Automatic Generation of Study Texts from Lecture Slides

William Massami Costa Harada[1], **Fabio Santos da Silva**[1]

[1]Escola Superior de Tecnologia – Universidade do Estado do Amazonas (EST-UEA)
Grupo de Pesquisa em Sistemas Inteligentes

{wmch.eng22, fssantos}@uea.edu.br

Abstract. *Generative artificial intelligence has expanded the potential for automated educational content generation to support teaching and independent learning. Among these resources, lecture slides play a central role, yet their concise visual format limits its use for asynchronous study by students. This work proposes a method to automatically transform slide decks into self-contained study documents. The solution integrates visual parsing, iterative discourse reconstruction, and knowledge retrieval to ground implicit lecture concepts. Experiments across four multimodal LLMs show the generation of coherent materials while highlighting operational trade-offs.*

1. Introduction

Recent advances in generative artificial intelligence, particularly Large Language Models (LLMs), have enabled intelligent systems capable of performing increasingly complex tasks across a wide range of domains. In education, these systems support learning by answering questions, explaining concepts, and guiding students throughout their studies [Chu et al. 2025]. When combined with the widespread availability of digital educational resources, LLMs can be further augmented to support both learning and teaching using course-specific materials [Swacha and Gracel 2025]. In particular, this capability assists instructors in preparing and delivering content, reducing administrative and pedagogical workload [Liang et al. 2025] while complementing, rather than replacing, human instruction [Moore and Lee 2024].

Among existing educational resources, slide presentations occupy an important role in undergraduate, graduate, and professional education because they are a product of the instructor's intellectual effort to synthesize the broader course syllabus and serve as a curated bridge between lectures and the course bibliography. However, because their primary function is to support oral delivery, slides rarely serve as effective stand-alone study resources, frequently containing short bullet points, tangential concepts that are not fully elaborated, and technical concepts introduced without explicit definitions. As a result, reading the slides in isolation does not convey the full depth of the material, primarily because critical nuances are delivered only through speech, and this spoken information tends to be retained less effectively by learners once slides are introduced [Wecker 2012]. Moreover, a substantial portion of their content is conveyed through the spatial organization of visual elements, requiring joint visual and textual reading to recover their intended meaning [Tanaka et al. 2023].

These characteristics expose an opportunity to leverage slides not only as presentation artifacts, but also as existing instructional resources from which intelligent systems can benefit to assist both instructors and students. While recent work has explored the use

of course-specific materials and retrieval-augmented approaches to support educational applications [Jacobs and Jaschke 2024], the potential of transforming widely used static lecture slides into complementary study resources to assist instructors remains underexplored. In this context, this work investigates how lecture slides can be leveraged through generative AI to support instructors in the preparation of supplementary educational materials. To this end, the proposed approach coordinates multiple specialized components, including multimodal layout parsing and targeted literature retrieval, to transform the information contained in lecture slides into structured learning materials.

To detail the proposed approach, this paper is organized as follows: Section 2 reviews related work, Section 3 describes the methodology details, Section 4 reports the experimental results, and Section 5 discusses the results, limitations and future work.

2. Related Work

The use of LLM-based agents to automate instructional design tasks has received increasing attention due to its potential to reduce the effort required to create educational materials. In this context, [Yao et al. 2026] propose a multi-agent framework aimed at the automatic generation of instructional materials such as syllabi, presentations, lesson scripts, and assessments. The architecture simulates collaboration among different roles present in the academic environment, such as professor, teaching assistant, and course coordinator, organizing interactions according to the ADDIE instructional framework [Branch and Varank 2009]. Additionally, the proposed solution offers different levels of automation and human supervision, ranging from fully autonomous execution to modes with iterative validation by specialists. The results obtained across five university courses indicated that the approach was capable of producing consistent and high-quality materials, highlighting especially the operating mode where the system previously receives a catalog containing the educator's preferences and requests feedback from them during inference time, as it presented the best human evaluations and demonstrated potential to significantly reduce the time dedicated to didactic planning. Whereas our work takes existing slides as input and convert them into written study texts, this system authors such materials from instructor specifications, leaving the slides as a resource to be presented rather than read independently.

Still in the context of automatic educational content generation, the use of agents has also been explored to generate presentations tailored to different contexts and target audiences. In this direction, [Rao et al. 2025] propose an agent responsible for the iterative creation and editing of slide presentations from natural language instructions, inferring the context of use, selecting appropriate presentation templates, and using information retrieval techniques to extract relevant content from user-provided documents while complementing it with information obtained from the web. Based on this material, the agent organizes the extracted information into a coherent presentation structure, adapting both the content and the visual design according to the intended audience and presentation objectives. The authors show that this approach is capable of producing visually consistent presentations aligned with the requested context while also supporting iterative editing cycles with minimal human intervention. In contrast to their focus on creating presentation-ready visuals, our work operates in the opposite direction, using slides as the source material for generating documents intended for asynchronous study.

Focusing on transforming presentations into teaching support materials, [Wang et al. 2025] propose a system aimed at the automatic generation of lecture scripts from multimodal presentations. The proposed architecture processes slides by extracting text, converting pages into images, and generating descriptions for visual elements. Then, it analyzes the material holistically, seeking to transform textual and visual information into a pedagogical narrative suitable for oral presentation. The results showed that the solution outperformed previous approaches that generated scripts from isolated and sequential slide analysis, producing more coherent and consistent narratives. The authors also observed that the explicit association between visual content and pedagogical narrative represents one of the most important components for system performance. Our work builds on this discourse-reconstruction idea, but treats the narrative as an intermediate representation required for the accuracy of the final generated text, adding a retrieval stage that grounds concepts the slides assume into a self-contained written document for asynchronous study.

Together, these works show that slides can serve as a structured source of knowledge amenable to automated processing. However, existing literature primarily focuses on authoring presentations or generating oral delivery scripts, leaving their transformation into asynchronous study materials largely underexplored. Inserted in this context, our work proposes a method for the automatic generation of study material from slide presentations, and conducts a quantitative analysis of the proposed approach.

3. Methodology

The proposed methodology employs an agentic framework to automatically generate comprehensive study materials from lecture slide presentations. We formulate this process as a coordinated sequence of reasoning and synthesis stages, in which the content of the original presentation is progressively transformed into intermediate representations to structure context, preserve factual grounding, and feed subsequent stages. Employing specialized components allows each phase of this process to operate with dedicated prompts, tools, and verification boundaries, effectively decoupling visual layout analysis from knowledge retrieval and text composition. Figure 1 presents an overview of the proposed method.

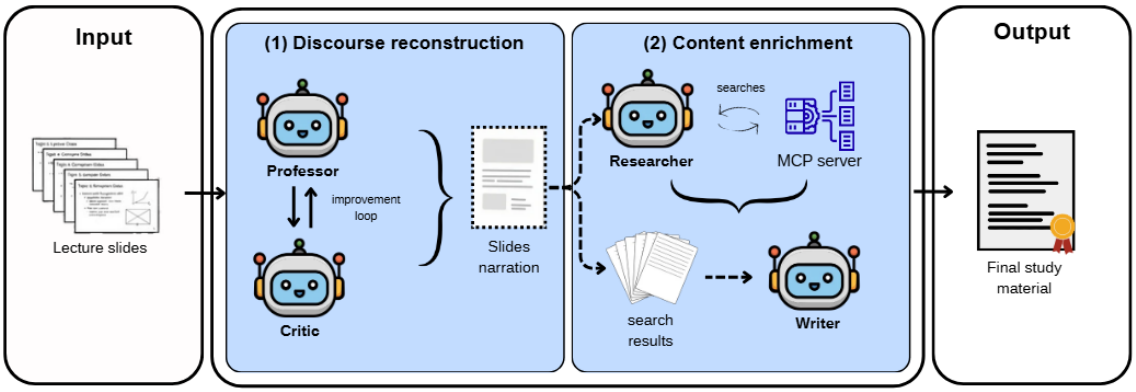

Figure 1. System architecture diagram.

The proposed architecture is organized as a directed graph of agents, executed over each window of consecutive slides, composed of a *Professor*, a *Critic*, and a *Researcher*,

followed by a *Writer* that unifies the artifacts produced from each window into the final document. The Professor reconstructs the lecture discourse for the group of slides, the Critic checks in an iterative loop that this reconstruction reads as a coherent spoken explanation, and the Researcher retrieves complementary information from external sources. Once every window of a lecture has been processed, the Writer synthesizes all of them into a single study material. The following sections describe each stage, together with the intermediate representations exchanged along the interaction of the agents.

3.1. Input Representation

To preserve the visual information present in slides, the presentation is received as a PDF and each page is rendered as an image, which is then read directly by a vision-language model. This approach leverages the capabilities acquired by these multimodal models during training on large collections of images and text, including recognizing printed text and visual elements, and relating them through their spatial arrangement on the page. Reading the slides in this form allows the model to operate directly on the original visual content without intermediate text extraction that could introduce noise. The rendered images are passed directly to the agents, ensuring that their elements are processed exactly as presented. To preserve narrative continuity, consecutive slides are grouped into windows and processed as a single unit.

The window size was determined through a preliminary sensitivity study in which windows of size $N \in \{1, 2, 3, 5, 8\}$ slides were evaluated by manual inspection and quantitative tracking of text density, which declined from 171 at $N = 1$ to 90 words/slide at $N = 8$. Larger windows produced more cohesive and less verbose discourse by allowing the model to resolve context across consecutive slides, however they also reduced the specificity of visual descriptions, with individual entities frequently being replaced by more general descriptions. A window of five slides was therefore adopted because it retains the cohesion benefits and factual accuracy while limiting the loss of visual details.

3.2. Lecture Narrative Reconstruction

Within each window, the first stage reconstructs the lecture discourse associated with the slides it contains. This is inspired by the work of [Wang et al. 2025], which converts multimodal presentations into a pedagogical narrative by making explicit the explanations and transitions that connect the slides. Since slide presentations are inherently shallow documents, downstream models struggle to generate detailed study notes directly from isolated bullet points. Expanding these sparse cues into an explicit verbal stream that also captures the visual arrangements provides the necessary contextual scaffolding for subsequent processing. However, since generating missing discourse with an LLM is probabilistic and prone to speculative drift, this reconstructed narrative is not treated as a final output. Instead, it serves as an intermediate representation subjected to iterative critique and refinement to ensure factual fidelity to the source presentation before final synthesis.

This reconstruction is performed by the *Professor*, an agent that receives the images of the slides in the window and, when the reconstruction is being revised, the critique produced in the previous iteration. This agent produces a single continuous explanation in the professor's voice covering the whole window, articulating the content strictly within the scope of what the slides present.

Professor Prompt

You are given the images of slides from a presentation, each labelled `SLIDE k:`. Reconstruct what the professor would say while presenting them, as one continuous, didactic narration in their voice that connects the slides in order and reads the text and visual elements into words.

Stay grounded in the slides, reading their text, numbers, and figures faithfully and keeping exact values and named results as shown. You may add a plausible framing or analogy that the slides invite, but never assert facts, numbers, or named results that have no basis in them. If given a critique of a previous attempt, fix exactly what it flagged.

{window_slide_images}
{previous_critique}

The reconstructed narrative is subsequently evaluated within a critic loop, in which the *Critic* agent examines it as a spoken explanation and determines whether it is internally consistent, considering whether it is cohesive, whether it follows a coherent order of reasoning as it relates one idea to the next, and whether it reads as a professor explaining the window aloud. This agent rejects the narrative only when one of these properties is violated, and it may additionally indicate improvements to its fluency, which are returned as feedback. Because language models tend to attribute higher quality to their own generations, a tendency known as self-preference bias [Wataoka et al. 2024], the *Critic* relies on a model distinct from the one driving the generative agents and is implemented as a fixed independent judge. In order to keep the loop consistent, this agent is also given the history of previous reconstructions and critiques, which prevents it from contradicting an earlier judgment or penalizing a revision that it had itself requested.

Critic Prompt

Prompt abbreviated.

You are an impartial judge of explanation quality. Read the reconstructed discourse and decide whether it works as a spoken explanation of the window: whether it is cohesive, whether it follows a coherent line of reasoning as it connects ideas, and whether it reads like a professor explaining the slides aloud.

Reject the discourse only when one of these breaks, namely when it is incoherent, when its reasoning is out of order, or when it does not read as a spoken lecture. You may also suggest ways to make it flow better, phrased as an ordinary feedback.

{window_slide_images}
{reconstructed_discourse}
{dialogue_history}

Output a verdict and, if shown earlier rounds, stay consistent: don't contradict past feedback or re-open what a prior attempt already fixed.

Because the *Critic* is implemented as a fixed independent judge, held constant across every configuration and distinct from the model under evaluation, the refinement process remains consistent across experiments. This separation also proved necessary in practice, since when the *Critic* shared the same model as the *Professor*, reconstructed narratives were almost always approved in the first iteration, leaving the refinement loop without useful corrective feedback. Whenever the *Critic* rejects a reconstruction, its feed-

back is returned to the *Professor*, which produces a revised narrative. The cycle continues until the reconstruction is approved or the maximum number of refinement rounds is reached. If the refinement limit is reached without approval, the most recent draft is preserved , and the window is flagged for manual review.

3.3. Knowledge Retrieval

A lecture frequently relies on background knowledge that is not explicit in presentations, so that a discourse anchored in the slides may still leave a student without the context required to follow it. To supply this background, the *Researcher* retrieves complementary information from external sources, following the ReAct (Reasoning and Acting) paradigm proposed by [Yao et al. 2022], in which the agent alternates between reasoning steps and tool invocations. The tools are exposed by a Model Context Protocol (MCP) server that mediates access to Wikipedia, providing one operation that searches for candidate pages and another that retrieves the content of a selected page. Reading the validated discourse, this agent determines for itself which concepts warrant grounding and queries Wikipedia through these tools.

Researcher Prompt

Prompt abbreviated.

You are a researcher responsible for retrieving reliable background information for the window. From the discourse, decide which concepts are worth grounding and search Wikipedia for them.

Read the candidates, use the discourse to disambiguate, and fetch only the article that matches the concept taught in the lecture; if none matches, fetch nothing, since a wrong page is worse than none. From each chosen article, keep its text and its own figures.

`{window_discourse}`

Based on these, output:

- `retrieved_articles`
- `available_figures`

A single term can match several articles, and to choose among the candidates the *Researcher* relies on the generated discourse, keeping only the article that corresponds to the concept as it was taught in the lecture. When none of the candidates fits, the Researcher abstains rather than attach a source that does not belong, because an incorrect reference would mislead the student more than a missing one. From the article it does select, the Researcher takes both the text needed to complement the lecture and the figures that the article itself contains, which are already on topic and come with a curated caption. With this, the per-window graph ends, and each window is left with its reconstructed discourse, the retrieved text, and the associated figures, while the material itself is written only afterwards.

3.4. Lecture Synthesis

Each window covers only a few consecutive slides, so the previous stages yield a set of reconstructed discourses and retrieved sources that together span one whole lecture but

remain split across windows. The *Writer* brings these pieces into a single document, receiving, for every window of the lecture, the reconstructed discourse along with the text and figures gathered by the *Researcher*, and writing the entire lecture in one step. Operating from these combined intermediate representations instead of revising segmented prose prevents the loss of fine-grained details that a secondary rewriting pass would introduce.

Writer

Prompt abbreviated.

You are the writer producing the final study material for a lecture. For every group of slides you receive the professor's narrative together with the retrieved relevant text and figures.

Write the lecture, organized by concept, merging recurring concepts and treating the retrieved text as substance (definitions, mechanisms, results, examples). Use LaTeX for mathematical notation (`$...$` inline, `$$...$$` in display). Place each figure inline, in its own centered paragraph within the section it illustrates, omit any figure without a matching section.

`{windows_source}`

Output the lecture as Markdown, with a title, sections, prose, mathematical notation, and figures.

The Writer organizes the material by concept, merging concepts that recur across windows and placing each figure next to the passage it illustrates. When the combined source fits within a single prompt, which is the common case since a lecture spans few windows, the material is written in one call, while longer lectures are written in parts, where the first part opens the lecture and the following ones continue it. The resulting document is then exported as a PDF through Pandoc [1] and Tectonic [2] libraries, with mathematical notation typeset in LaTeX and each figure centered and accompanied by a caption.

4. Results and Discussion

The architecture was evaluated on the MIT OpenCourseWare course *Introduction to Computational Thinking and Data Science* (15 lectures, 531 slides), using windows of five slides and four open-weights vision-language models: `qwen3-vl-8b`, `qwen3-vl-30b-a3b`, `qwen3-vl-235b` [Bai et al. 2025], and `qwen3.7-plus` [Qwen Team 2026]. In each configuration, the evaluated model served as the *Professor*, the *Researcher*, and the *Writer*, while the *Critic* was fixed as `gemini-2.5-flash` [Google DeepMind 2025], since a preliminary study identified it as the only candidate that consistently requested revisions, and its feedback showed the highest alignment with human evaluation.

Accordingly, system performance is primarily assessed through the convergence of the refinement loop. Since the model employed by the *Critic* produced judgments that were the most closely aligned with human evaluation, reaching its approval provides the main operational criterion for an acceptable reconstructed narrative. Performance is therefore measured by the number of refinement iterations required for approval and by whether approval is reached within the established refinement budget.

[1] `https://pypi.org/project/pandoc/`
[2] `https://github.com/tectonic-typesetting/tectonic`

4.1. Study material generation

We first analyze the convergence of the Professor–Critic refinement loop, focusing on the number of refinement rounds required for approval and the fraction of slide windows accepted before the refinement budget was exhausted, denoted by K. As shown in Figure 2, the evaluated models exhibit noticeably different convergence patterns. Roughly half of the windows generated by `qwen3.7-plus` were accepted after the first interaction with the Critic, whereas `qwen3-vl-30b-a3b` frequently required all available refinement rounds before its narrative was accepted, with the remaining models falling between these two extremes and generally converging under the same criterion while differing in the amount of feedback required to reach an acceptable reconstruction.

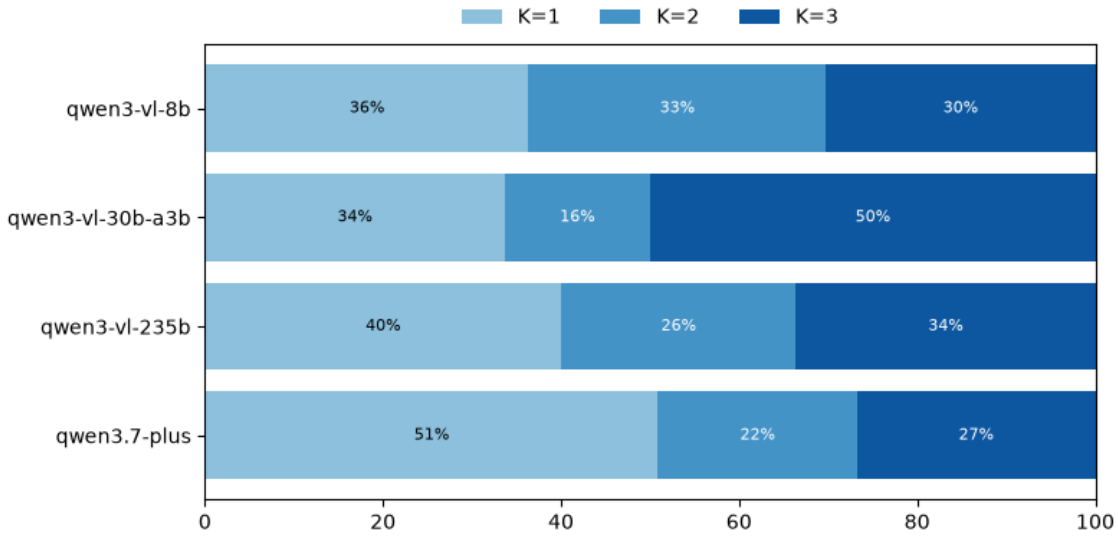

Figure 2. Distribution of refinement iterations required for narrative approval.

These differences in convergence reflect how each model produces a narrative that already reads as a coherent and fluent spoken explanation. A window approved on the first round is one whose initial reconstruction already satisfies the *Critic*, so that the loop merely confirms it, whereas a window that consumes every available round is one whose narration is repeatedly found to break in its cohesion or in its order of reasoning and must be revised before it is accepted. Under this reading, the early convergence of `qwen3.7-plus` indicates that its reconstructions tend to be well formed from the outset, while the concentration of `qwen3-vl-30b-a3b` at the round limit indicates that its initial narrations more often require restructuring, and that for part of the windows the available budget is not sufficient to reach an acceptable narrative.

Once a narrative is reconstructed, it is passed to the remaining stages and synthesized into the final study material, illustrated in Figure 3. Although the evaluated models preserve the same document structure, they differ in writing style, level of detail, and the amount of complementary information incorporated into the text. These structural differences are reflected in the aggregate statistics reported in Table 1, showing that `qwen3.7-plus` generated the longest materials and incorporated the largest number of supporting figures, whereas `qwen3-vl-235b` and `qwen3-vl-30b-a3b` produced documents of intermediate length and organization.

The models also differed in their capability to return the structured output required, highlighting a known limitation encountered when deploying smaller models in agent pipelines. Specifically, the smallest model, `qwen3-vl-8b`, was the only one that repeatedly struggled to follow the required output format, and therefore could not process all the lectures. Consequently, the metrics reported for this model cover only the *8* lectures it processed successfully.

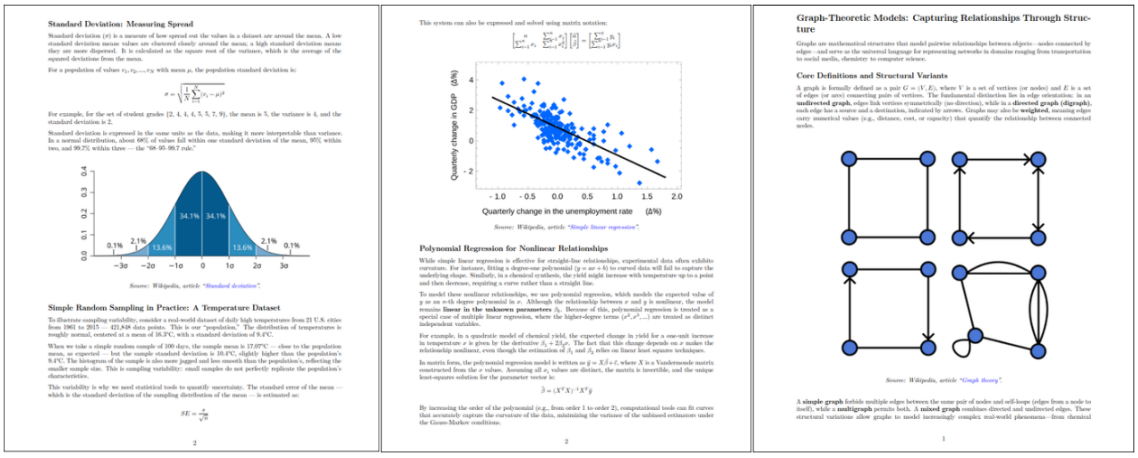

Figure 3. Examples generated by the system.

Table 1. Study materials generated for the course, averaged per lecture.

Model	Words/lecture	Sections/lecture	Figures/lecture
qwen3-vl-8b*	2336 ± 1086	10.9 ± 5.5	5.8 ± 4.2
qwen3-vl-30b-a3b	3615 ± 1297	17.9 ± 6.1	7.7 ± 3.2
qwen3-vl-235b	3218 ± 994	15.9 ± 4.4	7.3 ± 2.8
qwen3.7-plus	4763 ± 1595	16.4 ± 5.8	11.5 ± 5.5

* aborted early, averages over completed lectures.

Beyond the volume statistics reported in Table 1, we manually inspected the generated texts. The inspected texts were faithful to the source slides and cited references, with no gross hallucinations observed in the test set, and also followed a logical progression. The larger models (`qwen3-vl-235b` and `qwen3.7-plus`) produced the most coherent narratives, whereas the smallest (`qwen3-vl-8b`) generated comparatively shallow content. The main limitation, observed only in the longest materials, is the occasional repetition of facts retrieved independently by multiple windows, suggesting that the consolidation stage could benefit from concept-level deduplication.

4.2. Computational cost

The computational cost of the proposed method is characterized in terms of both processing time and token consumption, capturing the end-to-end cost of the generation stages.

Figure 4 reports the end-to-end processing time per slide. This measurement includes the entire refinement loop, accounting for all interactions between the *Professor* and the *Critic* until the narrative is approved or the refinement budget is exhausted. Additional refinement rounds further increase this cost because every rejected reconstruction

requires another interaction between the agents. Across models, however, inference latency remains the dominant factor, since `qwen3.7-plus` exhibits the longest execution times despite typically requiring fewer refinement rounds than the remaining models.

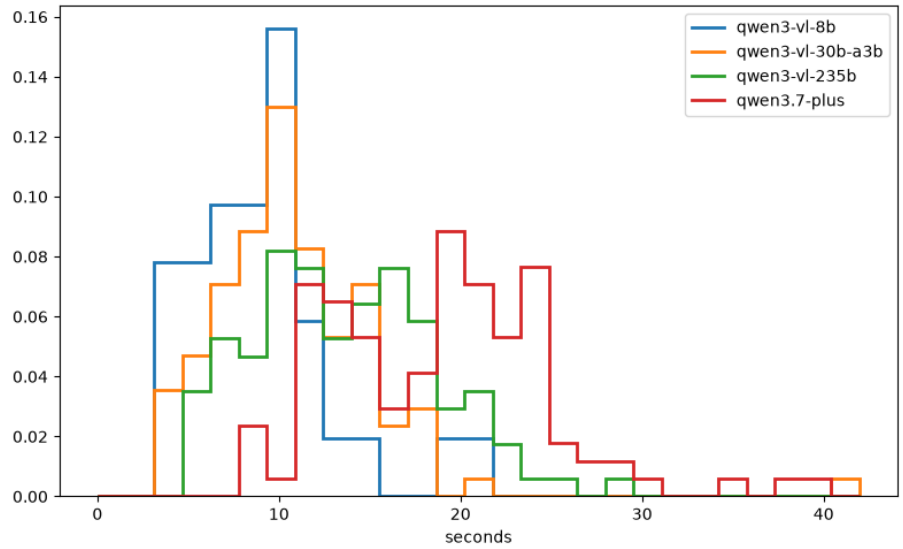

Figure 4. Execution time per processed window.

Although processing time characterizes the overall latency of the pipeline, it does not provide insight into how the computational effort is allocated across its constituent stages. To better understand this distribution, Figure 5 reports the token consumption of each agent throughout the processed lectures.

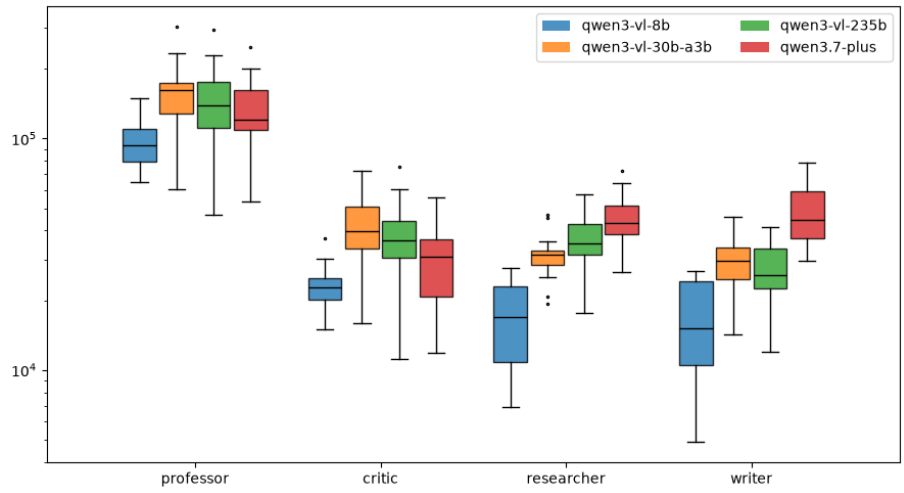

Figure 5. Token consumption grouped by agent.

Across all evaluated models, the *Professor* accounts for the largest share of token consumption, reflecting both the reconstruction of the narrative from the slide images and the possibility of multiple refinement iterations. The remaining agents consume substantially fewer tokens, with the relative consumption of the *Critic*, *Researcher*, and *Writer* varying across the evaluated models.

5. Final Considerations

This work presented an agent-based architecture for automatically transforming lecture slides into self-contained study materials. The proposed method organizes this process into complementary stages of multimodal processing, narrative reconstruction, iterative refinement, external knowledge retrieval, and content synthesis. Operating directly on slide images and progressively making implicit information explicit, the approach seeks to preserve the original lecture context while producing material suitable for asynchronous study.

The experimental evaluation showed that the proposed method can produce coherent and substantially expanded study materials, while also highlighting relevant differences among the evaluated models. Larger models generally generated more detailed and coherent texts, whereas the smallest model showed difficulties in consistently meeting the structured-output requirements of the pipeline. Computational cost was also influenced both by the number of refinement iterations and by the inference characteristics of each model, reinforcing that model selection involves a trade-off between output quality and processing cost.

Manual inspection showed that the generated materials remained consistent with the source slides and retrieved references, with no gross hallucinations observed in the evaluated set. Nevertheless, longer materials contained repeated information originating from different slide windows, because related concepts were processed independently across separate windows. In addition, the evaluation focused on generation quality, document characteristics, and computational cost, leaving the pedagogical effectiveness of the resulting materials to be investigated.

Future work will include evaluating the generated materials with students and instructors, improving cross-window consolidation, and extending the retrieval stage with additional trusted educational sources. Adaptive generation strategies may also be explored to produce materials suited to different levels of prior knowledge and learning objectives.

AI Transparency and Ethical Considerations

Grammatical review was performed for each section of this paper, assisted by Gemini. All automatic reviews were carefully checked by the authors to ensure text accuracy.

Code Availability

The full implementation of this work, as well as the generated artifacts, is publicly available at: `https://github.com/h4rad4/wesaac2026`.

Acknowledgements

The authors acknowledge the support provided by the Amazonas State Research Support Foundation (FAPEAM), as well as the support and infrastructure provided by the Intelligent Systems Laboratory (LSI) at Amazonas State University (UEA).

References

Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al. (2025). Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Branch, R. M. and Varank, İ. (2009). *Instructional design: The ADDIE approach*, volume 722. Springer.

Chu, Z., Wang, S., Xie, J., Zhu, T., Yan, Y., Ye, J., Zhong, A., Hu, X., Liang, J., Yu, P. S., et al. (2025). Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*, 2.

Google DeepMind (2025). Gemini 2.5 flash model card. `https://tinyurl.com/2cb7ftw5`. Model Card.

Jacobs, S. and Jaschke, S. (2024). Leveraging lecture content for improved feedback: Explorations with gpt-4 and retrieval augmented generation. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*, pages 1–5. IEEE.

Liang, J., Stephens, J. M., and Brown, G. T. (2025). A systematic review of the early impact of artificial intelligence on higher education curriculum, instruction, and assessment. In *Frontiers in Education*, volume 10, page 1522841. Frontiers Media SA.

Moore, R. L. and Lee, S. S. (2024). Harnessing generative ai (genai) for automated feedback in higher education: A systematic review. *Online Learning Journal*.

Qwen Team (2026). Qwen3.7-plus. `https://qwen.ai/blog?id=qwen3.7`. Accessed: 2026-06-20.

Rao, K., Coviello, G., Sankaradas, M., De Vita, C. G., Mellone, G., and Chakradhar, S. (2025). Slidecraft: Context-aware slides generation agent. In *2025 IEEE Conference on Pervasive and Intelligent Computing (PICom)*, pages 165–172. IEEE.

Swacha, J. and Gracel, M. (2025). Retrieval-augmented generation (rag) chatbots for education: A survey of applications. *Applied Sciences*, 15(8):4234.

Tanaka, R., Nishida, K., Nishida, K., Hasegawa, T., Saito, I., and Saito, K. (2023). Slidevqa: A dataset for document visual question answering on multiple images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13636–13645.

Wang, Y., Yu, J., Zhang-Li, D., Lim, J. J. Y., Tu, S., Li, H., Liu, Z., Liu, H., Hou, L., Li, J., et al. (2025). Educraft: A system for generating pedagogical lecture scripts from long-context multimodal presentations. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6153–6160.

Wataoka, K., Takahashi, T., and Ri, R. (2024). Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.

Wecker, C. (2012). Slide presentations as speech suppressors: When and why learners miss oral information. *Computers & Education*, 59(2):260–273.

Yao, H., Xu, W., Turnau, J., Kellam, N., and Wei, H. (2026). Instructional agents: Reducing teaching faculty workload through multi-agent instructional design. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4087–4109.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2022). React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.