Computer Science
&
Data Science

Capstone Report - Fall 2024

Kiwi-Data Structures: Empowering Personalized Learning with Multimodal AI System

Yifei Chen,
Xinwei Xie,
Zixuan Huang

supervised by
Hongyi Wen

Declaration

I declare that this senior capstone was composed entirely by myself with the guidance of my advisor, and that it has not been submitted, in whole or in part, to any other application for a degree. Except where it is acknowledged through reference or citation, the work presented in this capstone is entirely my own.

Preface

This report outlines the development of Kiwi-Data Structures, an AI-driven system designed to enhance personalized learning in computer science. As an improvement of Kiwi-ICP, this project is valuable for educators and researchers interested in AI-driven education, offering a more cohesive and effective learning experience for students.

Acknowledgements

A huge thanks to Professor Hongyi Wen for supporting our humble project despite his busy schedule, Utku from the library for helping us debug when we had problems deploying Kiwi on our local machines, Ke Ning, Professor Wen's former RA, for validating our ideas along the way.
My (Yifei Chen) personal thanks goes to Helen, John and various other fellow students who supported me to go through the toughest moments. And special thanks to Ulysse Villanueva for the mental support along the way.
I (Xinwei Xie) would like to extend my heartfelt gratitude to my roommate, Yiru, and my friend Shangqi, and many others for their unwavering support and encouragement.

Abstract

This project addresses the limitations of the Kiwi-ICP system in supporting the Data Structures course, specifically its inability to detect and integrate correlations across slides. This gap hinders students' ability to connect concepts and images critical for mastering complex topics. We tackle this by providing static and dynamic contextual information and additional knowledge base inputs to enable coherent cross-slide connections. Our approach significantly improved Kiwi's performance in answering image and concept based questions. Early results demonstrate better retriever performance and overall RAG performance, offering students a more cohesive and personalized learning experience. This work showcases how LLMs can be adapted for advanced, concept-heavy educational contexts. It will pave the way for integrating Kiwi into more advance courses such as Machine Learning.

Keywords

Personalized learning; Retrieval-augmented generation; Multimodal RAG; Contextual retrieval

Contents

1 Introduction

Kiwi is a research platform under active development at NYU Shanghai, whose goal is to create personalized learning experience for students as well as to facilitate teaching for our faculty. An instance, Kiwi-ICP, has already been developed and put into use for ICP 2024 Spring. It offers a comprehensive learning experience including: reviewing course material (slides and recitation quiz questions), obtaining real-time guidance and feedback form a LLM, and engaging in coding exercises with an IDE. To broaden the impact of this innovative multimodal AI system, we seek to extend Kiwi to Data Structures, an advanced CS course which builds on the coding foundations of ICP to explore how to efficiently manage and manipulate data.

Our project aims to address the limitations of the current Kiwi-ICP system in supporting the Data Structures course. Currently, Kiwi's RAG pipeline doesn't support correlation detection across slides, making it difficult for students to understand connections across images and concepts on multiple pages—a crucial aspect of mastering data structures. By expanding Kiwi's Retrieval-Augmented Generation (RAG) capabilities, we strive to enable coherent cross-slide connections, essential for grasping complex topics.

This is an interesting problem because it explores how LLM system can be adapted for advanced, personalized learning in challenging courses. To address Kiwi's limitations, we identified some key issues: lack of prompt context and insufficient contextual grouping. By adding relevant context to prompts, grouping related slides, we aim to enable Kiwi to better capture and connect information across slides, offering students a more cohesive learning experience in data structures.

2 Related Work

2.1 LLM-powered programming assistant

The use of large language models (LLMs) in educational tools has gained significant attention for their ability to provide real-time, personalized feedback. These systems assist students with understanding complex concepts, debugging code, and generating pseudo-code, making them particularly valuable in large-scale programming courses.

Kazemitabaar et al. (2023) proposed CodeAid, an LLM-powered assistant that provides conceptual explanations, pseudo-code, and annotated feedback to aid students without revealing full solutions [1]. Similarly, Chen et al. introduced GPTutor, a web application that personalizes learning content by aligning it with students' interests and career goals through Chain-of-Thought

prompting [2].

Beyond programming, personalized AI educational assistants leverage diverse approaches. Honglu's framework uses knowledge space theory to recommend tailored resources based on student behavior [3]. Park et al. integrate LLMs with student modeling to dynamically adapt exercises to individual progress [4]. These systems exemplify the growing emphasis on using AI to bridge knowledge gaps and enhance learning experiences.

2.1.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a framework that combines information retrieval with text generation to enhance the capabilities of large language models in handling knowledge-intensive tasks[5]. In the context of programming assistants, RAG can be used to retrieve relevant documentation or code snippets from external sources, such as large code repositories or knowledge bases, and incorporate them into the generation process. This enables LLM-powered systems to provide more accurate and contextually relevant feedback without solely relying on their internal parameters. The integration of retrieval systems allows the assistant to offer detailed explanations or debugging suggestions that are grounded in external, verified knowledge, making it particularly useful for complex coding tasks .

2.1.2 Prompting

Prompting techniques are key to guiding LLM-powered programming assistants in generating meaningful and accurate responses. Prompting methods such as Chain-of-Thought (CoT) prompting, where the model breaks down complex tasks into logical, intermediate steps, help in scaffolding the learning process for students. Few-shot prompting is also commonly used, where the assistant is given a few examples to better generate explanations or code snippets. These techniques enable the assistant to provide structured support for tasks like debugging or understanding advanced concepts, ensuring that the model's outputs align with the students' learning goals [6]. Additionally, techniques like self-criticism and verification allow the model to assess and refine its responses, improving the reliability of feedback in programming education.

2.1.3 Text Clustering

Zhang et al. (2022) presents a comparative study of neural topic modeling versus clustering methods for topic identification [7]. The research delves into how contextual embeddings can be

used for clustering text data to obtain topic words. This study demonstrates that with a proper sentence embedding and topic word selecting method, more accurate and diverse topics can be generated. Further, it is shown in the article that clustering-based model is robust to document length and topic number which could be useful when a student asks questions that might require materials from multiple slides and key concepts in Data Structures. Wu et al. pointed out that the problem with this method is that clustering methods are not topic models so they cannot infer topic distributions in the document.

Zhang et al. (2023) introduced ClusterLLM, a novel framework that leverages large language models (LLMs) to guide the clustering process [8]. This framework shows how LLMs can facilitate improved clustering granularity and perspective, making it relevant for the goal of identifying correlations between text chunks in RAG. But since the fine-tuning process is affected by the errors in LLM predictions, the computational cost might be high. This should be taken into account when we want to build a real-time interactive platform for students.

2.1.4 Contextual Retrieval and Reranking

Anthropic's article on Contextual Retrieval highlights the performance improvements made by contextual annotation by LLM and reranking after retrieval [9]. This ensures that only the most relevant chunks are passed to the model. It proves to provide better responses and reduce cost and latency because the model is processing less information. However, we have to fine-tune the chunk number for a trade-off between performance enhancement and lower latency and cost.

Glass et al. introduced RE2G which demonstrated the effectiveness of employing reranker in a RAG framework [10]. The method uses an interaction model to rerank the top-N passages retrieved by an initial retrieval model which is seen fit to our case when we want to select multiple correlated chunks for generation.

2.2 Evaluation Metrics for RAG

Outputs from RAG system will be evaluated. Some latest work on RAG evaluation metrics includes RAGChecker, which offers insights into retrieval errors, generator performance, and overall system quality[11]. We chose RAGChecker as our evaluation metric because it outperforms RAGAS and CRUD-RAG to be introduced next and has a well-rounded evaluation pipeline for retriever, generator and overall system.

Retrieval Augmented Generation Assessment (RAGAs) is a reference-free framework [12].

Faithfulness, answer relevance, and context relevance are measured without ground-truth annotations. It offers fine-grained evaluation but only on the end-to-end RAG system level.

CRUD-RAG is a comprehensive Chinese benchmark that classifies application scenarios into four categories (Create, Read, Update, Delete) and apply separate metrics accordingly [13]. It also provides insights into several crucial factors in RAG framework such as the top-k value and chunk size which might be useful for our purpose.

3 Solution

3.1 Data preparation

A huge difficulty we faced was securing a data set for testing purposes. It is hard to find high quality question-answer pairs based on a knowledge base. Since our focus is on Data Structures, we decided to use the exercises in the reference book with solutions relevant to the context of the book and hence the slides. To expand the varieties of our questions, we also came up with some image-based questions and concept-related questions and placed soft labels on them, generated from LLaMA model. Hence, we were able to group our data into three sets as follows.

- **Group 1: Questions related to algorithms**

  "Give an algorithm for finding the second-to-last node in a singly linked list in which the last node is indicated by a next reference of None."

- **Group 2: Questions related to concepts**

  "What are the advantage of Linked List?"

- **Group 3: Questions related to images**

  Why is 'curNode' needed in traversing? (This question is based on Figure 1)

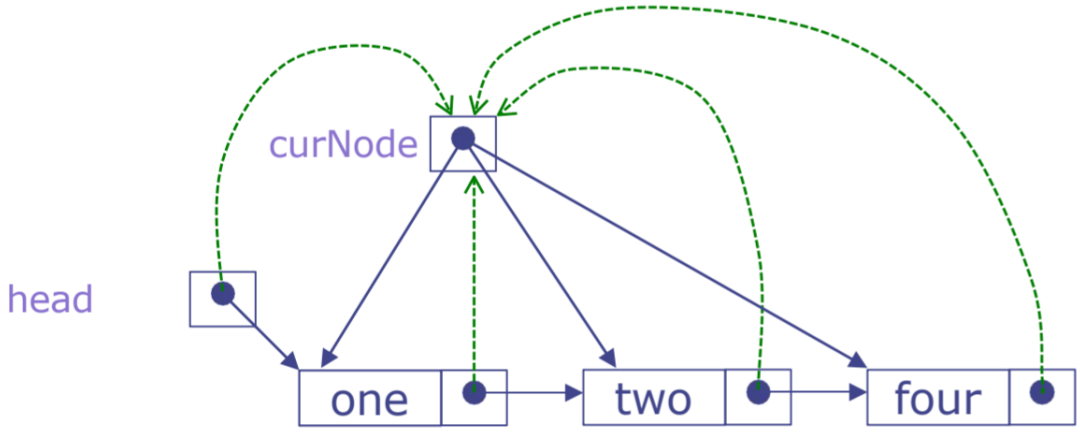


Figure 1: Image in Data Structures slides related to the question "Why is 'curnode' needed in traversing?".

3.2 RAG architecture

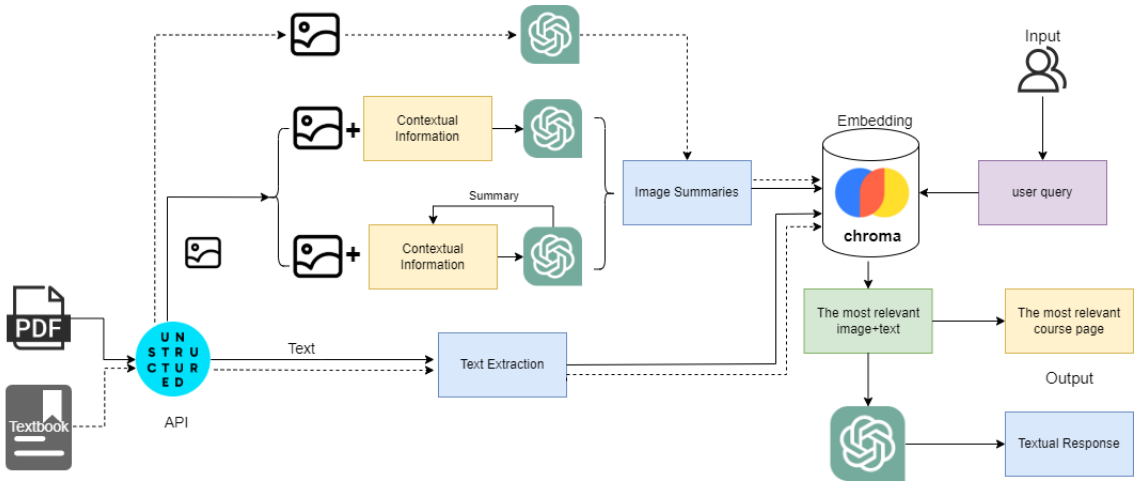


Figure 2: Architecture of RAG system in Kiwi-Data Structures

The RAG pipeline of Kiwi-Data Structures is shown in Figure 2. Unstructured API is used to chunk the text data and to extract all images and tables from the lecture slides. To convert the multimodal data to text, images and tables will be annotated by a LLM. The image summaries and texts will be embedded in chroma for later queries. When user queries the system, it will retrieve the most semantically relevant images, tables and text as well as the corresponding document pages as outputs.

Inspired by Anthropic's latest article on contextual retrieval, we proposed two different strategies to improve image chunk summaries based on our observation that Kiwi was only able to

describe the elements in the images but wasn't able to infer information.

Baseline Prompt: *"You are a helpful AI that summarizes images for later retrieval. Summarize this image given to you."*

- **Method 1: Naive Context (NC)**
  - **Contextual Prompt:** *"The following image is derived from Data Structures course slides on doubly linked lists. Your task is to summarize the information presented in the image."*

  This method involves augmenting the GPT prompt with additional context derived from the initial slide text. It adds static context information about the course itself and the topic of the current chapter. Adding this context helps the model generate summaries that are more relevant and aligned with the topic.

- **Method 2: Accumulative Summary (AS)**
  - **Accumulative Summary:** *"The following image is derived from Data Structures course slides on singly linked lists. Based on the knowledge in the previous images: 'context', please summarize the image below and show the connection between them."*

  In this approach, each image summary is fed into the next summary, creating an accumulative summary structure. This is a dynamic approach to add contextual information. By accumulating information progressively, this method produces more cohesive summaries that capture the relationships across sequential slides. This iterative process enhances the quality and relevance of the summaries.

Additionally we added the reference book to the knowledge base so that the system can have a more coherent narrative of the lecture content. The textbook will go through the same chunking, describing and embedding procedure as the lecture material.

3.3 Findings

In this section, we will introduce some findings through experimenting with different prompting strategies. Evaluation of all three methods will be included in Section 4.2.2.

3.3.1 Finding I: Accumulative Summary can better capture correlated information.

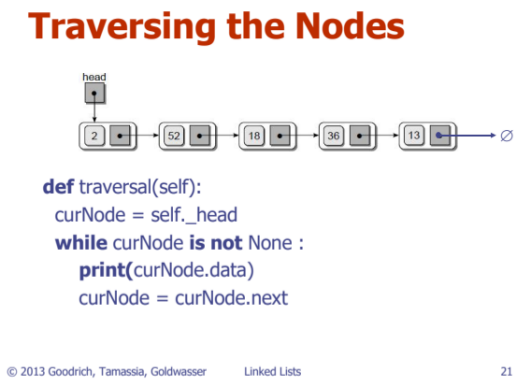


Figure 3: Page 21 in Data Structures slide, right before Figure 1

The image summaries produced for Figure 1 are as follows.

Baseline: The image depicts a linked list structure with nodes labeled "one," "two," and "four." There is a node labeled "curNode" with multiple arrows pointing to and from it, indicating connections to other nodes. Solid arrows represent direct links between nodes, while dashed green arrows suggest additional references or pointers. The structure appears to illustrate a complex node relationship or traversal in a data structure.

Naive Context: The image illustrates a singly linked list with nodes labeled "one," "two," and "four." The node labeled "curNode" is highlighted, indicating the current node in focus. The arrows represent pointers or links between nodes, showing the sequence of the list. The dashed green arrows suggest possible operations or movements of the "curNode" pointer to different nodes in the list.

Accumulative Summary: The image illustrates a singly linked list with nodes labeled "one," "two," and "four." The node labeled "curNode" is highlighted, indicating the current node in focus. The green dashed arrows suggest possible operations or movements within the list, such as traversing or updating pointers. In the context of singly linked lists, this image likely demonstrates how to navigate or manipulate the list by changing the current node's pointer to different nodes. This is a common operation in linked list algorithms, such as insertion, deletion, or traversal.

Accumulative Summary outperforms Baseline and Naive Context when the image itself is not self-explanatory. Supplemented by the image from previous page (Figure 3) which concerns the

same topic (traversing the nodes), the LLM was able to give information such as the intention of the image is to "update pointers" and the curNode is the current node is focus, etc. Among the three different methods, Accumulative Summary could infer most information out of Figure 1.

3.3.2 Finding II: Accumulative Summary might suffer from misleading information from irrelevant images.

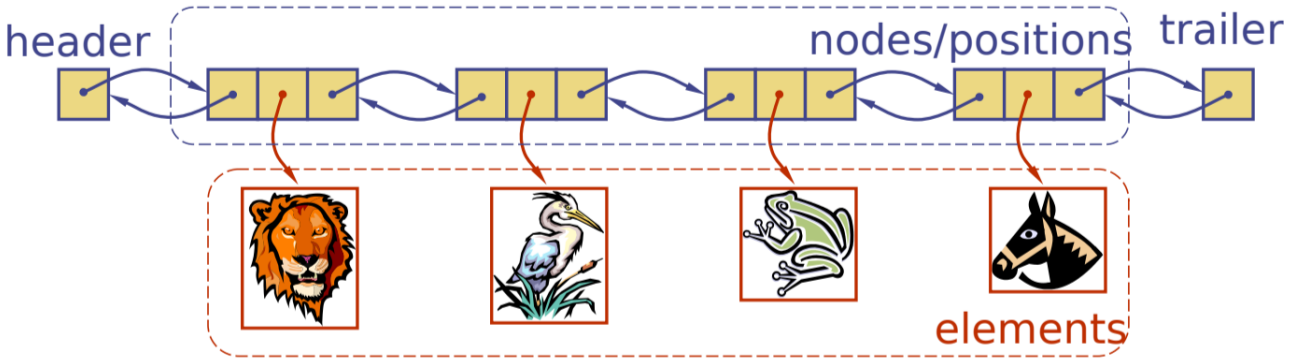


Figure 4: The figure explaining the concept of doubly linked lists

The image summaries generated for Figure 4 are as follows.

Baseline: The image illustrates a doubly linked list data structure. It includes a header and trailer, with nodes/positions in between. Each node is connected by arrows indicating the links between them. Below the nodes, there are images of a lion, a bird, a frog, and a horse, labeled as elements, showing the data stored in each node.

*Naive Context: The image illustrates a doubly linked list structure. It includes:- **Header and Trailer**: These are special nodes at the beginning and end of the list, respectively.- **Nodes/Positions**: Each node contains pointers to the previous and next nodes, allowing traversal in both directions.- **Elements**: Each node holds an element, represented here by images of animals (lion, bird, frog, horse). This structure allows efficient insertion and deletion of elements at both ends and at any position in the list.*

Accumulative Summary: The image illustrates a doubly linked list structure. It shows a sequence of nodes, each containing an element represented by an animal image (lion, bird, frog, horse). The nodes are connected by arrows indicating the bidirectional links between them, typical of a doubly linked list. The terms ̈headeränd ̈trailers̈uggest the presence of sentinel nodes at the beginning and end of the list, which are common in such data structures to simplify boundary conditions. The

nodes/positions hold the elements, which are the animal images in this case.

In this case, where the previous image summaries are irrelevant, Naive Context was able to infer more accurate information, in the context of Data Structures, compared to Accumulative Summary. For example, "bidirectional links", compared to "traversal in both directions" is out of context.

Hence, we concluded that Accumulative Summary has better inference performance out of all three. However, we need to carefully select the context to be put in the context. With regard to our future work, introduced in Section 5, we need to work on an algorithm to determine whether an image should be incorporated as the context based on their semantic similarity and relative distance.

4 Results

This section provides an overview of the evaluation of the RAG system's performance using RAGChecker, including metrics, experimental protocols, and outcomes. While the system demonstrates improvements, the enhancements are not consistent across all categories. However, it achieves significant gains in specific areas.

4.1 Evaluation Framework

To evaluate the performance of the RAG system, we implemented a rigorous experimentation pipeline with RAGChecker. RagChecker assesses the alignment between model-generated responses and the retrieved chunks by processing data into structured claims and comparing the claims and retrieved chunks against ground truth. The workflow is as follows (figure 5):

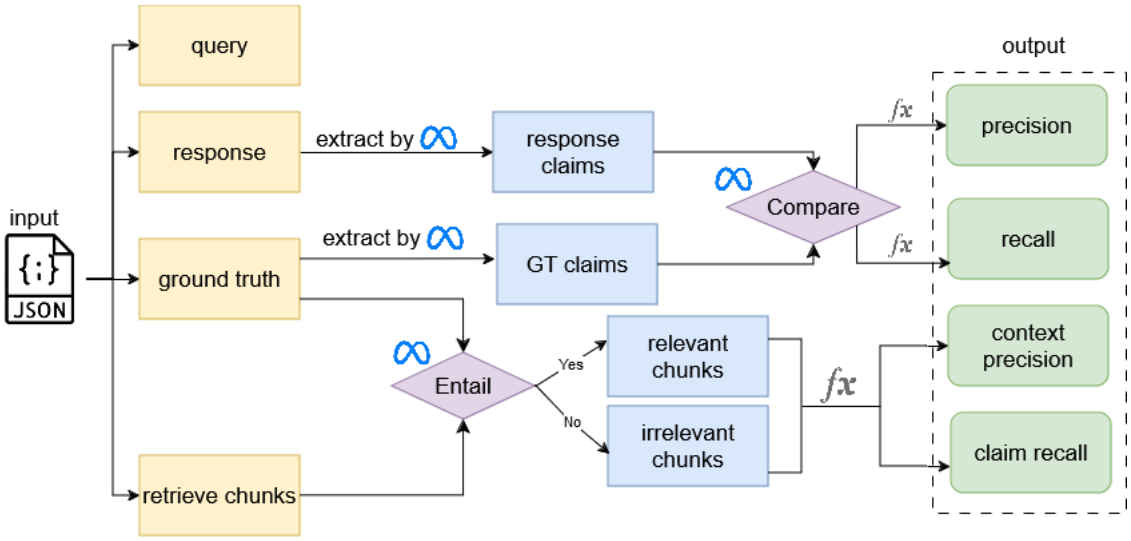


Figure 5: RAGChecker pipeline, an advanced automatic evaluation framework designed to assess and diagnose RAG systems.

1. Preparation of Evaluation Data:

As mentioned in section 3.1, a set of question pairs is clearly fully prepared from the textbook. These questions are used to generate model responses, which are then assessed for their alignment with the retrieved chunks.

2. Answer Relevance Evaluation:

The responses and ground truth are processed by the LLaMA model into structured bullet-point claims. This enables a detailed comparison between:

- **Correct Claims:** Claims present in both the response and the ground truth.
- **Missing Claims:** Claims present in the ground truth but absent in the response.
- **Incorrect Claims:** Claims present in the response but absent in the ground truth.

Using the above categories, RagChecker calculates the following evaluation metrics:

- **Precision (Correctness):**

$$\text{Precision} = \frac{\text{Correct}}{\text{Correct} + \text{Incorrect}}$$

- **Recall (Completeness):**

$$\text{Recall} = \frac{\text{Correct}}{\text{Correct} + \text{Missing}}$$

- **F1 Score (Overall Performance):**

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

2. Retrieval Ability Assessment:

The system's ability to retrieve relevant chunks was evaluated by cross-checking retrieved chunks against the ground truth. It examines if the information contained within the retrieved chunks is sufficient to fully support, confirm, or align with the claims in the ground truth.This involves categorizing the chunks into:

- **Relevant Chunks:** Chunks with ground truth claims.

- **Irrelevant Chunks:** Chunks without ground truth claims.

The proportion of relevant chunks is critical for assessing the logical sufficiency of the retrieval process. Metrics Context Precision and Claim Recall are used in this case to evaluate the effectiveness of retrieval.

- **Context Precision:** Measures how many retrieved chunks are relevant.

$$\text{Context Precision} = \frac{\text{Relevant Chunks}}{\text{Relevant Chunks} + \text{Irrelevant Chunks}}$$

- **Claim Recall:** Measures whether all relevant chunks are successfully captured.

$$\text{Claim Recall} = \frac{\text{Correct Claims}}{\text{Correct Claims} + \text{Missing Claims}}$$

4.2 Results and Comparisons

4.2.1 Tabular Results

We use the five metrics in section 4.1 to draw the table for the three methods in section 3.1.

Metric	**Baseline**	**NC**	**AS**	**TB**	**NC%**	**AS%**	**TB%**
Precision	74.6	69.9	68.1	68.5	-6.30%	-8.71%	-8.18%
Recall	95.2	70.8	85.7	74.1	-25.63%	-9.98%	-22.16%
F1	85.3	69.5	75.3	69.2	-18.52%	-11.72%	-18.87%
CR (Claim Recall)	49.2	66.7	66.7	49.7	35.57%	35.57%	1.02%
CP (Context Precision)	77.8	100	88.9	88.9	28.53%	14.27%	14.27%

*NC: Naive Context, AS: Accumulative Summary, TB: Textbook.

Table 1: Group 1: Algorithmic questions

Metric	Baseline	NC	AS	TB	NC%	AS%	TB%
Precision	54	75.2	69.7	67.5	39.26%	29.07%	25.00%
Recall	68.1	79.2	64.9	61.3	16.30%	-4.70%	-9.99%
F1	59.3	73.2	65.9	60.5	23.44%	11.13%	2.02%
CR (Claim Recall)	73.6	74.4	55.4	57.1	1.09%	-24.73%	-22.42%
CP (Context Precision)	77.8	88.9	88.9	88.9	14.27%	14.27%	14.27%

*NC: Naive Context, AS: Accumulative Summary, TB: Textbook.

Table 2: Group 2: Conceptual questions

Metric	Baseline	NC	AS	TB	NC%	AS%	T%
Precision	44.4	47.7	54.4	63.8	7.43%	22.52%	43.69%
Recall	71.4	69.5	68.1	71.4	-2.66%	-4.62%	0.00%
F1	51.7	54.5	57.8	57.9	5.42%	11.80%	11.99%
CR (Claim Recall)	53.6	79.2	79.2	81	47.76%	47.76%	51.12%
CP (Context Precision)	88.9	100	100	100	12.49%	12.49%	12.49%

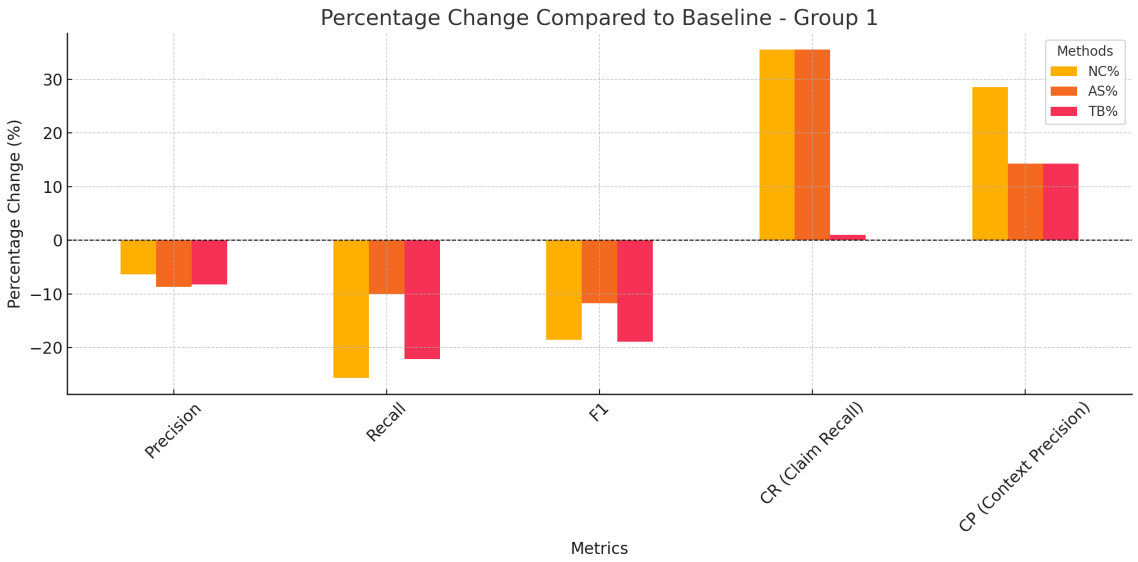
*NC: Naive Context, AS: Accumulative Summary, TB: Textbook.

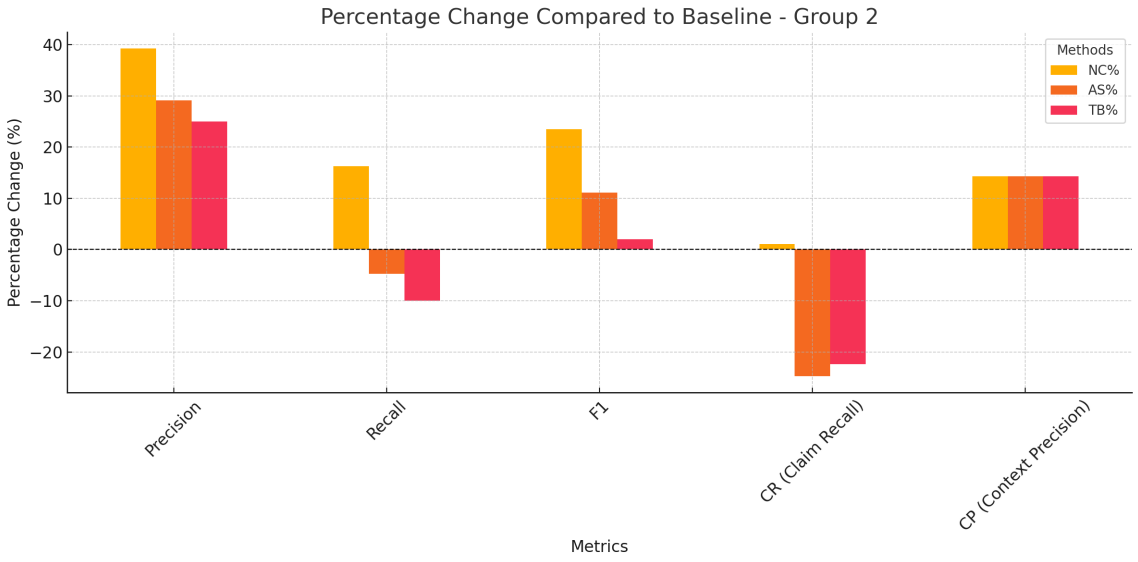
Table 3: Group 3: Image-based questions

4.2.2 Graphical Insights and Analysis

To evaluate which method consistently improves metrics across all groups, we analyzed the percentage changes in performance relative to the baseline for Naive Context, Accumulative Summary, and Textbook methods. The following observations were made: 6

1. **NC (Naive Context):**
    - NC demonstrates notable improvements in **Claim Recall (CR)** and **Context Precision (CP)**, particularly in Group 1, with increases of +35.57% and +28.53%, respectively.
    - However, significant declines are observed in **Recall** and **F1**, indicating challenges in maintaining comprehensive and accurate generation.
    - Importantly, Group 1 consists of coding-related questions, where the addition of naive contextual information provides limited benefits. Instead of improving responses, the expanded context often introduces hallucinations, thereby reducing the system's reliability. This highlights that adding generic contextual information is not always effective for problem-solving tasks with inherently concise answers, which are easily solvable with LLMs.


Percentage Change Compared to Baseline - Group 1
Methods
NC%
AS%
TB%
Percentage Change (%)
30
20
10
0
−10
−20
Precision
Recall
F1
CR (Claim Recall)
CP (Context Precision)
Metrics


Percentage Change Compared to Baseline - Group 2
Methods
NC%
AS%
TB%
Percentage Change (%)
40
30
20
10
0
−10
−20
Precision
Recall
F1
CR (Claim Recall)
CP (Context Precision)
Metrics

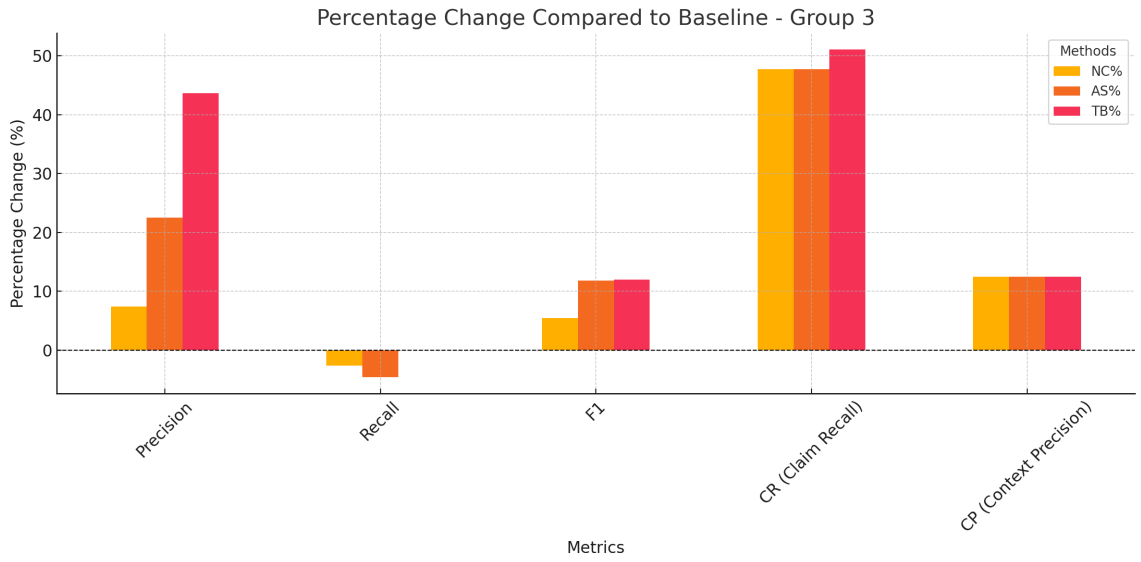


Figure 6: The three graphs for percentage of increase and decrease in metrics performance for the three methods compared to the baseline.

2. **AS (Accumulative Summary):**

   - AS consistently improves **Claim Recall (CR)** and **Precision**, achieving significant gains in Group 3 (image-based questions) with increases of +47.76% and +22.52%, respectively.
   - For conceptual questions in Group 2, AS shows moderate improvement in **F1** (+11.13%), benefiting from its ability to summarize accumulative evidence for concepts. However, these gains are less evident in other groups, indicating variability in its effectiveness.
   - Despite these strengths, AS struggles with retaining full recall, as seen in the slight declines for Group 2 (-4.70%) and Group 3 (-4.62%). This suggests that while AS provides high precision, its image summarization approach may sometimes omit key details necessary for complete coverage.

3. **TB (Textbook):**

   - TB achieves the most consistent and significant improvements across all groups, particularly in Group 3 (image-based questions), where it achieves a +51.12% improvement in **Claim Recall (CR)** and a +43.69% improvement in **Precision**.
   - TB demonstrates balanced performance, avoiding the significant declines in **Recall** and **F1** observed in the other methods. This stability highlights its ability to align both retrieval and generative components effectively.

- A key factor in TextBook's success lies in its integration of instructional context—capturing the lecturer's intent and classroom scenarios—which provides a highly relevant basis for both retrieval and generation. This approach not only reduces hallucinations but also enhances the specificity and relevance of answers, particularly for complex or multimodal tasks like image-based queries.

These results align with the findings in Section 3.3:

Among the three methods, **TB (Textbook)** emerges as the most reliable and effective due to its integration of structured and contextual information directly aligned with the teaching material. This is particularly evident in scenarios involving images that are not self-explanatory. This demonstrates that **TB** effectively mitigates the shortcomings of descriptive prompts by providing targeted, relevant context.

In contrast, **AS (Accumulative Summary)**, while capable of incorporating additional relevant information, provides information that is out of context, thus perform poorly at metrics such as **Claim Recall (CR)** and **Recall**. This highlights a need for more sophisticated integration mechanisms to refine accumulative summaries and prevent misleading.

NC (Naive Context) introduces broader benefits such as topic framing (e.g. the course title singly linked lists) but lacks specificity and consistency. While useful in some scenarios, the added context often fails to address the nuanced dependency on prior slides, which is critical for solving image-based questions.

The findings further indicate that adding **TextBook** and **Accumulative Summary** are particularly effective for image-based questions requiring comprehensive prior knowledge, as these methods provide the necessary depth of context.

These results underline the importance of aligning contextual integration with the specific needs of the task. Structured approaches like **TB** excel in reducing ambiguity and enhancing the pedagogical value of image summaries, while **AS** offers potential for broader applications if its hallucination issues can be addressed. Naive context methods like **NC** are limited by their inability to discern and prioritize critical contextual dependencies, making them less effective in scenarios demanding precision.

5 Discussion

Compared to related work, our approach significantly improves the integration of multimodal content, including images and text, while dynamically building contextual relationships across lecture materials.

As for limitations, the research of our scope is very specific. This made it very difficult for us to secure a good dataset. We had to use soft labels with answers generated by LLMs to perform the evaluation. This should be considered in earlier stage of our work.
Accummulative Summary still suffers from misleading information from irrelevant images. Therefore, in future work, we should consider scoring images in the context based on their semantic similarity and relative distance, and incorporate only the most relevant images in the context. Or instead of embedding text and images separately. We could treat each slide as a unit so that the information is more coherent.

As for evaluation, although RAGChecker represents one of the most advanced evaluation frameworks, it faces challenges when dealing with technical and academically oriented questions.[11] A fundamental trade-off exists between improving retrieval performance and introducing noise, as it remains difficult to ascertain whether weakly related answers are genuinely relevant, whether it is for evaluation or retrieval summarization. This balance underscores the complexity of optimizing retrieval strategies without compromising precision. Ultimately, enhancing retrieval performance is pivotal to improving both the accuracy and the reliability of the system's responses, ensuring it can effectively address the nuanced demands of academic inquiries.

6 Conclusion

The project enhanced Kiwi's RAG pipeline for the Data Structures course by improving prompt context, contextual slide grouping, and retrieval accuracy. These advancements enabled better cross-slide correlation, offering students a cohesive understanding of complex concepts in data structures.

The evaluation of the RAG system using RAGChecker demonstrated significant advancements in integrating multimodal content and aligning generated responses with retrieved chunks. While consistent improvements were observed across most categories, the system performed exceptionally well in image-related queries, achieving high Precision, Recall, and F1 Score in relevant metrics.

This work sets a foundation for further exploration of multimodal content integration in RAG systems while highlighting areas for refinement and scalability.

References

[1] M. Kazemitabaar, R. Ye, X. Wang, A. Z. Henley, P. Denny, M. Craig, and T. Grossman, "Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. Toronto, Ontario, Canada: University of Toronto, 2023.

[2] E. Chen, "Gptutor: A chatgpt-powered programming tool for code explanation," in *Artificial Intelligence in Education*. Springer, 2023.

[3] A. Honglu, "Personalized learning resource recommendation framework based on knowledge space theory," in *Proceedings of the International Conference on E-Learning Technologies*. Springer, 2023.

[4] M. Park, S. Kim, S. Lee, S. Kwon, and K. Kim, "Empowering personalized learning through a conversation-based tutoring system with student modeling," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. ACM, 2024.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[6] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff, P. S. Dulepet, S. Vidyadhara, D. Ki, S. Agrawal, C. Pham, G. Kroiz, F. Li, H. Tao, A. Srivastava, H. D. Costa, S. Gupta, M. L. Rogers, I. Goncearenco, G. Sarli, I. Galynker, D. Peskoff, M. Carpuat, J. White, S. Anadkat, A. Hoyle, and P. Resnik, "The prompt report: A systematic survey of prompting techniques," 2024. [Online]. Available: https://arxiv.org/abs/2406.06608

[7] Z. Zhang, M. Fang, L. Chen, and M. R. Namazi Rad, "Is neural topic modelling better than clustering? an empirical study on clustering with contextual embeddings for topics," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 3886–3893. [Online]. Available: https://aclanthology.org/2022.naacl-main.285

[8] Y. Zhang, Z. Wang, and J. Shang, "ClusterLLM: Large language models as a guide for text clustering," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 13 903–13 920. [Online]. Available: https://aclanthology.org/2023.emnlp-main.858

[9] Anthropic, "Contextual retrieval," 2023, accessed: 2024-10-05. [Online]. Available: https://www.anthropic.com/news/contextual-retrieval

[10] M. Glass, G. Rossiello, M. F. M. Chowdhury, A. Naik, P. Cai, and A. Gliozzo, "Re2G: Retrieve, rerank, generate," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 2701–2715. [Online]. Available: https://aclanthology.org/2022.naacl-main.194

[11] D. Ru, L. Qiu, X. Hu, T. Zhang, P. Shi, S. Chang, J. Cheng, C. Wang, S. Sun, H. Li, Z. Zhang, B. Wang, J. Jiang, T. He, Z. Wang, P. Liu, Y. Zhang, and Z. Zhang, "Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation," 2024. [Online]. Available: https://arxiv.org/abs/2408.08067

[12] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," 2023. [Online]. Available: https://arxiv.org/abs/2309.15217

[13] Y. Lyu, Z. Li, S. Niu, F. Xiong, B. Tang, W. Wang, H. Wu, H. Liu, T. Xu, and E. Chen, "Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2401.17043