



NATIONAL ECONOMICS UNIVERSITY

ICETAI 2026

INTERNATIONAL CONFERENCE ON EDUCATIONAL TECHNOLOGY AND ARTIFICIAL INTELLIGENCE

TriGen: A Unified RAG Pipeline for Multi-Format Learning Content Generation from Textbooks

Pham Xuan Lam · Nguyen Diep Anh · Nguyen Thi Huyen · Tran Thi My Diep

*Presented by **Nguyen Diep Anh** · National Economics University, Hanoi, Vietnam*



What this talk covers

01

Motivation

Two bottlenecks that keep chat-based LLM output out of the classroom.

02

Related work

Where TriGen sits versus prior RAG and question-generation systems.

03

Method

Structure-aware chunking, hierarchical indexing, retrieval, and three generation heads.

04

System in practice

What TriGen output looks like inside a real learning platform.

05

Deployment & results

A first-cohort analysis on a live university platform.

06

Limitations & conclusion

What's proven, what's deferred, and where this goes next.

Two bottlenecks block the classroom

Active recall (MCQs), spaced repetition (flashcards) and knowledge organization (mindmaps) beat passive study — yet chat-based LLM output stalls before it reaches learners.



Output is view-only

Generated items live in a chat window.

No scoring for MCQs. No spaced-repetition schedule for flashcards. No course navigation for mindmaps — because no learning platform consumes the output.



Prompt-competence gap

Learners can't prompt what they don't yet know.

Enforcing item-writing rules, balancing Bloom levels, imposing flashcard atomicity and guarding against hallucination are exactly the skills a student is still acquiring.

Where TriGen sits in the literature



Retrieval-Augmented Generation

- Lewis et al. introduced RAG: a dense retriever paired with a seq2seq generator [7].
- Structure-aware / graph RAG preserves document layout for higher-quality context [8].
- Self-RAG and agentic RAG interleave retrieval with reasoning and self-critique [16, 9].

TriGen's angle: chapters/sections as first-class retrieval units — for both chunk boundaries and hierarchical query filtering.



AI-generated learning materials

- MCQ generation pre-dates LLMs; weak distractors remain a recurring weakness [6, 10, 11].
- Flashcard generation is under-studied — mostly prompting with surface-level metrics [12].
- Mindmap generation is studied separately, as concept-map extraction [13].

TriGen's angle: the first to unify all three artifacts under one retrieval/indexing pipeline with a shared evaluation framework.

What we contribute



01

A unified RAG pipeline

Prompts encode item-writing, atomicity and coverage rules learners can't apply themselves — producing MCQs, flashcards and mindmaps from one retrieval backbone.



02

A first-cohort deployment analysis

Four Vietnamese textbooks, programming and non-programming: approval rates, option-quality vs. teacher items, atomicity, mindmap coverage, generation cost.

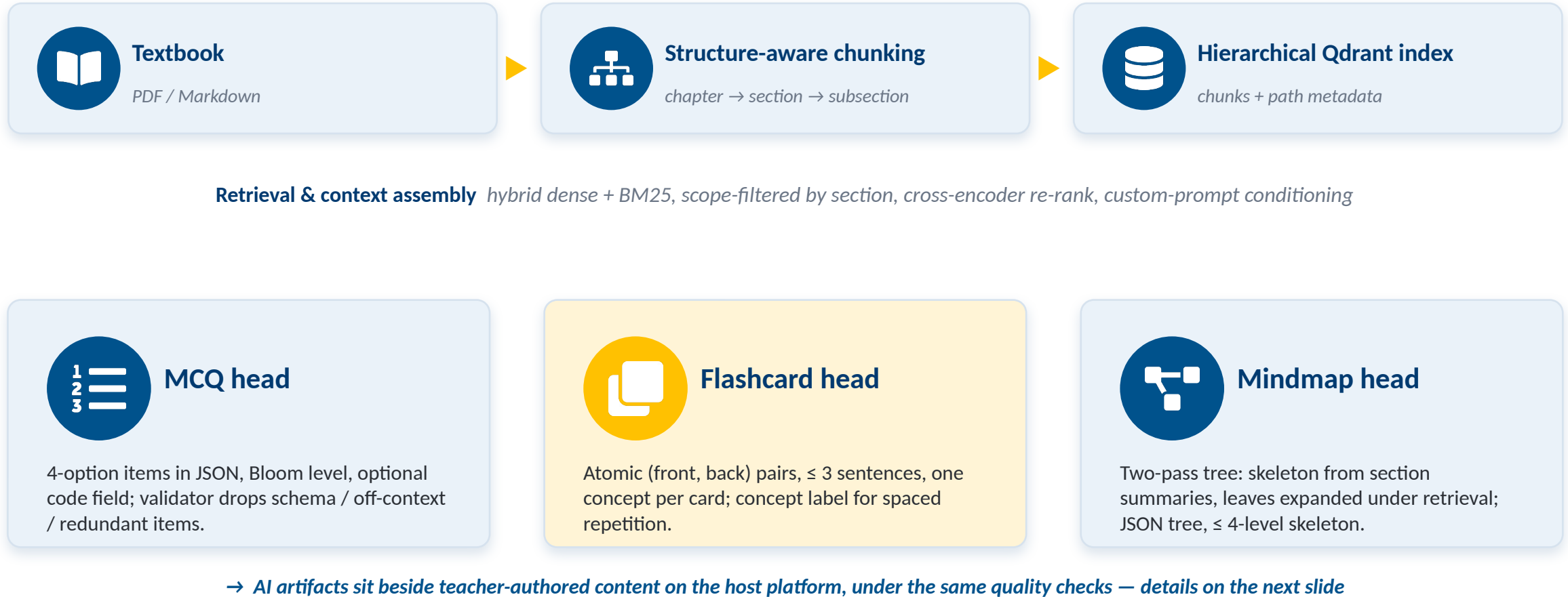


03

Structure-aware chunking & evaluation

Chapter/section/subsection hierarchy preserved through chunking and retrieval, paired with a framework combining retrieval fidelity and pedagogical quality.

The TriGen pipeline



Three generation heads



MCQ head

- 4-option JSON: stem, correct answer, 3 distractors, Bloom level, optional code field.
- 3 prompt rules: plausible distractors, no semantic overlap, shared surface form.
- Validator drops schema failures, off-context claims, or Jaccard > 0.8 option pairs.



Flashcard head

- Atomic (front, back) pairs — one definition, theorem or rule per card.
- Back side capped at 3 sentences by the prompt.
- Concept label emitted for future spaced-repetition scheduling.



Mindmap head

- Two-pass generation: skeleton from section summaries, then per-node expansion.
- Skeleton capped at 4 levels (Cognitive Load Theory).
- Post-expansion depth reaches ≈ 5.9 levels on the observed cohort.

Structure-aware chunking

Naïve fixed-window chunking (e.g. 512 tokens, 50-token overlap) fragments definitions across chunks and interleaves unrelated topics — hurting both retrieval precision and downstream quality in all three generation heads.



Extract a logical tree — PDF outline + font-size heuristics + section markers, or Markdown heading levels for .md/.txt sources.



Chunk only inside leaf nodes — DFS traversal; soft ≈ 700 -token target with sentence-aware boundaries — never splits mid-sentence.



Inherit the full hierarchical path — book $\rightarrow$ chapter $\rightarrow$ section $\rightarrow$ subsection travels with every chunk as metadata.



Normalise chapter identifiers — canonical Chapter-X form keeps Qdrant payload filters stable across textbooks with inconsistent labelling.

This hierarchy is the shared backbone: every retrieval query and all three generation heads depend on the same chunk metadata — get it wrong here, and the error propagates downstream.

Hierarchical chunk metadata

Textbook

Chapter-3

3.2 Section

3.2.1 Subsection

chunk · ≈ 700 tokens

3.3 Section

chunk · ≈ 700 tokens

Hierarchical indexing & retrieval



Hierarchical vector indexing

- Chunk-level embeddings carry parent chapter and section ids as Qdrant payload fields.
- A separate collection of section-level summaries serves as a coarse retrieval layer for mindmaps.
- Cosine-similarity deduplication across textbook editions avoids retrieval collapse.



Retrieval & context assembly

- A hybrid query — dense embedding + BM25 — is scope-filtered to the requested section.
- A cross-encoder re-ranks the top candidates before context assembly.
- Custom prompts condition retrieval: salient terms are appended to both queries.

Context carries explicit chapter/section headers, so the generator can attribute each fact to its source — improving faithfulness.

System in practice

The screenshot displays the ScoreUp dashboard for a student. The left sidebar contains navigation links for 'Sinh viên' (Student) and 'Giáo viên' (Teacher). The main content area is titled 'Tóm tắt luyện tập' (Practice Summary) and includes three summary cards: 'LƯỢT ĐÃ LÀM' (4), 'MÔN ĐÃ ÔN' (3), and 'ĐỘ CHÍNH XÁC TB' (25%). Below these is a table of recent practice activities with columns for 'Thời gian', 'Môn học', 'Điểm', and 'Chi tiết'.

Thời gian	Môn học	Điểm	Chi tiết
10:58, 21/04/2026	Lập trình Java	0/52	Xem
09:43, 11/04/2026	Lập trình Java	0/52	Xem
14:23, 10/04/2026	Cơ sở lập trình	0/70	Xem
22:11, 05/04/2026	Nhập môn Công nghệ thông tin	10/10	Xem

Below the table is a section titled 'Lịch sử theo môn học' (History by subject) with three cards for 'Nhập môn Công nghệ thông tin', 'Cơ sở lập trình', and 'Lập trình Java', each showing the time taken and score.

Scored, not view-only

Practice history and progress tracking — the bottleneck chat-based output cannot offer (Slide 3).

System in practice



Gợi ý

Thi

Studio BETA

Giáo viên

Lớp học

Studio BETA

Quản lý ca thi

Quản lý câu hỏi

Quản lý

Bảng điều khiển

Quản lý môn học

Quản lý khoa/viện

Quản lý lớp học

Quản lý tài liệu

Nội dung ca thi

Loại ca thi *

☐ Câu hỏi ngẫu nhiên

☒ Bộ câu hỏi đã tạo

Bộ câu hỏi *

Cơ sở lập trình 13 05 2026

Số lượng câu hỏi *

VD: 40

Thông tin ca thi

Tên ca thi *

VD: Thi giữa kì CSLT lớp 03,...

Thời gian bắt đầu

yyyy-mm-dd --:-- --

Thời gian kết thúc

yyyy-mm-dd --:-- --

Nếu không chọn thì ca thi lúc nào cũng làm được

Thời gian thi (phút)

Bảo mật & Giám sát

Mật khẩu ca thi (tùy chọn)

Nhập mật khẩu

☒ Bật chức năng giám sát thi

Tự động cảnh báo nếu thí sinh thoát toàn màn hình hoặc chuyển tab

☐ Vô hiệu hóa bàn phím

Ngăn thí sinh sử dụng bàn phím khi làm bài

☐ Ẩn điểm số sau khi nộp bài

Không hiển thị kết quả và điểm số trên trang kết quả

☐ Chế độ luyện tập

Cho phép sinh viên làm lại không giới hạn số lần; điểm chính thức lấy theo lần cao nhất

Danh sách sinh viên

Tải danh sách sinh viên

Reviewed items reach real exams

Instructors schedule live assessments drawing from the same approved item pool.

ICETAI 2026 · TriGen: A Unified RAG Pipeline

System in practice

Câu 12: This program has a bug. What is the problem?

▶ Chạy thử

Copy ↺ ≡

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int n = 5
5     printf("%d\n", n);
6     return 0;
7 }
```

Bình luận (0)

(A) printf needs a & before n

(B) Missing semicolon after int n = 5

(C) main must return void

(D) #include should use angle brackets only for user headers

✓ Thành thạo

Đánh dấu

⚠ Báo sai

The code field, live

Programming-content MCQs render an accompanying code snippet in a syntax-highlighted editor.

System in practice



Tiến độ làm bài

00:23:48

0102030405

0607080910

1112131415

1617181920

2122232425

2627282930

Nộp bài

Test

Môn thi: Lập trình Java

8/27/2026, 10:15:00 PM

Thông tin sinh viên

Họ và tên: Nguyễn Diệp Anh

Mã sinh viên: 11220315

Hủy đánh dấu

Câu 11: Khi nào nên sử dụng phương thức `orElse()` của `Optional`?

(A) Khi cần ném ngoại lệ nếu giá trị không tồn tại

(B) Khi cần trả về một giá trị mặc định nếu `Optional` trống

(C) Khi cần kiểm tra sự tồn tại của giá trị

(D) Khi cần chuyển đổi `Optional` sang kiểu dữ liệu khác

Đánh dấu

Câu 12: Phương thức nào của `Set<E>` chỉ giữ lại các phần tử có trong bộ sưu tập được chỉ định?

(A) `retainAll(Collection<E> coll)`

(B) `removeAll(Collection<E> coll)`

Exams, graded instantly

A learner sitting a live exam that is drawn from the same approved item pool as teacher-authored questions.

System in practice

The screenshot displays the ScoreUp application interface. On the left is a sidebar with navigation options: Dashboard, Bảng xếp hạng, Luyện tập, Gợi ý, Thi, and Studio (marked BETA). Below these are sections for 'Giáo viên' (Lớp học, Studio BETA, Quản lý ca thi, Quản lý câu hỏi) and 'Quản lý' (Bảng điều khiển, Quản lý môn học). The main area is titled 'EP17.CNTT1128 - Cơ sở lập trình' and shows a flashcard titled 'Flashcard Chương 9 - Tiếng Anh - medium' with 15 cards. The current card asks 'What are the common file modes in C?'. Below the question is a prompt box: 'Phím tắt Nhấn [phím cách] hoặc nhấp vào thẻ để lật'. At the bottom of the card, there are two buttons: 'ĐANG HỌC 0' (red) and 'ĐÃ BIẾT 1' (green). A progress bar at the bottom indicates '1 / 15 thẻ'. On the right side, there is a 'Công cụ' (Tools) section with 'Thẻ ghi nhớ' and 'Mindmap' buttons. Below that is a 'DANH SÁCH HỌC LIỆU' (Study List) section with 26 items, including 'Mindmap Chapter 1 - English versi...', 'Mindmap Chương 1', 'Flashcard Chương 9 - Tiếng Anh - ...', 'Flashcard Chương 8 - Tạo thẻ - m...', and 'Flashcard Chương 7 - Thẻ tiếng A...'. Each item has a date and a status icon.

Create, view and edit flashcards

Instructors create flashcards based on a chapter. Possible to specify topics or concepts within that chapter using prompt.

System in practice

The screenshot displays the ScoreUp application interface. On the left is a sidebar with navigation options: Dashboard, Bảng xếp hạng, Luyện tập, Gợi ý, Thi, and Studio (marked BETA). Below this is a section for 'Giáo viên' (Teacher) with options like Lớp học, Studio (BETA), Quản lý ca thi, and Quản lý câu hỏi. At the bottom is a 'Quản lý' (Management) section with options like Bảng điều khiển and Quản lý môn học.

The main content area is titled 'EP17.CNTT1128 - Cơ sở lập trình'. It features a 'Mindmap Chương 4' (Chapter 4 Mindmap) with a central node 'Chương 4' and several branches: '4.1. Thư viện chuẩn vào ra trong C', '4.2. Các hàm nhập dữ liệu từ bàn phím', '4.3. Các hàm xuất dữ liệu ra màn hình', '4.4. Một số chương trình ví dụ', and '4.5. Câu hỏi ôn tập và bài tập'. The '4.2' branch further sub-divides into '4.2.1. Hàm scanf' and '4.2.2. Một số hàm nhập dữ liệu khác gets, getch, getchar'.

On the right side, there is a 'Công cụ' (Tools) section with buttons for 'Thẻ ghi nhỏ' and 'Mindmap'. Below that is a 'DANH SÁCH HỌC LIỆU' (List of Learning Materials) section with 26 items, including 'Mindmap Chương 4', 'Mindmap Chương 3', and various English versions of chapters.

Create, view and edit mindmaps, live

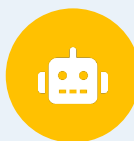
Instructors can create mindmaps for a chapter, a few chapters, or even the whole textbook

Experimental setup



Qwen3-8B via Ollama

primary generator across all three heads



GPT-4o

used for a subset of mindmaps, to inspect provider variance



Multilingual sentence-transformer

fine-tuned for Vietnamese semantic similarity (embeddings)

Four textbooks, mixed domain

Java Programming

Programming

Programming Fundamentals

Programming

IT Project Management

Project management

Population, Resources and Environment in Sustainable Development

Non-programming (generalization probe)

Integrated beside 1,209 existing teacher-authored / imported MCQs on the same platform · exports anonymized via HMAC-keyed research IDs.

Deployed on a live university platform

Qwen3-8B via Ollama as the primary generator (GPT-4o for a subset of mindmaps) · multilingual sentence-transformer embeddings · integrated beside 1,209 existing teacher-authored / imported MCQs



236

active learners



636

MCQ responses (60-day peak window, 2026)



4

Vietnamese textbooks (mixed domain)

First cohort of TriGen-generated artifacts

790

MCQs

414

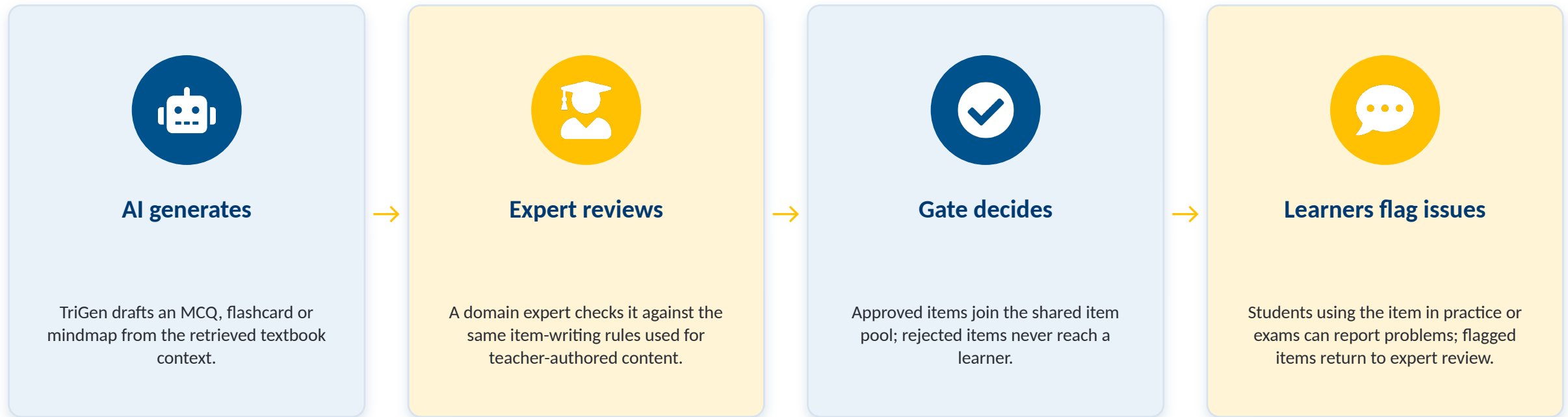
flashcards

14

mindmaps



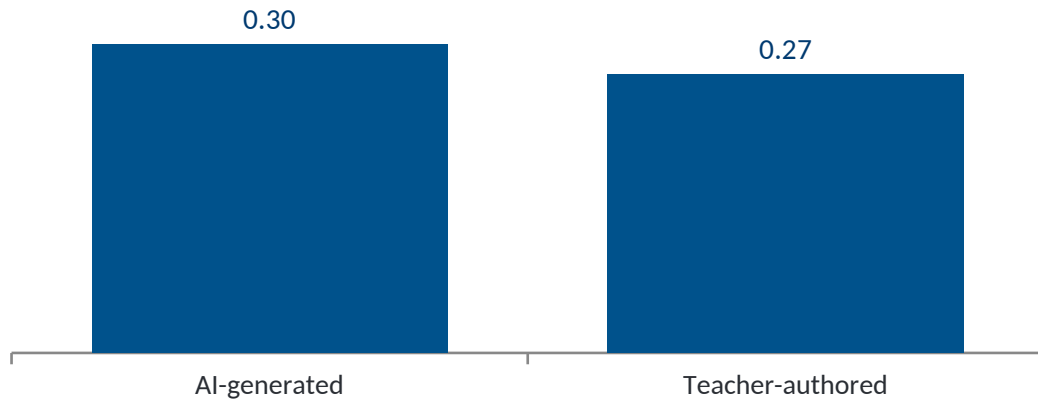
Every AI item passes expert review



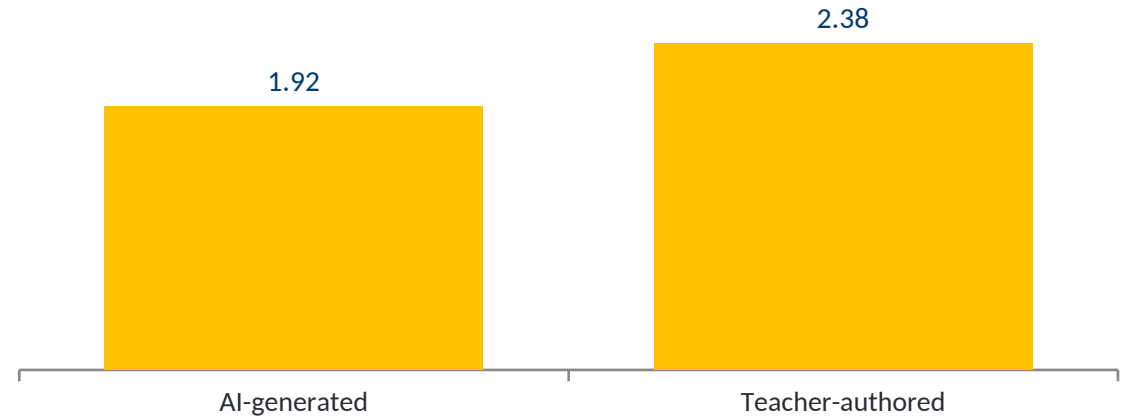
No AI item bypasses this gate, and review continues after deployment: approval is a property of the platform's shared quality pipeline, not of any single textbook or generation run — see Slide 16 for what we can and cannot yet quantify about review outcomes.

MCQ structural quality

Mean max-Jaccard overlap (4 options)



Option-length ratio (max / min chars)



3.8% of AI items exceed the engineered 0.8 redundancy threshold, versus **2.4%** of teacher-authored items — the validator already rejects items above that threshold before publication.

Caution: structural / format-compliance indicators only — not yet full pedagogical quality (factual correctness, distractor discrimination deferred to future work).

Flashcards & mindmaps



Flashcards (n = 414)

99.8%

comply with the ≤ 3 -sentence back-side constraint

91.5%

have a single sentence on the back (median 54 chars)

Atomicity is enforced at generation time by the prompt — prompt-encoded item-writing rules translate directly into compliance.



Mindmaps (n = 13 scored)

~19%

mean structural coverage per chapter-scoped tree (40.7% Java → 11.1% Prog. Fund.)

6.0

mean full-tree depth (levels)

66k / 11.4m

tokens & minutes per mindmap (transparent cost)

Partial per-tree coverage motivates an explicit multi-mindmap composition strategy for full-book coverage.

Limitations & ongoing work



Deployment-in-progress

- A flat-chunking baseline is specified but not yet run in parallel.
- A faculty-curated ground-truth corpus for the RAGAs evaluation is still under construction.
- Per-item response samples remain small — item-level psychometrics are deferred.



Where we go next

- Close the loop: feed learner responses back into generation parameters.
- A code-fidelity evaluation on the programming subset (code field).
- Cross-lingual (English) generation, evaluated beyond the Vietnamese cohort.

Conclusion

01

A unified, structure-aware pipeline

TriGen produces MCQs, flashcards and mindmaps from a shared retrieval backbone, deployed on NEU's live ScoreUp platform — 236 learners and 636 MCQ responses in a 60-day window.

02

A first-cohort analysis that holds up

Across four Vietnamese textbooks: 47.7% expert MCQ approval with sharp per-textbook variance; option-quality on par with teacher-authored items; 97.6% single-sentence flashcard atomicity; ~20% mindmap coverage per tree.

03

Larger claims, scoped honestly

A flat-chunking ablation, a RAGAs evaluation, a learner-cohort psychometric study, and a code-fidelity evaluation are ongoing work, not yet reported here.

References

11 core sources cited in this talk — full 23-reference list in the paper

- [6] Bitew et al. — Learning to reuse distractors. IEEE TLT (2024)
- [7] Lewis et al. — Retrieval-augmented generation for NLP. NeurIPS (2020)
- [8] Edge et al. — From local to global: a graph RAG approach. arXiv (2024)
- [9] Singh, Shukla & Mehrotra — Agentic RAG: a survey. arXiv (2025)
- [10] Kurdi et al. — Systematic review of automatic question generation. IJAIED (2020)
- [11] Alhazmi et al. — Distractor generation: a survey. EMNLP (2024)
- [12] Lohr et al. — LLMs for course-specific learning objects. arXiv (2024)
- [13] Wang et al. — Concept maps from textbooks. CIKM (2016)
- [14] Haladyna, Downing & Rodriguez — MC item-writing guidelines. AME (2002)
- [15] Sweller, van Merriënboer & Paas — Cognitive architecture. EPR (2019)
- [16] Asai et al. — Self-RAG: learning to retrieve & critique. ICLR (2024)

Thank you

Presented by [Nguyen Diep Anh](#) · Pham Xuan Lam · Nguyen Thi Huyen · Tran Thi My Diep — National Economics University

