

A Local-First Course-Grounded Learning System Using Parent-Child Retrieval and Quantized Large Language Models

Nyoman Sarasuartha Mahajaya^{1*}, I Dewa Gede Aristana², Anak Agung Gede Oka Kessawa Adnyana¹, I Nyoman Darma Kotama³, and I Gusti Made Ngurah Desnanjaya¹

¹ Faculty of Informatics and Technology, Institut Bisnis dan Teknologi Indonesia, Denpasar, Indonesia

² Faculty of Business and Creative Design, Institut Bisnis dan Teknologi Indonesia, Denpasar, Indonesia

³ Department of Information and Communication Systems, Okayama University, Okayama 700-8530, Japan

Article Info29

Article history:

Received September 2, 2026

Revised September 30, 2026

Accepted October 15, 2026

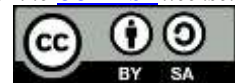
Keywords:

local-first generative AI
course-grounded learning
retrieval-augmented generation
parent-child retrieval
large language model
lecturer governance

ABSTRACT

Selecting a local language model for course-grounded learning requires evidence about response quality and computing demand under a shared retrieval workflow. This study develops a local-first system that retrieves child passages, restores parent sections as generation context, and preserves source links for answers and lecturer-reviewed quiz candidates. Five quantized 7B and 14B models were evaluated using 20 questions derived from six meetings of a single Problem Solving and Debugging course, with 300 performance measurements and 100 responses for RAG evaluation. The 7B models achieved 57.14-58.95 tokens/s using approximately 5 GB of VRAM. Qwen3 14B obtained the highest Faithfulness (0.882) and Answer Relevancy (0.906), requiring 10.19 GB of VRAM and a mean response time of 60.05 s. Across models, Context Precision ranged from 0.828 to 0.840 and Context Recall from 0.776 to 0.789. The contribution is a source-traceable parent-child workflow and a measured quality-resource trade-off for local model selection. The evaluation was limited to 20 questions from a single course and did not include a child-only retrieval baseline. Therefore, the findings cannot establish the generalizability of the results across courses or the independent benefit of restoring parent sections.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Nyoman Sarasuartha Mahajaya, Faculty of Informatics and Technology, Institut Bisnis dan Teknologi Indonesia, Denpasar, Indonesia; Email : nyoman_sarasuartha@instiki.ac.id; Orcid Id : <https://orcid.org/0009-0007-3209-213X>

1. INTRODUCTION

Learning Management Systems (LMSs), such as Moodle, allow lecturers to distribute files, books, and web resources [1] and organize assignments, quizzes, and discussions [2]. These functions provide a foundation for course delivery. The design problem addressed here is how to provide conversational explanations that remain traceable to lecturer-approved materials while running on institution-controlled computing resources. Large Language Models (LLMs) can support learning assistance, but their educational use requires attention to inaccurate output, privacy, and responsible use [3].

For institutions that require course documents and student interactions to remain on their own infrastructure, external inference services introduce a deployment constraint. Running inference, storage, and supporting services locally can reduce dependence on external processing. Quantization lowers model memory requirements [4], [5], but local execution alone does not establish whether answers are supported by the selected course material.

Retrieval-Augmented Generation (RAG) supplies source content to a language model before generation [6], with document indexing, retrieval, generation, and evaluation forming the main pipeline [7]. Educational research has examined tutoring and content generation [8], self-regulated learning [9], and broader educational applications [10]. More specifically, Levonian et al. evaluated safety guardrails and RAG in mathematics conversations [11]. Zhang et al. evaluated retrieval-supported answers about assessment documentation using GPT-3.5 and GPT-4o [12]. Zheng et al. examined failures when textbook content conflicts with model knowledge [13]. These studies establish the relevance of grounding and evaluation, but they do not directly answer which quantized local model offers a suitable quality-resource balance when retrieved child items are expanded into course sections for generation.

The gap addressed in this study is therefore deployment-specific evidence about answer grounding, latency, and memory demand under a shared course-grounded retrieval configuration. Quantized-model studies establish the feasibility of reduced-memory inference [4], [5], [14], while assessment-generation studies emphasize

the need for human review [15], [16], [17]. For the present course setting, the unresolved question is how five quantized 7B and 14B models differ when using the same parent-child pipeline and local inference conditions. Parent-child retrieval is the architectural choice examined within this system; its independent advantage over flat retrieval is not established by the model comparison.

The objective of this study is to develop a local-first course-grounded learning system and quantify the trade-off between local inference performance and response grounding across five quantized LLMs. The system searches smaller child items, restores their parent sections as generation context, and preserves source links for conversational answers and lecturer-reviewed quiz candidates. The evaluation uses 20 course-grounded questions to compare generation speed, latency, VRAM use, and four RAGAS metrics. The contribution is an implemented, source-traceable parent-child workflow and a descriptive model comparison within that workflow. In this paper, course-grounded refers to using identifiable course materials as the basis for generated content; it does not denote a guided-learning pedagogy or a demonstrated improvement in learning outcomes.

2. METHOD

The research followed four stages adapted from the 4-D development model: Define, Design, Develop, and Disseminate. Define identified learning requirements, course materials, and available computing resources. Design specified the architecture, knowledge base, interface, and evaluation instruments. Develop integrated the local LLM, RAG pipeline, visual retrieval module, and progress tracker. The modified Disseminate stage evaluated the prototype in a laboratory environment and through initial user testing.

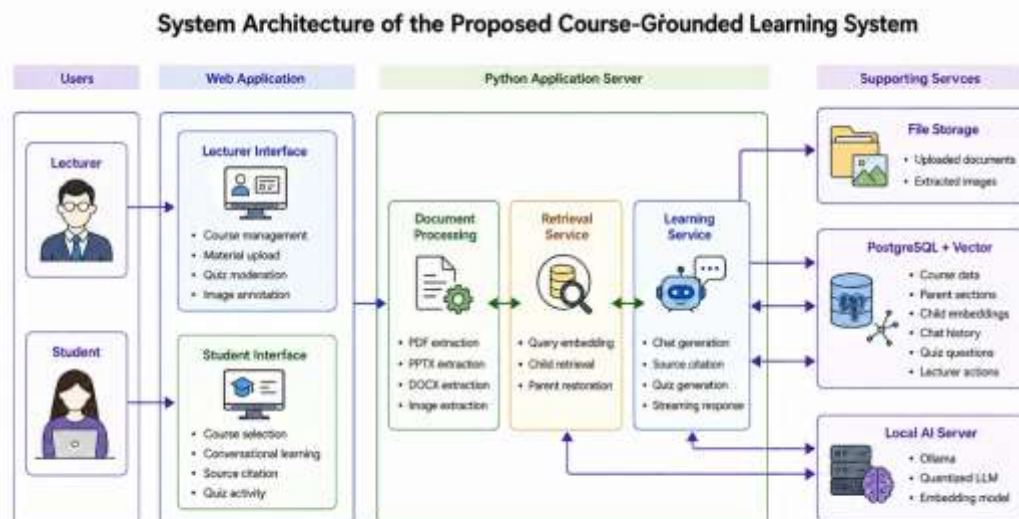


Figure 1. Research procedure of the proposed course-grounded learning system.

2.1. System Architecture

The proposed system consists of users, a web application, a Python application server, and supporting data and local AI services (Figure 2). Lecturer functions include course management, material upload, quiz moderation, and image annotation. Student functions include course selection, course-grounded conversation, source citation, and quiz activities.

The web application sends requests to the Python server, which processes course materials, retrieves relevant content, generates answers and quiz questions, and records interactions. File storage holds uploaded documents and extracted images. PostgreSQL with vector support stores course data, structured sections, embeddings, conversations, quizzes, and lecturer actions. Ollama runs the quantized language and embedding models locally, enabling processing within institution-controlled infrastructure.

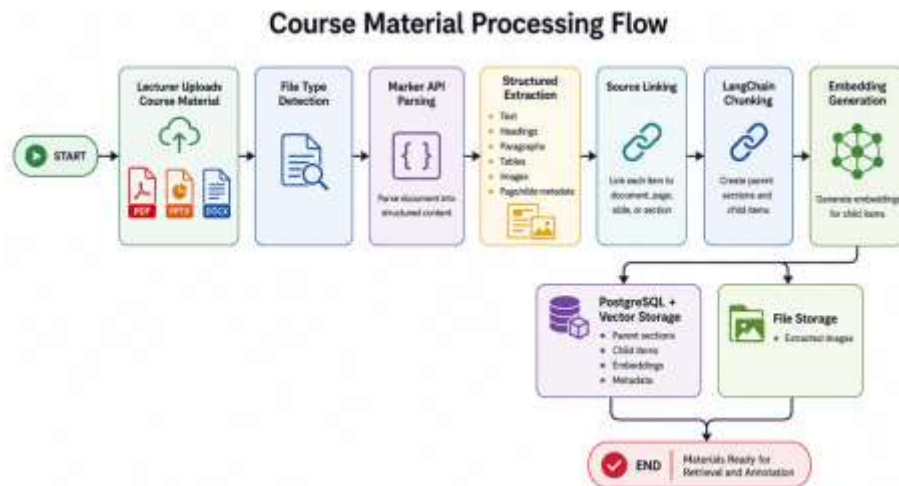


Figure 2. System architecture of the proposed course-grounded learning system.

2.2. Course Material Processing and Parent-Child Retrieval

The system accepts PDF, PPTX, and DOCX course documents. After upload, the processing service detects the format and sends the document to the Marker API, which converts it into structured text, headings, paragraphs, tables, images, and page or slide metadata [18]. Each item remains linked to its original document and location. LangChain then creates complete parent sections and smaller child retrieval units [19]. Embeddings are generated for the child items and stored with the source metadata; after a child item is retrieved, its parent identifier restores the complete section used as generation context. This separation allows a short passage to act as the search target while its enclosing section provides the surrounding explanation. In Table 1, the three P01 child items refer to the same parent section, so a match to one item can recover that section and its source links. The table illustrates the representation rather than an experimental comparison with other chunking methods

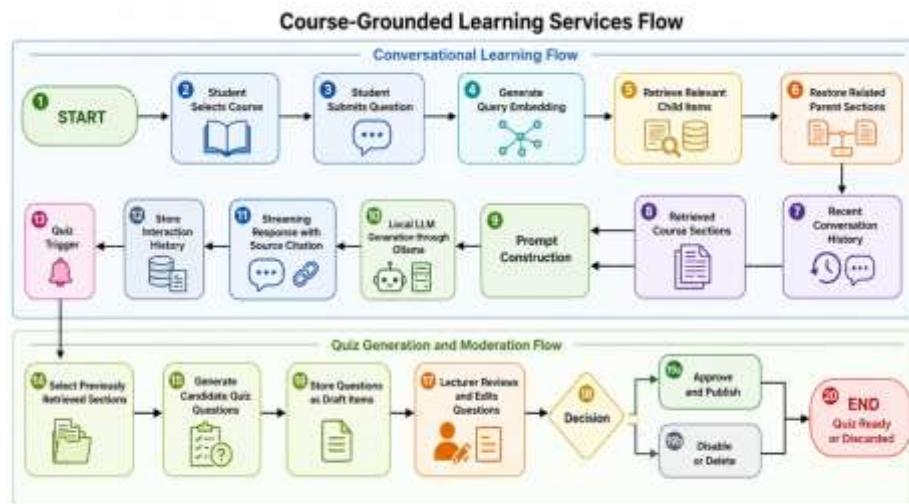


Figure 3. Course material processing flow in the proposed course-grounded learning system.

Table 1. Example of parent-child document representation for the proposed RAG process.

Parent ID	Parent Section	Child Item	Source
P01	Introduction to Machine Learning	Machine learning allows a computer system to learn patterns from training data.	Document A, p. 2
P01	Introduction to Machine Learning	Supervised learning uses labeled data to train a prediction model.	Document A, p. 2
P01	Introduction to Machine Learning	Common supervised tasks include classification and regression.	Document A, p. 3
P02	Neural Network Architecture	A neural network consists of input, hidden, and output layers.	Document A, p. 5
P02	Neural Network Architecture	Each neuron applies a weighted sum followed by an activation function.	Document A, p. 5
P02	Neural Network Architecture	Hidden layers learn intermediate representations from input data.	Document A, p. 6

2.3. Course-Grounded Learning Services

A student selects a course and submits a question. The system embeds the query, retrieves relevant child items, maps them to their parent sections, and separates the restored course context from recent conversation history during prompt construction. The prompt is sent to the local LLM through Ollama, and the response is streamed with references to the related document, page, slide, or section. The system records the question, answer, retrieved sections, citations, and interaction time.

When the quiz trigger is reached, previously retrieved sections become the source for candidate questions. Each draft stores the question, answer options, correct answer, explanation, and source metadata. A lecturer reviews and edits every item before approving it; unsuitable items are disabled or deleted. Adaptive assessment therefore means interaction-scoped question generation, not psychometric adaptive testing or automatic mastery estimation.

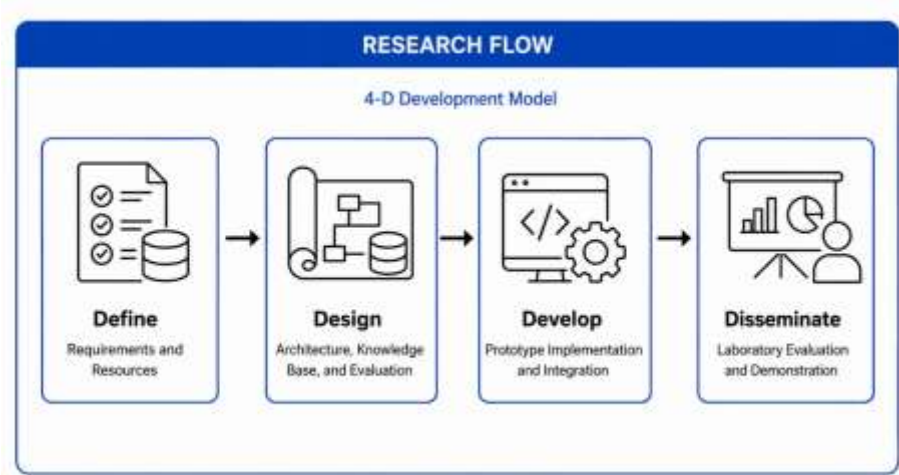


Figure 4. Course-grounded conversational learning and quiz moderation flow.

2.4. Data Storage and Local AI Services

File storage contains original course documents and extracted images, each linked to its source document and location. PostgreSQL stores course information, accounts, parent sections, child items, source metadata, conversations, quiz questions, and lecturer actions [20]. The pgvector extension stores child embeddings and supports similarity search [21]. Ollama serves the local AI components [22]; Nomic Embed Text generates vector representations [23], while Qwen3-14B produces grounded answers, summaries, and quiz candidates [24]. The tested configuration keeps documents, retrieved content, responses, and lecturer actions within institution-controlled infrastructure, but the study does not claim formal privacy or regulatory compliance.

2.5. Evaluation Setup

2.5.1. Dataset and Models

The evaluation used 20 questions derived from six meetings of a Problem Solving and Debugging course. Topics included technical problem identification, systematic troubleshooting, root-cause analysis, and error

analysis. Each question had a reference answer derived from the corresponding course material. For performance testing, every question was processed three times by each of five models, yielding 300 measurements. For RAG evaluation, each model generated one answer per question, yielding 100 course-grounded responses.

Table 2. Quantized language models used in the evaluation.

ID	Model	Size	Quantization
LM01	Mistral Instruct	7B	Q4_K_M
LM02	Qwen2.5 Instruct	7B	Q4_K_M
LM03	Qwen3	14B	Q4_K_M
LM04	Phi-4	14B	Q4_K_M
LM05	DeepSeek-R1	14B	Q4

All models ran locally through Ollama with temperature 0 and a 2048-token context. The same prompts, questions, and retrieval process were used across models in one common local computing environment.

2.5.2. Evaluation Metrics

Local inference evaluation measured generation speed, Time to First Token (TTFT), total response time, output length, and peak VRAM use. Mean generation speed, TTFT, response time, and VRAM use were calculated for each model; output tokens were examined because response length affects total duration.

Table 3. Metrics used for local inference performance.

ID	Metric	Unit	Purpose
PM01	Generation speed	token/s	Token generation rate
PM02	Time to first token	s	Initial response delay
PM03	Total response time	s	Complete generation duration
PM04	Output length	token	Generated response length
PM05	VRAM use	GB	Graphics-memory requirement

RAG quality was evaluated with four RAGAS metrics [25]. Context Precision and Context Recall assess the retrieved course material, whereas Faithfulness and Answer Relevancy assess how the model uses that context and addresses the question. The average score for each metric was computed over the 20 questions for each language model

Table 4. Metrics used for parent-child RAG evaluation.

ID	Metric	Evaluated aspect
RM01	Context Precision	Relevance of retrieved context
RM02	Context Recall	Coverage of reference information
RM03	Faithfulness	Support of response by retrieved context
RM04	Answer Relevancy	Relation of response to the question

3. RESULTS AND DISCUSSION

3.1. Prototype Implementation

The prototype implements lecturer functions for course-material processing and student functions for course-grounded conversation and quiz activities. Lecturers upload course documents, monitor parent-child processing, and annotate extracted images. Students select a course, ask questions, inspect source-linked answers, and complete lecturer-approved quizzes. The same interaction history is used to choose the source sections for candidate quiz generation.

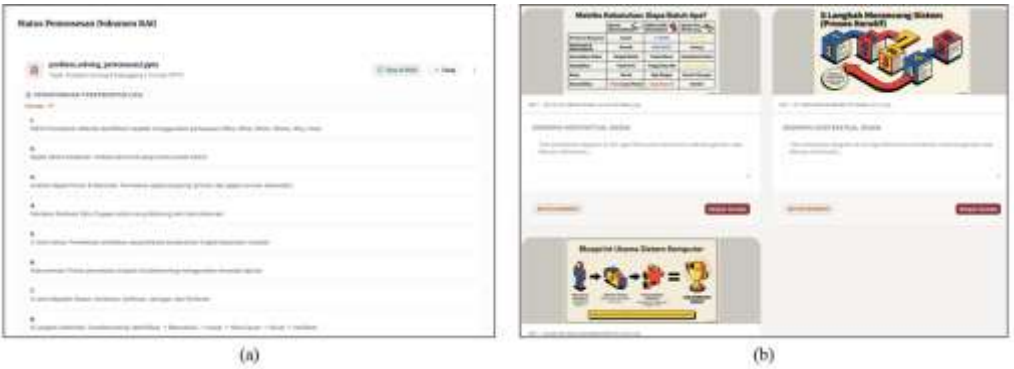


Figure 5. Lecturer interface for material processing and image annotation.

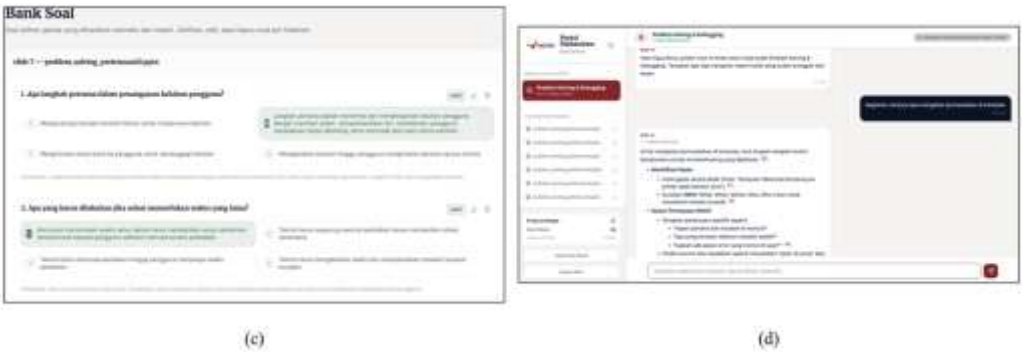


Figure 6. Student interface for quiz activities and course-grounded conversation.

3.2. Local Inference Performance

Table 5 reports the averages from 60 measurements per model. Mistral 7B and Qwen2.5 7B achieved the highest generation speeds, 58.95 and 57.14 tokens/s, while using about 5 GB of VRAM. The 14B models generated approximately 28-29 tokens/s and required approximately 10 GB. Qwen3 14B had the longest responses (1,678.27 tokens) and highest mean response time (60.05 s), illustrating how output length contributes to total latency.

Table 5. Local inference performance of the evaluated language models.

Model	TPS	TTFT (s)	Mean RT (s)	Median RT (s)	Tokens	VRAM (GB)
Mistral 7B	58.95	0.23	12.74	7.91	709.63	5.42
Qwen2.5 7B	57.14	0.29	13.41	12.66	727.62	5.11
Qwen3 14B	28.19	0.38	60.05	58.95	1678.27	10.19
Phi-4 14B	29.14	0.28	28.97	28.00	823.95	10.16
DeepSeek-R1 14B	29.28	0.32	32.77	31.51	935.75	10.06

3.3. Parent-Child RAG Quality

Under the shared parent-child pipeline, Qwen3 14B achieved the highest mean Faithfulness (0.882) and Answer Relevancy (0.906). Context Precision ranged from 0.828 to 0.840 and Context Recall from 0.776 to 0.789. Phi-4 and Qwen3 shared the highest Context Precision (0.840) and Context Recall (0.789). These aggregate scores describe the complete evaluated configurations; they do not isolate the effect of parent-child retrieval.

Table 6. Parent-child RAG quality of the evaluated language models.

Model	Context Precision	Context Recall	Faithfulness	Answer Relevancy
Mistral 7B	0.828	0.776	0.797	0.844
Qwen2.5 7B	0.832	0.783	0.830	0.866
Qwen3 14B	0.840	0.789	0.882	0.906
Phi-4 14B	0.840	0.789	0.859	0.888
DeepSeek-R1 14B	0.840	0.784	0.845	0.877

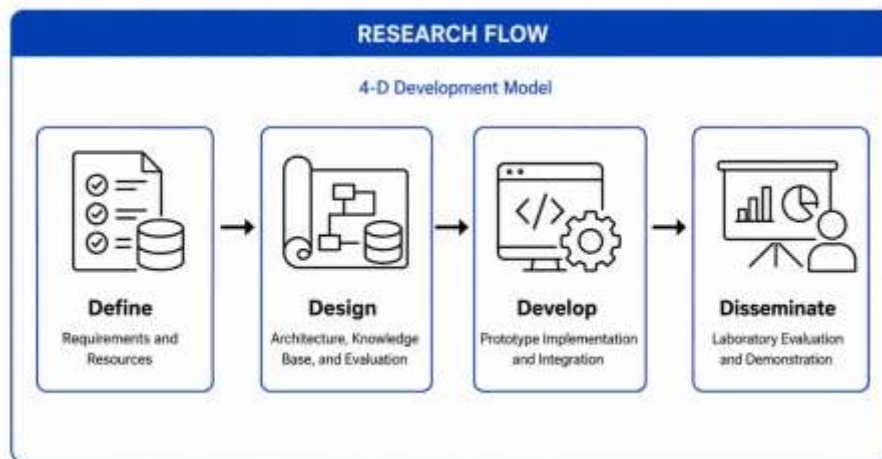


Figure 7. Comparison of parent-child RAGAS scores across evaluated models.

3.4. Discussion and Limitations

The evaluation was limited to 20 questions derived from six meetings of a single Problem Solving and Debugging course, five quantized models, and one local computing environment. The 7B models were faster and used roughly half the VRAM of the 14B models, while the 14B models obtained higher Faithfulness and Answer Relevancy. Qwen3 14B was selected for the proposed system because it provided the strongest response grounding under the tested configuration, despite requiring about 10 GB of VRAM and producing longer responses.

The parent-child workflow contributes a source-traceable link between passage-level retrieval and section-level generation context. Restored sections also provide the source scope for quiz candidates, which remain subject to lecturer approval. This is a functional contribution of the implemented architecture. Although the retrieval-level scores occupy a narrow range, the shared pipeline alone does not explain their differences across models; per-question retrieved contexts and evaluator settings would be needed to determine the cause. The higher response-level means for Qwen3 support its selection within this experiment, without establishing statistical significance or superiority in other courses.

The evaluation was limited to 20 questions from one course, five quantized models, and one local computing environment. No flat-chunk, child-only, or no-retrieval baseline was evaluated, so the independent benefit of restoring parent sections cannot be quantified. The study also did not measure student learning outcomes, lecturer assessment, concurrent-user performance, or sensitivity to retrieval parameters. Future work should compare retrieval strategies under matched settings, report uncertainty across questions, and extend testing to more courses and educational outcomes.

4. CONCLUSION

This study contributes a source-traceable parent-child workflow for local course-grounded generation and an empirical characterization of its quality-resource trade-off across five quantized LLMs. Child items support semantic matching, while restored parent sections provide the course context and source links used for answers and lecturer-reviewed quiz candidates. Within this workflow, Context Precision ranged from 0.828 to 0.840 and Context Recall from 0.776 to 0.789. These results characterize the implemented pipeline but do not demonstrate an improvement over flat retrieval. The 7B models achieved 57.14-58.95 tokens/s using approximately 5 GB of VRAM, whereas Qwen3 14B obtained the highest Faithfulness (0.882) and Answer Relevancy (0.906), with 10.19 GB of VRAM and a mean response time of 60.05 s. The findings inform model selection for this local configuration. Controlled retrieval comparisons and broader classroom evaluation are needed to establish the specific benefit of parent restoration and its educational value.

5. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support provided by the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia (Kemendikristek) through the Penelitian Dosen Pemula (PDP) program administered via BIMA. The authors also thank Institut Bisnis dan Teknologi Indonesia for its institutional support during the development and evaluation of this research.

REFERENCES

- [1] Moodle, "Resources," MoodleDocs, Jul. 1, 2024. [Online]. Available: <https://docs.moodle.org/405/en/Resources>. Accessed: Sep. 8, 2026.
- [2] Moodle, "Activities," MoodleDocs. [Online]. Available: <https://docs.moodle.org/405/en/Activities>. Accessed: Sep. 8, 2026.
- [3] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, 2023, <https://doi.org/10.1016/j.lindif.2023.102274>.
- [4] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "LLM.int8(): 8-bit matrix multiplication for transformers at scale," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 30318-30332.
- [5] J. Lin, J. Tang, H. Tang, S. Yang, X. Dang, C. Gan, and S. Han, "AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration," *Proc. Machine Learning and Systems*, vol. 6, pp. 87-100, 2024.
- [6] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459-9474.
- [7] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," *arXiv:2312.10997*, 2023, <https://doi.org/10.48550/arXiv.2312.10997>.
- [8] W. Gan, Z. Qi, J. Wu, and J. C.-W. Lin, "Large language models in education: Vision and opportunities," *arXiv:2311.13160*, 2023, <https://doi.org/10.48550/arXiv.2311.13160>.
- [9] D. T. K. Ng, C. W. Tan, and J. K. L. Leung, "Empowering student self-regulated learning and science education through ChatGPT: A pioneering pilot study," *British Journal of Educational Technology*, vol. 55, no. 4, pp. 1328-1353, 2024, <https://doi.org/10.1111/bjet.13454>.
- [10] S. Wang et al., "Large language models for education: A survey and outlook," *arXiv:2403.18105*, 2024, <https://doi.org/10.48550/arXiv.2403.18105>.
- [11] Z. Levonian, O. Henkel, C. Li, and M.-E. Postle, "Designing safe and relevant generative chats for math learning in intelligent tutoring systems," *Journal of Educational Data Mining*, vol. 17, no. 1, pp. 66-97, 2025, <https://doi.org/10.5281/zenodo.14751365>.
- [12] T. Zhang et al., "Ask NAEP: A generative AI assistant for querying assessment information," *Journal of Measurement and Evaluation in Education and Psychology*, vol. 15, special issue, pp. 378-394, 2024, <https://doi.org/10.21031/epod.1548128>.
- [13] T. Zheng, W. Li, J. Bai, W. Wang, and Y. Song, "Assessing the robustness of retrieval-augmented generation systems in K-12 educational question answering with knowledge discrepancies," *arXiv:2412.08985v1*, 2024, <https://doi.org/10.48550/arXiv.2412.08985>. [Online]. Available: <https://arxiv.org/abs/2412.08985v1>.
- [14] B. Malisetty and A. J. Perez, "Evaluating quantized Llama 2 models for IoT privacy policy language generation," *Future Internet*, vol. 16, no. 7, p. 224, 2024, <https://doi.org/10.3390/fi16070224>.
- [15] X. Duan, F. Nwanganga, and C. Wang, "CODE-GEN: A human-in-the-loop RAG-based agentic AI system for multiple-choice question generation," *arXiv:2604.03926*, 2026, <https://doi.org/10.48550/arXiv.2604.03926>.
- [16] S. Elkins, E. Kochmar, J. C. K. Cheung, and I. Serban, "How useful are educational questions generated by large language models?" *arXiv:2304.06638*, 2023, <https://doi.org/10.48550/arXiv.2304.06638>.
- [17] B. Tarun, H. Du, D. Kannan, and E. F. Gehringer, "Human-in-the-loop systems for adaptive learning using generative AI," *arXiv:2508.11062*, 2025, <https://doi.org/10.48550/arXiv.2508.11062>.
- [18] Datalab, "Marker: Convert documents to Markdown, JSON, chunks, and HTML," 2026. [Online]. Available: <https://github.com/datalab-to/marker>. Accessed: Jul. 30, 2026.
- [19] LangChain, "Text splitter integrations," 2026. [Online]. Available: <https://docs.langchain.com/oss/python/integrations/splitters>. Accessed: Jul. 30, 2026.
- [20] PostgreSQL Global Development Group, "PostgreSQL documentation," 2026. [Online]. Available: <https://www.postgresql.org/docs/>. Accessed: Jul. 30, 2026.
- [21] pgvector, "Open-source vector similarity search for PostgreSQL," 2026. [Online]. Available: <https://github.com/pgvector/pgvector>. Accessed: Jul. 30, 2026.
- [22] Ollama, "Ollama documentation," 2026. [Online]. Available: <https://docs.ollama.com/>. Accessed: Jul. 30, 2026.
- [23] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar, "Nomic Embed: Training a reproducible long context text embedder," *arXiv:2402.01613*, 2024, <https://doi.org/10.48550/arXiv.2402.01613>.
- [24] A. Yang et al., "Qwen3 technical report," *arXiv:2505.09388*, 2025, <https://doi.org/10.48550/arXiv.2505.09388>.
- [25] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, "RAGAS: Automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics: System Demonstrations*, 2024, pp. 150-158, <https://doi.org/10.18653/v1/2024.eacl-demo.16>.