Interactive Textbook Learning with Retrieval-Augmented Generation: The Role of Instructional Structure

Anonymous ACL submission

Abstract

This paper investigates how retrieval-augmented generation (RAG) and instructional structure can transform static textbook chapters into interactive learning experiences. To study this, we introduce Textbook Tutor, an interactive AI tutoring system built on a reproducible Textbook-to-Interaction pipeline. The system converts open educational resources into multiple instructional modes, including storytelling, business case simulations, and interactive challenges grounded in retrieval-augmented generation. Through pipeline-level ablations and a mixed-method user study (N=62), we find that structured instructional formats improve short-term learning proxies and can partially compensate for smaller models under fixed retrieval settings. Guided by usability findings from an early prototype, we transitioned the system to a web-based deployment supporting both curated single-textbook tutoring and flexible multi-book workflows. Together, our results demonstrate that effective interactive textbook tutoring can be achieved with lightweight retrieval and modest computational resources.

1 Introduction

Traditional textbook-based learning often struggles to support sustained engagement and conceptual understanding (Woo, 2009), as static chapters provide limited opportunities for interaction, feedback, or adaptive guidance.

Recent advances in large language models (LLMs) have created new opportunities for educational support, including automated tutoring, summarization, and question answering. Systems such as Newtbot (Lieb and Goel, 2024) and Ruffle&Riley (Schmucker et al., 2023) demonstrate the potential of LLM-based tutoring, but primarily focus on dialogue-based interaction and rely on cloud-scale infrastructure.

As a result, two important questions remain underexplored. First, it is unclear how different *instructional structures* (e.g., storytelling, scenario-based reasoning, and structured challenges) interact with retrieval-augmented language models when transforming textbook content into interactive learning experiences. Second, most existing tutoring systems assume access to large models and GPU infrastructure, leaving open whether effective tutoring experiences can be achieved under *low-resource deployment constraints* (OpenStax, 2023b; Lewis et al., 2021; Qin et al., 2023). Traditional textbooks are also often written in dense formats, motivating structured instructional approaches such as storytelling and scenario-based learning.

We study how retrieval-augmented generation (RAG) interacts with instructional structure when transforming textbook chapters into interactive learning experiences. We hypothesize that, in resource-constrained settings, structured instructional formats combined with lightweight retrieval can offset limitations in model capacity. To examine this hypothesis, we introduce Textbook Tutor, an interactive AI tutoring system built around a reproducible Textbook-to-Interaction (T2I) pipeline. The pipeline converts chapter-structured open educational resources into multiple instructional formats, including storytelling modules, business case simulations, and structured challenges grounded in retrieval-augmented generation (Lewis et al., 2021). Content creation is supported by a curated chapter-aligned dataset generated using DeepSeek (AI, 2023a) and manually validated for fidelity. This enables systematic analysis of retrieval grounding, instructional structure, and model capacity.

We first evaluated an early prototype in Google Colab to study feasibility under constrained settings. Based on usability feedback, we redesigned the system as a web-based platform to improve accessibility for non-technical users. Low-resource

deployment is a critical constraint in many educational settings, including schools with limited computational infrastructure, unreliable internet access, or cost restrictions that make cloud-based API usage impractical. These constraints motivate the need for tutoring systems that operate efficiently on local hardware while maintaining instructional quality. We address the following research questions:

1. How effectively can retrieval-augmented generation combined with DeepSeek-generated drafts transform static textbook content into coherent and pedagogically aligned instructional material?

2. How do multiple interactive learning modes, including storytelling, business case simulations, and structured challenges, influence learner engagement and short-term learning perceptions?

3. Is it feasible to deploy an AI-powered tutoring system in resource-constrained environments while maintaining usability and accessibility for non-technical users?

Contributions. This work makes three contributions. (1) We introduce a reproducible Textbook-to-Interaction (T2I) pipeline that converts textbook chapters into multiple instructional formats beyond dialogue-based tutoring. (2) We provide empirical analysis of the relative roles of retrieval grounding, model capacity, and instructional structure, showing that structured instructional formats can partially compensate for smaller models under fixed retrieval conditions. (3) We demonstrate the feasibility of a low-resource deployment using lightweight lexical retrieval and CPU-based generation, validated through a mixed-method user study (N=62) measuring usability, engagement, and short-term learning proxies.

2 Related Work

Intelligent tutoring systems (ITS) aim to provide adaptive and personalized learning experiences (Woo, 2009). Recent advances in large language models (LLMs) have enabled conversational tutoring, retrieval-grounded feedback, and dynamic content generation. However, many existing systems emphasize dialogue-centric interaction, assume cloud-based deployment, and provide limited support for chapter-aligned instruction over open educational resources (OER), leaving open questions around extensibility, reproducibility, and deployment feasibility.

LLM-powered tutoring systems. Conversational tutors such as **Newtbot** (Lieb and Goel, 2024), **Ruffle&Riley** (Schmucker et al., 2023), and **Khanmigo** (Khan Academy Labs, 2023) demonstrate the effectiveness of LLM-based dialogue for learning support, but primarily focus on conversational interaction and rely on cloud-scale or proprietary infrastructure. More recent work integrates retrieval-augmented generation (RAG) into tutoring systems: **Slade et al.** (Slade et al., 2024) and **Németh et al.** (Németh et al., 2024) report usability gains in coding and statistics education, yet provide limited analysis of how retrieval interacts with alternative instructional structures beyond dialogue or how such systems generalize to constrained deployment settings.

Interactive textbooks and retrieval-augmented generation. Prior to LLMs, **Inquire Biology** (Chaudhri et al., 2013) illustrated the value of interactive textbooks grounded in curated knowledge bases, highlighting the importance of aligning instruction with source material. However, these systems relied on manually curated knowledge bases rather than retrieval-augmented language models and did not examine how instructional structure interacts with generative models in lightweight tutoring pipelines. In parallel, retrieval-augmented generation has been shown to improve factual grounding in language models (Lewis et al., 2021). Broader retrieval-augmented and multimodal models, such as RAMM (Mialon et al., 2023), explore architectural advances but are not designed for educational pipelines or chapter-level instructional design. Collectively, these works establish the promise of retrieval grounding, yet leave open how it can be combined with pedagogical structures such as storytelling, scenario-based reasoning, and structured challenges tied to OER.

Positioning. Prior work demonstrates the effectiveness of LLM-based tutors and retrieval grounding, but typically focuses on dialogue-only interaction and assumes substantial computational resources, limiting accessibility and extensibility. Few systems systematically examine how multiple instructional modes interact with retrieval grounding within a single reproducible pipeline, or how such architectures perform under practical low-

System / Paper	Modes	OER	Compute	Artifacts	Eval.	Notes
Newtbot (Lieb and Goel, 2024)	Dialogue	○	Heavy / Cloud	○	Usability	Secondary physics
Ruffle&Riley (Schmucker et al., 2023)	Dialogue	○	Heavy / Cloud	○	Mixed-methods	Design insights
Khanmigo (Khan Academy Labs, 2023)	Dialogue + tools	○	Heavy / Proprietary	○	Case studies	Large-scale platform
Slade et al. (Slade et al., 2024)	Dialogue + RAG	○	Mixed	Code	Usability	Coding courses
Németh et al. (Németh et al., 2024)	Dialogue + RAG	○	Mixed	○	Pilot	Statistics education
Inquire Biology (Chaudhri et al., 2013)	QA over KB	(non-LLM)	Lightweight	○	Classroom	Pre-LLM interactive textbook
RAMM (Mialon et al., 2023)	Multimodal gen.	○	Heavy / Research	Code	Benchmarks	Not education-specific
This work (T2I)	Story, Cases, Multiple Challenges + RAG (single/multi)	(OER + textbook PDFs)	Low local compute (CPU-based)	Dataset + Study Instruments + Code	User study + Retrieval	Web-based deployment

Table 1: Comparison of representative tutoring and RAG-in-education systems. "OER" indicates direct alignment with open educational resources.

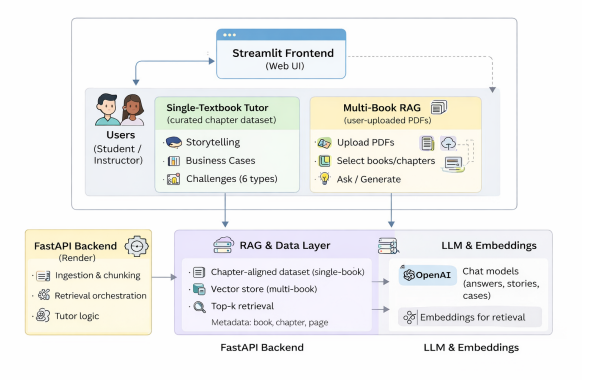

Figure 1: System overview of the Textbook Tutor platform.

resource deployment constraints.

How our work extends this landscape. In contrast to prior work, we treat instructional structure as a first-class design variable within a controlled RAG pipeline. Rather than focusing solely on dialogue-based interaction or model scaling, we systematically vary retrieval grounding, model capacity, and instructional format within a single reproducible framework. This enables analysis of how pedagogical structure interacts with RAG under low-resource constraints, which has not been explicitly studied in prior LLM-based tutoring systems.

3 Methodology

Textbook Tutor is a retrieval-grounded system for converting textbook chapters into interactive learning experiences under resource constraints.

Figure 1 summarizes the system architecture and data flow.

3.1 Data Preparation and Pipeline Overview

For the curated workflow, we extract chapters from the OpenStax *Workplace Software and Skills* textbook (OpenStax, 2023b) using PyMuPDF (Team, 2023). We generate chapter summaries and up to five question–answer pairs per chapter using DeepSeek (AI, 2023a), followed by manual validation. The resulting dataset supports storytelling, challenge generation, and guided question answering.

For the multi-book workflow, users upload PDFs through the web interface, which are processed using the same extraction and chunking pipeline with metadata preserved at the book, chapter, and page level.

Content is indexed using lightweight lexical retrieval. In the curated workflow, retrieval relies on chapter-aligned summaries and validated question–answer pairs; in the multi-book workflow, uploaded content is indexed using TF–IDF representations (Salton and Buckley, 1988). Retrieved content is passed to a generation layer, and learners interact through selectable learning modes while session activity is logged for analysis.

Session activity logs. Session logs record learner actions (e.g., selected mode, completed activities, hints used) and are used to reconstruct interaction sequences for post-hoc analysis.

3.2 Interactive Learning Modes

The system includes storytelling modules (sma, 2023), business case simulations (wri, 2023), and six challenge types: flashcards, multiple-choice questions, fill-in-the-blank exercises, matching tasks, timed questions, and scenario-based prompts with hints (Dicheva et al., 2015). All activities are grounded in validated chapter-aligned content.

Chapter-aware question answering. The system also provides a question answering interface grounded in chapter summaries and validated question–answer pairs, with controllable answer styles (e.g., concise, step-by-step, or example-based).

3.3 Deployment Architecture

The system follows a client–server architecture with a Streamlit frontend and FastAPI backend, enabling browser-based access and CPU-based operation using lightweight models such as Falcon-RW-1B (Institute, 2023). The deployment removes notebook setup and relies on TF–IDF retrieval with template-driven generation. This deployment was

not evaluated in a separate user study and is presented as an engineering extension of the prototype. Further deployment details and cost comparisons are provided in Appendix C.

3.4 Prompting and Retrieval Strategy

We use task-specific prompt templates for summaries, storytelling, challenges, and question answering. Prompts are conditioned on retrieved content and constrained to remain grounded in the source text.

Retrieval uses a fixed top-k similarity search. If no content exceeds a similarity threshold, the system abstains and prompts the learner to revisit relevant sections.

4 Model Development

The model development of Textbook Tutor followed a staged design process. We first employed high-capacity language models during development to generate and validate instructional content. We then evaluated lightweight baseline models to test feasibility under constrained computational settings. Finally, the system was optimized for public deployment with an emphasis on accessibility and efficiency. This staged approach ensures that instructional quality is maintained without requiring heavy computation at runtime.

4.1 Development and Prototype Models

During early development, we employed large instruction-following models, including `mistralai/Mistral-7B-v0.1` (AI, 2023b) and `deepseek-ai/deepseek-coder-6.7b-instruct`, to generate chapter summaries, question–answer pairs, and draft instructional material. These experiments were conducted on a university high-performance computing cluster (64 CPU cores, 256 GB RAM, and up to 8 GPUs), enabling rapid iteration over prompt design and content validation.

The primary role of these models was offline content generation and dataset construction. All generated outputs were manually reviewed and refined before being incorporated into the instructional pipeline. These models are not used during learner-facing interaction. To evaluate feasibility in constrained environments and support early user studies, we implemented a notebook-based prototype in Google Colab using `tiiuae/falcon-rw-1b` (Institute, 2023). This model was selected for its relatively small size and stable performance on free-tier CPU and GPU resources. We applied 4-bit quantization using BitsAndBytes (Dettmers et al., 2022) to reduce memory usage and improve loading reliability. While Falcon-RW-1B provides weaker generative performance than larger models, it enabled interactive experimentation and empirical evaluation without specialized hardware, serving as a baseline for deployment feasibility.

4.2 Deployed System and Lightweight Generation

The final deployed system prioritizes accessibility, robustness, and scalability. Rather than relying on large language models at runtime, Textbook Tutor adopts a lightweight, retrieval-grounded generation strategy. For the curated single-textbook workflow, responses are constructed from chapter-aligned summaries and validated question–answer pairs. For the multi-book workflow, user-uploaded PDFs are indexed using TF-IDF-based retrieval, and responses are grounded in the most relevant retrieved chunks. This approach avoids dependence on dense vector indexes or continuous model inference, while preserving grounding and instructional fidelity.

4.3 Conversational Assistant

In addition to predefined learning modes, the system includes a chat-based, chapter-aware question answering assistant that allows learners to ask free-form questions about the selected textbook chapter. For each query, the assistant constructs a compact context from the chapter summary, validated question–answer pairs, and recent chat history, and generates responses using a hosted large language model via the OpenAI API. While the chapter content serves as the primary grounding signal (Lewis et al., 2021), the assistant may additionally use general world knowledge to provide intuitive real-world examples and analogies.

To accommodate different learning preferences, learners can control the answer style, requesting concise explanations, step-by-step reasoning, or real-world illustrative examples while keeping the chapter context fixed. The assistant is implemented using the `gpt-4o-mini` model and is intended to support exploratory learning. This component was not used in the user study, retrieval evaluations, or deployment analysis reported in this paper. We describe this component for completeness, but exclude it from all evaluations to avoid confounding instructional effects with proprietary models.

5 Evaluation

Evaluation Overview. We separate evaluation into (i) controlled pipeline ablations that isolate system components, and (ii) a human-centered user study evaluating usability and short-term learning outcomes.

Note. An earlier ARR cycle included a 30-participant pilot; this revision reports the subsequent main study (N=62; 57 complete).

5.1 Pipeline-Level Ablation and Component Analysis

These ablations are designed to isolate the relative contributions of retrieval grounding, model capacity, and instructional structure within the Textbook-to-Interaction pipeline. We conduct pipeline-level ablations to isolate the effects of (i) retrieval grounding, (ii) model capacity, and (iii) retrieval strength, while holding the instructional structure fixed across conditions.

5.1.1 Prompt Grounding Ablation (RAG ON vs. OFF)

To isolate the effect of retrieval grounding in generation, we compare responses generated with retrieval enabled against generation-only baselines using identical prompts, models, and decoding parameters. Retrieval-augmented generation significantly improved rubric scores (mean total 3.9 vs. 2.7 out of 5; $\Delta = 1.2$, $p < 0.001$, $r = 0.62$), with substantial inter-rater agreement ($\kappa = 0.71$).

5.1.2 Model Capacity Ablation

To assess the impact of model scale, we compare a lightweight model (Falcon-RW-1B) with a larger model (Mistral-7B) under identical prompts, retrieval inputs, and decoding settings. Larger models outperformed lightweight baselines on faithfulness (exact-overlap 68% vs. 51%) and overall rubric scores (+0.6/5; $p = 0.02$).

5.1.3 Retrieval Strength Ablation

To evaluate whether lightweight retrieval is sufficient for grounding, we compare TF–IDF, BM25, and hybrid lexical retrieval strategies while holding generation fixed. Results show consistently high chapter-level recall and precision across all methods, indicating that lightweight lexical retrieval is sufficient for reliable chapter-level grounding.

Single-Book We evaluate the retrieval component independently using a chapter-aligned dataset derived from the curated single-textbook workflow. The evaluation set consists of **486 queries** generated from validated chapter-level questions. Each query is associated with a gold chapter, and retrieval is considered successful if at least one retrieved chunk belongs to the gold chapter.

We compare three lightweight retrieval strategies designed for low-resource settings: TF–IDF, BM25, and a hybrid lexical retriever combining both. Results are shown in Table 2.

Multi-Book Routing We evaluate multi-book routing using two complementary textbooks: *Beyond Coding* (CAUL, 2023) and *Computer Science* ((OpenStax, 2023a)). We construct manually curated query sets of **100 queries** and **250 queries**, respectively, each labeled with its gold source textbook. Retrieval is considered correct if at least one of the top-k retrieved chunks originates from the correct textbook.

We report **Recall@k** for $k \in \{1, 3, 5\}$ and compare three lightweight retrieval strategies suitable for low-resource deployment: TF–IDF, BM25, and a hybrid lexical retriever. Results are shown in Table 3.

Across both datasets, Recall@5 exceeds **0.80** in all configurations and approaches **1.0** for the more structured *Computer* textbook. TF–IDF performs best at rank 1 for semantically phrased queries, BM25 provides stronger coverage at higher ranks, and the hybrid strategy consistently balances these strengths, achieving the best or near-best Recall@1.

Analysis of tradeoffs. Across ablations, we observe a consistent tradeoff between model capacity and instructional structure. While increasing model size improves generation quality, the magnitude of improvement is smaller than the improvements observed from retrieval grounding and structured instructional formats. This suggests that improvements from structured instructional formats are comparable in magnitude to gains from increased model capacity under fixed retrieval conditions. This pattern is consistent across all ablation settings, suggesting that the observed effect is not tied to a specific configuration but reflects a broader property of retrieval-grounded instructional systems.

Table 2: Chapter-level retrieval performance on the merged chapter dataset (486 queries).

Retrieval Method	Recall			Precision		
Precision / Recall	@1	@3	@5	@1	@3	@5
TF–IDF	0.947	0.986	0.992	0.947	0.947	0.945
BM25	**0.977**	**0.992**	**0.992**	**0.977**	**0.979**	**0.980**
Hybrid (TF–IDF + BM25)	0.965	0.988	0.992	0.965	0.964	0.963

Table 3: Multi-book retrieval performance (Recall@k) across two textbooks.

Method	Recall@1	Recall@3	Recall@5
Beyond Coding (CAUL, 2023)(100 queries)			
TF–IDF	0.294	0.686	0.765
BM25	0.098	0.765	0.843
Hybrid	0.235	0.725	0.804
Computer (OpenStax, 2023a) (250 queries)			
TF–IDF	0.504	0.972	0.984
BM25	0.504	**0.984**	**0.996**
Hybrid	**0.516**	0.976	0.992

5.2 User Study (Human-Centered Evaluation)

5.2.1 Study Design

We conducted a mixed-method evaluation to assess usability, engagement, and instructional effectiveness. The study was approved by the university IRB (IRB2025-0182), and all participants provided informed consent. A total of 62 participants accessed the system, with 57 completing both pre- and post-assessments. Participants were students or early-career professionals in computing-related fields, recruited via university mailing lists, social media, and faculty announcements.

Each participant completed a 45-minute session in Google Colab consisting of a pre-assessment, free exploration across learning modes, and a post-assessment. Participants selected learning modes freely, so the study was not between-subjects. For exploratory analysis, session logs were used to compare MCQ-only and MCQ+multi-mode usage patterns among participants with complete pre/post concept quizzes.

We collected Likert-scale ratings, open-ended feedback, and system logs. Pre/post items measured short-term learning proxies (*familiarity* and *comfort*), while post-only items measured usability and engagement. The conversational assistant was disabled during the study to keep the interaction flow consistent across participants. All participants were exposed to the same system and were free to select learning modes during the session; no controlled intervention or randomized assignment was used.

Table 4: Post-assessment descriptive statistics (N = 57). Likert-scale items rated 1–5.

Feedback Item	Mean	Median	SD
Overall experience	4.70	5.0	0.82
Ease of using the interface	4.60	5.0	0.94
Helpfulness of walkthroughs	4.70	5.0	0.82
Engagement of learning modes	4.70	5.0	0.50
Content clarity and quality	4.76	5.0	0.47
Ease of session completion	4.78	5.0	0.46
Would use again (1–5)	4.71	5.0	0.50

5.2.2 Usability and Engagement

Post-assessment feedback (Table 4) indicated strong positive reception. Mean ratings across all Likert-scale items exceeded **4.5**, with participants reporting high ease of use (**4.60**), content clarity (**4.76**), and satisfaction with guided walkthroughs (**4.70**). Engagement of learning modes (**4.70**) and overall experience (**4.70**) were similarly rated. A total of **55 out of 57 participants (96%)** reported improved understanding, and willingness to reuse the tutor scored **4.71**, indicating strong perceived value. Exploratory factor analysis revealed two latent factors: *Usability* and *Engagement and Content Quality*. The Kaiser–Meyer–Olkin value was **0.674**, and Bartlett's test was significant ($\chi^2 = 114.66$, $p < 0.001$), confirming data suitability. Together, the factors explained **50.27%** of the variance. Internal consistency was excellent for Usability ($\alpha = 0.895$), while Engagement and Content Quality showed modest reliability ($\alpha = 0.449$). Given the heterogeneous nature of this construct and the small item set, we treat this factor structure as exploratory and report full loadings in Appendix F.

5.2.3 Objective Learning Assessment (Concept Quiz)

This analysis is part of the user study and should be interpreted as exploratory, as participants were not randomly assigned to instructional conditions. To complement the self-reported measures, we analyzed performance on an objective concept quiz administered before and after the session. The quiz consisted of eight short conceptual questions related to retrieval-augmented generation and system design (see Appendix E.6).

Participants were free to explore different learning modes and were not randomly assigned to instructional conditions. For an exploratory, post-hoc analysis we used session activity logs to classify participant behavior into two groups: *MCQ-only* (at least one completed MCQ activity and

Table 5: Participant flow for the concept-quiz subgroup analysis.

Population	Count
Participants consented	62
Completed pre/post assessments	57
Included in concept-quiz subgroup analysis	20
MCQ-only	8
MCQ + multi-mode	12

no completed activities in any other mode) and *MCQ+multi-mode* (at least one completed MCQ and at least one completed activity in another instructional mode, e.g., storytelling or scenario-based challenges). We applied these inclusion criteria to construct the analysis set. Participants with incomplete quiz data or ambiguous logs were excluded, resulting in $n = 20$ (Table 5). We then compared change-score distributions Δ (post$-$pre) between the two independent groups using a two-sided Mann–Whitney U test (Wilcoxon rank-sum). Participants exposed to multi-mode learning showed larger median gains than MCQ-only participants (median $\Delta = +2$ vs. $\Delta = +0$); the comparison yielded $p = 0.004$ with effect size $r = 0.49$. Because assignment to modes was self-selected and this comparison was post-hoc, the result should be interpreted as exploratory rather than causal.

This exploratory result suggests that instructional structure may be associated with measurable learning differences beyond self-reported metrics, addressing a key limitation of prior work relying primarily on subjective measures.

5.2.4 Learning Outcomes

Wilcoxon signed-rank tests demonstrated significant improvements between pre- and post-assessments (Table 6). Self-reported familiarity increased from **2.39** to **2.77** ($p < 0.001$), while comfort increased from **2.38** to **2.82** ($p < 0.001$), indicating improved conceptual understanding and confidence. In addition, we conducted a post-hoc comparison based on interaction logs to examine differences between participants who interacted only with multiple-choice quizzes (MCQ) and those who engaged with multiple instructional modes (e.g., storytelling and scenario-based challenges). No significant differences were observed across user subgroups based on initial familiarity or comfort (all $p > 0.42$), and effect sizes were negligible ($|\delta| < 0.15$), suggesting consistent improvements regardless of prior experience. Most

Table 6: Pre- vs. post-assessment comparisons (Wilcoxon signed-rank tests).

Comparison	Pre	Post	Statistic	p-value
Familiarity to Understanding	2.39	2.77	88.0	0.00003
Comfort to Experience	2.38	2.82	102.0	0.00061

participants showed non-negative gains on the concept quiz, and the full sample improved from pre- to post-assessment. This suggests that the observed gains are consistent with a broader improvement trend associated with system use.

6 Discussion

We interpret the results in relation to the three research questions, focusing on retrieval grounding, instructional structure, and deployment feasibility.

RQ1: Retrieval-grounded generation. Pipeline evaluations show that retrieval significantly improves factuality and grounding over generation alone (Table 11). Chapter-level retrieval achieves consistently high recall and precision across both curated and multi-book settings (Table 2), indicating that lightweight lexical retrieval is sufficient for reliable, chapter-aligned generation.

RQ2: Role of instructional structure. User study results show strong engagement and usability across learning modes (Table 4). The concept quiz comparison further suggests that multi-mode interaction leads to larger gains than MCQ-only use (Table 12). These findings highlight the importance of jointly considering retrieval design and instructional structure in resource-constrained educational systems.

Beyond prior learning science findings. While prior work in learning science has established that instructional structure improves engagement and comprehension, our results extend this insight to retrieval-augmented language models under low-resource constraints. Specifically, we show that structured instructional formats can compensate for limited model capacity in RAG-based tutoring systems, a setting not explored in traditional educational research. This highlights a distinct design principle for NLP-based tutoring systems: improving instructional structure may be a more effective lever than increasing model scale when computational resources are limited.

RQ3: Deployment feasibility. The Colab prototype enabled controlled evaluation but introduced

usability friction. The web-based deployment removes setup requirements and runs on CPU with lightweight retrieval and template-based generation, suggesting practical feasibility for non-technical users.

Cost-aware design implications. Compared to prior tutoring systems that rely on cloud-based large language models, our approach demonstrates that comparable instructional functionality can be achieved using CPU-based retrieval and lightweight generation. This suggests that, in low-resource educational settings, system design choices can significantly reduce deployment cost without sacrificing usability or engagement.

Implications. Taken together, the results suggest that effective interactive tutoring can be achieved without heavy infrastructure when retrieval grounding and instructional design are jointly considered. The Textbook-to-Interaction pipeline provides a practical framework for building accessible, resource-efficient learning systems. We note that prompt design may play a similar role to instructional structure, as both impose organization on generated content; exploring structured prompting strategies is an important direction for future work.

Overall, these findings highlight the importance of jointly considering retrieval design and instructional structure when building resource-efficient educational systems.

7 Limitations and Future Work

This work has several limitations. First, the user study was conducted on a notebook-based prototype, and we did not run a second quantitative study after redeploying the system as a web application. Second, learning outcomes were measured mainly through short-term self-report proxies, supplemented by a small objective concept quiz, so they do not capture long-term learning or retention. Third, the system uses constrained retrieval and generation to preserve grounding and reliability, which may limit coverage for highly open-ended or cross-document queries. Finally, we do not include direct comparisons with other tutoring systems or commercial AI assistants, because deploying comparable baselines under identical low-resource constraints would be difficult.

Future work will focus on longitudinal and classroom-based evaluations, improved retrieval strategies that better balance flexibility and grounding, comparative studies with alternative tutoring approaches, a more responsive conversational assistant, and web-platform extensions such as persistent user accounts, instructor analytics, and support for additional subjects and multimodal content. We also leave the effect of alternative prompting strategies, such as chain-of-thought and plan-and-solve prompting, for future study.

Because the user study was not randomized and relies partly on self-reported measures, we do not make causal claims about instructional effectiveness.

8 Conclusion

This paper presented a retrieval-grounded interactive tutoring system that transforms textbook chapters into structured learning experiences using a reproducible Textbook-to-Interaction pipeline. Through controlled ablations and a user study, we show that lightweight retrieval supports reliable grounding and that structured instructional formats are associated with improved engagement and modest gains on short-term concept assessments. These findings suggest that effective interactive tutoring is possible under constrained computational settings without relying on large-scale models at runtime. More broadly, this work highlights instructional structure as a key design dimension in retrieval-augmented educational systems, particularly in low-resource settings. Future work will focus on stronger causal evaluation, prompt design, and longitudinal learning outcomes.

Ethics Statement

The user study was conducted with voluntary participation under university-approved IRB protocol. No personally identifiable information was collected.

Broader Impact

This work aims to improve access to interactive learning using open educational resources and lightweight AI deployments. It can benefit learners in resource-constrained environments by enhancing personalization and engagement. Limitations include dependency on internet access and evolving AI model reliability.

References

2009. Copyright page. In Beverly Park Woolf, editor, *Building Intelligent Interactive Tutors*, page iii. Morgan Kaufmann, San Francisco.

2023. How to write a business case. Accessed: 2025-04-28.

2023. Sell your product with powerful storytelling. Accessed: 2025-04-28.

DeepSeek AI. 2023a. Deepseek coder: Code large language model. https://huggingface.co/deepseek-ai/deepseek-coder-6.7b-instruct. Accessed: 2025-04-28.

Mistral AI. 2023b. Introducing mistral 7b. https://mistral.ai/news/introducing-mistral-7b/. Accessed: April 26, 2025.

CAUL. 2023. *Beyond Coding: Professional and Soft Skills for Computing*. OER Collective.

Vinay K. Chaudhri, David Gunning, Hannah Lane, Peter Clark, Ken Barker, Simon Chaw, Mark Greaves, Benjamin Grosof, Alex Leung, Suchi Mishra, Julio Pacheco, Bruce Porter, Aaron Spaulding, Dan Tecuci, and Peter Z. Yeh. 2013. Inquire biology: A textbook that answers questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 1624–1631. AAAI Press.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. 2015. Gamification in education: A systematic mapping study. *Educational Technology & Society*, 18(3):75–88.

Technology Innovation Institute. 2023. Falcon-rw-1b model card. https://huggingface.co/tiiuae/falcon-rw-1b. Accessed: 2025-04-28.

Khan Academy Labs. 2023. Introducing Khanmigo: AI-Powered Tutor and Teaching Assistant. https://www.khanacademy.org/khan-labs. Accessed: 2025-04-25.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Anna Lieb and Toshali Goel. 2024. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Preprint*, arXiv:2302.07842.

Renáta Németh, Annamária Tátrai, Miklós Szabó, and Árpád Tamási. 2024. Using a rag-enhanced large language model in a virtual teaching assistant role: Experiences from a pilot project in statistics education. *Hungarian Statistical Review*, 7(2):3–27.

OpenStax. 2023a. *Computer Science*. Rice University.

OpenStax. 2023b. *Workplace Software and Skills*. OpenStax, Rice University.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *Preprint*, arXiv:2302.06476.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2023. Ruffle&riley: Towards the automated induction of conversational tutoring systems. In *NeurIPS 2023 Workshop on Generative AI for Education*.

Joseph J. Slade, Alina Hyk, and Regan A. R. Gurung. 2024. Transforming learning: Assessing the efficacy of a retrieval-augmented generation system as a tutor for introductory psychology. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 68, pages 1827–1830, Los Angeles, CA, USA. SAGE Publications.

MuPDF Team. 2023. Pymupdf: Lightweight pdf parsing and rendering. https://github.com/pymupdf/PyMuPDF. Accessed: 2025-04-26.

A Appendix

B Core Textbook-to-Interaction Pipeline

Textbook PDF T Interactive AI Tutor Experience

Step 1: Chapter Extraction

Extract chapter-wise content $C = \{c_1, c_2, ..., c_n\}$ from T using PDF parsing tools.

Step 2: QA Dataset Generation

chapter $c_i \in C$ Generate question–answer pairs QA_i and summary S_i using DeepSeek Coder-6.7B Store initial dataset $D = \{(QA_i, S_i, c_i)\}$

Step 3: Manual Review and Cleaning

Filter for hallucinations, formatting issues, and factual errors to produce cleaned dataset D^*

Step 4: Semantic Indexing
Embed QA_i with MiniLM and store in FAISS vector index V

Step 5: Retrieval-Augmented Generation
Use LangChain with index V to enable context-aware content retrieval and generation

Step 6: Model Integration
Internal high-resource environment Load Mistral-7B Low-resource environment (e.g., Colab) Load Falcon-RW-1B with 4-bit quantization using BitsAndBytes

Step 7: User Interaction Layer
Initialize user profile and preferences
If user is new or walkthrough requested **then**
Launch guided walkthrough with:

- Step-by-step preview of all challenge types
- Navigation controls (next/back/skip)
- Visual prompts and tooltips

Else
Launch interactive learning modules:

- Storytelling modules
- Business case simulations
- Interactive challenges:
  - Flashcards
  - Multiple-choice quizzes (MCQs)
  - Fill-in-the-blank tasks
  - Matching exercises
  - Timed questions
  - Scenario-based challenges with hints

Step 8: Gamification and Feedback
Track XP, level progression, and badge collection
Provide real-time feedback and update learner dashboard

C Deployment Details

C.1 Deployment Analysis

While the Colab prototype enabled controlled evaluation under constrained resources, it introduced practical barriers including session time limits, dependency setup, and notebook interaction. Based on these findings, we redeployed the system as a web-based platform using a Streamlit frontend and a FastAPI backend. The deployed version supports one-click browser access, both curated single-textbook and flexible multi-book workflows, and CPU-based operation using lightweight TF–IDF retrieval and template-driven generation.

System	Model Type	Hardware	Relative Cost
Khanmigo	GPT-4 class models	Cloud GPUs	High
Newtbot	LLM API	Cloud	High
Ruffle&Riley	LLM-based tutor	Cloud GPUs	High
Textbook Tutor (ours)	TF–IDF + small LM	CPU	**Low**

Table 7: Estimated comparison of deployment requirements across tutoring systems.

C.2 Deployment Cost Comparison

To contextualize the practical advantages of lightweight retrieval-based tutoring, we compare the deployment requirements of representative tutoring systems with Textbook Tutor. Many LLM-powered tutors rely on cloud-hosted large language models or proprietary APIs, which require GPU infrastructure or per-query API costs. In contrast, Textbook Tutor operates using lightweight lexical retrieval and CPU-based generation, enabling deployment in environments without specialized hardware.

D Operational metrics

E Baseline and Ablation Details

E.1 Rubric (0–5 scale)

We evaluated generated responses on three criteria: (1) **Factuality** (0–2), measuring correctness relative to the textbook source; (2) **Grounding/Traceability** (0–2), measuring explicit linkage to retrieved spans; and (3) **Clarity** (0–1), assessing readability and coherence. Total possible score: 5. Table 10 details the rubric.

E.2 Rater Agreement

Two independent raters scored all outputs using the rubric above. Inter-rater reliability was $\kappa = 0.71$ (95% CI [0.62, 0.80]), indicating substantial agreement. Disagreements were resolved through adjudication.

E.3 RAG ON vs. OFF Results

E.4 Single vs. Multi-Mode Results

E.5 Model Comparison Results

E.6 Concept Quiz Items (Objective)

Below are representative items from the 8-question concept quiz used in the single- vs. multi-mode study.

- Q1: What is retrieval-augmented generation (RAG) and how does it improve factual grounding?

Template	Intent / Inputs	Full Prompt Text (Simplified)
QA Generation	*Intent:* Create up to *N* factual QA pairs from a chapter excerpt. *Inputs:* {chapter_title}, {chapter_excerpt}, {N} (default 5)	**System:** You are an educational content author. Produce concise, factual question–answer pairs strictly grounded in the provided chapter excerpt. Avoid speculation. **User:** Chapter: {chapter_title}. Excerpt: {chapter_excerpt}. Task: Generate up to {N} QA pairs. Each pair must be answerable solely from the excerpt. Use JSON schema: `[ { "question": "...", "answer": "...", "evidence_span": "..." }, ... ]`. If information is insufficient, omit that pair.
Summary Generation	*Intent:* Produce a short chapter summary. *Inputs:* {chapter_title}, {chapter_excerpt}, {max_words} (default 150)	**System:** You are summarizing textbook material for learners. Stay faithful to the source; no external facts. **User:** Chapter: {chapter_title}. Excerpt: {chapter_excerpt}. Task: Write a concise summary ({max_words} words) capturing key definitions, processes, and takeaways. Include no references beyond the excerpt.
Storytelling Scaffold	*Intent:* Transform a validated summary into a short workplace scenario. *Inputs:* {chapter_title}, {validated_summary}, {persona_A}, {persona_B}	**System:** You are an instructional designer. Create a realistic, workplace-relevant micro-story aligned with the provided summary. **User:** Summary: {validated_summary}. Task: Write a 120–180 word narrative featuring {persona_A} and {persona_B}. The scenario must exemplify 2–3 core ideas from the summary (**bold key terms on first mention**). End with a reflective question answerable from the narrative. Output JSON: `{"story": "...", "reflective_question": "..." }`.

Table 8: Prompt templates used in the Textbook-to-Interaction (T2I) pipeline. Braced fields are filled at runtime.

Setting / Rule	Public (Falcon-RW-1B)	Dev (Mistral-7B)	Notes
Decoding temperature	0.7	0.3–0.5	Higher for storytelling; lower for QA
Top-p (nucleus)	0.9	0.9	Paired with moderate temperature
Top-k	50	50	Conservative cap to reduce drift
Max new tokens	256	256 (QA), 180 (story)	Enforced at pipeline level
Repetition penalty	1.10	1.10	Mitigates loops
Stop sequences	`{"\n\n"}`	same	Prevent runaway JSON / paragraphs
Grounding mode	Required	Required	Answers must cite retrieved span IDs
Similarity threshold θ	0.35 cosine	0.35 cosine	If $\max \text{sim} < \theta$, abstain
Abstention response	"I don't have enough context to answer that."	same	Followed by one clarifying question
Evidence requirement	At least one quoted / cited span per key claim	same	Regex check before output
External web access	Disabled	Disabled	Prevent leakage beyond OER
Story length guard	120–180 words	120–180 words	Keeps mode parity
JSON schema checks	Strict (pydantic)	Strict	QA / story outputs validated

Table 9: Guardrails and decoding defaults. Values shown are the defaults used in our experiments unless otherwise noted.

Criterion	Scoring Description
Factuality (0–2)	0 = Incorrect, 1 = Partially correct, 2 = Fully correct
Grounding (0–2)	0 = No evidence cited, 1 = Partial citation, 2 = Explicit and correct citation
Clarity (0–1)	0 = Unclear or incoherent, 1 = Clear and coherent

Table 10: Rubric used to score generated outputs in baseline evaluations.

- Q2: Describe two benefits of 4-bit quantization in low-resource deployments.
- Q3: In the storytelling mode, what instructional purpose do reflective questions serve?
- Q4: What role does the FAISS index play in semantic retrieval?

Metric	RAG ON	RAG OFF
Mean total score (0–5)	3.9	2.7
Factuality	1.8	1.2
Grounding	1.7	0.8
Clarity	0.4	0.7

Table 11: Comparison of assistant performance with and without retrieval grounding (N=40 questions).

Measure	MCQ Only	MCQ+ Story+ Scenario
Median concept quiz Δ	+0	+2
p-value (Wilcoxon)	0.004	
Effect size r	0.49	

Table 12: Learning gains comparing single-mode (MCQ-only) and multi-mode interaction (MCQ + storytelling + scenario). Values represent the median change (Δ) between post-test and pre-test concept quiz scores. Between-group comparison uses a two-sided Mann–Whitney U test. Effect size r is reported. The comparison uses the subgroup of participants who completed both quizzes and whose session logs clearly indicated MCQ-only or MCQ+multi-mode usage ($n = 20$).

F Exploratory Factor Analysis Details

F.1 Factor Loadings and Communalities

Table 14 reports the factor loadings for post-assessment items based on principal axis factoring with varimax rotation. Two factors emerged, explaining a total of **50.27%** of the variance (**37.65%** for Factor 1 and **12.62%** for Factor 2). Factor 1 corresponds primarily to *Usability*, while Factor 2 captures *Engagement and Content Quality*.

Metric	Falcon-RW-1B	Mistral-7B
Exact overlap with retrieved spans	51%	68%
Mean rubric total (0–5)	3.2	3.8
p-value	0.02	

Table 13: Model comparison on 30 held-out prompts.

Item	Factor1 (Usability)	Factor2 (Engagement /Content)
Overall experience	0.885	0.026
Ease of using the interface	0.881	0.224
Helpfulness of walkthroughs	0.833	-0.160
Engagement of learning modes	-0.011	0.599
Content clarity and quality	0.076	0.512
Ease of session completion	-0.009	0.246

Table 14: Exploratory factor loadings for post-assessment items ($N = 57$). Principal axis factoring with varimax rotation. Total variance explained: 50.27%.

F.2 Communalities and Reliability

Given the heterogeneous nature of the "Engagement/Content Quality" construct and the small item set, internal consistency was modest ($\alpha = 0.45$). We therefore treat the two-factor solution as exploratory. Future work will refine the item pool and validate the structure on a larger sample.

G Pre-Assessment Questionnaire

Participants completed the following pre-assessment form prior to using the AI Tutor. The form collected demographic information, prior experience with textbook learning and AI tools, learning preferences, and expectations from the system.

G.1 Section 1: Basic Information

- What best describes your current role?
  - High school or less
  - Associate's degree
  - Bachelor's degree
  - Master's degree
  - PhD degree
  - Instructor/Professor
  - Working Professional
  - Other
- How familiar are you with using AI-based educational tools? (Very familiar / Somewhat familiar / Not familiar at all / Other)

G.2 Section 2: Previous Experience with Textbook Learning

- How often do you use textbooks for learning? (Daily / Weekly / Occasionally / Rarely / Never)
- Have you used any interactive or digital learning platforms before? (Yes / No)

G.3 Section 3: Experience with Digital Learning Platforms

- Please list any digital learning platforms you've used.

G.4 Section 4: Learning Preferences & Expectations

- Which learning methods do you prefer? (Reading text / Watching videos / Interactive quizzes / Case-based learning / Storytelling / Group discussions)
- What are your main learning goals with the AI Tutor?
- What do you expect from an AI-powered educational tool?

G.5 Section 5: Technical Readiness

- How comfortable are you with using tools like Google Colab or Jupyter Notebooks? (Very comfortable / Somewhat comfortable / Not comfortable)
- Have you used any AI-driven tools before? (Yes / No / Maybe)

G.6 Section 6: Experience with AI Tools

- If yes, please describe the tools you've interacted with and your experience using them.

G.7 Section 7: Expectations from the AI Tutor

- What kind of content do you expect the AI Tutor to help you with? (Subject-specific knowledge / Study skills / Test preparation / Real-world applications / Problem-solving)
- Do you prefer a personalized or standardized learning experience? (Personalized / Standardized / No preference)
- What features would you want in an AI-powered learning tool?

H Post-Assessment Questionnaire

After completing their interaction with the AI Tutor, participants filled out the following post-assessment questionnaire. The form was designed to evaluate usability, user engagement, content quality, technical performance, and overall experience. Responses informed our analysis of short-term learning impact and user satisfaction.

H.1 Section 1: Basic Information

- What best describes your current role?
  - High school or less
  - Associate's degree
  - Bachelor's degree
  - Master's degree
  - PhD degree
  - Instructor / Professor
  - Working Professional
  - Other
- How did you access the AI Tutor?
  - Through Google Colab directly
  - Researcher sent me the link
  - Other

H.2 Section 2: Overall Experience

- How would you rate your overall experience with the AI Tutor?
  (1 = Very Poor . . . 5 = Excellent)
- How easy was it to use the AI Tutor interface?
  (1 = Very Difficult . . . 5 = Very Easy)
- How helpful were the guided walkthroughs in getting started?
  (1 = Not helpful . . . 5 = Very helpful)
- How engaging did you find the different learning modes (Business Case, Storytelling, Challenges)?
  (1 = Not Engaging . . . 5 = Very Engaging)

H.3 Section 3: Challenge Types

- Which challenge types did you try?
  - Flashcards
  - Multiple-Choice Questions (MCQ)
  - Fill in the Blanks
  - Match the Answers
  - Timed Questions

- Scenario-Based Challenges

- Which challenge type was the most effective for your learning? Why?

- Was the feedback (XP points, correct/incorrect indicators, progress tracking) helpful in understanding your performance? (Yes / No / Maybe)

H.4 Section 4: Performance and Content

- How would you rate the quality and clarity of the content presented in the AI Tutor? (1 = Very Poor . . . 5 = Excellent)

- Overall, how easy was it to complete your learning session with the AI Tutor? (1 = Very Difficult . . . 5 = Very Easy)

- Did the AI Tutor help improve your understanding of the concepts?
  - Yes, significantly
  - Yes, somewhat
  - Neutral
  - No, not really
  - No

- How would you describe the speed and performance of the AI Tutor in Google Colab?
  - Fast and smooth
  - Acceptable
  - A bit slow
  - Very slow or laggy
  - Didn't work properly

H.5 Section 5: Suggestions and Future Use

- What features would you like to see added in the future?

- Would you use the AI Tutor again if it were available as a public website? (Yes / No / Maybe)

- Any other comments or suggestions to improve the AI Tutor?

I XP Schedules and Level Thresholds

I.1 XP Reward Schedule

Table 15 shows the XP reward structure used in the prototype implementation. Points are awarded for completing different interactive challenge types, and their share of total XP reflects relative contribution to learner progression.

Challenge Type	XP per Attempt	Share of Total XP (%)
Flashcards (Flip)	5	7.5
MCQ quiz	10	14.9
Fill-in-the-Blank	10	14.9
Match-the-Answers	12	17.9
Timed questions	15	22.4
Scenario-based (Hint)	15	22.4

Table 15: XP reward schedule for interactive challenges.

I.2 Level Progression Thresholds

Learners advance through levels as they accumulate XP. Table 16 lists the XP requirements for each level and the cumulative total needed for progression.

Level	XP to Next	Cumulative XP
$1 \rightarrow 2$	50	50
$2 \rightarrow 3$	75	125
$3 \rightarrow 4$	100	225
$4 \rightarrow 5$	125	350
$5 \rightarrow 6$	150	500
$6 \rightarrow 7$	175	675
$7 \rightarrow 8$	200	875
$8 \rightarrow 9$	225	1100
$9 \rightarrow 10$	250	1350
$10 \rightarrow 11$	275	1625

Table 16: XP thresholds for each level in the gamification engine.