Smarter Questions, Smaller Models: RAG-Enhanced Multiple-Choice Question Generation for POSCOMP

José Robson da Silva Araujo Junior[1], Leandro Balby Marinho[1], Lívia Sampaio Campos[1], Kemilli Nicole dos Santos Lima[1], David Eduardo Pereira[1], Helen Bento Cavalcanti[1], Ana Luíza Cavalcante Ramos[1], Eliane Cristina de Araújo[1]

[1]Departamento de Sistemas e Computação – Universidade Federal de Campina Grande
Av. Aprígio Veloso, 882 - 58.109-970, Campina Grande, PB – Brasil

{joserobson,david.pereira,helen.cavalcanti}@copin.ufcg.edu.br,

{lbmarinho,livia,eliane}@computacao.ufcg.edu.br,

{kemilli.nicole.santos.lima,ana.luiza.cavalcante.ramos}@ccc.ufcg.edu.br

Abstract. *Generating high-quality multiple-choice questions (MCQs) for specialized exams like POSCOMP remains a complex and labor-intensive task. This paper proposes a Retrieval-Augmented Generation (RAG) approach to support MCQ creation using Large Language Models (LLMs). We introduce a novel data set of 1,340 past POSCOMP questions, enriched with LLM-classified themes that show strong agreement with human annotations. The RAG method was compared to a few-shot baseline across five LLMs, generating 120 MCQs evaluated by human experts and an LLM-as-a-judge using a detailed rubric. Results show that the RAG approach improves question quality in up to half of the evaluated criteria, highlighting its potential for educational assessment tasks.*

1. Introduction

Generative Artificial Intelligence (GenAI) significantly impacts people's lives, with applications ranging from text synthesis to video generation. Education is no exception: this technology can support both teachers and students by assisting in solving complex problems, providing access to clear explanations, and facilitating the development of educational materials [Silvestre et al. 2023, Marques and Morandini 2024]. The field of multiple-choice question (MCQ) generation, a crucial method for assessing students' knowledge on a subject, has also benefited from the use of Large Language Models (LLMs) leveraged by recent GenAI advancements [Tran et al. 2023, Hang et al. 2024].

Despite the relevance of MCQs, particularly in high-stakes exams such as the National Exam for Admission to Graduate Studies in Computing (POSCOMP)[1] and the Brazilian University Admission Exam (ENEM)[2], their preparation remains a time-consuming and complex task due to the wide range of content and the high-quality standards required [Ch and Saha 2018], as effective MCQ generation demands more than merely producing grammatically correct items [Yao et al. 2024].

Although LLMs can facilitate the process of generating MCQs, they still pose several challenges. Key issues include ensuring alignment with the target exam, maintaining

[1] https://www.sbc.org.br/poscomp/

[2] https://www.gov.br/inep/pt-br/

DOI: 10.5753/sbie.2025.12854

the relevance and clarity of the content, and upholding high standards of question quality. LLMs must demonstrate a deep understanding of the subject matter and be able to formulate questions that effectively assess students' knowledge. Furthermore, many existing studies overlook the specific assessment context in which these questions will be applied, which can compromise their pedagogical relevance and practical effectiveness.

This work proposes a method for generating POSCOMP-style MCQs by combining LLMs with the Retrieval-Augmented Generation (RAG) technique. We aim to investigate whether the internal knowledge of LLMs is sufficient for this specialized task or if their performance significantly benefits from access to domain-specific content, particularly in the context of small language models (SLMs), which may lack the capacity to encode niche knowledge. We hypothesize that grounding LLMs with relevant context retrieved from a curated dataset of past POSCOMP exams improves question quality compared to a zero-shot baseline, and that this effect is more pronounced for SLMs.

To assess quality, we designed a detailed evaluation rubric applied by human evaluators and an LLM-as-a-judge. The result shows that the RAG-based approach significantly improves the quality of the MCQ, with the RAG-generated questions meeting up to half more rubric criteria than their zero-shot counterparts. As a secondary contribution, we present a new dataset built by extracting MCQs from historical POSCOMP exam PDFs and enriching them with metadata. This dataset is used to drive the retrieval process in RAG and plays a crucial role in improving the effectiveness of the generation pipeline.

The remainder of this paper is organized as follows: Section 2 reviews related work, focusing on MCQ generation using RAG-powered LLMs. Section 3 discusses the POSCOMP exam, introducing relevant concepts. Section 4 describes the methodology adopted, including the research questions, the collection and structuring of the POSCOMP dataset, the setup for MCQ generation, and the approaches used for quality evaluation. Section 5 presents and discusses the results. Section 6 addresses potential threats to validity and the measures taken to mitigate them. Finally, Section 7 concludes the paper and outlines directions for future research.

2. Related Work

Automatic question generation (QG) has been explored since the late 1990s [Madri and Meruva 2023], with applications spanning educational assessment, adaptive learning, and tutoring systems. The QG pipeline typically involves text preprocessing, question formulation, generation of distractors (incorrect options), and answer validation. Although some studies propose end-to-end solutions, others focus on specific stages of this pipeline [Ch and Saha 2018].

With the advent of LLMs, automatic QG has undergone a significant transformation [Meißner et al. 2024], with recent studies increasingly leveraging their advanced language understanding and generation capabilities [Das et al. 2021]. Current approaches span from prompt-based techniques to more sophisticated strategies that incorporate RAG and multi-agent systems [Li et al. 2024c, Jiang and Feng 2025]. For instance, Tran et al. [Tran et al. 2023] evaluated GPT-3 and GPT-4 for generating MCQs and plausible distractors tailored to Computer Science education. Similarly, Pawar et al. [Pawar et al. 2024] proposed a prompt-driven pipeline using the Gemini model, in which users provide a topic and input text to guide both question and distractor generation.

To address issues such as generic, shallow, or hallucinated outputs, RAG has been introduced to inject contextual specificity at inference time. [Gopi et al. 2024] demonstrated GPT-4 with RAG for quiz generation in engineering education, showing improved relevance through the retrieval of a curated dataset. [Pradeesh et al. 2025] used GPT-4 in conjunction with PDF-based content retrieval to improve contextual alignment.

More structured frameworks such as MQGen [Hang et al. 2024] extend this idea by combining RAG with Chain-of-Thought (CoT) prompting and self-refinement. Their system dynamically incorporates external data (focused on math and programming) from a dual-source dataset—questions created by instructors and students—enhancing diversity and pedagogical quality. Evaluations were performed with various LLMs, including GPT-3.5, Llama-2, and PaLM 2.

Finally, RAG techniques have been explored in the context of Brazilian national exams, particularly ENEM. Studies have demonstrated RAG's effectiveness in solving both general and domain-specific questions—such as mathematics—through a methodology that categorizes questions by knowledge area [Campos Taschetto and Fileto 2024, Superbi et al. 2024].

Our work advances the use of LLMs and Retrieval-Augmented Generation (RAG) for MCQ generation by targeting a high-stakes, domain-specific assessment: POSCOMP. In contrast to prior studies focused on general education or open-ended tasks, we tackle the specific challenge of producing high-quality, exam-aligned MCQs. Leveraging a curated dataset of past POSCOMP questions, we investigate whether incorporating external retrieval improves question quality—particularly for smaller models, which may lack sufficient domain expertise.

3. The POSCOMP Exam

The POSCOMP is a theoretical and objective test designed to evaluate candidates applying for graduate programs in the field of Computer Science (CS). Organized by the Brazilian Computer Society (SBC)[3] since 2002, a total of 21 editions were held annually up to 2024, except for 2020 and 2021, during which it was suspended due to the COVID-19 pandemic.

The test comprises 70 multiple-choice questions covering topics commonly addressed in undergraduate CS courses, divided into three areas of knowledge: Mathematics (*Matemática*), Computational Foundations (*Fundamentos da Computação*), and Computational Technology (*Tecnologia de Computação*). According to the POSCOMP syllabus, each area of knowledge is divided into broader areas, which are further split into specific topics, referenced in this paper as subareas. Table 1 illustrates an example of an area and some of its subareas for each area of knowledge.

Table 1. Area and subarea examples for each area of knowledge

Area of knowledge	Area example	Subarea examples
Mathematics (MT)	Combinatorics	Distribution; Permutations; Combinations
Computational Foundations (CF)	Graph Theory	Coloring; Spanning Tree; Topological Sorting
Computational Technology (CT)	Databases	Data Model; Query Languages; Data Mining

[3] https://www.sbc.org.br/

4. Materials and Methods

This section presents the methodology adopted in this study, detailing the process from the research questions to the experimental setup. All implementation steps are publicly available in a companion repository[4], ensuring the reproducibility of the results. Figure 1 provides an overview of the methodology employed in this work.

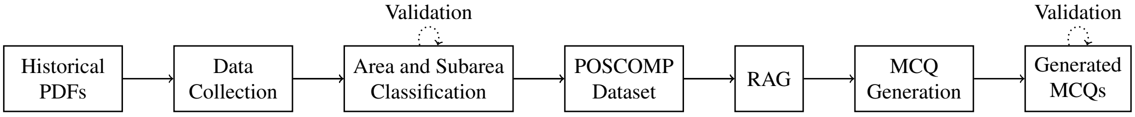

Figure 1. Overview of this work's methodology

4.1. Research questions

Two research questions were formulated to guide this study. **RQ**$_1$**:** What is the level of agreement between the area and subarea classifications assigned by a given LLM to the POSCOMP question dataset and those provided by human evaluators? **RQ**$_2$**:** To what extent does incorporating the POSCOMP dataset through the proposed RAG strategy enhance the quality of generated multiple-choice questions (MCQs)?

Given that the proposed RAG strategy depends on accurate area and subarea classification to retrieve relevant context, **RQ**$_1$ serves as an important step for validating the classification process and informing the analysis of **RQ**$_2$.

4.2. Data collection

As no previous publicly available and structured POSCOMP dataset was identified, an extensive data collection step was necessary to build one. For this purpose, a Python script was developed to extract the text content from the historical exam PDF files, covering the editions from 2002 to 2024. Depending on their encoding, the text was either read directly from the file or its pages were converted into images for applying OCR. The extracted text was transformed using regular expressions tailored to the different layouts used over the years. After extraction, the data underwent automatic cleanups (mainly for encoding normalization) and was organized into a structured format.

To ensure the accuracy of the content for each question, manual adjustments were performed by three of this paper's authors. By the end of this process, questions were represented using a combination of LaTeX (for mathematical notation) and Markdown (for text formatting), as modern LLMs have demonstrated a strong ability to work effectively with structured content [Achiam et al. 2023]. Since this study focuses on the generation of text-only questions, image-based questions were not included in the final dataset. Additionally, the official answer keys were parsed and added as question metadata.

4.3. Area and subarea classification

A relevant piece of information for each question is the specific topic it addresses. However, the only thematic grouping that could be inferred automatically was the area of knowledge, which is very general. The only exceptions are the 2019 and 2022 editions of

[4] https://github.com/Agents4Good/POSCOMP-RAG

POSCOMP, for which the official area labels for each question were released alongside the answer keys. Nevertheless, as described in Section 3, the exam provides a syllabus listing the areas and subareas it covers. This document was adopted as the reference for generating the area and subarea classifications, which were performed sequentially following the process outlined in Figure 2.

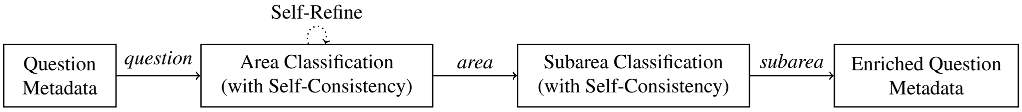
Self-Refine
Question Metadata
question
Area Classification (with Self-Consistency)
area
Subarea Classification (with Self-Consistency)
subarea
Enriched Question Metadata

Figure 2. Overview of the area and subarea classification

Given the strong performance of LLMs on text classification tasks [Kostina et al. 2025], we adopted an LLM-based approach to assign area and subarea labels using **Llama 3.3-70B**, which has shown competitive results even in zero-shot settings [Zhao et al. 2024]. The model was prompted to classify each question based on its underlying area of knowledge. To enhance reliability, we applied an adapted Self-Consistency (SC) technique [Wang et al. 2022], prompting each question up to five times and selecting the majority label.

By comparing the official area labels for the 2019 and 2022 editions of the exam with the model's output, a list of commonly confused areas was identified. The exam syllabus was also consulted to further inform and justify these choices. Using this information, the model was prompted once again to validate or revise its initial classification, being shown the current classification and possible alternatives, mirroring a Self-Refine (SR) technique [Madaan et al. 2023]. In this step, a list of subareas inside each area option was included in the prompt to support the model's reasoning.

Once the area classification was completed, the model was prompted once again to classify each question into one of the subareas within its assigned area. As in the previous step, the SC technique was used to improve the reliability of the classification. However, given this is a more granular task and no detailed descriptions were available for each subarea, the SR approach was not employed after this step.

4.4. The POSCOMP dataset

The POSCOMP dataset[5] comprises 1,340 questions, selected from a total of 1,470 questions available across all previous editions of the exam. The remaining 130 questions, which rely on images, were not included in this version of the dataset. Of the 1,340 questions, 391 are from Mathematics (MT), 507 from Computational Foundations (CF), and 442 from Computational Technology (CT).

Each question in the dataset is represented by a set of metadata: `year` (the year the exam was held), `number` (the item number in the test), `stem` (the question stem), `options` (a list containing all answer choices, including the correct one and distractors), `key` (the correct answer), `knowledge_area` (its associated area of knowledge), `area` (a division within the knowledge area), and `subarea` (a subdivision within the area). Except for the last two fields, all metadata was formatted during the data collection process (Section 4.2).

[5] `https://github.com/Agents4Good/POSCOMP-RAG/blob/main/app/data/poscomp-dataset.csv`

4.5. Area and subarea validation

After generating the model's area and subarea labels for each question, these classifications were compared with human-annotated labels to assess and support confidence in their accuracy. As with the generation process, validation was conducted separately for each classification level. The evaluators for each step were five of this paper's authors, all of whom are either current or former CS students familiar with the POSCOMP exam.

The validation of area classification began with the selection of a representative sample of the dataset questions ($n = 125$, approximately 10%), including every area based on their distribution and excluding the editions with ground-truth labels (i.e., 2019 and 2022). Each question was initially classified independently by two evaluators, who had access only to the question stem and options, its corresponding area of knowledge, and the exam syllabus for context. In cases of disagreement, a third evaluator independently annotated the question to serve as a tiebreaker. The final area was considered the human label and used as the reference for comparison with the LLM-generated labels.

Subarea classification followed a similar approach, with the selection of a sample ($n = 130$, approximately 10%) that preserved the subarea distribution. While it was not feasible to include all existing subareas, the sample was carefully constructed to ensure representation from every area, being slightly larger to account for underrepresented subareas. Each evaluator initially annotated their assigned questions independently, being allowed to select up to two subareas per question: a primary choice and a secondary one (if relevant). This strategy aimed to improve inter-evaluator agreement, given the finer granularity and potential overlaps in subarea classification. A third evaluator was involved when needed to reach consensus, resulting in a single subarea label per question.

4.6. Retrieval-Augmented Generation (RAG)

The RAG approach used in this paper is grounded in the area and subarea classifications described in Section 4.3; these classifications are used to filter and retrieve thematically related questions from the POSCOMP dataset, ensuring topical relevance in the retrieved context. This strategy also helps prevent the retrieval of questions that share surface-level terminology but are conceptually unrelated. For instance, Mathematical Logic and Discrete Mathematics share certain vocabulary while addressing distinct topics. This approach also aims to reduce the likelihood of generating irrelevant or unfocused questions.

The semantic storage and retrieval of POSCOMP exam questions is implemented using the ChromaDB[6] vector database. Each question is represented by a 384-dimensional embedding (numerical representation) generated by the `all-MiniLM-L6-v2`[7] model, based on a thematic classification field—a combination of all area-related classifications. The embeddings are stored in ChromaDB along with the rich metadata described in Section 4.5, and questions are organized into collections according to their area of knowledge.

For retrieval, a user query is converted into an embedding and compared against the vectors in the database via cosine similarity. ChromaDB executes a k-Nearest Neighbors search, and the results are filtered by a similarity threshold (i.e., ≥ 0.4, determined through empirical testing) and grouped to ensure diversity.

[6] https://www.trychroma.com/

[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2/

A total of five questions are retrieved and provided as context for the MCQ generation described in Section 4.7. This number was selected based on its common usage as an average in few-shot prompting, balancing prompt length and contextual relevance. In cases where few or no questions exist for a given subarea, the rest of the topic hierarchy is used to locate questions from related subareas within the same area. When more than five questions are retrieved, a random sample is selected to ensure variation and diversity.

4.7. Multiple-choice question generation

The MCQ generation prompt is constructed using a set of questions retrieved during the RAG process, instructing the model to use them as inspiration to generate a new question on the same topic[8]. To serve as a baseline for comparison, a zero-shot version of the prompt was also developed: "*Generate one (and only one) POSCOMP-style question related to the topic '`<TOPIC>`'.*".

Table 2 presents the list of models used in the MCQ generation experiments, sorted from smaller to larger based on their number of training parameters[9]. The selection includes four open-source models (accessed via the Deep Infra API[10]) and one proprietary model (GPT-3.5 Turbo), enabling broader comparisons and providing a more comprehensive view of the RAG strategy's impact across models of varying scales.

Table 2. Models used for MCQ generation with associated usage costs via API

Model name and version	Creator	Cost (per 1M tokens)
Llama 3.2 3B	Meta	\$0.01 / \$0.02 in/out
Gemma 3 4B	Google	\$0.02 / \$0.04 in/out
Qwen3-32B	Alibaba Cloud	\$0.06 / \$0.12 in/out
Llama 3.3-70B	Meta	\$0.23 / \$0.40 in/out
GPT-3.5 Turbo	OpenAI	\$0.50 / \$1.50 in/out

Each model was tasked with generating MCQs for the same set of 12 topics, each from a different area (covering 48% of the total areas). These topics were selected to simulate scenarios in which the RAG process would either have sufficient contextual data for retrieval or be forced to search for related topics due to limited availability of relevant questions. By combining the 12 topics with different experimental configurations (five models, each tested with and without RAG), a total of 120 distinct MCQs were generated.

4.8. Question validation

To ensure consistency across evaluations, a quality rubric was carefully designed. This instrument was inspired by previous rubrics for evaluating LLM-generated MCQs [Wang et al. 2025], but was extended to include criteria based on item-writing flaws (IWFs), aspects that deviate from common item-writing guidelines. The subset of IWFs was selected from a previous work by [Moore et al. 2023], pinpointing aspects that are

[8] https://github.com/Agents4Good/POSCOMP-RAG/blob/main/app/question_generator.py

[9] The exact number of parameters for GPT-3.5 Turbo is undisclosed, and has been the subject of ongoing discussion; however, for the purposes of this study, it was considered the largest among the selected models.

[10] https://deepinfra.com/

easy to identify and consistent with POSCOMP, allowing for a fast and comprehensive assessment of question quality. The rubric items are detailed in Table 3; an extended version is available in the repository[11], including the associated IWFs for each. Items were rated on a 3-point discrete scale: 1 for "no", 2 for "more or less", and 3 for "yes".

Table 3. Rubric items used to evaluate MCQ quality

Rubric Item	Definition
stem_clarity	Is the question stem phrased clearly and unambiguously, facilitating the understanding of what is being asked?
stem_conciseness	Is the question stem free of unnecessary information for answering the question?
options_clarity	Are all options phrased clearly and unambiguously?
distractors_plausibility	Are the incorrect options feasible to students with partial understanding, yet clearly incorrect for well-prepared students?
grammatical_consistency	Are all options grammatically consistent with the stem?
option_uniformity	Are all options similar in length and structure, avoiding cues that could lead to the correct answer?
single_answer	Is there exactly one, and only one, option that is clearly the best or correct answer among those provided?
absence_cues	Is the correct answer identified by knowledge on the topic instead of repeating words found in the question stem and not in distractors?
topic_related	Is the question related to the topic given in the prompt?
grammatical_correctness	Is the entire question grammatically correct and free from typographical errors?
complexity	Does the question require more than basic fact recall/memorization?
poscomp_aligned	Is the question aligned with the format usually covered in past POSCOMP exams?

This rubric was subsequently employed by both LLM-as-a-judge and human evaluators. The term *LLM-as-a-judge* refers to the use of LLMs as evaluators across various tasks, leveraging their reasoning capabilities to approximate the feedback typically provided by human experts. Prior studies have demonstrated high agreement between human and LLM evaluators [Li et al. 2024a, Li et al. 2024b]. In this study, the GPT-4.1 model was selected to assign scores to each generated MCQ based on the quality rubric. GPT-4 Turbo was identified as the best-performing model for LLM-as-a-judge in a recent survey [Gu et al. 2024]; however, according to OpenAI's benchmarks, the more recent GPT-4.1 model exhibits superior intelligence compared to that version.

For human evaluation, the five evaluators involved in the area and subarea classification participated, along with two external evaluators—CS graduates familiar with POSCOMP. The inclusion of external evaluators aimed to mitigate confirmation bias, as the other evaluators were directly involved in the research. During the evaluation process, each evaluator had access only to the question and its corresponding area and subarea.

Each question was independently evaluated by two evaluators. In cases of disagreement, a third evaluator was requested to resolve conflicts by scoring only the rubric items in dispute, resulting in a finalized annotation. These finalized scores were then compared to those assigned by the LLM, enabling calculation of agreement between the two approaches. The results were further analyzed to assess the quality of questions generated by each model, both with and without the use of RAG, allowing a comparative evaluation of model performance and the impact of the RAG strategy on MCQ quality.

[11] `https://github.com/Agents4Good/POSCOMP-RAG/blob/main/reports/extended_rubric.pdf`

5. Results and Discussion

This section presents and discusses the results obtained in this study, with a focus on addressing the two research questions: the agreement levels in area and subarea classifications (**RQ**$_1$), and the quality analysis of the generated questions (**RQ**$_2$).

5.1. RQ$_1$: Agreement in the areas and subareas classification

To evaluate the agreement between the LLM-based and human classifications, Cohen's kappa (κ) was the primary metric employed. Additionally, inter-evaluator agreement was assessed based on the classifications provided by the first two human evaluators, allowing for a comparative analysis. The interpretation of Cohen's kappa values in this study follows the widely used guidelines proposed by [Landis and Koch 1977].

The area classification task found agreement on 120 out of 125 questions, with κ = 0.958, indicating almost perfect agreement. When evaluating by area of knowledge, the highest agreement was observed in Computational Technology (κ = 1.000), followed by Mathematics (κ = 0.968); Computational Foundations had the lowest agreement, with κ = 0.905. Notably, all of these scores exceeded the corresponding inter-evaluator agreement values, which followed the same ranking from highest to lowest.

Subarea classifications were evaluated similarly, with agreement observed in 94 out of 130 cases (κ = 0.718, substantial agreement). Among these, questions labeled under Mathematics showed the highest agreement (κ = 0.858), followed by Computational Technology (κ = 0.742), while Computational Foundations exhibited the lowest value (κ = 0.572). Interestingly, inter-evaluator classification showed higher agreement for Computational Foundations compared to Computational Technology. Table 4 shows detailed results for agreements for area and subarea classifications.

Table 4. Agreement (κ) between human evaluators and between LLM and human labels for areas and subareas, grouped by area of knowledge.

	Area Classification		Subarea Classification	
Area of knowledge	**Inter-Evaluator**	**LLM-Human**	**Inter-Evaluator**	**LLM-Human**
Mathematics (MT)	0.679	0.968	0.776	0.858
Computational Foundations (CF)	0.602	0.905	0.656	0.572
Computational Technology (CT)	0.944	1.000	0.544	0.742
Overall	**0.758**	**0.958**	**0.665**	**0.718**

For the subarea classification, grouping by area was also conducted to identify cases that might be obscured by overall values or disproportionately contribute to the agreement levels[12]. Four out of the 25 areas showed perfect agreement ($\kappa = 1$): one from CF (Algorithms and Data Structures) and three from MT (Combinatorics, Differential and Integral Calculus, and Linear Algebra). Most areas (52%) exhibited fair to substantial agreement ($0.20 < \kappa \leq 0.8$) with Databases (CT) achieving near-perfect agreement.

Two areas resulted in $\kappa = \text{NaN}$ because only one subarea was represented for each in the sample. As all instances were correctly classified, Cohen's kappa is undefined

[12] Detailed results: `https://github.com/Agents4Good/POSCOMP-RAG/blob/main/reports/subarea_agreement.pdf`.

in these cases due to the absence of label variability. The remaining areas (four from CF and one from MT) showed slight to poor agreement ($\kappa \leq 0.20$):

- **Programming Languages (CF) and Mathematical Logic (MT):** Both included some of the most frequent subareas in the complete dataset, causing an overrepresentation in the sample which influenced the κ values. In the case of Mathematical Logic, which was represented by a single subarea in the sample, 75% of classifications were correct, but $\kappa = 0$ because of the imbalanced label distribution.
- **File and Data Organization (CF) and Programming Techniques (CF):** Both had very low representation (≤ 3 each) and were often misclassified.
- **Programming Techniques (CF):** This area was found particularly prone to confusion due to overlapping subareas; in most cases, multiple could plausibly apply.

The difference in agreement levels between subarea and area classifications can be attributed to three main factors. First, since classifications are performed sequentially (area then subarea) errors may propagate between steps, affecting subarea agreement even when area agreement is high. Second, subarea labels are inherently more specific than area labels, and some questions may relate to multiple subareas simultaneously, increasing the difficulty of reaching consensus. Finally, while each area branches into seven to ten subareas, some areas include up to 20 subareas, making classification inherently harder.

RQ$_1$: Overall, agreement between the LLM and human labels exceeds that between human evaluators, supporting the LLM's reliability as an evaluator and validating the final labels attributed by the model. Despite some challenges, area and subarea classification provide, in most cases, thematically aligned labels. The observed agreement, especially for area classification, supports the usage of these labels in the Retrieval-Augmented Generation (RAG) process for retrieving thematically grouped questions.

5.2. RQ$_2$: Quality of the generated MCQs

Initially, since both LLM-as-a-judge and human evaluations were conducted for the 120 generated MCQs, agreement between them was assessed. To this end, the quadratic version of Cohen's kappa (denoted as κ) was employed, as it is better suited for ordinal classifications such as those used in the evaluation rubric. The agreement between human evaluators (in the initial annotation) was $\kappa = 0.512$, while the agreement between human and LLM evaluations was higher, at $\kappa = 0.641$. These findings support the viability of using LLM-as-a-judge for evaluating MCQs in future studies employing this rubric.

For the primary analysis of MCQ quality, the final human evaluation scores were adopted as the ground truth. This choice ensures a consistent basis for analysis grounded in rigorously adjudicated human judgments. To complement this, all evaluations were also replicated using the LLM-as-a-judge approach[13], and the results were generally consistent with those presented in this subsection, particularly regarding relative differences.

Given that rubric items are rated in an ordinal scale, an approach for summarizing overall question quality was devised, computing the frequency of each score class per question. These metrics are referred to as `count_1` (number of items rated "1", no), `count_2` ("2", more or less), and `count_3` (3, "yes"). For instance, comparing

[13] Full report: `https://github.com/Agents4Good/POSCOMP-RAG/blob/main/reports/llm-as-a-judge-quality-summary.pdf`

count_3 values can quickly inform on question quality: questions with more rubric items rated as 3 are considered of higher quality than those with fewer.

The effectiveness of the few-shot (FS) approach using RAG was evaluated by comparing it to a zero-shot (ZS) baseline that relied solely on the base LLM's knowledge. On average, questions generated through FS met more quality criteria, receiving a mean of 10.6 rubric items rated as 3 ("yes"), compared to 7.3 for the ZS baseline. Another notable difference was observed in failure rates: ZS generations had a higher average of items rated as "no" (3.7) than FS generations (0.4). This indicates that the RAG strategy effectively benefits in reducing quality flaws and mitigating common generation failures. From the ZS-generated questions, nearly half (48%) failed to meet at least half of the rubric items, whereas over 88% of FS questions satisfied at least 9 out of 12 (75%) criteria.

In the inter-model comparison, Qwen3-32B consistently achieved the best results, showing almost no difference between the FS and ZS settings in terms of count_3; this can be attributed to the model's inherent reasoning capabilities, which are active by default. All other models benefited from the RAG strategy, particularly GPT-3.5 Turbo, Llama 3.2 3B, and Llama 3.3-70B, with increases in count_3 ranging from 4.5 to 6 items. Notably, Gemma 3 4B, despite being one of the smallest models evaluated, achieved an average of 9.75 rubric items rated as 3 in the ZS setting, increasing to 11 under FS prompting. Table 5 presents the average number of rubric items (out of 12) rated as 3, 2 and 1, respectively, for each model in both settings, along with overall averages.

Table 5. Average count of rubric items per rating, model and setting

	count_3 ("yes")			**count_2 ("more or less")**			**count_1 ("no")**		
Model	**Zero-Shot**	**Few-Shot**	**Average**	**Zero-Shot**	**Few-Shot**	**Average**	**Zero-Shot**	**Few-Shot**	**Average**
Llama 3.2 3B	3.500	8.167	5.833	1.167	2.667	1.917	7.333	1.167	4.250
GPT-3.5 Turbo	4.833	10.917	7.875	0.500	0.583	0.542	6.667	0.500	3.583
Llama 3.3-70B	6.583	11.333	8.958	1.417	0.250	0.833	4.000	0.417	2.208
Gemma 3 4B	9.750	11.000	10.375	1.583	0.833	1.208	0.667	0.167	0.417
Qwen3-32B	11.750	11.500	11.625	0.250	0.500	0.375	0.000	0.000	0.000

Additionally, the same human evaluators who conducted the quality assessment also evaluated the usability of the generated questions as a quick quality check. In the FS setting, 46 out of 60 questions (over 75%) were deemed usable as-is—meaning they had the correct number of options, were clearly worded, and included a single correct answer—while 10 required only minor adjustments. In contrast, under the ZS setting, more than half of the questions (33) were considered unusable, often due to improper formatting as multiple-choice questions. A model-level analysis revealed that all questions generated by GPT-3.5 Turbo in the ZS setting were judged unusable, whereas in the FS setting, every question it produced was deemed usable with minor or no revisions.

A final comparison was conducted to evaluate cost-efficiency. Generation costs were collected and summarized for all models and configurations. As expected, the FS strategy incurred higher costs than the ZS baseline due to longer prompts, with increases ranging from 75% to over 11× (the latter for GPT-3.5 Turbo, mainly due to its output token pricing). Although the cost of generating a single question was generally low (Qwen3-32B was the most expensive, at just over \$0.01 per question, due to its ”thinking” tokens), these costs scale rapidly when generating larger volumes. From a cost-efficiency perspective, the Gemma 3 4B model stood out: when paired with the RAG strategy, it achieved quality comparable to Qwen3-32B while being more than 35× cheaper.

RQ$_2$: Based on human evaluations, questions generated using the RAG approach met, on average, 27% more quality criteria compared to the zero-shot baseline, while also reducing the same proportion of flaws. This improvement was consistent across nearly all models, with increases ranging from 4.5 to 6 out of 12 criteria. These findings indicate that the RAG strategy effectively enhances MCQ quality, primarily by serving as a guardrail that mitigates common generation failures, specially for smaller models.

6. Threats to Validity

Several potential threats to the validity of this study were identified and addressed. One primary concern involves construct validity in the dataset structuring. Since no publicly labeled POSCOMP dataset was available, an LLM-based approach was used for area and subarea classification, which introduces a risk of misclassification. To mitigate this, Self-Consistency (SC) and Self-Refine (SR) techniques were applied to enhance label reliability. Another limitation concerns the scope of validation: although the sample was selected to reflect the overall distribution of the dataset, it was not feasible to include questions from all subareas, which may limit the generalizability of the agreement findings.

Threats related to internal validity were also identified during the evaluation of the generated MCQs. To minimize potential confirmation bias among researchers in this part of the study, two evaluators had no prior involvement with this work. Moreover, to ensure the reliability of automated evaluations, the LLM-as-a-judge was deliberately not used in the generation phase (GPT-4.1), mitigating potential self-enhancement bias. Lastly, to address external validity concerns stemming from reliance on a single model, the impact of the RAG strategy was assessed across a diverse set of LLMs.

7. Conclusion and Future Work

Crafting high-quality, domain-specific MCQs for high-stakes assessments like POSCOMP is a significant challenge. This paper addressed that challenge by proposing and evaluating a comprehensive methodology that leverages LLMs for automated question generation. A key contribution of this study is the creation of a new dataset comprising 1,340 past POSCOMP questions, enriched with area and subarea classifications generated by an LLM and validated through human adjudication.

The RAG-based strategy at the core of this work demonstrated strong effectiveness in enhancing question quality and reducing generation flaws. The average number of rubric items rated as “no” (indicating failure to meet a criterion) decreased from 3.7 in the zero-shot setting to just 0.4 in the few-shot setting, underscoring RAG’s value in bridging knowledge gaps and improving output reliability. Small models like Gemma 3 4B, when combined with RAG, delivered quality comparable to larger models at 35× lower cost, reinforcing the feasibility of adopting this strategy in scalable, cost-sensitive applications.

Several promising directions emerge for future work. First, the evaluation rubric could be further refined and validated for greater consistency and applicability. The proposed methodology could also be tested on other assessments (e.g., ENEM) to explore its generalizability across different domains. Additionally, it may serve as a foundation for more robust MCQ generation pipelines, including those using multi-agent systems.

Acknowledgements

This work was partially supported by the **First Partnership Agreement – Instituto Kunumi/UFCG-CEEI Embrapii/PaqTcPB** via the **Agents4Good** project. The authors also thank Rodrigo Eloy Cavalcanti and Mariana Marques dos Santos Dela-Bianca for their contributions to the evaluation process.

Artifacts Availability

The artifacts generated from this study (including code, scripts, and the POSCOMP dataset) are available in a GitHub repository: `https://github.com/Agents4Good/POSCOMP-RAG/`.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Campos Taschetto, L. and Fileto, R. (2024). Using retrieval-augmented generation to improve performance of large language models on the brazilian university admission exam. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 799–805. SBC.

Ch, D. R. and Saha, S. K. (2018). Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies*, 13(1):14–25.

Das, B., Majumder, M., Phadikar, S., and Sekh, A. A. (2021). Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning*, 16(1):5.

Gopi, S., Sreekanth, D., and Dehbozorgi, N. (2024). Enhancing engineering education through llm-driven adaptive quiz generation: A rag-based approach. In *2024 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., et al. (2024). A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Hang, C. N., Tan, C. W., and Yu, P.-D. (2024). Mcqgen: A large language model-driven mcq generator for personalized learning. *IEEE Access*.

Jiang, Z. and Feng, S. (2025). Usmlegpt: An ai application for developing mcqs via multi-agent system. *Software Impacts*, 23:100742.

Kostina, A., Dikaiakos, M. D., Stefanidis, D., and Pallis, G. (2025). Large language models for text classification: Case study and comprehensive review. *arXiv preprint arXiv:2501.08457*.

Landis, J. R. and Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, pages 363–374.

Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., et al. (2024a). From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Li, H., Dong, Q., Chen, J., Su, H., Zhou, Y., Ai, Q., Ye, Z., and Liu, Y. (2024b). Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Li, R., Jiang, Y.-H., Wang, Y., Hu, H., and Jiang, B. (2024c). A large language model-enabled solution for the automatic generation of situated multiple-choice math questions. In *Conference Proceedings of the 28th Global Chinese Conference on Computers in Education (GCCCE 2024). Chongqing, China: Global Chinese Conference on Computers in Education*, pages 130–136.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. (2023). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Madri, V. R. and Meruva, S. (2023). A comprehensive review on mcq generation from text. *Multimedia Tools and Applications*, 82(25):39415–39434.

Marques, D. and Morandini, M. (2024). Uso do chatgpt no contexto educacional: Uma revisão sistemática da literatura. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação*, pages 1784–1795, Porto Alegre, RS, Brasil. SBC.

Meißner, N., Speth, S., Kieslinger, J., and Becker, S. (2024). Evalquiz–llm-based automated generation of self-assessment quizzes in software engineering education. In *Software Engineering im Unterricht der Hochschulen 2024*, pages 53–64. Gesellschaft für Informatik eV.

Moore, S., Nguyen, H. A., Chen, T., and Stamper, J. (2023). Assessing the quality of multiple-choice questions using gpt-4 and rule-based methods. In *European conference on technology enhanced learning*, pages 229–245. Springer.

Pawar, P., Dube, R., Joshi, A., Gulhane, Z., and Patil, R. (2024). Automated generation and evaluation of multiplechoice quizzes using langchain and gemini llm. In *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, volume 1, pages 1–7. IEEE.

Pradeesh, N., Remya, T., MG, T., Pranav, V., et al. (2025). Retrieval-augmented generation for multiple-choice questions and answers generation. *Procedia Computer Science*, 259:504–511.

Silvestre, A., Amaral, E., Holanda, M., and Canedo, E. (2023). Students' perception about chatgpt's impact on their academic education. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 1260–1270, Porto Alegre, RS, Brasil. SBC.

Superbi, J., Pinto, H., Santos, E., Lattari, L., and Castro, B. (2024). Enhancing large language model performance on enem math questions using retrieval-augmented generation. In *Anais do XVIII Brazilian e-Science Workshop*, pages 56–63, Porto Alegre, RS, Brasil. SBC.

Tran, A., Angelikas, K., Rama, E., Okechukwu, C., Smith, D. H., and MacNeil, S. (2023). Generating multiple choice questions for computing courses using large language models. In *2023 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.

Wang, J., Xiao, R., and Tseng, Y.-J. (2025). Generating ai literacy mcqs: A multi-agent llm approach. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*, pages 1651–1652.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yao, Z., Parashar, A., Zhou, H., Jang, W. S., Ouyang, F., Yang, Z., and Yu, H. (2024). Mcqg-srefine: Multiple choice question generation and evaluation with iterative self-critique, correction, and comparison feedback. *arXiv preprint arXiv:2410.13191*.

Zhao, H., Chen, Q. P., Zhang, Y. B., and Yang, G. (2024). Advancing single-and multi-task text classification through large language model fine-tuning. *arXiv preprint arXiv:2412.08587*.