

Artificial Intelligence-Based Study Notes Generation Using Generative AI and Retrieval-Augmented Generation: A Comprehensive Review of Retrieval Methods, Prompting, Educational Evidence, and System Architecture

Korade Apeksha Maruti, Prof. Bangar Abhishek Popat, Prof. Bhosale S.B.

Dept. of Computer Engineering
Jaihind College of engineering, Pune, India

Abstract — Learners often spend a great deal of time condensing lecture slides, textbook chapters and research papers into brief revision notes. Large language models (LLMs) from the generative AI family can draft such notes within seconds, but a model that is not grounded in a source may write confident text that the material never states. Retrieval-augmented generation (RAG) lowers this risk by supplying the model with passages fetched from the learner's own documents. This paper offers a focused narrative review of verified literature on AI-driven study notes generation with generative AI and RAG. Three linked areas are covered: retrieval and RAG techniques (chunking, dense embeddings, retrievers and advanced pipelines); generative models and prompting (LLMs, structured prompt frameworks and hallucination); and empirical evidence on generative AI in engineering and computing courses. The reviewed work indicates that RAG was proposed to anchor generation in retrieved text, that dense retrieval outperformed a strong sparse baseline in open-domain question answering, and that in a quasi-experiment with 26 students, structured prompts produced fuller answers than free-style prompts. Even so, most educational studies capture student perceptions instead of learning outcomes, rely on small samples, and none of the reviewed studies directly measures note quality, faithfulness or retention. The paper contributes a comparison of representative studies, a conceptual seven-layer architecture that links ingestion, retrieval, grounded generation, verification and learner feedback, and a ranked list of research gaps.

Keywords— Study Notes Generation, Generative AI, Retrieval-Augmented Generation, Large Language Models, Prompt

I. INTRODUCTION

Notes help learners arrange, shorten and revisit what they study, but writing them from long slide decks and textbook chapters takes a lot of effort. Generative AI (GenAI) tools built on large language models (LLMs) can summarise and explain material when asked. A recent scoping review lists several educational benefits, including instant feedback, personalised learning, help with explaining difficult concepts and greater learner engagement [1].

That review, together with the studies it covers, also records drawbacks. In one study, students noticed hallucinations, missing details and misleading answers, and ChatGPT performed poorly on certain technical topics [9]. Such failures are serious for a notes generator, because the notes are used later for revision. Retrieval-augmented generation (RAG) [2] offers a natural remedy: rather than depending only on what the model memorised during training, the system fetches passages

from the learner's own documents and writes the notes from them.

This review looks at three strands together: (i) retrieval and RAG methods, (ii) generative models and prompting, and (iii) educational evidence on how students and teachers use GenAI. It asks how well established the evidence is for AI-written study notes and puts forward a conceptual architecture aimed at the weaknesses found.

II. REVIEW METHODOLOGY

The work is a focused narrative review and does not follow a systematic protocol. The evidence comes from two groups of sources. The first is a recent PRISMA-ScR scoping review of GenAI in engineering and computing education, which screened 146 records from Scopus, Web of Science and ERIC and kept 24 empirical studies [1]. Details of the educational studies discussed here, including their designs and outcomes, are taken from the tables and text of that review and were not

re-extracted from the original papers. The second group consists of standard technical references on RAG, dense retrieval, sentence embeddings and hallucination [2]–[5], [8]. No additional database search was performed for this version, and no study counts are claimed. Work on RAG-based educational tools is therefore under-represented, a limitation acknowledged in Section VIII.

III. RETRIEVAL-AUGMENTED GENERATION FOR STUDY MATERIAL

1. Role of RAG in Study Notes Generation

RAG joins information retrieval with text generation. In a study notes generator, the system first looks through the student's uploaded material for content tied to a chosen topic. The passages found are handed to an LLM, which turns them into clear, well-organised notes. Because the notes are built from the available learning material, they are tied to it and do not rest solely on knowledge acquired during model training [2], [15]. Suppose a student asks for notes on one topic from a set of lecture slides. The system can locate the matching sections and summarise them, and it can attach source references so the student can check each point. Retrieval on its own, however, does not make every generated statement correct, so the notes should still be compared with the original material [8].

2. Document Processing, Embeddings and Retrieval

Uploaded documents must first be turned into plain text. The extracted text is split into smaller units, usually called chunks, and each chunk is converted into a numerical vector known as an embedding and kept in a searchable index. When a student types a topic or question, the system compares the query vector with the stored vectors and picks the closest passages [3], [4], [15].

Retrieval quality depends in part on how documents are split and how passages are chosen. A chunk that is too large may mix several topics, while one that is too small can separate key details from the explanation around them. The chunking approach should therefore keep enough surrounding context intact. Sentence embeddings let the system match related content even when the query uses different wording from the source [4].

3. Improving the Retrieval Process

A basic RAG pipeline fetches relevant sections and passes them straight to the language model. Stronger pipelines refine the search query, mix keyword-based and semantic search, and rank the retrieved passages by relevance [5], [15].

These refinements suit notes generation well. A student might enter a broad heading such as “Introduction”, which does not point to a specific topic. Query refinement can make such a search more precise, and reranking can bring the most useful passages to the top. Once retrieval is finished, the language model writes structured notes from the chosen content. The system should also say so when the available material is not sufficient to answer the request.

In short, RAG is a practical way of linking generative AI to a student's own study resources. Its value rests on the quality of document processing, retrieval and the notes finally produced. More evaluation is needed to establish how faithfully RAG-based notes reflect the source and how well they support learning.

IV. GENERATIVE MODELS AND PROMPTING

1. LLMs and Structured Prompts

GenAI tools rely on LLMs such as GPT-4, Gemini, Claude and LLaMA 2, and users steer them through prompts. One framework splits a prompt into four parts: context (the role assigned to the model), instructions, input content and output format [1]. An education-focused set of parts consists of a verb, a focus, a context, alignment with the intended goals, and constraints [1]. The CRISPE framework has five stages: capacity and role, insight, statement, personality and experiment [1]. Applied to notes generation, the role could be a tutor, the insight is the retrieved context, the statement is the note-writing task, the personality is the tone, and the experiment is producing more than one draft.

Evidence on prompt structure is thin but points in one direction. In a quasi-experiment with 26 undergraduate computer science students, prompts written with CRISPE elements drew more complete and detailed answers than free-style prompts, and the authors advised using them under an instructor's guidance [6]. A mixed-methods study of 47 students found that those who iterated and reworded their requests were more successful, and that the usefulness of the help received was tied to how clear and precise the prompts were [7].

2. Hallucination and Trust

A survey of hallucination in natural language generation separates output that contradicts its source from output that cannot be checked against the source [8]. In a quasi-experiment with 29 software engineering students, the ChatGPT group reported significantly more frustration, no productivity effect was found, and ChatGPT was weak on specific topics [9].

Another study, of 84 students working on programmable logic controllers, found that 97.37% were unsure about the answers ChatGPT gave [10]. This kind of low trust translates directly into a design requirement: every statement in the notes should be traceable to its source.

V. EDUCATIONAL EVIDENCE

Benefits noted in the literature include structured, condensed help with programming problem-solving [14], immediate and detailed information [7], and improved written reports [12], [13]. The CS50 Duck, a GPT-4 assistant limited to course rules and guidelines, was found beneficial by most of the 500 students in the second stage, and its authors suggest that tools built by teachers may curb misconduct such as plagiarism [11]. An AI bot that analysed StandUp reports raised report quality and communication skills among 179 software engineering students [12]. In a pilot involving seven students, ChatGPT strengthened the introduction and conclusion of laboratory reports, while critical thinking, reading and composing improved the least, and teacher supervision was recommended [13].

Reported concerns include over-reliance, a drift away from learning goals, and teachers who are not prepared for the technology [1], [13]. The scoping review further notes that many studies record perceptions while few show how GenAI tools help the learning process, and that many do not report students' earlier experience with GenAI [1]. Course-specific tools [11] resemble a grounded notes generator, which hints that students accept generation restricted to course material. No reviewed study tests this for note generation, however, so the point remains an inference.

VI. COMPARATIVE ANALYSIS

Table I presents representative studies. The retrieval sources [2], [3] supply the technical foundation for grounded generation, whereas the educational studies [6], [7], [9], [11]–[14] show how students react to GenAI tools in their coursework. None of the rows evaluates RAG-generated study notes, which is the central gap taken up in Sections VII and VIII.

VII. PROPOSED CONCEPTUAL ARCHITECTURE

Fig. 1 shows a conceptual architecture drawn from the reviewed sources; it has not been validated experimentally. Study sources supplied by the learner (layer 1) are processed through text or OCR extraction, cleaning and chunking, with source metadata retained (layer 2). The chunks are embedded and saved in a vector index (layer 3) [3], [4]. Every topic or heading is turned into a query, which is rewritten and re-ranked when necessary (layer 4) [5]. The LLM then writes notes from the retrieved context using a structured prompt containing role, context, task and format (layer 5) [1], [2]. A verification layer adds source citations and marks statements that the retrieved text does not back up (layer 6), targeting the hallucination and trust issues reported in [8]–[10]. The learner edits, rates and exports the notes (layer 7). Feedback adjusts ingestion, retrieval and prompts only after independent review, so that the teacher or learner stays in control, as suggested in [6], [13].

Table 1: Comparison of Representative Studies Relevant to AI-Based Study Notes Generation

Author, year	Objective	Method / tool	Major reported findings	Limitations
Lewis et al., 2020 [2]	Anchor generation in retrieved text	Pretrained generator with a retriever over an external index	Blended model memory with retrieved knowledge; strong open-domain QA results	QA benchmarks; not an educational study
Karpukhin et al., 2020 [3]	Fetch passages for question answering	Dual-encoder dense retrieval compared with BM25	Better top-20 retrieval accuracy than the BM25 baseline	QA benchmarks; not tested on study material
Wang et al., 2024 [6]	Effect of prompt structure in a flipped class	ChatGPT with CRISPE prompts; quasi-experiment	CRISPE prompts produced fuller, more detailed answers; 26 students	Single topic (TCP sockets); small sample
Denny et al., 2024 [14]	Prompt-driven programming exercise	ChatGPT plus Promptly tool; 255 students	Structured support; some students hesitant and anxious about code generation	Programming task, not note writing

Author, year	Objective	Method / tool	Major reported findings	Limitations
Hou et al., 2024 [7]	Help-seeking behaviour with GenAI	Survey and interviews; 47 students	Clear, precise prompts associated with better help	Small, self-reported sample
Choudhuri et al., 2024 [9]	GenAI in software engineering tasks	Quasi-experiment; 15 vs 14 students	More frustration; no productivity effect; weak on specific topics	29 students; one kind of task
Liu et al., 2024 [11]	Course-specific AI assistant	CS50 Duck (GPT-4); 70 students, then 500	Most students found the tools useful; may reduce misconduct	Single course; based on perceptions
Neyem et al., 2024 [12]	AI feedback on StandUp reports	Slack bot; 179 students; mixed methods	Improved report quality and communication	Concerns reports, not study notes
Kim et al., 2024 [13]	ChatGPT in lab report writing	Qualitative; 7 students	Introduction and conclusion improved most; critical thinking least	7 students; supervision required

Reported values cannot be compared directly because tasks, datasets and designs differ. Educational study details are as reported in [1].

VIII. RESEARCH GAPS AND FUTURE DIRECTIONS

Gaps supported by the reviewed evidence. (i) Note quality is not evaluated directly: the educational studies assess perceptions, report quality or programming tasks rather than generated study notes [1], [12], [13]. (ii) Samples are small, with several studies enrolling between 7 and 47 students [6], [7], [9], [13]. (iii) Hallucination and low trust continue to be reported [8]–[10]. (iv) Many designs focus on perceptions and do not show how the tools influence learning [1]. (v) Background information is missing, since students' previous GenAI experience is often unreported [1]. (vi) Studies are geographically concentrated, with 11 of the 24 included studies from the United States [1]. (vii) Coverage here is limited, because RAG-based educational tools were not searched systematically.

Directions requiring investigation. Priorities include benchmark collections of study material paired with expert reference notes; faithfulness and citation-accuracy metrics for generated notes; controlled studies that use quizzes to measure retention instead of relying on perceptions; comparisons between RAG-based and ungrounded LLM notes; chunking and retrieval settings tuned to lecture material; support for multilingual and scanned sources; privacy handling of uploaded material; and teacher-in-the-loop designs that reduce over-reliance [1], [13].

IX. CONCLUSION

AI-based study notes generation draws on well-developed components: dense retrieval, sentence embeddings and RAG on the technical side, and structured prompting for generation. The educational evidence suggests that students appreciate quick, structured help yet stay wary of its accuracy, and that tools tied to a course are received favourably. Still, the evidence is dominated by small, perception-based studies of general-purpose chatbots, and none evaluates RAG-generated notes. A grounded, citation-based pipeline with a verification layer and human review, as described in the proposed architecture, is a sensible design direction, but it needs empirical validation with larger samples and direct measures of note quality and learning.

REFERENCES

1. J. Á. Ariza, M. B. Restrepo, and C. H. Hernández, "Generative AI in engineering and computing education: A scoping review of empirical studies and educational practices," *IEEE Access*, vol. 13, pp. 30789–30810, 2025, doi: 10.1109/ACCESS.2025.3541424.
2. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.
3. V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6769–6781.
4. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992.

5. Y. Gao et al., “Retrieval-augmented generation for large language models: A survey,” 2023, arXiv:2312.10997.
6. M. Wang, M. Wang, X. Xu, L. Yang, D. Cai, and M. Yin, “Unleashing ChatGPT’s power: A case study on optimizing information retrieval in flipped classrooms via prompt engineering,” *IEEE Trans. Learn. Technol.*, vol. 17, pp. 629–641, 2024.
7. I. Hou, S. Mettelle, O. Man, Z. Li, C. Zastudil, and S. MacNeil, “The effects of generative AI on computing students’ help-seeking preferences,” in *Proc. 26th Australas. Comput. Educ. Conf.*, Jan. 2024, pp. 39–48.
8. Z. Ji et al., “Survey of hallucination in natural language generation,” *ACM Comput. Surv.*, vol. 55, no. 12, Art. no. 248, 2023.
9. R. Choudhuri, D. Liu, I. Steinmacher, M. Gerosa, and A. Sarma, “How far are we? The triumphs and trials of generative AI in learning software engineering,” in *Proc. IEEE/ACM 46th Int. Conf. Softw. Eng.*, Apr. 2024, pp. 1–13.
10. S. Stankovski, G. Ostojic, S. Tegeltija, M. Stanojevic, M. Babic, and X. Zhang, “Generative AI applications and tools in engineering education,” in *Proc. 23rd Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, Mar. 2024, pp. 1–4.
11. R. Liu, C. Zenke, C. Liu, A. Holmes, P. Thornton, and D. J. Malan, “Teaching CS50 with AI: Leveraging generative artificial intelligence in computer science education,” in *Proc. 55th ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2024, pp. 750–756.
12. A. Neyem, J. P. Sandoval Alcocer, M. Mendoza, L. Centellas-Claros, L. A. Gonzalez, and C. Paredes-Robles, “Exploring the impact of generative AI for StandUp report recommendations in software capstone project development,” in *Proc. 55th ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2024, pp. 951–957.
13. D. Kim, A. Majdara, and W. Olson, “A pilot study inquiring into the impact of ChatGPT on lab report writing in introductory engineering labs,” *Int. J. Technol. Educ.*, vol. 7, no. 2, pp. 259–289, Mar. 2024.
14. P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves, “Prompt problems: A new programming exercise for the generative AI era,” in *Proc. 55th ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2024, pp. 296–302.