OneNote: tổ chức ghi chú đọc, liên kết nguồn và ý tưởng nghiên cứu.
(Research notes)

Ví dụ dẫn nguồn (In text citation)

Transformer sử dụng cơ chế attention [1].

Danh mục (Bibliography)

[1] A. Vaswani et al., Attention Is All You Need, arXiv:1706.03762, 2017.

Kiểm tra tác giả, năm, tiêu đề, DOI / URL trước khi trích dẫn.
(Verify source metadata)

Cấp tiêu đề, vị trí caption và citation style tuân theo mẫu của đơn vị (Institutional template).