Sandra Haas, BSc, MA

Leveraging Contemporary LLMs and Knowledge Graph-based Retrieval Augmented Generation for Quiz Content Creation in the Context of Environmental Protection

Master's Thesis to achieve the university degree of Master of Science
Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor Assoc.Prof. Dipl.-Ing. Dr.techn. Roman Kern

Institute of Machine Learning and Neural Computation
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Legenstein

Graz, May 2025

Acknowledgments

This thesis would not have been possible without the support, encouragement, and guidance of many individuals to whom I am deeply grateful.

First and foremost, I would like to express my sincere gratitude to Assoc. Prof. Dipl.-Ing. Dr.techn. Roman Kern for his supervision, insightful guidance, and support throughout every stage of this project.

I would also like to extend my thanks to Smart-Study for generously providing the server infrastructure necessary to host my evaluation application. Furthermore, I am grateful to the evaluators who contributed their time and thoughtful feedback, which greatly enriched this research.

Finally, I wish to express my heartfelt appreciation to my friends and to my family for all the support they offered during my studies and this thesis. Without your support, this thesis would not have been possible — thank you!

Abstract

This thesis explored the use of LLaMa 3.1 for generating and structuring questions to quizzes within a gamified learning platform aimed at promoting environmental awareness. To enhance the quality, relevance, and coherence of the generated questions, two methods were employed: Retrieval-Augmented Generation (RAG) and Graph-based Retrieval-Augmented Generation (GRAG). By integrating knowledge graphs, the study aimed to improve both the factual accuracy of questions and the logical consistency of full quizzes.

A two-step evaluation assessed the generated content at both the question and quiz levels. The findings show that LLaMa 3.1 can generate contextually relevant and educationally useful questions from open-source environmental materials. However, occasional issues such as ambiguous phrasing and inconsistent difficulty levels indicate the need for careful prompt design and quality assurance.

Although GRAG did not outperform standard RAG at the question level, it demonstrated clear advantages at the quiz level, producing quizzes with higher overall quality and reduced redundancy. These benefits were most evident when knowledge graphs were smaller and semantically focused, while larger, fragmented graphs hindered performance.

Overall, the comparison of RAG and GRAG underscores the importance of balancing content relevance with structural coherence in automated quiz generation. While GRAG shows promise for enhancing quiz-level organization, further refinement of knowledge graph construction and integration is required to fully realize its potential in educational content generation.

Contents

List of Figures

1 Introduction

Memory research has consistently shown that the processes of encoding new information, storing it over time and retrieving it are interdependent. This is exemplified by the testing effect, which demonstrates that retrieving information from memory can significantly enhance retention (Rowland, 2014). Quizzes are a practical application of this effect, making them crucial for students. By regularly engaging in quizzes, students can reinforce their learning, track their progress and identify areas that need improvement, ultimately leading to better academic performance and deeper understanding of the material (Mcdaniel et al., 2007). Online quizzes are regarded as effective learning tools for formative assessments, particularly for students who prefer independent study, as they provide immediate feedback on their understanding of key concepts (Aravinthan and Aravinthan, 2010).

However, the process of creating questions which assess the user's understanding of a piece of text can be labour-intensive (Kumar et al., 2018). As a result, automatic question generation (AQG) has emerged as a crucial research topic. AQG techniques have been developed to address the challenges test developers face in creating a substantial number of high-quality questions. AQG focuses on designing algorithms that generate questions from knowledge sources, which can be either structured, like knowledge bases, or unstructured, such as text (Kurdi et al., 2020a). Over the past few years, various Natural Language Processing (NLP) techniques have been developed to automatically generate questions from provided material using language models (Lopez et al., 2021; Bhat et al., 2022; Diwan et al., 2023; Iusztin et al., 2024). These methods have made significant strides in automating question generation, but they have often been limited in their scope, focusing mainly on generating factual questions rather than evaluating deeper levels of understanding (Lohr et al., 2024). Building on the developments in AQG, this research aims to explore the proficiency of contemporary Large Language Models (LLMs) in both AQG and the structuring of learning content within the context of climate change topics. Specifically, it seeks to evaluate how effectively these models can generate meaningful questions and structure quiz questions, while examining their potential to improve the overall educational experience in this domain.

Why Large Language Models? The usage of LLMs enables straightforward and adaptable question generation through a "plug-and-play" mechanism. These

Pre-trained Language Models (PLM), having been pretrained on extensive web-scale data, possess a wealth of linguistic knowledge compared to earlier models. Moreover, they can be effortlessly customized for different generation tasks using the "prompting" technique, wherein users specify the desired task as a prompt (Wang et al., 2023).

Why climate change as context? Climate change serves as an ideal context for this study due to two primary reasons:

- Relevance and Significance: Climate change is a critical issue of our era, with global implications for the health of our planet and future generations. By focusing on this topic—a subject that is both timely and significant—the research is positioned to have practical and impactful applications.
- Proof of Concept: Limiting the context to climate change allows to demonstrate the feasibility and effectiveness of this approach. This focused scope provides a clear and manageable framework for testing and validating the underlying concepts, making it easier to draw concrete conclusions and develop robust methodologies.

To enhance the quality and relevance of the generated content, this research will employ Retrieval-Augmented Generation (RAG) and Graph-based Retrieval-Augmented Generation (GRAG) techniques. Unlike traditional fine-tuning methods, which are often resource-intensive and ill-suited for dynamic datasets, RAG decouples knowledge retrieval from the model's fixed parameters. Instead of relying solely on the model's internalized knowledge, RAG retrieves relevant information from external sources and incorporates this into the generation process. This approach helps mitigate hallucinations and enables real-time adaptability, making it particularly effective for dynamic learning environments (Iusztin et al., 2024). This research aims to provide a more comprehensive analysis of automatically generated quizzes by examining them as a cohesive unit rather than focusing solely on individual question quality. While existing studies often evaluate quiz questions in isolation, this work emphasizes the overall structure and effectiveness of the quiz as a whole. Additionally, this study explores the integration of knowledge graphs to further enhance the quality and relevance of the generated quizzes, an area that has been underexplored in previous AQG research. By leveraging knowledge graphs, this work seeks to improve not only the factual accuracy of the questions but also the logical consistency and educational value of the quizzes, offering a more holistic approach to automatic question generation.

Having established the significance of quizzes and the potential of LLMs in generating educational content, this research seeks to answer several critical ques-

tions regarding the effectiveness and structure of automatically generated quizzes. While the integration of LLMs, RAG and knowledge graphs offers a promising approach, several factors remain to be explored, such as the specific capabilities of LLMs in structuring content, the impact of knowledge graphs on content quality and the role of preprocessing as well as prompt engineering in enhancing the generated content.

1.1 Research Questions

The goal of this thesis is to utilize LLMs for the generation and structuring of content within a gamified learning platform aimed at promoting environmental awareness. Leading to the following research questions:

Research Question 1: To what extent can contemporary LLMs (Large Language Models) adeptly generate questions and effectively structure learning content in the context of environmental protection topics?

To achieve this, LLaMa 3.1. is utilized for the generation and structuring of content within a gamified learning platform aimed at promoting environmental awareness. Data collection involves sourcing open-access materials related to environmental protection. Users of the learning platform will have the capability to upload their own materials. Using LLMs, questions are generated from the provided materials and organized into levels, progressively increasing in difficulty to enhance the learning experience Gamification is employed to elevate the learning experience.

Research Question 2: Can Knowledge-Graph-based Retrieval Augmented Generation enhance the quality of the generated questions?

To address this research question a knowledge graph is constructed for each provided resource and used in the query to the LLM. The base model uses Retrieval Augmented Generation (RAG). RAG effectively mitigates the issue of generating factually incorrect content and reduces hallucinations (Gao et al., 2024). RAG-based retrievers primarily focus on individual documents, selecting relevant candidates based on text similarity. However, in many cases, important correlations among documents are also significant and here knowledge graphs may prove useful (Hu et al., 2024).

Research Question 3: To what extent is it necessary to preprocess the documents before prompting the LLM?

Preprocessing documents before prompting LLMs can significantly impact the quality and relevance of the generated questions. Previous studies have shown that effective preprocessing can help in focusing the model on the most relevant information, thus improving the accuracy and coherence of the outputs. For instance,

Scaria et al. (2024) found that reducing the complexity and amount of information in prompts led to better performance in LLMs, particularly in generating educational content.

Research Question 4: In what way can prompt engineering enhance the quality of the generated questions?

Prompt engineering plays a critical role in optimizing the performance of LLMs. Techniques such as Chain of Thought (CoT) prompting, providing examples and including detailed instructions have been shown to guide models in producing higher quality outputs. Lee et al. (2023) highlighted that using CoT prompting and including specific examples in prompts helped in generating more structured and valid educational questions. Additionally, Scaria et al. (2024) observed that prompt strategies incorporating Bloom's taxonomy levels improved the cognitive depth of the questions produced by LLMs, aligning them more closely with educational objectives.

1.2 Glossary

AI Artificial Intelligence

AQG Automated Question Generation

AEQG Automated Educational Question Generation

CoT Chain of Thought

GRAG Graph-based Retrieval-Augmented Generation

GPT-2 Generative Pre-trained Transformer 2

KB Knowledge Base

KG Knowledge Graph

LLM Large Language Model

ML Machine Learning

NLP Natural Language Processing

NER Named Entity Recognition

PLM Pre-trained Language Model

QA Question Answering

QG Question Generation

RAG Retrieval Augmented Generation

SQuAD The Stanford Question Answering Dataset

T5 Text-to-Text Transfer Transformer

UI User Interface

2 Background

The following chapter provides an overview of key concepts and technologies relevant to the study of Automatic Question Generation (AQG) using Large Language Models (LLMs). The first section introduces AQG, outlining its significance in educational contexts and its potential applications. The subsequent sections explore the foundational aspects of LLMs and their underlying principles in Natural Language Processing (NLP), detailing their structure and capabilities. Special attention is given to Prompt Engineering, a critical technique for optimizing LLM performance, which is further enriched by few-shot learning and output structuring methods. Moving beyond basic LLM capabilities, the chapter also discusses Retrieval-Augmented Generation (RAG) and Graph-based RAG (GRAG) techniques, which offer solutions to limitations in model knowledge and reasoning, enabling more accurate, up-to-date and context-aware responses (Lohr et al., 2024).

Together, these section lays the groundwork for understanding how AQG systems and LLMs work.

2.1 Automatic Question Generation (AQG)

AQG systems have emerged as valuable tools for producing educational questions from instructional materials, minimizing the need for extensive input from instructors and domain experts (Nguyen et al., 2022). Over time, multiple strategies have been developed to address the complexities of AQG, with recent advances highlighting the potential of LLMs (Lee et al., 2023; Scaria et al., 2024; Li et al., 2023).

AQG addresses the challenges of creating high-quality test questions by developing algorithms to generate questions from knowledge sources, which can be either structured (e.g., knowledge bases) or unstructured (e.g., text) (Kurdi et al., 2020b). Kurdi et al. (2020b) reviewed AQG literature between 2015 and early 2019 and identified three generation tasks: Preprocessing, Question Construction, Postprocessing.

Preprocessing in AQG involves two types, namely standard preprocessing and QG-specific preprocessing. Standard preprocessing involves common NLP tasks like segmentation, sentence splitting, tokenisation, POS tagging, coreference reso-

lution, named entity recognition (NER) and relation extraction (described in 2.2.1). QG-specific preprocessing focuses on optimizing inputs for question generation by applying techniques such as sentence simplification, sentence classification, and content selection (Kurdi et al., 2020b).

Question Construction - the core task in AQG —encompasses a range of processes that vary depending on the question type (e.g., multiple-choice, true/false, or gap-fill) and the desired response format. Key components of this task include generating the question stem and correct answer, constructing plausible distractors (i.e., incorrect options), producing informative feedback, and managing the overall difficulty level of the generated questions (Kurdi et al., 2020b).

Post-processing aims on improving the output questions, typically accomplished by two processes: Verbalisation and Question ranking (filtering). Verbalisation is defined as any process aimed at enhancing the surface structure of questions (e.g., improving grammaticality and fluency) or generating variations of questions through paraphrasing. An over-generate and rank approach is applied by several of the evaluated generators to prioritise good quality questions. (Kurdi et al., 2020b)

Building on these foundational AQG components, more recent approaches have shifted towards leveraging deep learning and, in particular, LLM, which offer new capabilities for context-aware and scalable question generation (Lohr et al., 2024). Given their central role in modern AQG systems, it is important to first provide an overview of large language models (LLMs) and the underlying concepts that enable their capabilities.

2.2 Large Language Models (LLMs)

Large Language Models are the product of combining natural language processing (NLP), deep learning (DL) concepts and generative AI models as shown in Figure 2.1.

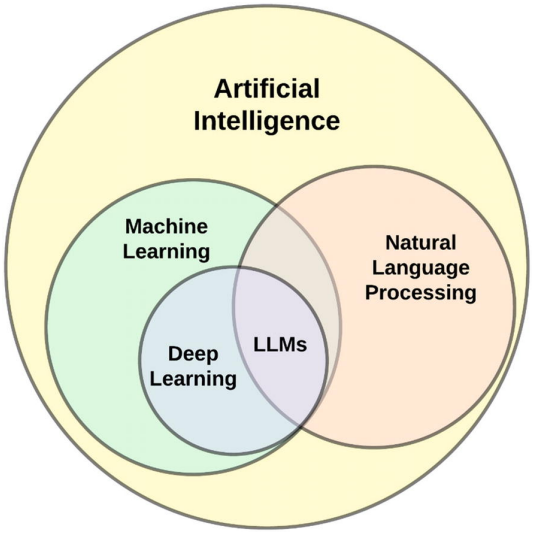


Figure 2.1: **LLMs in the AI landscape**. **LLMs in the AI landscape**. Where LLMs are in the AI landscape. Reprinted from *Understanding Large Language Models* by Thimira Amaratunga (2023). Copyright by Thimira Amaratunga, 2023.

2.2.1 Natural Language Processing (NLP)

Textual data is inherently unstructured; however, it typically adheres to the syntax and semantics of a particular language. Whether a single word, a sentence, or an entire document, all text data is rooted in some form of natural language. To grasp text analytics and natural language processing, it is essential to understand what qualifies a language as "natural." Simply put, a natural language is one that has organically developed and evolved through human interaction and communication, rather than being artificially constructed, such as programming languages (Sarkar, 2019).

LLMs are rooted in NLP, relying heavily on algorithms and representations developed through decades of research. Consequently, a deep understanding of LLMs requires familiarity with core principles of NLP. These basic concepts involve Tokenization, Stopword removal, Part-of-speech (POS) tagging, Parsing, Word embeddings, Named Entity Recognition (NER), Stemming and lemmatization, and Language models (Amaratunga, 2023).

Tokenization, the process of breaking down text or character sequences into smaller parts, called tokens is a crucial preprocessing step. Tokens are the building blocks for language processing tasks. There are different methods to perform tokenization according to Amaratunga (2023) which include:

- Whitespace tokenization: a process where text is split into tokens based on whitespace like spaces, tabs and newlines.
- Punctuation tokenization: punctuation marks like periods, commas or exclamation marks are used to split the text.
- Word tokenization: language-specific rules are used to split text into tokens.
- Subword tokenization methods, such as byte-pair encoding (BPE) and SentencePiece, break words into subword units, enabling models to manage out-of-vocabulary words and handle rare or unseen words more efficiently.

Word embeddings are dense vectors that represent words in a continuous vector space, with similar words positioned closer to each other, and capture semantic relationships. These semantic relationships enable NLP models to understand word meanings in relation to their context. Similar words are close to each other in the vector space which allows vector arithmetic for anologies like 'man is to woman as king is to queen'. Furthermore, word embeddings have a reduced dimensionality compared to one-hot encodings. One-hot encodings are binary vectors with the size of the vocabulary whereas word embeddings typically have a smaller dimension regardless of the vocabulary size. Additionally, word embeddings generalize across words since words that share similar contexts typically have similar embeddings, enabling models to infer the meaning of new words based on their contextual relationships. Word embeddings are continuous which enables interpolation and exploration of relationships between words. For example adding the vector "Spain" to "capital" and subtracting "France" leads to a vector close to "Madrid" (Amaratunga, 2023).

Stopword removal is the process of removing stopwords. Stopwords are frequently used terms (such as 'the', 'is' and 'and') that contribute minimal semantic value to a text. Eliminating them can reduce noise and enhance computational efficiency by reducing the dimensionality of input representations and eliminating frequent but semantically neutral words (Amaratunga, 2023).

Part-of-speech (POS) tagging assigns grammatical labels—such as noun, verb, or adjective—to words in a sentence, identifying their syntactic function (Amaratunga, 2023).

Parsing refers to the process of examining a sentence's grammatical structure to identify how words and phrases are related. Common approaches include dependency parsing and constituency parsing (Amaratunga, 2023).

Named Entity Recognition (NER), also known as entity chunking or entity extraction, is a widely used technique in information extraction that focuses on detecting, segmenting and categorizing named entities into predefined classes. In

textual documents, certain terms stand out as they provide more specific information and possess a distinct context compared to the rest of the text. These terms, referred to as named entities, typically represent real-world objects such as individuals, locations, organizations and similar concepts, often indicated by proper names. Identifying these entities involves analyzing noun phrases within the text. Various different frameworks exist for this task for example spaCy [1] (Sarkar, 2019).

Stemming and lemmatization are methods for transforming words into their base or root forms. For instance, words like 'running', 'runs' and 'ran' can all be reduced to the root word 'run' (Amaratunga, 2023). A lemma is the standard or dictionary form of a group of related words, collectively known as a lexeme. It serves as the base or headword that represents all variations of that word. Word forms are the inflected versions of the lemma that appear in actual usage. For example, in the lexeme (eating, ate, eats), "eat" is the lemma, while the others are word forms (Sarkar, 2019).

Language models are predicting the likelihood of a sequence of words occurring in a language by design, hence a probability distribution over sequences of words. Language models are fundamental to LLMs. There are two categories of language models: generative language models and predictive language models. Generative models generate new text by identifying patterns learned from their training data. Starting from an initial sequence, they produce the next word or sequence of words incrementally. Generative models are widely used for tasks like text generation, story generation, and poetry writing. Predictive models use previous words as input, predict the most probable next word based on their learned patterns from training data, and can be used in tasks like autocomplete, next-word prediction and machine translation (Amaratunga, 2023).

Transformers are the basis for many state-of-the-art language models like Generative Pre-trained Transformer (GPT), Bidirectional Encoder Representations from Transformers (BERT) and LLaMa. With the utilization of self-attention mechanisms they are able to process words in parallel, enabling them to process long-range dependencies efficiently. During training the model's weights are updated using backpropagation and gradient descent to minimize the prediction error (Amaratunga, 2023).

The attention mechanism used in transformers is inspired by the human cognitive process of selective attention on certain elements of sensory information over others and allows models to focus on specific parts of the input data. The classic attention mechanism has three main components: the queries (Q) which is a query vector representing the current element for which the attention is being computed, keys (K), which are vectors that represent other elements in the se-

[1] https://spacy.io/

quence and values (V), which are vectors containing information associated with each element in the sequence. Attention scores are typically computed using the dot product between query and key vectors and quantify the similarity between query and key vectors. To obtain the attention weights a softmax function is applied to the attention scores, converting the scores into a probability distribution where the weights sum up to 1(Amaratunga, 2023).

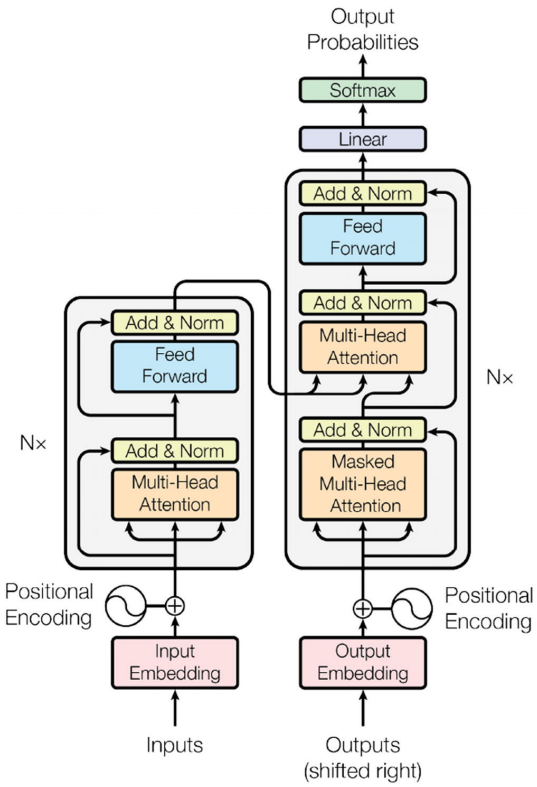


Figure 2.2: **The transformer architecture**. Reprinted from *Understanding Large Language Models* by Thimira Amaratunga (2023). Copyright by Thimira Amaratunga, 2023.

As shown in Figure 2.2 the transformer architecture consists of tokenizers, embedding layers and transformer layers. The tokenizers convert text into tokens, the embedding layers convert tokens into semantically meaningful representations and the transformer layers perform reasoning. The transformer layers can be either encoder or decoder. The original architecture consisted of both encoder and decoder whereas more recent variations use one or the other, for example like GPT models. **The encoder**, which is shown in the left side of Figure 2.2, consists of a stack of N identical layers (typically N=6), each containing a multihead

self-attention mechanism and a feed-forward network. Positional encodings, generated using sine and cosine functions, are added to input embeddings to incorporate word position information, as transformers lack inherent recurrence. The encoder's role is to extract meaningful representations from the input for downstream tasks or for use in the decoder. **The decoder**, shown in the right side of Figure 2.2, is composed of a stack of N identical layers (N=6 in the original paper), each with three sublayers. The first sublayer applies multihead self-attention to the output from the previous decoder stack, focusing only on preceding words by using a masking mechanism to prevent attention to future words in the sequence. The second sublayer performs multihead attention by combining the previous sublayer's queries with the encoder's output, allowing the decoder to attend to all words in the input sequence. The third sublayer is a fully connected feed-forward network, similar to the one in the encoder. Like the encoder, each sublayer in the decoder includes residual connections, followed by normalization layers. Positional encodings are added to input embeddings. The output embeddings are shifted by one position, ensuring predictions depend only on previously known outputs. The transformer architecture also introduces key concepts like scaled dot product and multihead attention. The scaled dot product is used to overcome the vanishing gradient problem, which occurs when the gradient backpropagation gets too small preventing the network to learn further. Multihead attention is used to linearly project the queries, keys, and values h times using a different learned projection for each of them. Then single attention is applied to each of the h projections in parallel producing h outputs, which are later concatenated and projected to produce the result. Multihead attention drives the computational costs down closer to a single-head attention which improves the training efficiency (Amaratunga, 2023).

Having outlined the fundamentals of NLP and traditional language models, the following section introduces LLMs, which build upon these foundations.

2.2.2 Large Language Models

A large language model typically refers to a language model with a significant number of parameters trained on an extensive dataset. Parameters are the elements of the model that are learned from the training data throughout the training process and typically determine how much, or rather how complex representations of data, a model can learn. The training process of large language models demands significant computational resources. It is possible to fine-tune large language models on specific tasks or datasets once its trained (Amaratunga, 2023). Large language models can be built using different architectures like Transformers, Recurrent Neural Networks, Convolutional Neural Networks. LLMs can be

categorized by their training objectives into autoregressive models, autoencoding models, sequence-to-sequence models or hybrid models (Amaratunga, 2023).

Since LLaMA 3, which is employed in this research for AQG and knowledge graph construction, is an autoregressive language model, it is important to explain the core principles of autoregressive modeling to provide the necessary conceptual foundation.

Autoregressive models are trained to generate one token at a time using previously generated words as context for future words. This allows them to maintain context in a text which makes them suitable for dialogue generation, storytelling or code writing. Additionally autoregressive models are capable of handling long-range dependencies in text allowing for more coherent and contextually relevant text generation. Limiting factors in autoregressive models are speed, repetition, lack of revision and context limit. Autoregressive models can be slower for generation tasks compared to parallel models, they can get stuck in a loop and generate repetitive text, can accumulate errors in long sequences since once a token is generated it can't be changed and there's a maximum sequence length (Amaratunga, 2023).

While LLMs show great promise, their capabilities can be further enhanced through targeted approaches like prompt engineering. This process helps refine how LLMs interpret and respond to tasks, ensuring greater accuracy and efficiency (Amaratunga, 2023).

Prompt Engineering

Prompt engineering is the practice of designing effective inputs to influence the behavior and output quality of LLMs. Since the performance of LLMs heavily depends on how input queries are phrased, well-crafted prompts can reduce ambiguity, steer the model's behavior, and customize its responses. This includes specifying the task clearly, providing contextual information, offering examples (few-shot learning), rephrasing inputs, or imposing constraints on outputs (Amaratunga, 2023).

Amaratunga (2023) outlines several key principles for optimizing prompts:

- **Explicitness:** It is important to use precise and clear instructions. For example, instead of asking "Tell me about apples," you might say, "Provide a 200-word summary of the nutritional benefits of apples."

- **Examples as Guidance**: Examples can illustrate the expected output. For instance, if you're instructing the model to convert sentences into questions, you could offer an example: "Transform the following sentences into questions. Example: 'It is raining' becomes 'Is it raining?'"

- **Iterative Refinement**: Prompt engineering often requires refining inputs based on the model's output. If a specific phrasing doesn't yield the desired results, rephrasing or adding more context can improve the outcome.
- **Controlling Verbosity and Complexity:** Using phrases like "in simple terms," "briefly explain," or "in detail" can direct the model's verbosity and complexity in its responses.
- **Systematic Variations:** Experimenting with different prompt formulations helps identify the most effective approach for a particular task.

To engineer prompts effectively when working with LLMs, several techniques can be employed:

- **Prompt templates:** involve using fixed structures with variable sections to maintain consistency, which is useful for tasks like data extraction (Amaratunga, 2023).
- **Prompt concatenation:** combines multiple prompts or instructions in a sequence to guide the model more effectively (Amaratunga, 2023).
- **Question decomposition:** breaks down complex queries into simpler parts, often leading to more accurate answers (Amaratunga, 2023)
- **Prompt priming:** introduces context or primes the model with a specific role, such as asking it to respond as if it were a history teacher, to improve the quality of the response (Amaratunga, 2023).

Despite its benefits, prompt engineering has limitations. It often requires trial and error, can be computationally expensive, and may not be sufficient for domain-specific or complex tasks—where fine-tuning might be necessary. Moreover, over-reliance on highly specific prompts may reduce generalizability and due to inherent randomness, even well-designed prompts might not yield consistent outputs (Amaratunga, 2023).

While prompt engineering focuses on shaping the model's response through carefully crafted inputs, techniques like few-shot learning provide additional structure, allowing LLMs to learn more from a limited number of examples. These methods can be particularly valuable when the task requires specificity or domain expertise (Ozdemir, 2024).

Few-Shot Learning Few-shot learning involves providing a language model with a few examples of a task within the prompt to help it generalize the pattern and produce more accurate results. This is especially useful for tasks that involve specific tone, structure, or domain-specific terminology. By learning from a small number of examples, the model adapts to expectations more effectively (Ozdemir, 2024).

Output Structuring To ensure consistency and facilitate integration with other systems, prompts can request specific output formats. For example, LLMs can be instructed to respond in bullet points, numbered lists, or structured formats like JSON. Structured outputs are easier to parse, analyze, or use in downstream applications (Ozdemir, 2024).

Chain-of-Thought Prompting Chain-of-thought prompting is a method that guides LLMs to reason through a series of steps, breaking down complex tasks into smaller sub tasks. This approach leads to more structured, transparent, and precise outputs, since the model addresses each sub task in a step-by-step manner. It also allows for intermediate outputs, making it easier to identify and debug issues. Additionally, this method improves the interpretability and transparency of the model's responses, offering insights into the reasoning process and promoting trust in the model's decisionmaking (Ozdemir, 2024).

In summary, prompt engineering offers a powerful yet iterative approach to guide LLM behavior and when combined with techniques like few-shot learning, output structuring, and chain-of-thought prompting, it can enhance the accuracy, consistency, and usefulness of model outputs (Amaratunga, 2023; Ozdemir, 2024). However, despite these advancements, LLMs still face limitations when dealing with scenarios where access to up-to-date or domain-specific knowledge is crucial. To bridge this gap, Retrieval-Augmented Generation (RAG) emerged as an effective solution, enabling the model to pull in real-time external data to enrich its responses and overcome the constraints of static training knowledge (Iusztin et al., 2024).

Retrieval-Augmented Generation (RAG)

RAG is a technique designed to provide LLMs with access to external data in order to perform specific tasks more effectively (Iusztin et al., 2024). RAG solves two core problems: hallucinations and private/unknown information (Iusztin et al., 2024).

When working with data the LLM wasn't trained on RAG offers a compelling alternative to fine-tuning, which is an extremely costly operation. Since LLMs are bound to understanding data they were trained on, which is commonly known

as parametrized knowledge, access to the latest data or other external sources is not possible. RAG overcomes these limitations by providing access to external data(Iusztin et al., 2024). Li et al. (2024) report benefits of combining parametric and nonparametric memory especially with the generation for knowledge-intensive tasks.

RAG can increase the accuracy and reliability of generative AI models by fetching information from external sources. RAG enhances the trustworthiness of the answers of a LLM because it enforces the LLM to always answer solely based on the introduced context. The LLM functions as the reasoning engine, whereas the external information retrieved through the RAG framework serves as the definitive source of truth for the generated output (Iusztin et al., 2024).

As shown in Figure 2.3 a RAG system is composed of three main components: Ingestion pipeline, Retrieval pipeline, and Generation pipeline.

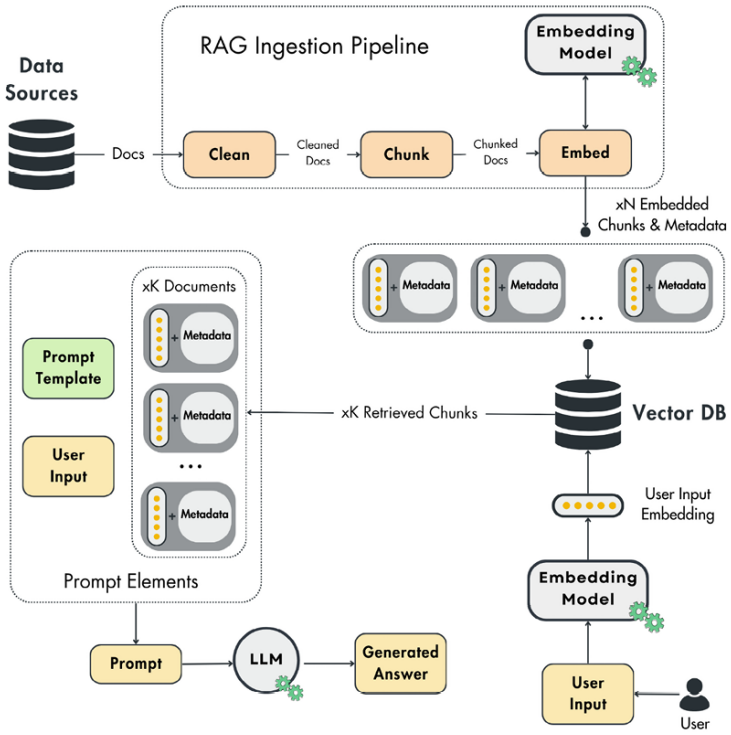


Figure 2.3: **Vanilla RAG architecture**. Reprinted from *Engineer's Handbook: Master the Art of Engineering Large Language Models from Concept to Production* by P. Iusztin, M. Labonne, J. Chaumond, H. Tahir, and A. Gulli, 2024, Birmingham: Packt Publishing. Copyright by Packt Publishing, 2024.

The **RAG ingestion pipeline** first extracts raw documents from data sources. Then in a cleaning process it standardizes and removes unwanted characters from the data. Next it chunks the data, splitting it into smaller sections to ensure the model's input maximum size isn't exceeded. The embedding model project the chunk's content into a dense vector packed with semantic value embeds the documents. Ultimately, the ingestion pipeline loads the embedded chunks into a vector storage (Iusztin et al., 2024).

The **RAG retrieval pipeline** takes an input (text, image, audio, etc.), embeds it and then performs a query to the vector DB for similar vectors to the input. The main purpose of the retrieval step is to map the input into the same vector space as the embeddings stored in the vector database. That way the top K most similar entries can be identified by comparing the stored embeddings with the vector representation of the input. These selected entries are then used to enhance the prompt provided to the LLM for generating a response. To compare two vectors a distance metric is used. Te most popular distance metric is the cosine distance, which is equal to 1 minus the cosine of the angle between two vectors (Iusztin et al., 2024).

The **RAG generation pipeline** takes the input, retrieves data, passes it to an LLM and generates a valuable answer. The final prompt is generated by combining a system and prompt template with the user's query and the retrieved context(Iusztin et al., 2024).

In summary, RAG enhances LLM performance by combining real-time information retrieval with generative reasoning, enabling accurate, context-aware outputs without requiring costly model retraining (Iusztin et al., 2024).

Graph-based Retrieval-Augmented Generation (GRAG)

While traditional RAG systems help mitigate challenges such as hallucinations and data updating in LLMs, certain limitations persist. For instance, retrieval from unstructured text may struggle to capture deeper semantic relationships. To address this limitation, incorporating knowledge graphs (KGs) into the retrieval process has emerged as a promising solution. KG store real-world factual knowledge in a structured format, effectively addressing the weaknesses of RAG technology. By integrating structured knowledge into the retrieval process, RAG models can better grasp the semantics of input queries and perform more effective relationship inferences using graph-based structures (Xu et al., 2024a).

Knowledge Graphs (KGs) are a type of graph where the emphasis lies on the contextual understanding. Graphs are simple structures where nodes (or vertices) are connected by relationships (or edges) to create models of a domain. The most

popular model for modern graph databases is the property graph model. It describes nodes as entities in a domain and relationships as how those entities interrelate in the domain. (Barrasa and Webber, 2021).

A labeled property graph has the following characteristics according to Robinson (2015):

- It contains nodes and relationships.
- Nodes contain properties (key-value pairs).
- Nodes can be labeled with one or more labels.
- Relationships are named and directed and always have a start and end node.
- Relationships can also contain properties

Architecture

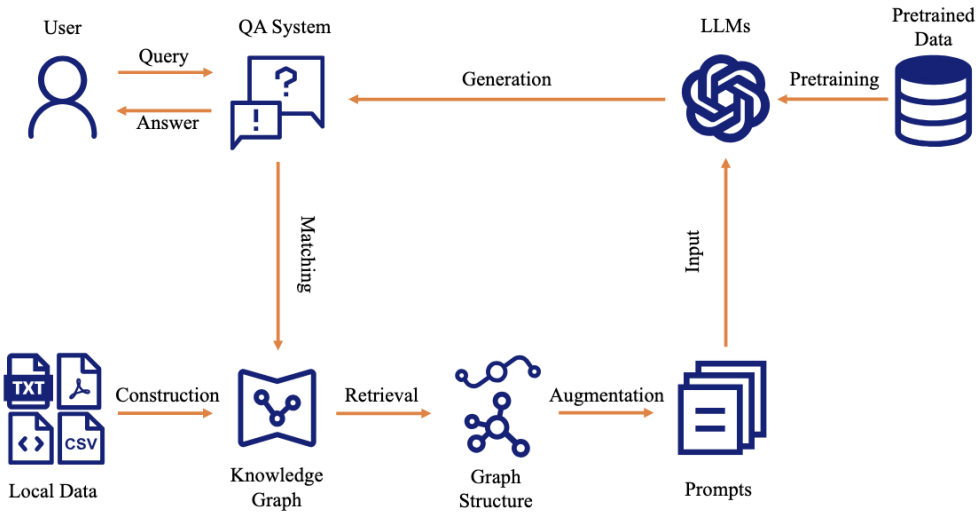


Figure 2.4: **Knowledge Graph RAG architecture**. Reprinted from *Retrieval-Augmented Generation with Knowledge Graphs: A Survey* by R.Chen, 2025. Copyright by R.Chen, 2025.

As shown in Figure 2.4 the architecture of RAG with knowledge graphs can be divided into three essential stages: graph construction, graph retrieval, and augmented generation (Chen, 2025).

During the **graph construction phase**, various types of local files are transformed into a knowledge graph using designated methodologies. The effectiveness of RAG is closely tied to the quality of the underlying knowledge graph. Consequently, a critical challenge at this stage is constructing a dynamic knowledge graph from diverse data sources while ensuring its accuracy and up-to-date information (Chen, 2025).

In the **retrieval phase**, the user query is semantically compared with elements in the knowledge graph, enabling the selection of the most relevant subgraphs, which may vary in granularity (Chen, 2025).

Finally, in the **augmented generation phase**, the retrieved graph segments are reformatted into data structures compatible with LLMs and combined with the original query as input prompts to produce a generated response (Chen, 2025).

In summary, Graph-based Retrieval-Augmented Generation systems enhance traditional RAG pipelines by integrating structured, semantically rich knowledge graphs, enabling more precise retrieval and context-aware generation in LLM applications (Chen, 2025).

3 Related Work

Automatic Question Generation is a prominent research topic. There have been various Natural Language Processing (NLP) approaches to automatically generate questions from provided material using language models over the past years for example by Lopez et al. (2021); Bhat et al. (2022); Diwan et al. (2023); Iusztin et al. (2024).

3.1 Methodology of Literature Review

To establish a strong theoretical foundation for this research, a structured literature review was conducted on Automatic Question Generation (AQG) with Large Language Models (LLMs), including Retrieval-Augmented Generation (RAG) and Graph-based Retriveal-Augmented Generation (GRAG).

3.1.1 Search Strategy

Relevant literature was identified through academic search engines, including Google Scholar and TU Graz Library Search. To refine the search, a combination of keywords and Boolean operators was used shown in Table 3.1. The results were filtered to include only literature published after 2019, ensuring the relevance and currency of the selected sources in relation to recent advancements in the field. In the TU Graz Library Search, the filter was applied to only include peer-reviewed journals. Resources focusing on Question Answering were excluded. The search terms described in Table 3.1 were selected to capture both the theoretical development of AQG systems and their practical applications in educational and technological contexts.

The search results were screened in multiple stages. First, title and abstract screening were conducted to remove irrelevant papers. Next, full-text reviews were performed to assess methodological rigor and relevance. Finally, key studies were analyzed in depth, focusing on their contributions to AQG, LLM-based approaches and educational applications.

Additionally, the citation list of the studied publications were analyzed to identify further relevant literature, ensuring a comprehensive review of related re-

Search term	GS	TGLS
”Automatic Question Generation” AND (”Large Language Models” OR ”LLaMa” OR ”RAG” OR ”G-RAG”)	231	13
”Prompt Engineering” AND (”Question Generation” OR ”Educational AI”)	350	5
(”Retrieval-Augmented Generation” OR ”RAG”) AND ”Automatic Question Generation”	119	1

Table 3.1: Total number of publication found with Google Scholar (GS) and TU Graz Library Search (TGLS)

search.

A summary of the retrieved and selected publications, detailing the total number of papers initially found and the used search terms, is provided in Table 3.1. The final selection of publications deemed relevant after screening for in-depth analysis are listed in Table 3.2.

3.2 State of the Art

In the area of AQG, early rule-based systems relying on predefined templates have been progressively replaced by sequence-to-sequence networks and transformer-based architectures, driven by recent advancements in deep learning and neural networks. The emergence of LLMs in recent years has unlocked new possibilities for overcoming the limitations of earlier Automatic Question Generation systems, which predominantly generated fact-based questions rather than evaluating deeper understanding, by leveraging semantics-based approaches (Lohr et al., 2024).

Table 3.2 provides an overview of the key publications studied in the domain of AQG, summarizing the models used, datasets employed, evaluation metrics, and key findings.

Table 3.2: Summary of Key Studies in Automatic Question Generation (AQG)

Study	**Model(s) Used**	**Dataset(s)**	**Evaluation Metrics**	**Key Findings**
Lopez et al. (2021)	GPT-2 (124M)	SQuAD	BLEU, ROUGE, METEOR	best-performing model outperforms previous models. Transformer-based AQG mainly generated factual questions. 88.26% were identification-type.
Bhat et al. (2022)	T5 Transformer	learning materials from a graduate-level introductory data science course at an R1 university in the northeastern United States	Information Score, GPT-3 classification and Human evaluation	Used hierarchy extraction to score relevance. 74.38% of questions useful. Generated mostly descriptive answers.
Diwan et al. (2023)	GPT-2 (Fine-tuned)	Custom (40GB text)	Human evaluation	Introduced Definition Generator for quiz creation. BERT performed best for keyphrase extraction. 79.06% of questions are relevant, 84.68% semantically correct and 83.12% grammatically correct.

Continued from previous page

Study	Model(s) Used	Dataset(s)	Evaluation Metrics	Key Findings
Bulathwela et al. (2023)	T5 (Fine-tuned)	SQuAD, SciQ	BLEU, ROUGE, METEOR	EduQG models surpassed the baseline Leaf model in nearly all evaluation measures. Fine-tuning with SciQ dataset improved question quality.
Lee et al. (2023)	ChatGPT (Prompt Eng.)	two passages from the Korean SAT exam	Human evaluation (expert review)	Prompting improved question validity. Struggled with cloze and yes/no questions.
Scaria et al. (2024)	GPT-4, GPT-3.5, LLaMa-2, Mistral 7B, Palm 2	Graduate Data Science	Human + LLM evaluation (Bloom's taxonomy)	GPT-4 with CoT performed best (94.1% quality). Too much context reduced performance.
Lewis et al. (2020)	BART-Large (400M)	Open-domain QA datasets	Human evaluation (factual accuracy)	RAG outperformed BART in factual accuracy (42.7% cases).
Reddy et al. (2017)	RNN AQG Model	QA pairs dataset	BLEU	Early usage of knowledge graphs for AQG.
Li et al. (2023)	KG-Enhanced LM	Custom KG dataset	BLEU-4, Human evaluation	KGEL improved multi-hop question generation, surpassing GPT-2 by 27% BLEU-4.
Lohr et al. (2025)	GPT-4-turbo with RAG	Custom educational datasets	Human evaluation	Generated learning questions were contextually relevant but required refinement for pedagogical accuracy.

Continued from previous page

Study	Model(s) Used	Dataset(s)	Evaluation Metrics	Key Findings
Sayed et al. (2024)	Falcon, Orca, LLamA-2, Google Gemini	User-uploaded PDF documents (Linux Server Administration)	Human evaluation (contextual relevance)	Google Gemini outperformed all models in question quality and relevance.

The studies summarized in Table 3.2 can be broadly grouped into three key categories based on their methodological approaches: (1) neural and transformer-based models, (2) large language models (LLMs), and (3) retrieval-augmented generation (RAG) and knowledge graph (KG)-enhanced methods. The following sections provide a detailed discussion of the publications within each of these categories, highlighting their unique contributions, methodologies, and findings.

3.2.1 Neural and Transformer-based Models

With the advent of neural networks and transformer-based architectures, question generation (QG) has seen significant advancements utilizing models with deeper semantic understanding to capture intricate relationships within the input text. The adoption of deep learning and neural network models enabled the generation of more contextually relevant, diverse, and grammatically coherent questions, moving beyond simple extraction-based approaches to more sophisticated generative techniques. Traditional rule-based and statistical methods have given way to deep learning approaches that leverage large-scale pretraining and fine-tuning techniques (Lohr et al., 2024). In particular, transformer models like GPT-2 and T5 have demonstrated strong capabilities in generating fluent and contextually relevant questions (Lopez et al., 2021; Bhat et al., 2022; Diwan et al., 2023). This section reviews several key studies that employ neural and transformer-based models for automatic question generation.

One of the influential works in the area of transformer-based architectures is by Lopez et al. (2021) who explored the application of GPT-2 for question generation on The Stanford Question Answering Dataset (SQuAD). Lopez et al. (2021) used the 124M parameter GPT-2 as their base model on The Stanford Question Answering Dataset (SQuAD) showing that transformer models can be used to create question generation systems. They finetuned six Question Generation (QG) models and trained for 3 epochs using the Adam optimizer. For the paragraph-level question generation they used the top-p nucleus sampling method with a value of p=0.9 and

a temperature of 0.6. They evaluated the performance using automatic evaluation metrics such as BLEU_1, BLEU_2, BLEU_3, BLEU_4, ROUGE_L ad METEOR on two approaches: One Question Per Line and All Questions Per Line. In their experiments, the One Question Per Line model achieved the best scores, although the difference compared to the other approach was relatively small. Their best performing model outperforms previous more complex RNN-based Seq2Seq models, with an 8.62 and a 14.27 increase in METEOR and ROUGE L scores. They observed failed generations characterized by repetitive loops of the last three words and instances where the question was prematurely truncated. As a possible explanation for the first type of failure, they propose that the model's attention weights may become evenly distributed across a set of seemingly random positions. Rather than guiding the model toward selecting the most appropriate next token, the attention mechanism becomes ineffective—introducing noise and leading to confusion in the generation process. The second type of failure is believed to occur when the model reaches the maximum generation length while still copying text from the context, resulting in incomplete or abruptly ended outputs. Their analysis suggests that a context length of around ten sentences yields the best results (Lopez et al., 2021). Their approach shows limitations, since the majority of generated questions showcase only a straightforward extraction from the provided context because 88.26% of questions fall under the identification type (Diwan et al., 2023).

While Lopez et al. (2021) demonstrated the potential of transformer-based architectures like GPT-2 for generating fluent questions from SQuAD, their approach primarily focused on direct extraction-based questions with limited conceptual depth. Building upon this foundation, Bhat et al. (2022) extended the capabilities of transformer models by incorporating a structured concept hierarchy and a more comprehensive evaluation framework. Their work utilized the T5 model for question generation and a hierarchy extraction model on the text content to score the generated questions based on their relevance to the extracted key concepts. As dataset Bhat et al. (2022) used the learning materials from a graduate-level introductory data science course at an R1 university in the United States. The hierarchy of concepts is extracted using the MOOCCubeX pipeline, which enables weakly supervised, fine-grained concept extraction from a given corpus— without the need for expert-labeled data or manual intervention. For the task of Question Generation they finetuned T5 on SQuAD, a reading comprehension dataset. Bhat et al. (2022) performed a three way evaluation using Information Score (IS), GPT-3 classification and Expert evaluation. To evaluate the quality of the generated questions, a combination of automated and human-centered metrics was applied. A custom metric, the Information Score, was introduced to measure how closely each question aligns with key concepts extracted during the concept hierarchy phase. For each question qq, the score is computed as the proportion of its tokens

that match the identified concept set CC, thereby normalizing for question length. In addition, GPT-3 was used for binary classification of questions as either useful for learning or not useful. A question was considered useful if it directly assessed domain knowledge, while questions related to prerequisites or vague topics were labeled as not useful. The GPT-3 model was fine-tuned on the LearningQ dataset, which contains over 5,600 annotated student questions. To further ensure quality, an expert evaluation was conducted. Two data science instructors with over five years of teaching experience independently rated each question using the same classification criteria. The inter-rater reliability (Cohen's K = 0.425) indicated moderate agreement, with a 75.59% match rate. Disagreements were resolved through discussion, ensuring that every question was reviewed and classified by both human evaluators and the GPT-3 model. In this study, a total of 203 questions were generated using a hierarchical content structure across module, unit, and title levels. Of these, 151 questions (74.38%) were classified as useful for learning by a fine-tuned GPT-3 model. A comparison with expert ratings showed agreement in 135 cases (66.5%), with the remaining differences analyzed qualitatively. The authors found that expert raters tended to favor questions that either established a clear context or targeted specific concepts. Conversely, questions that were vague or overly general—despite being relevant in theme—were often deemed not useful by experts but still rated positively by the model. This revealed GPT-3's limitations in discerning conceptual specificity (Bhat et al., 2022). Diwan et al. (2023) however stated that the questions mostly lead to descriptive answers which is not ideal for multiple choice question types.

While Bhat et al. (2022) advanced transformer-based question generation by integrating concept hierarchies and multi-faceted evaluation techniques to enhance alignment with learning objectives, further refinements in semantic depth and question variety were explored by Diwan et al. (2023). Their work shifted focus toward the generation of more diverse quiz formats—namely Overview and Reflection quizzes—by combining multiple semantic extraction methods with a custom natural language generation component built on GPT-2. This approach emphasized not only question relevance but also the generation of well-structured definitions and key concepts, aiming to broaden the scope and applicability of automatically generated quizzes. Their pipeline consists of multiple components based on different semantic models and a natural language generation (NLG) component which uses GPT-2. They propose a NLG model called Definition Generator which creates natural language text in form of definitions for a given keyword or concept. For their Definition Generator they fine-tuned GPT-2 on 40GB of Internet text. For the task of generating overview quizzes, they first extract key-phrases and topical anchors from the resources, identity the main theme and passed the description of the main theme to the Definition Selector. For the key-phrase extraction Di-

wan et al. (2023) evaluated different extraction methods, namely EmbedRank, TF-IDF, Nopund keyphrase extraction and BERT based key-phrase extraction. BERT perfomed best on their corpus. The top ranked key-phrase represents the topical anchor for the learning resource in their approach. Their human evaluation concluded that 79.06% of the generated questions are relevant to the learning source, 84.68% are semantically correct, and 83.12% are grammatically correct. The human evaluation of the options of the MCQ resulted 88.6% in a good clarity, 61.4% of the semantic options are close enough to the correct option, 54.2% of the syntactic options are close enough to the correct option (Diwan et al., 2023).

In parallel to these efforts, Bulathwela et al. (2023) pursued a different direction by focusing on improving both the predictive accuracy and linguistic quality of educational questions through their Leaf system. Their work centered on fine-tuning a T5 model with the SQuAD dataset and further pretraining it on the SciQ dataset—a collection of crowd-sourced science exam questions—aiming to enhance the generation of coherent and diverse questions for educational quizzes. Bulathwela et al. (2023) used the Leaf system which finetunes T5 with SQuAD dataset as baseline and additionally pretrained it with the SciQ dataset, which is a collection of crowd-sourced scientific exam questions covering physics, chemistry, and other science. Their evaluation of the Question Generation (QG) models considered two key quality aspects: (i) prediction accuracy and (ii) linguistic quality of the generated questions. To measure prediction accuracy, the BLEU score and F1 score, were employed. For assessing linguistic quality, the perplexity, and diversity scores were used. A lower perplexity indicates greater coherence in the generated questions, while the diversity score reflects the variability of the vocabulary. Higher diversity values, when coupled with lower perplexity, suggest that the model generates questions with both grammatical precision and a richer vocabulary. The study demonstrated that the proposed EduQG models (Small and Large) consistently outperformed the baseline Leaf model across both predictive and linguistic quality metrics. In particular, EduQG Large achieved the highest results across most predictive metrics (e.g., BLEU-1 = 29.19, F1 = 33.18) and attained improved diversity (0.749), with statistically significant improvements ($p < 0.01$). Overall, the EduQG models surpassed the baseline Leaf model in nearly all evaluation measures when generating educational questions on the SciQ test dataset (Bulathwela et al., 2023).

Building on the foundational work of transformer-based models, recent advancements have been marked by the emergence of LLMs. These models have substantially redefined the landscape of AQG, offering enhanced capabilities in both scalability and contextual understanding, as evidenced in the works of Lee et al. (2023); Scaria et al. (2024) and Li et al. (2023).

3.2.2 Large Language Models (LLMs)

The rise of LLMs has significantly transformed the processes of question generation and answering, with these models increasingly taking on the responsibility of Automatic Question Generation (AQG), as demonstrated in the works of Lee et al. (2023); Scaria et al. (2024) and Li et al. (2023).

Lee et al. (2023) aimed to develop a validated AQG system using LLMs like ChatGPT, enhanced through prompt engineering techniques. To evaluate their system, they conducted a combination of interviews and questionnaires, assessing both the prompting manual and the generated questions with input from experts and English teachers. In the study, the authors selected two passages from the Korean SAT exam for crafting English questions using ChatGPT. The passages chosen were structured and consistent in format, making them ideal templates for the automated generation of questions. The study specifically used one literary passage and one non-literary passage to ensure a comprehensive evaluation that captures potential variations in question types. Their findings indicate that integrating LLMs with prompt engineering results in statistically significant improvements in question validity. However, certain question types, such as cloze and yes-no questions, were found to be less effective. For cloze items, ChatGPT often misplaced blanks, highlighting a limitation in handling such formats. Additionally, the generated yes-no questions tended to be overly complex, making them difficult to understand. Lee et al. (2023) attribute this issue to ChatGPT's tendency to produce hallucinations—plausible but incorrect responses—particularly in complex tasks. Furthermore, open-ended questions intended for yes-no responses received lower evaluation scores. Reviewers noted that these questions often focused on less relevant passages and exhibited unclear structures, reducing their quality, which mirrors the issues with yes-no and cloze questions (Lee et al., 2023).

While Lee et al. (2023) demonstrated the potential of leveraging LLMs like ChatGPT combined with prompt engineering to enhance the validity of automatically generated questions, their work primarily focused on English reading comprehension tasks within a narrowly defined context. Expanding on this foundation, Scaria et al. (2024) conducted a broader, comparative evaluation across multiple state-of-the-art LLMs, incorporating diverse prompting strategies and targeting higher-order educational question generation aligned with Bloom's taxonomy. Their study not only explored the influence of model size and prompt complexity but also assessed the applicability of LLMs in domain-specific, graduate-level instructional contexts.

Scaria et al. (2024) examined the ability of five state-of-the-art LLMs of different sizes, namely Mistral 7B, LLaMa 2 70B, Palm 2, GPT 3.5, GPT 4 for the task of automated educational question generation (AEQG) for a graduate-level data science

course. They set the temperature to 0.9 to enhance variety and diversity in generating the questions. They examined different prompt styles/strategies each differing in complexity. They used zero-shot and few-shot techniques as well as Chain of Thought (CoT) prompting. The first prompt strategy was to augment prompts with CoT instructions aiming to make the LLM think sequentially and include the persona of a graduate-level university course instructor creating questions for their students. The second prompt strategy included definitions of the six cognitive levels of the Bloom's taxonomy aiming to enhance the quality of the questions. The third strategy included an example for each of Bloom's taxonomy level. They performed a human evaluation as well as a LLM evaluation and concluded that the human evaluation proves to be more effective in their case. In their human evaluation they presented experts with LLM-generated questions in random order along with the course topic and let the experts assess the questions with a hierarchical nine-item rubric which includes the items Understandable, TopicRelated, Grammatical, Clear, Rephrase, Answerable, Central, WouldYouUseIt, Bloom's Level. If any of the items is assessed with no, none of the other subsequent items are evaluated to streamline the evaluation process and minimize time as well as effort required for the experts. For Scaria et al. (2024) the generated questions are considered as high quality if the criteria Understandable, Grammatical, Clear and Answerable are marked with yes and WouldYouUseIt is marked with yes or maybe by the experts. Their experiments concluded that LLMs can produce high-quality and diverse educational questions aligned with Bloom's taxonomy but the performance varies based on the size of the model and the prompting style. In the human evaluation GPT 4 scored highest of all models using a CoT & skill explanation prompt with 94.12% in Quality and 89.53% in Skill. LLaMa 2 70B scored received the highest score in the human evaluation using the CoT & example prompt with 77.45% in Quality and 45% in Skill. In the automated evaluation of the automatically generated questions GPT 3.5 with a simple prompt technique scored highest in Quality with 82.35% and Palm 2 using Cot & skill explanation in the prompt scored highest in Skill with 49.25%. LLaMa 2 70B scored highest using a CoT & example prompt with 80.39% Quality and 40.24% Skill. They also concluded that using a lot of information significantly reduces the performance of LLMs, especially for open-source models. Scaria et al. (2024) describe limitations of the models in the understanding of specific domains and conclude that Retrieval-Augmented Generation (RAG) can address those limitations (Scaria et al., 2024).

3.2.3 Retrieval-Augmented Generation (RAG) and Knowledge Graphs

Traditional fine-tuning methods, although effective, are often resource-intensive and impractical for dynamically changing datasets or domain-specific applications. RAG offers a compelling alternative by decoupling knowledge retrieval from the model's fixed parameters. Instead of relying solely on the model's internalized (parametrized) knowledge, RAG architectures retrieve relevant context from external sources and condition the generation process on this retrieved information. This approach not only mitigates hallucinations but also enables real-time adaptability without retraining the model (Iusztin et al., 2024).

Lewis et al. (2020) explored the efficiency of RAG for knowledge-intense NLP tasks using BART-large, a pre-trained seq2seq transformer with 400M parameters. The authors propose two RAG model variants: one that conditions on the same retrieved passages for the entire generated sequence and another that allows different passages for each token. Both variants leverage dense retrieval (using DPR) to fetch relevant documents and then generate text by attending to the retrieved information. The models are evaluated on a range of tasks including open-domain question answering, abstractive question answering, Jeopardy question generation and fact verification. Evaluators found that BART was more factual than RAG in just 7.1% of cases, whereas RAG was more factual in 42.7% of cases. Additionally, both RAG and BART were factual in a further 17% of cases. This clearly demonstrates RAG's effectiveness on the task compared to the state-of-the-art generation model. Furthermore, evaluators noted that RAG's outputs were significantly more specific (Lewis et al., 2020).

Building on the foundational demonstration of RAG's effectiveness for knowledge-intensive NLP tasks by Lewis et al. (2020), subsequent work by Lohr et al. (2025) extended the application of RAG architectures to the educational domain. By integrating RAG with GPT-4-turbo, their study shifted the focus from general factual generation to the more nuanced task of producing pedagogically meaningful learning questions, thereby evaluating the model's suitability for instructional content creation. Lohr et al. (2025) conducted an expert-based evaluation of RAG integrated with GPT-4-turbo, investigating its capacity to generate learning questions that are not only contextually relevant but also pedagogically meaningful. Their system generated five questions per topic. Their findings indicate that the generation of structural and semantic annotations by LLMs was largely effective. In contrast, the performance declined notably when generating relational annotations. The resulting questions frequently failed to meet established educational quality standards. These outcomes suggest that, while LLMs hold promise as tools for augmenting the development of learning materials, their

current capabilities necessitate substantial human oversight and refinement to ensure pedagogical appropriateness and content validity (Lohr et al., 2025).

In a more application-focused approach, Sayed et al. (2024) leveraged RAG for the task of AQG, utilizing advanced LLMs including Falcon, Orca, LLaMA-2 and Google Gemini. Their system enables users to upload PDF documents, from which a knowledge base is automatically constructed through document embedding. This knowledge base then serves as the retrieval component, supplying relevant context to the language model during prompt generation, thereby enhancing the contextual accuracy and relevance of the generated questions. The dataset employed in this study was derived from a user-uploaded file, which served as the primary input for the model's output generation. Specifically, the investigation utilized the PDF version of Linux Server Administration by Wale Soyinka as the source material for generating questions. A Linux domain expert conducted a manual review of approximately 1,400 questions generated by the models, assigning scores for both overall quality from 1 to 5 and contextual relevance from 0 to 1. The evaluation revealed notable performance differences among the models. Falcon received the lowest average rating (1.82) and relevance score (0.45), indicating poor alignment with user expectations. Orca demonstrated moderate improvement, achieving a rating of 3.58 and a relevance score of 0.87. LLaMa-2 further improved with a 3.68 rating, although its relevance score (0.85) was slightly lower than that of Gemini. Google Gemini outperformed all other models, attaining the highest average rating (3.92) and relevance score (0.90), reflecting superior question quality and contextual fit (Sayed et al., 2024).

This section has highlighted the growing role of RAG as a scalable and adaptable alternative to traditional fine-tuning methods in AQG. By retrieving external knowledge at generation time, RAG mitigates hallucinations and enhances contextual relevance, as demonstrated across diverse tasks—from open-domain QA to educational content creation. Recent studies have applied RAG-integrated systems with advanced LLMs, such as GPT-4-turbo, LLaMa-2 and Google Gemini, to produce high-quality and contextually grounded learning questions. According to Lewis et al. (2020), RAG-generated outputs were found to be more specific compared to traditional generation methods, and Lohr et al. (2025) highlighted that, while LLMs show promise in assisting the creation of learning materials, they still require significant human involvement and refinement to ensure that the questions are pedagogically sound and valid. Despite these advancements, challenges remain in ensuring the generation of relationally accurate questions.

One promising direction for enhancing AQG systems involves the integration of Knowledge Graphs (KGs), which offer structured, domain-specific knowledge that can enrich RAG systems by improving factuality and enabling more complex reasoning.

Knowledge Graphs in Question Generation

Structured knowledge sources such as Knowledge Graphs (KGs) provide a promising extension to RAG systems in the area of AQG, particularly for enhancing factuality and enabling complex reasoning as demonstrated in the work of Li et al. (2023).

While Knowledge Graph-based Retrieval-Augmented Generation has been predominantly applied to Question Answering (QA) tasks—such as in the works of He et al. (2024), Xu et al. (2024b) and Wiratunga et al. (2024)—its potential for educational Question Generation (QG) with LLMs remains underexplored. Reddy et al. (2017) employed an RNN-based model to generate QA pairs from KGs, while Li et al. (2023) proposed a multi-head attention generation module integrated with KGs.

Recent advancements in natural language processing have significantly enhanced the capabilities of AQG. Traditional QG models predominantly focus on single-hop reasoning, wherein questions are derived from isolated sentences. However, such models fall short in capturing the complexity of multi-hop reasoning, which involves synthesizing information across multiple sentences or documents. This limitation has prompted the development of more sophisticated models that can emulate human-like reasoning. In response to this challenge Li et al. (2023) proposed a novel framework namely the Knowledge Graph-Enhanced Language Model (KGEL), specifically designed to facilitate multi-hop question generation (MHQG). The model is grounded in a three-phase architecture: (1) context understanding, (2) reasoning and (3) generation. Initially, a pre-trained GPT-2 model is employed to encode the input context and corresponding answer, yielding a rich semantic representation. Subsequently, named entities within the context are extracted using a BERT-based named entity recognition (NER) model and a knowledge graph is constructed to represent the relational structure among these entities. To enhance reasoning over this structured information, the authors introduced an answer-aware graph attention network (GAT). This module selectively emphasizes entities relevant to the target answer, thereby simulating a human-like chain of reasoning. A bi-attention mechanism further updates the answer representation by aligning it with entity-level information. The final phase involves a multi-head attention-based decoder, which leverages the enriched contextual embeddings to generate fluent, logically consistent questions. Experimental evaluations were conducted on the HotpotQA dataset, a benchmark for multi-hop QA and QG tasks. The KGEL model demonstrated substantial improvements over baseline models, including standard GPT-2 and other seq2seq variants. Notably, KGEL achieved a 27% increase in BLEU-4 score relative to GPT-2, alongside improvements in ROUGE-L and METEOR metrics. Human evaluations corroborated these

findings, highlighting significant gains in fluency, completeness, and answerability. An ablation study revealed that the removal of core components—such as the graph attention mechanism or the multi-head attention module—led to a marked decline in performance, underscoring the effectiveness of the proposed architecture. Despite its strengths, the authors acknowledged certain limitations, particularly in terms of dataset diversity and the model's capacity to discern subtle semantic relationships. Future work may benefit from integrating external commonsense knowledge bases, employing copy mechanisms and expanding to more diverse training corpora to further enhance the model's reasoning capabilities. Overall, KGEL represents a promising advancement in the field of question generation, offering a structured and interpretable approach to multi-hop reasoning through the integration of knowledge graphs and pre-trained language models (Li et al., 2023).

Seyler et al. (2017) use knowledge graphs for automatic question generation. Their question generation system consists of three major components: query generation, difficulty estimation, and query verbalization. For the generation of multiple choice questions a fourth component is added which can generate distractors and quantify their impact on question difficulty. The approach proposed by Seyler et al. (2017) begins by selecting a named entity from the knowledge graph to serve as the correct answer. It then constructs a structured triple-pattern query, which is formulated to retrieve only that specific entity. In the case of multiple-choice question generation, additional entities are selected as distractors. Finally, a template-based verbalization method is employed to convert the structured query into a natural language question. Seyler et al. (2017) employed YAGO2s as the reference knowledge graph, which comprises approximately 2.6 million entities and 300,000 types organized in a hierarchical structure. It includes over 100 distinct predicates, collectively forming more than 48 million factual assertions. YAGO entities are linked to corresponding Wikipedia entries, while YAGO types are aligned with WordNet synsets or Wikipedia categories. To estimate question difficulty, the approach utilizes the ClueWeb09 and ClueWeb12 document collections, along with FACC annotations provided by Google. These annotations offer semantic disambiguation of named entities from Freebase, which are subsequently mapped to YAGO2s using their associated Wikipedia articles. Thirteen evaluators participated in the study, each assessing an average of 92.5 questions. The difficulty rankings generated by the classifier showed a moderate level of agreement with the human judgments (Seyler et al., 2017).

3.2.4 Summary

The integration of LLMs into AQG, has significantly advanced the development of educational tools. Early studies, such as those by Lee et al. (2023), demon-

strated that LLMs like ChatGPT, when paired with prompt engineering techniques, can generate valid and contextually relevant questions. However, the research highlighted limitations, particularly in generating certain question types such as cloze and yes-no questions, where issues like misplaced blanks and overly complex structures were prevalent (Lee et al., 2023).

Further investigations, such as those by Scaria et al. (2024), revealed that the quality of questions generated by LLMs can be significantly influenced by the complexity of the prompts used. While these studies focused primarily on evaluating the individual quality of each question, they still face the limitation that no work explicitly evaluated the generation of entire quizzes—a necessary extension to ensure that these questions work cohesively in larger educational assessments. The task of ensuring that a series of generated questions are coherent, pedagogically aligned, and balanced at the quiz level remains unexplored in these studies.

Retrieval-Augmented Generation (RAG) offers an important advancement by decoupling the fixed parameters of traditional LLMs from the task of dynamically retrieving relevant knowledge from external sources. This approach reduces hallucinations and enables models to adapt in real time without the need for retraining (Iusztin et al., 2024). Studies such as Lewis et al. (2020) have demonstrated RAG's ability to generate more factual and contextually specific outputs compared to traditional sequence-to-sequence models. Similarly, Lohr et al. (2025) showed that when LLMs like GPT-4-turbo are augmented with external retrieval, they are capable of producing structurally and semantically relevant questions. However, these evaluations were again conducted on the individual questions, not on the ability to generate a full set of questions that would constitute a well-balanced and coherent quiz.

The practical applications of RAG for AQG, as seen in Sayed et al. (2024), further underscore the value of integrating domain-specific retrieval systems. Their findings revealed that models like Google Gemini were particularly effective in producing questions that were both high in quality and relevance to the source material. Nonetheless, their analysis, like others, focused on evaluating question-level outputs rather than assessing how well these questions could function together within an integrated quiz or assessment framework. Thus, the need for holistic evaluation at the quiz level remains an open challenge in the field.

In addition to RAG, Knowledge Graphs (KGs) have shown promise in improving multi-hop reasoning in AQG. The Knowledge Graph-Enhanced Language Model (KGEL), proposed by Li et al. (2023), represents a significant step forward by integrating graph-based reasoning with LLMs to generate multi-hop questions. This approach outperformed standard models in terms of fluency and logical consistency. However, like all prior works, this research was limited to question-level evaluation and did not assess the model's ability to generate full quizzes with in-

terconnected questions that test a range of cognitive skills or cover diverse aspects of the subject matter.

Although Knowledge Graphs also hold potential for improving AQG, as seen in the work by Seyler et al. (2017), which employs KGs for multiple-choice question generation, the research still focused primarily on generating individual questions, without investigating how these isolated items could be combined into a larger educational context. This point is critical, as generating individual questions that are valid and relevant is only one aspect of creating comprehensive, balanced, and effective quizzes.

In summary, while the integration of LLMs, RAG and KGs has substantially advanced the field of AQG, the research to date has largely been limited to the evaluation of individual questions rather than fully constructed quizzes. As a result, this study aims to explore how these models can be utilized not only to generate individual questions but also to ensure that they work together cohesively as part of a larger, pedagogically sound quiz. Moreover, further work is needed to improve the overall coherence and quality of quizzes, ensuring they are aligned with educational goals, represent a balanced range of cognitive skills and are suitable for diverse learning contexts.

3.3 Research Gap

Despite the advancements in AQG using models like OpenAI's GPT-3, existing solutions are often cloud-based, raising concerns regarding privacy and the high costs associated with long-term usage. While GPT-3 has proven effective, these issues limit its accessibility and scalability for widespread use, particularly in private or sensitive contexts. This thesis aims to address these concerns by leveraging LLaMa-3, an open-source language model that can be deployed locally. LLaMa-3 not only offers a more cost-effective solution but also mitigates privacy concerns, enabling secure and sustainable experimentation in AQG tasks.

Additionally, while RAG has shown promise in enhancing the factuality and adaptability of question generation systems, its use in AQG remains underexplored, particularly in terms of leveraging domain-specific knowledge through the integration of Knowledge Graphs (KGs). This study aims to bridge this gap by investigating the potential of KGs to further enrich RAG systems, improving both the factual accuracy and the complexity of questions generated. In terms of evaluation, most existing studies primarily focus on basic metrics like grammatical correctness and topic relevance. However, the absence of a comprehensive evaluation framework specific to quiz question generation, especially for educational purposes, limits the understanding of how well these systems perform in

real-world, applied scenarios. Automated evaluation metrics, while widely used, often fall short in assessing the pedagogical and contextual relevance of the generated questions. This research proposes a more nuanced evaluation framework that emphasizes the quality of questions within a quiz context, considering expert-based evaluations to ensure questions are not only relevant but also meaningful for educational settings.

Lastly, datasets like SQuAD are highly generalized and may not capture the nuances of specialized domains. To address this, this thesis uses the MIT Explainer dataset, offering a more focused context for evaluating question generation in technical fields. This allows for a deeper exploration of how question generation models can be tailored to domain-specific knowledge, enhancing the relevance and applicability of the generated questions.

By addressing these gaps, this thesis aims to contribute to the development of more accurate, useful, and contextually relevant question generation systems, particularly in the educational domain.

4 Use Cases & Requirements

Use cases are essential in translating user needs and system requirements into actionable functionality. In the context of the Automatic Question Generation (AQG) system, identifying and understanding the personas and use cases helps ensure that the system is designed to meet the real-world needs of its users, including teachers and students. These use cases guide the development process, ensuring that the system is both functional and intuitive, while addressing the specific challenges faced by users in educational environments. Furthermore, well-defined use cases are critical in establishing clear requirements, validating system functionality and ensuring that the system aligns with user expectations. Research has shown that user-centered design, based on detailed use case scenarios, leads to more effective and engaging software solutions (Nielsen, 1994; Sharp et al., 2007).

4.1 Personas

4.1.1 Teacher

The teacher persona represents individuals who are responsible for delivering educational content and assessments. Teachers may use the AQG system to create quizzes, enhance learning materials and evaluate student comprehension. They prioritize efficiency in producing high-quality, relevant questions and value tools that allow them to adjust quiz content and difficulty to align with the diverse needs of their students.

- Name: Christina, University Science Teacher
- Age: 35
- Goals:
  - Create quizzes aligned with curriculum goals.
  - Save time in generating diverse and challenging questions.
  - Improve student engagement through interactive quizzes.
- Challenges:

  - Difficulty in finding time to manually create and curate quiz questions.
  - Need for personalized, adaptive learning tools for diverse student groups.

4.1.2 Student

The student persona represents individuals engaged in learning, either in a formal or informal context. They will interact with the quizzes generated by the AQG system to assess their knowledge and enhance their learning. Students may value quizzes that are relevant to the material, offer appropriate difficulty and provide meaningful feedback.

- Name: Nina, University Student
- Age: 21
- Goals:
  - Improve knowledge and comprehension of course material.
  - Engage with quizzes that challenge but don't frustrate them.
  - Receive feedback after quizzes to track their progress.
- Challenges:
  - Struggles with too easy or irrelevant quiz questions.
  - Requires guidance on areas where improvement is needed.

4.2 Use Cases

4.2.1 Use Case 1: Teacher

Scenario: Christina, a university science teacher, wants to quickly generate quizzes for her students based on the content covered in recent lessons. Using the AQG system, she inputs a set of educational articles or lessons and the system automatically generates a series of questions that are aligned with the material. She is able to customize the difficulty of the questions and review the generated output before presenting it to her students.

Actors: Teacher (Christina), AQG System

Preconditions:

Teacher has input educational material (e.g., articles, textbooks) into the system.

Teacher has access to the AQG system and is logged in.

Steps:
Teacher uploads content or provides links to lessons.
AQG system generates quiz questions.
Teacher reviews, edits and customizes the questions.
Teacher generates and distributes the final quiz to students.
Postconditions:
Teacher has a ready-to-use quiz with appropriate questions for the students.

4.2.2 Use Case 2: Student

Scenario: Nina, a university student, uses the AQG system to practice for an upcoming exam. After studying the course material, she accesses the generated quizzes based on the topics she has learned. The system presents her with a variety of questions and upon completion, she receives feedback to understand which areas need more focus.

Actors: Student (Nina), AQG System
Preconditions:
Student has access to the AQG system.
Student has already studied the relevant course material.
Steps:
Student selects a quiz based on the material they wish to review.
Student completes the quiz.
Student receives feedback on their answers and areas for improvement.
Postconditions:
Student has reinforced knowledge and identified weak areas to focus on.

4.3 Requirements

4.3.1 Functional Requirements

A functional specification defines what a system should do, grounded in a clear understanding of the domain and organizational context. This ensures that stakeholders—customers, users and developers—share a common understanding of the problem the system is intended to solve (Loucopoulos and Karakostas, 1995). Based on this definition, the functional requirements specify the system's core capabilities to support its educational purpose and meet the needs of teachers and students. These include:

- **Automatic Question Generation:** The system must be able to automatically generate quiz questions based on a provided text, such as educational articles.
- **Customizable Question Difficulty:** The system should allow teachers to set the difficulty level of questions, adjusting the complexity based on the target audience.
- **Review and Edit:** Teachers should be able to review, modify and customize the generated questions before finalizing them.
- **Feedback Generation for Students:** The system should provide students with feedback after they complete a quiz, indicating correct/incorrect answers and areas for improvement.

4.3.2 Non-functional Requirements

Ameller et al. (2012) conducted interviews with 13 experienced software architects to identify the most significant non-functional requirements commonly prioritized in software systems. The study highlighted the following as particularly relevant: performance, usability, security, availability, interoperability, maintainability, accuracy, fault-tolerance, reusability, scalability, modularity, monitoring and portability.

Building on the insights of Ameller et al. (2012), the key non-functional requirements for the automatic question generation and quiz application developed in this thesis are outlined below. Each requirement has been selected based on its relevance to the system's intended functionalities and user interactions:

- **Performance:** The system should generate questions and deliver quizzes promptly, ensuring minimal waiting time for users. High performance is critical to maintain user engagement, especially when processing lengthy texts or managing multiple concurrent quiz participants.
- **Scalability:** The system should be able to handle a large number of users and large input resources without significant performance degradation.
- **Usability:** The interface should be intuitive and user-friendly for both teachers and students.
- **Security:** The system must ensure that data, especially sensitive student information, is securely handled and stored.
- **Accessibility:** The system should be accessible to users with disabilities, adhering to Web Content Accessibility Guidelines (WCAG) guidelines.

4.3.3 Contextual Requirements

Contextual requirements refer to system needs that arise directly from the environment in which the software will be used. They are elicited within the user's actual work setting to ensure that the software aligns with real-world usage conditions and constraints (Keller, 2011). These requirements help capture critical environmental, organizational and technical factors that conventional elicitation methods may overlook. The following are key examples of contextual requirements for this system:

- **Compatibility**: The system should work across common web browsers (Chrome, Firefox, Safari) and devices (PCs, tablets, smartphones).
- **Localization**: The system should be adaptable to various languages, enabling users to interact with it in their language of choice.

4.4 Quality Attributes

4.4.1 Usability

Usability is a key quality attribute for this system, particularly since both teachers and students will interact with it frequently. The system must offer an intuitive user interface that requires minimal training. According to Nielsen (1994), *learnability* is one of the most fundamental usability attributes. Features such as the quiz creation tool, difficulty adjustments and progress tracking should be clearly accessible. The inclusion of tutorials or help sections will support users unfamiliar with the platform. Another important usability attribute identified by Nielsen (1994) is *efficiency*, which ensures users can perform tasks quickly and effectively. Additional attributes include *memorability*, so returning users can easily reestablish proficiency after a period of not using the system; *error management*, where the system should minimize the occurrence of errors and help users recover from them easily; and *satisfaction*, meaning the platform should offer a pleasant and engaging user experience.

4.4.2 Gamification

Gamification elements, such as leaderboards, scoring systems and achievement badges, should be integrated to enhance user engagement, particularly for students. These elements may provide motivation and foster a sense of achievement, making the learning process more interactive and enjoyable. The leaderboard will allow students to track their progress and compare their performance

with peers, while achievement badges will reward them for reaching milestones in their learning journey. In their systematic review of nine studies examining the impact of gamification on student motivation and academic performance, Jaramillo-Mediavilla et al. (2024) found that 56% of the reviewed articles reported that incorporating gamification into educational contexts enhances student engagement, participation and motivation. These studies highlighted benefits such as increased autonomy, knowledge acquisition, competence, collaboration and the development of academic skills aligned with students' basic psychological needs. Additionally, 33% of the studies implemented gamified elements like missions, badges, points, challenges, levels and avatars in their activities (Jaramillo-Mediavilla et al., 2024).

4.5 User interface

The user interface (UI) of the AQG system plays a crucial role in bridging the interaction between users and the underlying automated question generation capabilities. It has been thoughtfully designed to balance functionality with ease of use, catering to the diverse needs of both teachers and students within an educational setting. The UI supports core workflows such as quiz creation, quiz participation and performance feedback, ensuring that users can seamlessly navigate the system to accomplish their goals.

4.5.1 Homepage

The homepage shown in 4.1 serves as the central hub of the application. When logging in, users are greeted with a dashboard summarizing available quizzes. For teachers, the homepage provides quick access to quiz creation, editing tools and analytics on student performance.

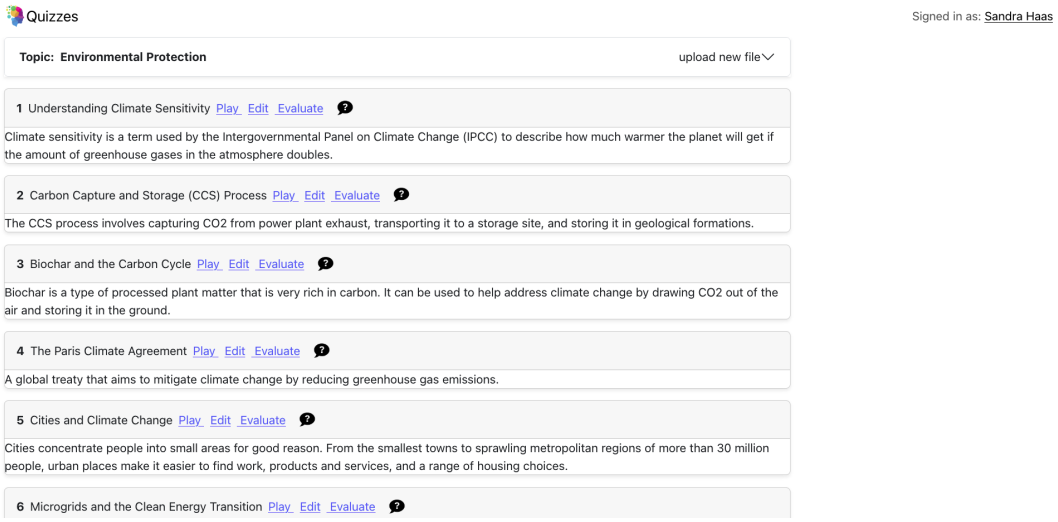


Figure 4.1: UI Homepage with Quizzes

The homepage serves as the central hub of the application. When logging in, users are greeted with a dashboard summarizing available quizzes. For teachers, the homepage provides quick access to quiz creation, editing tools and analytics on student performance.

Students are presented with quizzes sorted by topic and completion status. Visual cues and progress bars allow users to immediately assess their ongoing learning progress. The layout emphasizes simplicity and responsiveness across devices, supporting contextual requirements such as compatibility and non-functional requirements such as accessibility.

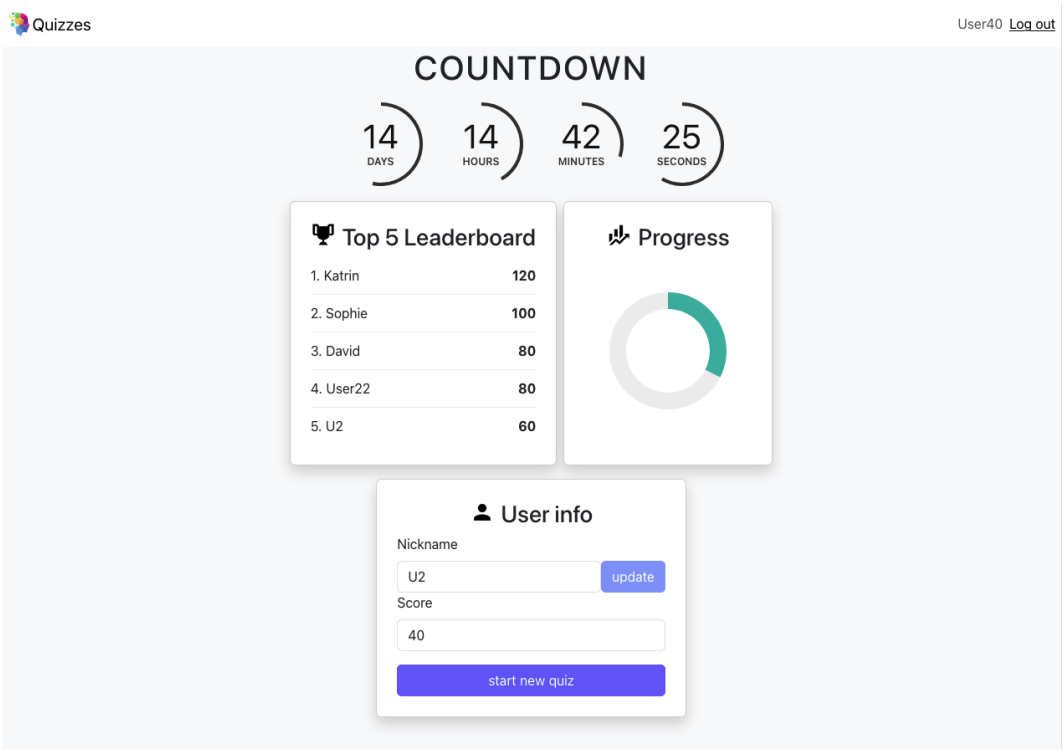
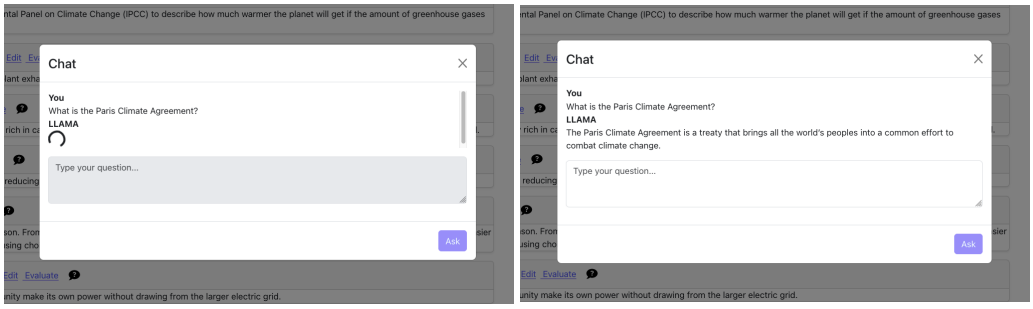


Figure 4.2: UI Student Dashboard

The student dashboard shown in Figure 4.2 offers a dynamic and engaging interface that includes a leaderboard displaying top-performing peers, a progress pie chart visualizing the student's completion status across assigned quizzes and a countdown timer indicating the remaining time until quiz deadlines. Additionally, students can personalize their experience by changing their nickname which is shown in the leaderboard.



(a) (b)

Figure 4.3: UI Chat Window Homepage

The chat feature integrated into the homepage shown in Figure 4.3 allows users to ask questions related to the content of the quiz, providing real-time clarification and support. A loading indicator is displayed while the system processes the query, as it communicates with the LLM to generate a response. For teachers, this feature could help track frequently asked questions or common areas of confusion, providing useful data to adjust quiz content and identify topics that may need further emphasis in future lessons.

4.5.2 Editing Page

The editing page shown in Figure 4.4 is tailored for the teacher persona, enabling full control over the generated content. After initial automatic generation, teachers can review the list of questions, edit or delete any that do not meet pedagogical goals and reorder questions for optimal flow. The interface supports bulk editing and previewing the quiz in student mode. This feature addresses functional requirements such as customizability and review capabilities.

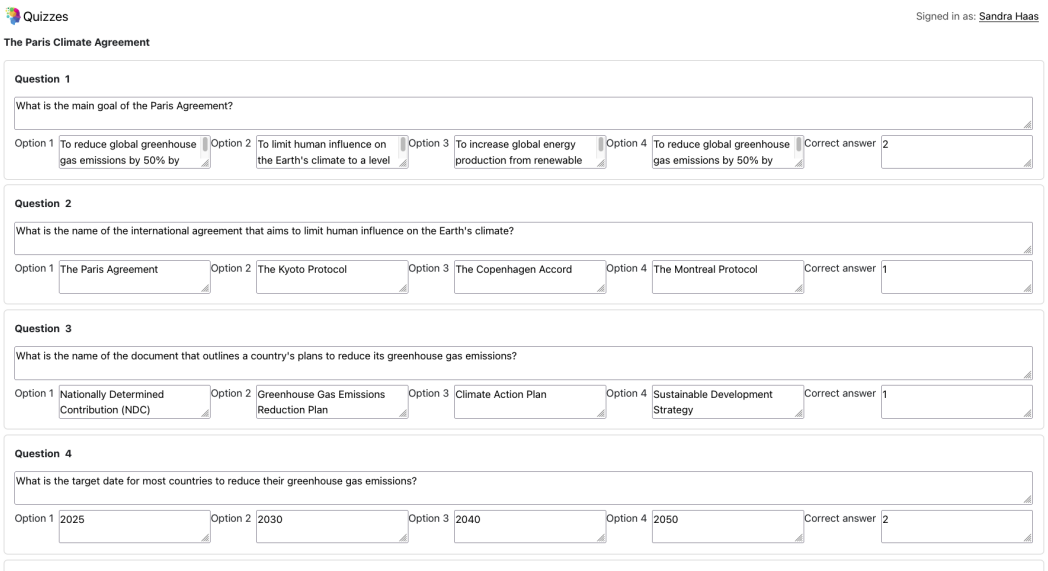


Figure 4.4: UI Editing Page of Quiz

4.5.3 Game Page

The game page shown in Figure 4.5 is styled to resemble the popular game show Millionenshow and transforms quiz-taking into an interactive experience, especially targeted at students like Nina. It incorporates gamification elements such

as scoring and progress milestones. Users receive immediate visual and auditory feedback for each answer — cheering sounds and celebratory animations reinforce correct responses, while subtle, empathetic sounds indicate incorrect ones. Points are awarded based on performance, contributing to the user's overall score and progress, thereby possibly enhancing motivation through instant recognition and reward.

Figure 4.5: UI Game Page of Quiz

During gameplay, students can access an embedded chat assistant shown in Figure 4.6 for hints, clarifications, or explanations related to the quiz content. This feature not only supports real-time formative feedback but also fosters a sense of guided learning.

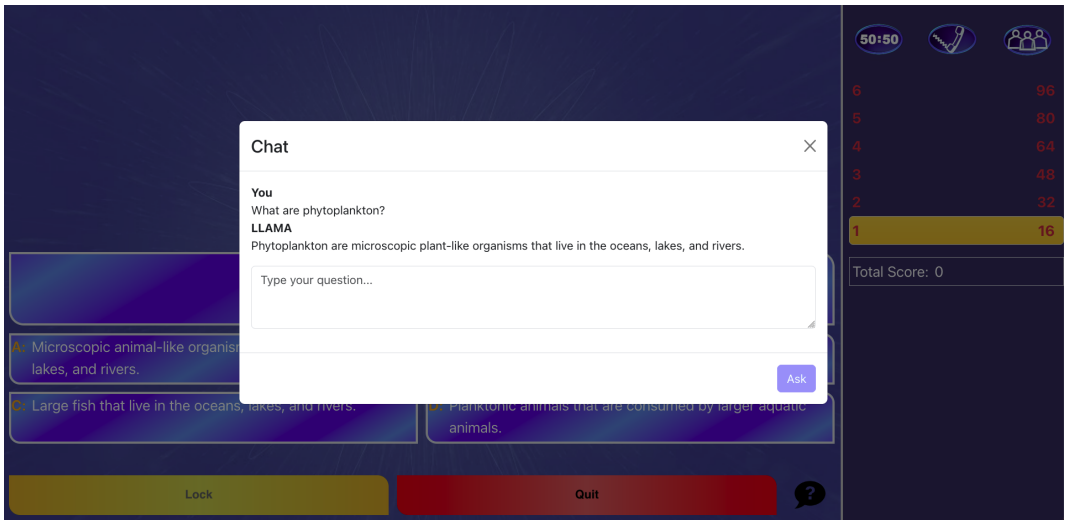


Figure 4.6: UI Chat Window in Game Page of Quiz

The user interface has been deliberately designed to fulfill the functional, non-functional and contextual requirements outlined earlier in this chapter. Functionally, it enables automatic question generation, teacher review and editing and detailed feedback for students—all presented through intuitive components such as the quiz editor, chat interface and feedback modules. Non-functionally, the UI prioritizes usability through a clean, responsive layout; performance through lightweight, fast-loading components; and accessibility by following best practices such as high-contrast visuals and keyboard navigability. Additionally, it supports scalability through modular page design and ensures compatibility across devices and browsers. Gamified elements like the Millionenshow-style quiz game, leaderboard and progress tracking directly address user motivation and engagement, aiming at enhancing the system's educational impact.

5 Problem Statement

Given a set of materials:

$$M = \{m_1, m_2, \ldots, m_n\}$$

where each m_i represents a document or piece of content related to climate change, the task is to develop a system that generates a set of quizzes:

$$\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_p\}$$

where each quiz Q_i consists of a set of quiz items:

$$Q_i = \{q_1, q_2, \ldots, q_k\}.$$

We define a mapping function:

$$\varphi : Q \to M$$

where $\varphi(q_i)$ maps each quiz item $q_i \in Q$ to exactly one material $m_n \in M$. This indicates the specific document or content that informs quiz item q_i.

We introduce the following sets:

- $D = \{1, 2, 3, 4, 5\}$ — the set of possible difficulty ratings (from 1 = not difficult to 5 = very difficult), where $d_i \in D$ is the difficulty of quiz item q_i.
- $T = \{\text{MCQ},\ \text{True/False}\}$ — the set of question types.
- $\mathcal{D}$ — the set of possible distractors for multiple choice questions. For each quiz item q_i of type MCQ, the distractors set is $\mathcal{D}_i = \{z_1, z_2, z_3, z_4\}$, where exactly one element is the correct answer a_i and the remaining three are plausible distractors.
- A — the set of correct answers derived from m_n, where for true/false questions, $a_i \in \{\text{True}, \text{False}\}$ and for multiple-choice questions, $a_i \in \mathcal{D}_i$.

For each $q_i \in Q_i$:

$$\psi(q_i) = (\varphi(q_i), d_i, t_i, a_i, \mathcal{D}_i)$$

where:

- $\varphi(q_i) = m_n \in M$ — the actual question derived from m_n
- $d_i \in D$ — difficulty rating
- $t_i \in T$ — question type (multiple choice (MCQ) or True/False)
- $\mathcal{D}_i \subseteq \mathcal{D}$ — distractors (for MCQs)
- $a_i \in A$ — correct answer derived from m_n, where:
  - $a_i \in \{\text{True}, \text{False}\}$ for true/false questions,
  - $a_i \in \mathcal{D}_i$ for multiple-choice questions, with one being the correct answer

Each quiz item q_i should meet the following criteria:

- **Relevance:** The generated question should be relevant to the document.
- **Educational Usefulness:** The generated question should be relevant and useful to the learning objectives of the material.
- **Answer Quality / Plausibility:** The generated answer should be relevant to the learning objectives of the material and distractors should be plausible.
- **Difficulty Rating** The generated question should be assigned a difficulty level on a scale from 1 (not difficult) to 5 (very difficult), indicating the cognitive challenge it presents to enable content structuring.
- **Question Type** The generated question should be categorized as either multiple choice or true/false, depending on which format best assesses the intended knowledge or skill.

The entire quiz set Q_i should meet the following criteria:

- **Educational Value:** The generated quiz should enhance the learner's understanding of the material. It should reinforce key concepts, stimulate critical thinking and support deeper engagement with the course material, making it a valuable educational tool in both self-study and instructional settings.
- **Redundancy:** The generated quiz should consist of a diverse set of questions with minimal repetition. In the best case, each question offers a unique angle or concept, contributing new value to the learner's assessment experience without duplicating content.

- **Progressiveness:** The questions should be arranged in a pedagogically meaningful order—starting with basic factual recall and gradually increasing in complexity to include conceptual reasoning and applied understanding.
- **Overall Quality:** The generated quiz should be clear, well-balanced, free of errors, and engaging. It should be suitable for real-world deployment in educational environments, demonstrating strong alignment with instructional goals and learner needs.

6 Methods

This chapter outlines the methodology used in developing the system, with a focus on ensuring reproducibility and clarity. The structure follows a top-down approach, beginning with conceptual foundations and progressing to detailed implementation steps. This organization enables a thorough understanding of both the theoretical basis and practical execution of the system.

6.1 Concepts

The recent adoption of large language models (LLMs) has significantly advanced the field of Automatic Question Generation (AQG), allowing for the creation of high-quality, contextually relevant questions with minimal human oversight. However, deploying LLMs in domain-specific applications typically requires fine-tuning, which is often computationally expensive and demands substantial annotated data (Iusztin et al., 2024). To address these challenges, this study adopts a Retrieval-Augmented Generation (RAG) approach, enhancing the base model's capabilities without the need for retraining, as demonstrated in prior works such as Lewis et al. (2020); Lohr et al. (2025); Sayed et al. (2024).

6.1.1 Retrieval-Augmented Generation (RAG)

RAG integrates a retrieval component with a generative model, allowing the system to dynamically query external knowledge sources during inference. This method enriches the model's context with relevant data, thereby improving the factuality and specificity of generated content. In doing so, RAG minimizes the risk of hallucinated outputs, a known issue in artificial intelligence-generated content (AIGC) systems (Zhao et al., 2024a). Additional technical details and background on RAG can be found in Chapter 3 and Section 2.2.2.

6.1.2 Graph-Retrieval-Augmented Generation (GRAG)

While RAG has demonstrated strong performance and is widely adopted across fields, it encounters notable limitations in real-world settings. Traditional RAG

often treats textual information as isolated pieces, failing to capture structured relational knowledge, such as citation links between academic papers, which go beyond simple semantic similarity. Additionally, by concatenating numerous text snippets into prompts, RAG can produce overly long contexts where crucial information gets lost, a problem known as the ”lost in the middle” effect. Moreover, since RAG typically retrieves only a subset of documents, it struggles to gain a comprehensive understanding of global information, which hampers its effectiveness in tasks like Query-Focused Summarization (QFS) (Peng et al., 2024).

To further enhance semantic reasoning and entity understanding, the system extends RAG by incorporating knowledge graphs. This approach, referred to as Graph-based Retrieval-Augmented Generation (GRAG), facilitates deeper semantic associations between context elements and augments retrieval quality which has been shown in prior work by Li et al. (2023).

Property Graph Index

A Property Graph Index consists of nodes and edges, each annotated with arbitrary key–value properties and labels. This graph-based representation allows for enriched querying and structured retrieval (Robinson, 2015). More information is provided in Section 2.2.2.

6.2 System Implementation

This section details the implementation architecture and associated technologies used to realize both the RAG and GRAG systems.

6.2.1 LLaMa 3.1 (8B)

The project employs the LLaMa 3.1 (8B) model in GGUF quantized form[1], employing the the Meta-Llama-3.1-8B-Instruct-Q5_K_M.gguf variant with a size of 4.92GB for efficient inference. This quantized version leverages Q5_K_M quantization to strike a balance between performance and efficiency, significantly reducing memory and compute requirements with minimal loss in accuracy. The model is deployed locally using llama.cpp[2], a high-performance C++ inference engine designed for efficient execution on resource-limited systems. This configuration en-

[1] https://huggingface.co/bartowski/Meta-Llama-3.1-8B-Instruct-GGUF
[2] https://github.com/ggerganov/llama.cpp

ables fast and cost-effective inference, making it well-suited for experimentation and production in constrained environments (Dhar, 2023).

Figure 6.1 illustrates the llama.cpp configuration used in this research.

```
llm = LlamaCPP(
        model_path=model_path,
        temperature=0.1, #0.5, #0.1,
        max_new_tokens=2000, #256,
        context_window=7900, #3900,
        # kwargs to pass to __call__()
        generate_kwargs={},
        # kwargs to pass to __init__()
        # set to at least 1 to use GPU
        model_kwargs={"n_gpu_layers": -1},
        # transform inputs into Llama3 format
        messages_to_prompt=messages_to_prompt,
        completion_to_prompt=completion_to_prompt,
        verbose=True,
        )
```

Figure 6.1: llama.cpp Configuration

LLaMA (Large Language Model Meta AI) is a family of transformer-based foundation models ranging from 7B to 65B parameters, designed to deliver state-of-the-art performance using only publicly available datasets. LLaMA demonstrates that open, efficiently trained models can outperform or match closed models like GPT-3 and Chinchilla, despite having fewer parameters (Touvron et al., 2023). The latest iteration, LLaMA 3 (2024), further extends this family, offering models with 8B and 70B parameters and introducing significant improvements in reasoning, code generation and instruction-following. LLaMA 3.1 (2024) refines the models with even larger context windows (128K) and further enhancements in alignment and safety (Meta AI, 2024a).

LLaMA builds upon the Transformer architecture which is described in chapter 2 with several modern optimizations (Touvron et al., 2023):

Feature	Change	Inspiration
Pre-normalization	LayerNorm applied to sub-layer inputs	GPT-3
RMSNorm	RMS normalization instead of LayerNorm	Zhang & Sennrich (2019)
SwiGLU Activation	Replaces ReLU with SwiGLU	PaLM
Rotary Positional Embeddings	Removes absolute positional embeddings	GPT-Neo
Causal Multi-Head Attention	Memory- and compute-efficient attention	Dao et al. (2022), xformers

Table 6.1: LLaMA Architecture Optimizations. Adapted from Touvron et al. (2023)

LLaMA is trained on publicly available datasets. LLaMA 1 uses a 1.4 trillion token dataset, while LLaMA 3.1 scales this up to a **15 trillion token corpus** (Touvron et al., 2023; Meta AI, 2024a). The dataset composition is visualized in Table 6.2. LLaMa models are trained using the AdamW optimizer (Loshchilov and Hutter, 2017), with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.95$. A cosine learning rate schedule is employed, where the learning rate decays to 10% of its maximum value by the end of training. A weight decay of 0.1 and gradient clipping at 1.0 are applied. The training setup includes 2,000 warmup steps, with both the learning rate and batch size scaled according to the model size (Touvron et al., 2023).

Dataset	% of Data	Notes
CommonCrawl	67%	Cleaned with CCNet pipeline
C4	15%	Processed CommonCrawl variant
GitHub	4.5%	Code under permissive licenses
Wikipedia	4.5%	20 languages
Gutenberg + Books3	4.5%	Public domain and The Pile
arXiv	2.5%	Scientific papers
StackExchange	2%	High-quality Q&A

Table 6.2: LLaMA 1 Pretraining Dataset Composition. Adapted from LLaMA: Open and Efficient Foundation Language Models, by H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, 2023, arXiv (https://arxiv.org/abs/2302.13971).

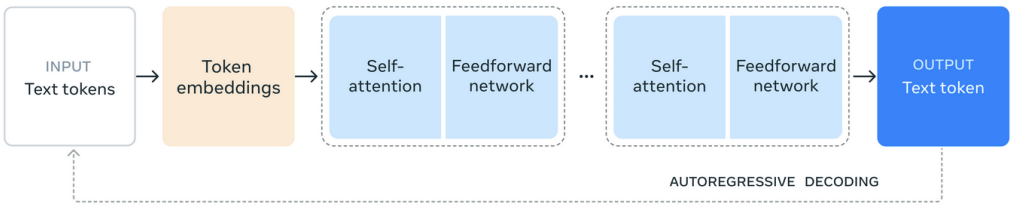


Figure 6.2: LLaMa 3.1 Architecture. Reprinted from Introducing Llama 3.1: Our most capable models to date, by Meta AI, 2024, Meta AI Blog (https://ai.meta.com/blog/meta-llama-3-1/). Copyright 2024 by Meta Platforms, Inc.

LLaMA 3 introduces an enhanced tokenizer (improved handling of non-English text), a 128K context window in LLaMA 3.1 and optimized architecture for efficiency and factuality (Meta AI, 2024a).

LLaMA 3.1 is an auto-regressive language model (see more information in chapter 2) built on an optimized transformer framework, shown in Figure 6.2. The instruction-tuned variants are aligned with human preferences for helpfulness and safety through supervised fine-tuning and reinforcement learning guided by human feedback (Meta AI, 2024b).

6.2.2 RAG

This subsection details the implementation of the Retrieval-Augmented Generation (RAG) system, focusing on the technologies and processes used to enhance question generation by incorporating external retrieval mechanisms.

Technology stack

The following outlines the technology stack utilized for building the RAG system, covering frontend and backend frameworks, database management tools and model deployment strategies.

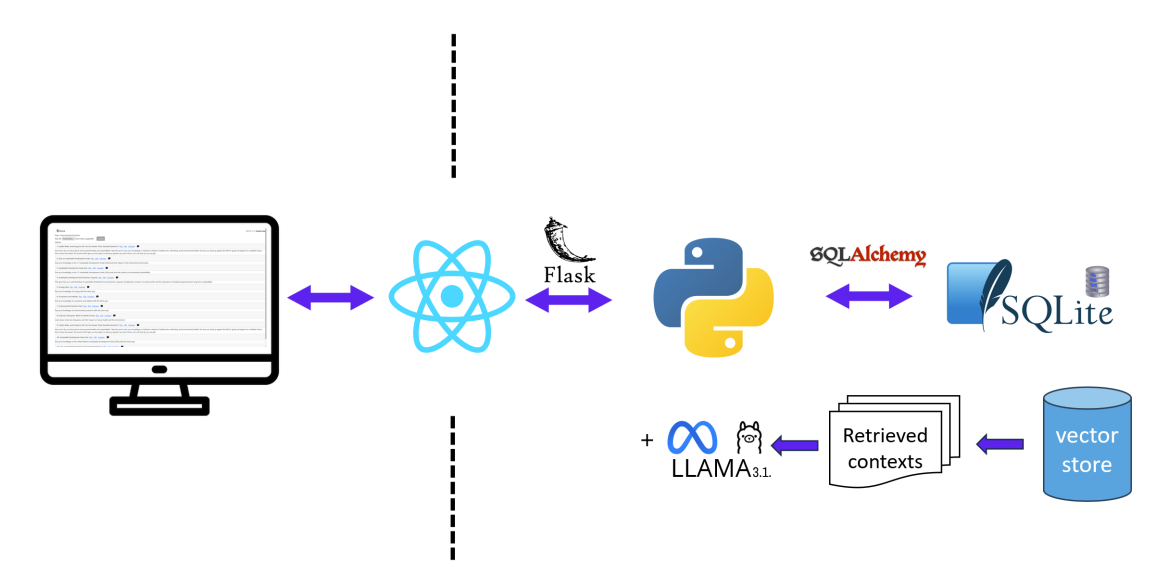


Figure 6.3: Technology Stack RAG

As illustrated in Figure 6.3, the architecture includes a React[3]-based frontend and a Python[4]-powered backend using Flask[5]. The backend integrates an SQLite[6] database managed through SQLAlchemy[7] for efficient database interactions. For question generation, LLaMA 3.1 is utilized, coupled with a FAISS[8] vector store to enhance retrieval capabilities and optimize performance.

Workflow

The workflow outlined in this section describes the sequential steps involved in document ingestion, retrieval and automated question generation, providing a detailed view of the system's operational process.

[3] https://react.dev/
[4] https://www.python.org/
[5] https://flask.palletsprojects.com/en/stable/
[6] https://www.sqlite.org/
[7] https://www.sqlalchemy.org/
[8] https://github.com/facebookresearch/faiss

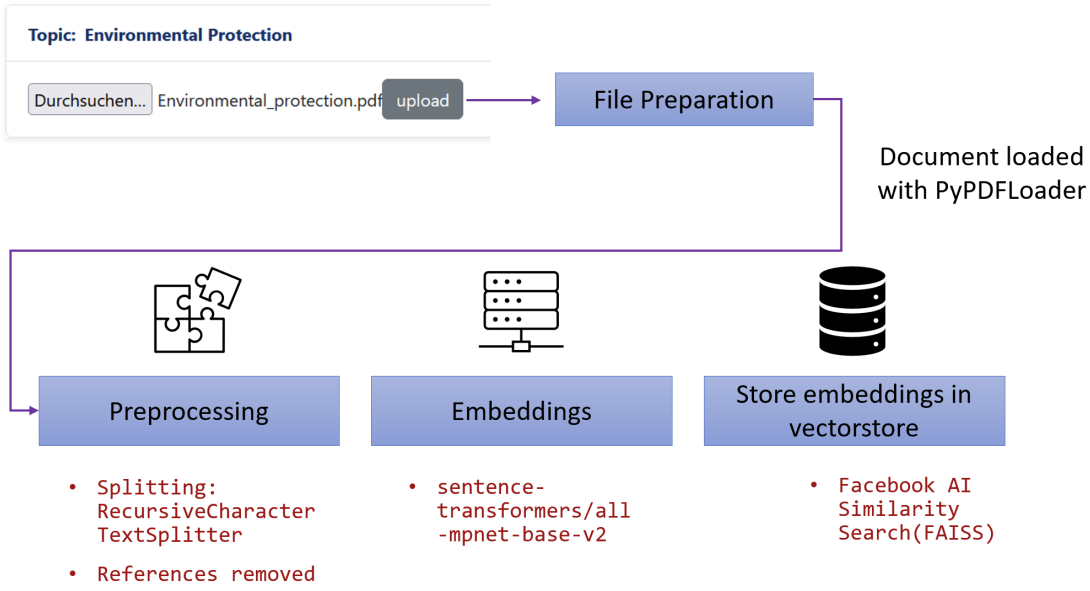


Figure 6.4: Workflow RAG

The RAG pipeline, depicted in Figure 6.4, is implemented using LangChain[9]. LangChain facilitates document ingestion, preprocessing, embedding and orchestration of retrieval queries.

File preparation and index creation As shown in figure 6.4 the PDF document is loaded with PyPDFLoader. In the **preprocessing step** the document is split using RecursiveCharacterTextSplitter. It sequentially attempts to divide the text using these characters until the resulting segments are sufficiently small. By default, it prioritizes splitting at paragraph breaks ("\n\n"), followed by line breaks ("\n"), spaces ("␣") and finally individual characters. This approach helps preserve the semantic structure of the text, keeping paragraphs, sentences and words intact whenever possible to maintain coherence (LangChain, 2024). Then the reference list is removed since including it leads to useless questions.

Then sentence-transformers/all-mpnet-base-v2 [10] is used to create embeddings from the split text.

The embeddings and splits are then used to create a FAISS vectorstore.

[9] https://python.langchain.com/docs/introduction/
[10] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Question generation For the task of question generation LangChain's ConversationalRetrievalChain with the vectorstore as a retriever is used.

LlamaGrammar is used to restrict the output to JSON.

Prompting

The process of prompt generation involved a step-by-step refinement approach, where the initial prompt was incrementally enhanced through desktop research and iterative testing. This iterative process ensured that the final prompt was well-tuned to generate contextually relevant and high-quality questions. The refined version of the prompt, which serves as the foundation for the question generation process, is presented in Figure 6.5.

```
<|start_header_id|>system<|end_header_id|>

You are a tutoring system which uses provided documents for automatic question generation.
You generate quiz items from the provided documents in different question modi.
The possible modi include Multiple Choice and True_False. Additionally you evaluate the difficulty of every
question ranging from 1 = not difficult to 5 = very difficult.
Please preserve the formatting and overall template that I provide.
This is the template for the Multiple Choice question modus: "type": "multiple_choice", "question": X,
"option1": X, "option2": X, "option3": X, "option4": X, "correct_answer": X as the number of correct option,
"difficulty": X .
This is the template for True_False question modus: "type": "true_false", "correct_answer": X as true or
false, "difficulty": X . This is the overall template: "questions": [ X ].
Respond only with valid JSON. Do not write an introduction or summary.

<|eot_id|>
<|start_header_id|>user<|end_header_id|>

Create at least 10 meaningful quiz questions from the context documents.
Please return the questions in JSON in the following format {"questions": [ X ]}. Only output JSON.

<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

Figure 6.5: Question Generation Prompt RAG

6.2.3 GRAG

This subsection describes the implementation of the Graph-based Retrieval-Augmented Generation (GRAG) system, an extension of the traditional RAG approach that integrates knowledge graphs to enhance semantic reasoning and retrieval accuracy.

Technology stack

The GRAG architecture extends the RAG pipeline by integrating knowledge graph functionality through LlamaIndex[11]. LLaMa 3.1 is employed to extract entities and relationships from PDF resources and construct corresponding knowledge graphs, which are subsequently stored in NebulaGraph[12].

Workflow

In this task LlamaIndex is used to load documents, construct the knowledge graph and automatically generate quiz questions by querying LLaMa 3.1.

File preparation and graph construction The pdfs are converted to Markdown files. Documents are ingested using the MarkdownReader component of LlamaIndex. Preliminary evaluations demonstrated that converting PDFs to Markdown significantly improved content fidelity, particularly for tabular data. This preprocessing step preserved semantic structures such as headers and tables, which were frequently misinterpreted or omitted during direct PDF parsing, thereby enhancing the overall quality of document embeddings and retrieval. Then a PropertyIndex is generated from the loaded document, the knowledge graph is constructed with the use of LLaMa 3.1 with SimpleLLMPathExtractor and a configuration of maximum 10 paths per chunk.

The SimpleLLMPathExtractor generates a foundational knowledge graph without relying on a fixed schema. While it can uncover a broad variety of relationships, it may show inconsistencies in how entities and relations are named. The resulting graph often contains the widest variety of entities and connections. It is best suited for exploratory tasks where the goal is to identify as many potential relationships as possible for RAG applications, without focusing on specific entity types (LlamaIndex Contributors, 2025).

The MarkdownNodeParser[13], which splits the document into nodes using Markdown header-based splitting logic is used as node parser in the PropertyIndex.from_documents functionality.

The prompt proposed by LlamaIndex shown in Figure 6.6 is used to extract important entities and relationships.

[11] https://docs.llamaindex.ai/en/stable/
[12] https://www.nebula-graph.io/
[13] https://docs.llamaindex.ai/en/v0.10.17/api/llamaindex.core.nodeparser.MarkdownNodeParser.html

<|start_header_id|>system<|end_header_id|>

You are a helpful, respectful and honest assistant. Always answer as helpfully as possible and follow ALL given instructions. Do not speculate or make up information. Do not reference any given instructions or context.

<|eot_id|>
<|start_header_id|>user<|end_header_id|>

You are a knowledge graph maker who extracts relevant concepts, entities and their relations from a given context.Some text is provided below. Given the text, extract up to {max_knowledge_triplets} entity-relation (knowledge) triplets in the form of (subject, predicate, object). Avoid stopwords.
The triplets should be relevant, meaningful, factual correct and logically consistent.
Return the extracted entities and relationships in the following format: (entity1, relationship, entity2)
Text: {text}
Triplets:

<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

Figure 6.6: Knowledge Graph Generation Prompt

Llamaindex provides functionality for constructing and querying a graph with the PropertyGraphIndex [14] class. The PropertyIndex is then used as a retriever for LLaMa 3 via the QueryEngine[15] interface.

Question generation For the task of automated question generation the query engine interface is used which provides functionality to query large language models. The query engine uses LLMSynonymRetriever and VectorContextRetriever (if embeddings are enabled) as sub retriever per default. The LLMSynonymRetriever processes the query by generating relevant keywords and synonyms to retrieve associated nodes along with the paths connected to those nodes. The VectorContextRetriever dentifies nodes by their vector similarity and subsequently retrieves the paths linked to those nodes (LlamaIndex Developers, 2024).

To identify the most influential nodes within the knowledge graph, the Degree Centrality algorithm is applied. This algorithm measures the importance of a node based on the number of direct and indirect connections it has within a specified number of steps. Nodes with higher degree centrality are considered more central and influential in the network structure (NebulaGraph, 2023).

The following Cypher query is executed to calculate the degree centrality:

[14] https://docs.llamaindex.ai/en/stable/moduleguides/indexing/lpgindexguide/
[15] https://docs.llamaindex.ai/en/stable/moduleguides/deploying/queryengine/

```
MATCH (n)-[r]->(m)
RETURN n, COUNT(r) AS connections
ORDER BY connections DESC
LIMIT 5
```

The top five nodes with the highest centrality scores are selected and incorporated into the question generation prompt. This guides the language model toward focusing on the most significant concepts, thereby improving the relevance and depth of the generated questions.

Prompting

The prompt used for question generation, as shown in Figure 6.7, closely mirrors the structure employed in the RAG pipeline. However, in the GRAG approach, it is further enriched by incorporating key topics identified through the Degree Centrality algorithm from the knowledge graph. By integrating these central concepts, the prompt ensures that the GRAG pipeline can leverage the relational data from the knowledge graph.

<|start_header_id|>system<|end_header_id|>

You are a tutoring system which uses provided documents for automatic question generation.
You generate quiz items from the provided documents in different question modi.
The possible modi include Multiple Choice and True_False. Additionally you evaluate the difficulty of every question ranging from 1 = not difficult to 5 = very difficult.
Please preserve the formatting and overall template that I provide.
This is the template for the Multiple Choice question modus: "type": "multiple_choice", "question": X, "option1": X, "option2": X, "option3": X, "option4": X, "correct_answer": X as the number of correct option, "difficulty": X .
This is the template for True_False question modus: "type": "true_false", "correct_answer": X as true or false, "difficulty": X . This is the overall template: "questions": [X].
Respond only with valid JSON. Do not write an introduction or summary.

<|eot_id|>
<|start_header_id|>user<|end_header_id|>

Create at least 10 meaningful quiz questions covering the whole context document with the key topics:

<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

Figure 6.7: Question Generation Prompt GRAG

6.3 Methodological Limitations

While the RAG and GRAG systems proposed in this study represent significant advancements in automated question generation, certain limitations must be acknowledged. First, the retrieval and knowledge graph construction processes are inherently dependent on the quality and completeness of the source documents. Incomplete, noisy, or biased input data could adversely affect both retrieval accuracy and the semantic relevance of generated questions (Zhao et al., 2024b).

Moreover, the construction of the knowledge graph itself, which relies on the LLaMa 3.1 model to extract entities and relationships, introduces an additional layer of variability. Errors or incomplete extractions during the extraction phase—such as missed entities, incorrect relationship mappings, or hallucinated links—can propagate through the graph and ultimately influence the relevance, coherence and factual correctness of the generated questions. Since the LLM-based extraction is probabilistic and context-sensitive, its outputs may occasionally misinterpret nuanced domain-specific information, especially in complex or technical documents (Zhu et al., 2024)

Furthermore, although the adoption of LLaMa 3.1 (8B) and optimized retrieval strategies mitigates computational burdens, real-time scalability for significantly larger datasets may still present practical challenges.

7 Evaluation

7.1 Evaluation methodology

This chapter describes the evaluation methodology used to assess the effectiveness of the proposed approach for automatic question generation (AQG).

7.1.1 Human Evaluation

A two-phase human evaluation framework was implemented to assess both the quality of individual questions and their collective performance within structured quizzes.

1. **Question-Level Evaluation:** In the first step, 36 evaluators independently assessed the quality of individual questions. Each of the 120 generated quizzes was rated by two evaluators. The goal was to determine which of the two methods produced higher-quality questions based on predefined evaluation criteria. Each evaluator was presented with the source article alongside the generated questions from both methods, shown individually in randomized order to minimize bias and ensure a fair comparison.

2. **Quiz-Level Evaluation:** After identifying the method that produced the best results, a secondary evaluation was conducted at the quiz-level. In this phase, a competent evaluator assessed a subset of the generated quizzes, focusing on their overall quality, coherence, and effectiveness as complete assessments. The subset included the five highest-rated quizzes as well as the five lowest-rated quizzes for each method respectively. This step ensured that the evaluation moved beyond assessing individual questions in isolation, instead considering how well they functioned together within a structured quiz format.

This two-phase evaluation approach provided both insights at the question-level and a holistic assessment of the quiz structure.

7.1.2 Evaluation on Question-Level

This section presents the evaluation framework for assessing the quality of questions generated by the AQG system. It details the selection of evaluators, the criteria employed to assess question quality and the procedural steps taken to ensure consistency, reliability, and unbiased results in the evaluation process.

Evaluator Selection

The human evaluation on question-level was carried out by a group of 36 evaluators with diverse academic and professional backgrounds. All evaluators were at least at the college level, holding a Matura or equivalent qualification and many had experience working in the software development field. The group included teachers, technicians, and scientists, each bringing unique perspectives to the evaluation process. Importantly, all evaluators had undergone extensive learning themselves at various stages of their academic and professional careers, making them well-suited to judge the relevance, difficulty, quality, and educational value of the generated questions. This combination of diverse backgrounds and first-hand learning experience ensured a broad and balanced approach to assessing the quality of the generated material.

Evaluation Criteria

As highlighted by Kurdi et al. (2020b), the establishment of precise evaluation criteria is fundamental for producing accurate and meaningful insights during expert review processes.

Kurdi et al. (2020b) conducted a review of 93 papers published between 2015 and early 2019 that focused on AQG for educational purposes. This study summarized the evaluation metrics employed in these papers, highlighting the most commonly used criteria.

Building upon the work of Kurdi et al. (2020b), the most frequently used evaluation criteria from these studies were reviewed, as shown in Table 7.1 and those most relevant to the present research were selected. Grammatical correctness was deliberately excluded from the evaluation, based on the assumption that it would be less critical, since modern Large Language Models (LLMs) generally demonstrate high grammatical proficiency.

Criterion	Frequency
Statistical difficulty and reviewer rating of difficulty	19
Question acceptability	17
Grammatical correctness	14
Semantic ambiguity	11
Educational usefulness (i.e., usability in a learning context)	10
Relevance to the input	8
Domain relevance	6
Fluency	6
Distractor quality or plausibility	16
Answer correctness or distractor correctness	4

Table 7.1: Evaluation criteria and their frequency of use, adapted from Kurdi et al. (2020b).

Therefore the following criteria were used, each rated by evaluators using a 5-point Likert scale:

- **Relevance:** This measures how well the question aligns with the source material or topic it is intended to assess. A relevant question should directly reflect key concepts, facts, or themes from the associated content. Irrelevant questions may focus on tangential or unrelated information. Guiding Question: Does the question test knowledge or skills directly related to the topic?

  1 = Not relevant, ... , 5 = Highly relevant

- **Difficulty:** This evaluates how challenging the question is for the intended audience (college-level learners). It considers the complexity of the concepts, the level of reasoning required and familiarity of the terminology. Guiding Question: Is the question appropriately challenging for the target audience?

  1 = Simple, ... , 5 = Difficult

- **Educational Usefulness:** This assesses whether the question effectively supports learning objectives. An educationally useful question reinforces key concepts and promotes critical thinking. Guiding Questions: Does the question help reinforce important learning outcomes? Would answering or discussing this question help deepen understanding?

  1 = Not useful, ... , 5 = Highly useful

- **Answer Quality / Plausibility:** This measures the quality of the answer in general, incorrect answer choices (distractors). Good distractors should be plausible enough to challenge learners but clearly incorrect with sufficient knowledge. Poor distractors may be too obviously wrong, irrelevant, or misleading. Guiding Question: Are the distractors believable and relevant to the topic?

  1 = Low, ... , 5 = High

Additionally a checkbox for flawed items was integrated into the evaluation interface, accompanied by the following instruction:

> Please check this box if the quiz item does not make sense, is factually incorrect, the supposed answer is incorrect or the article does not provide enough material to answer the question.

This checkbox functioned as a true/false option, allowing evaluators to easily flag items with issues. It combined the evaluation criteria of grammatical correctness, semantic ambiguity, answer, and distractor correctness and fluency, as identified in the literature review by Kurdi et al. (2020b), into a single, streamlined action.

Evaluation Procedure

For the human evaluation on the question-level, a dedicated website was hosted, which was password-protected to ensure that only authorized evaluators could participate. Evaluators were asked to complete a series of evaluations, with the option to proceed to another evaluation once they finished the current one. This approach was designed to ensure that evaluators remained focused and attentive for each task, with the ability to stop the evaluation at any time if they no longer wished to continue.

To support engagement and evaluator motivation, the platform incorporated a progress bar, providing real-time visual feedback on completion status. This feature helped evaluators track their progress and motivated them to finish each evaluation.

Additionally, a leaderboard and scoring system were implemented to encourage participation and foster a sense of friendly competition. The leaderboard, which can be seen in the appendix, displayed evaluators' scores and rankings based on their score. To respect privacy and autonomy, evaluators were given the option to opt out of the leaderboard and could choose a custom nickname, ensuring their participation could remain anonymous if desired. This system was designed to motivate evaluators to complete multiple evaluations while maintaining a positive and engaging environment.

To ensure an unbiased comparison of the two generation methods described in chapter 6, the evaluators were presented with all generated questions of each method for each article in a randomized order. This randomization helped mitigate any potential bias that could arise from the order in which the questions were presented. Each evaluator was asked to carefully read the entire article before proceeding to evaluate the generated questions.

After reading the article, the evaluators went through the questions one by one and assessed them based on the predefined evaluation criteria. This step-by-step approach was aimed at helping the evaluators focus on the quality of each individual question while keeping the context of the article in mind, ensuring a thorough and consistent evaluation of the generated questions.

7.1.3 Evaluation on Quiz-Level

This section outlines the evaluation process for assessing the overall quality of quizzes generated by the AQG system. It details the selection of an expert evaluator, the criteria used for assessment and the procedure followed to ensure a comprehensive evaluation of the quizzes' content, structure and pedagogical effectiveness.

Evaluator Selection

The evaluation at the quiz-level was conducted by a psychologist with extensive experience as a tutor. This professional evaluated the quizzes on a qualitative basis, providing expert insights into the content, structure, and overall effectiveness of the quizzes in promoting learning. The evaluator assessed factors such as the clarity of questions, the cognitive load imposed on students, and the alignment of quiz items with learning objectives. This qualitative evaluation aims to ensure that the quizzes are not only pedagogically sound but also support meaningful engagement and knowledge retention.

Evaluation Criteria

To assess the holistic quality of quizzes, a set of evaluation criteria focused on structural coherence, pedagogical effectiveness and informational content was designed. Each criterion was rated by evaluators using a 5-point Likert scale, where:

1 = Low 2 = Fairly Low, 3 = Fair, 4 = Fairly High, 5 = High

- **Coverage:** Measures the extent to which the quiz reflects the breadth and depth of the source article. A score of 5 indicates that the quiz addresses all

major and minor concepts comprehensively, while a score of 1 indicates that the quiz covers only a narrow or superficial portion of the content.

- **Overall Educational Value:** A subjective judgment of the quiz's utility in a learning context. A high score reflects strong alignment with pedagogical goals, critical thinking stimulation, and potential use in instructional settings.
- **Redundancy:** Assesses the degree of repetition among questions. A score of 1 indicates no noticeable redundancy and diverse question content, whereas a score of 5 suggests that multiple questions are repetitive or ask the same thing in slightly different ways.
- **Progressiveness:** Evaluates whether the quiz follows a logical sequence, ideally starting with simpler questions, and advancing to more complex or inferential ones. A top score indicates a well-structured progression that aids cognitive flow.
- **Overall Quality:** An overarching rating that considers clarity, balance, engagement and technical soundness. A score of 5 reflects a highly polished quiz suitable for educational deployment, while a score of 1 denotes a quiz with major structural or content issues.

Evaluation Procedure

The evaluation procedure on quiz-level was designed to ensure a comprehensive and systematic assessment of the quizzes. Initially, the evaluator was asked to read the assigned article in its entirety to gain a thorough understanding of the content. Once the article was reviewed, the evaluator assessed it based on several key aspects, including **Key Concepts, Content, Educational Usefulness and General Observations**. These preliminary evaluations provided context for how well the article aligned with the intended learning outcomes.

After reviewing the article, the focus shifted to evaluating the quiz itself. The evaluator assessed the quiz according to five primary criteria: **Coverage, Overall Educational Value, Redundancy, Progressiveness and Overall Quality**. These criteria were selected to assess both the structural and pedagogical effectiveness of the quiz in an educational context.

By incorporating both the article evaluation and a detailed review of the quiz, the procedure aimed to provide a thorough assessment of the quality and educational effectiveness of the generated quizzes.

7.1.4 Potential Biases and Limitations

In any evaluation methodology, there are inherent biases and limitations that may affect the validity and generalizability of the results. The evaluation process employed in this study, while robust, is not without such challenges. Below, some of the key biases and limitations that may be present in the evaluation methodology are highlighted:

- **Evaluator Bias:** The evaluators in this study, while diverse in terms of their academic and professional backgrounds, still bring their own subjective judgments and interpretations to the evaluation process. Even with predefined criteria and structured scales, the evaluation of question quality remains partially influenced by individual preferences, experiences and knowledge (Hosking et al., 2024). The absence of complete objectivity could potentially lead to discrepancies in the ratings, particularly on more nuanced criteria such as educational usefulness or difficulty.
- **Question Presentation Bias:** In the Question-Level evaluation, the questions generated by the AQG system were presented to evaluators in a randomized order, which helps mitigate potential order effects. However, the order in which evaluators view questions might still influence their evaluation, especially if they perceive certain question types as easier or more engaging based on their placement in the list. This effect may lead to inconsistencies in the evaluation of questions, especially for those evaluators who may be influenced by fatigue or the difficulty of earlier questions.
- **Scope of Evaluation:** The evaluation focused exclusively on a subset of criteria deemed most relevant for assessing the generated questions, such as relevance, difficulty, and educational usefulness. While these criteria are fundamental to evaluating AQG performance, they may not fully cover all dimensions of question quality. Other important factors, such as the alignment with cognitive learning outcomes or the broader impact of questions on long-term retention, were not considered in the current evaluation methodology.
- **Human Evaluation Constraints:** Despite the use of a structured 5-point Likert scale and explicit evaluation guidelines, human evaluators may still struggle with consistent interpretation of certain criteria, particularly those that are subjective, like educational usefulness or question clarity. Variability in the evaluators' assessment standards or their understanding of the content could introduce inconsistencies in ratings.

- **Limitations in Evaluator Engagement:** To maintain engagement, evaluators were given the option to stop at any time and incentives like leaderboards were used. However, this could introduce biases based on evaluator motivation. Some evaluators may have rushed through the questions for the sake of completing the task quickly, which could compromise the quality of their evaluations. Additionally, those who opted out of the leaderboard may not have been as motivated to evaluate a large number of questions, potentially affecting the distribution of evaluations.

7.2 Dataset

The dataset used in this study consists of 60 articles from the MIT Climate Portal's "Explainers" section [1] with a total word count of approximately 30,000 words. This size is substantial enough to ensure a variety of questions can be generated for analysis while remaining manageable for detailed review. The articles are categorized into multiple themes, including climate science, climate policy and proposed solutions, offering a balanced view across diverse aspects of the climate change field. This collection offers accessible, scientifically grounded overviews of various topics related to climate change science, solutions and policy, authored by experts in the field. The decision to use this dataset was driven by several factors:

- Comprehensive Coverage: The collection provides a wide range of content, encompassing key issues in climate science and related fields.
- Accessibility: The articles are freely available and well-structured, making them suitable for educational applications.
- Manageable Scope: The number of articles in this collection is sufficiently large to support a meaningful qualitative analysis, while still being manageable in terms of the depth and scale of the evaluation.

This dataset is thought to be valuable for generating educational questions, as it contains fact-based, well-researched and easily understandable content that seem ideal for assessing the effectiveness of AQG models in an educational context.

To create the dataset, a web scraper was developed to extract the article text from the MIT Climate Portal. The extracted content was then saved as PDFs to demonstrate that the system can process and generate questions from PDF documents.

[1] https://climate.mit.edu/explainers

7.3 Results

This chapter presents the results of the evaluation conducted on the two quiz generation methods: Retrieval-Augmented Generation (RAG) and Graph-based Retrieval-Augmented Generation (GRAG). The objective of this evaluation is to quantitatively assess and compare the quality, difficulty and relevance of questions generated by both methods as well as the prevalence of flaws within these quiz items.

7.3.1 Question-Level Evaluation

This section reports the evaluation of 1248 individual multiple-choice and true/-false questions generated using the RAG and GRAG methods. The analysis systematically examines three key dimensions: overall question quality, question difficulty, and the incidence of flawed questions.

Quality

The aggregated results for the quality of the generated questions across the two methods, RAG and GRAG, are presented in Table 7.2. RAG outperformed GRAG in all metrics, achieving notably higher scores for *educational usefulness* (3.53 vs 2.74), *relevance* (3.92 vs 3.21), and *overall quality* (3.67 vs 2.99) which is the mean of the metrics *educational usefulness*, *relevance*, and *distractor quality/answer plausibility* combined. Additionally, the average *flawed* percentage per quiz was lower for RAG (15%) vs GRAG (22%).

Method	Educational Usefulness	Relevance	Distractor Quality	Overall Quality	Flawed %
RAG	3.53	3.92	3.57	3.67	15.01
GRAG	2.74	3.21	3.03	2.99	22.22

Table 7.2: Mean Evaluation Scores for Generated Questions by Method

Distribution of Quality Scores: RAG vs. GRAG

Figure 7.1 presents the kernel density estimation plots comparing the distribution of quality scores across four evaluation metrics — *educational usefulness*, *relevance*, *distractor quality*, and *overall quality*—for both the RAG and GRAG methods.

Across all four metrics, GRAG exhibits a noticeable leftward shift in its score distributions compared to RAG, suggesting that the quality of outputs generated by GRAG was generally perceived to be lower. The distributions for RAG are more

concentrated around higher scores (typically between 3.5 and 5), indicating a consistent performance in generating high-quality content. Specifically, *Educational usefulness* and *relevance* show clear separation between the two methods, with RAG achieving higher peak scores and tighter density curves, suggesting more reliable performance. *Distractor quality/answer plausibility* displays some overlap, but RAG still maintains a peak further to the right, indicating better distractor construction. *Overall quality* follows a similar pattern, with GRAG skewed toward lower scores and RAG maintaining a strong peak around scores of 4.

These trends imply that RAG not only outperformed GRAG on average but also demonstrated greater consistency in generating educationally effective and relevant questions.

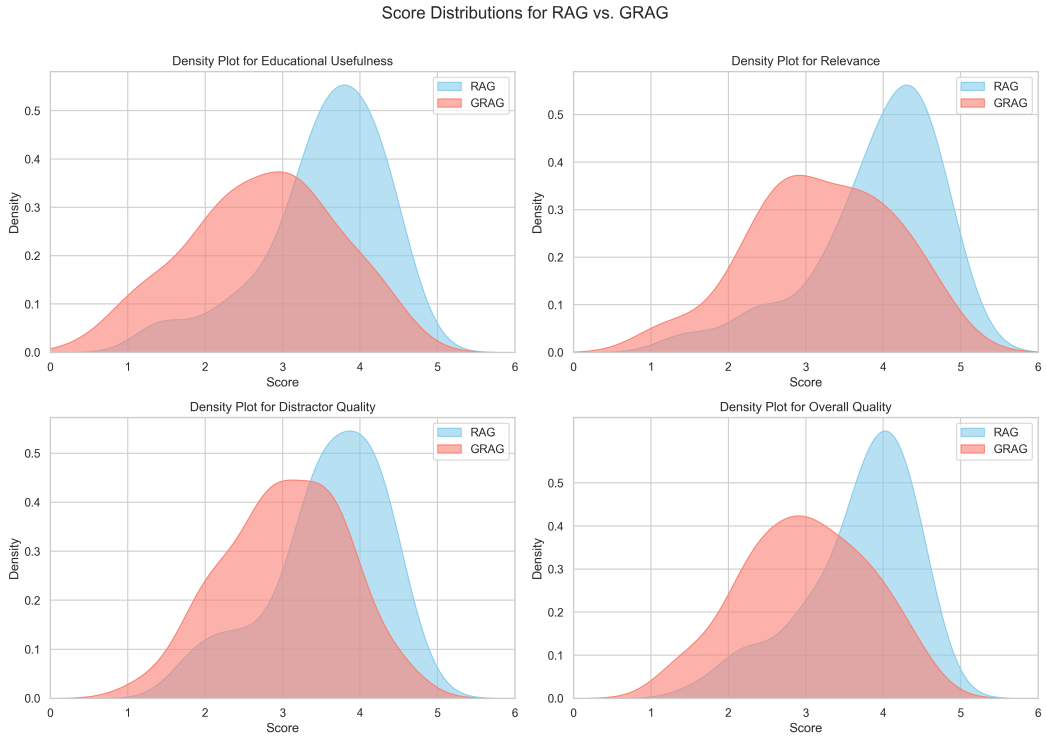


Figure 7.1: Score Distributions for RAG vs. GRAG

Tables 7.3 and 7.4 summarize the question-level evaluation metrics for the RAG and GRAG methods, respectively. Each table reports descriptive statistics (mean, standard deviation, minimum, quartiles and maximum) across five key metrics: *educational usefulness*, *relevance*, *distractor quality*, *overall quality* and *flawed count* as well as *flawed percentage*.

For the RAG method (Table 7.3), the average scores for *educational usefulness* (3.53), *relevance* (3.92), *distractor quality* (3.57), and *overall quality* (3.67) indicate moderately high question quality across all dimensions. The relatively low mean *flawed count* (2.78) corresponding to 15.01%, suggests that RAG-generated questions exhibited fewer detectable flaws on average.

In contrast, the GRAG method (Table 7.4) shows lower average scores across all quality dimensions, with *educational usefulness* (2.74), *relevance* (3.21), *distractor quality* (3.03), and *overall quality* (2.99). The mean *flawed percentage* (22.22%) is notably higher compared to RAG (15.01%), indicating that GRAG-generated questions contained more issues per question on average.

These descriptive statistics suggest that the RAG method generally outperforms the GRAG method in terms of question quality and reliability.

Metric	Mean	Std	Min	25%	50%	75%	Max
Educational Usefulness	3.53	0.78	1.36	3.25	3.66	4.08	4.71
Relevance	3.92	0.82	1.44	3.67	4.12	4.47	5.0
Distractor Quality	3.57	0.74	1.73	3.29	3.69	4.13	4.78
Overall Quality	3.67	0.72	1.60	3.25	3.84	4.21	4.68
Flawed Count	2.78	3.39	0.0	0.0	1.5	4.25	14.0
Flawed %	15.01	17.09	0	0	7.67	25	58.33

Table 7.3: Summary of Evaluations for RAG Method

Metric	Mean	Std	Min	25%	50%	75%	Max
Educational Usefulness	2.74	0.94	0.6	2.09	2.83	3.36	4.5
Relevance	3.21	0.92	1.1	2.6	3.18	3.91	5.0
Distractor Quality	3.03	0.77	1.15	2.57	3.05	3.65	4.65
Overall Quality	2.99	0.81	1.26	2.36	2.98	3.57	4.52
Flawed Count	4.68	3.75	0.0	2.0	3.5	8.0	13.0
Flawed %	22.22	18.27	0	9.77	15	35	65

Table 7.4: Summary of Evaluations for GRAG Method

To emphasize the questions with the highest quality, Table 7.5 displays the five quizzes with the highest individual question evaluation scores for each method. Most of the highest-rated questions were generated using RAG, with some GRAG questions achieving similarly strong scores, particularly regarding *relevance*.

Table 7.5 shows the five highest-rated quizzes generated by the RAG method, with scores ranging from 4.31 to 4.71 for *educational usefulness* and from 4.50 to

4.96 for *relevance*. *Distractor quality/answer plausibility* scores range from 4.17 to 4.61, while *overall quality* ranges from 4.48 to 4.68. Notably, the percentage of *flawed* questions for RAG-generated items is mostly 0 or low, with one exception where it reaches 11.11%.

Table 7.5 also includes the five highest-rated quizzes generated by the GRAG method. Here, the *educational usefulness* scores range from 3.86 to 4.30, while *relevance* scores vary between 4.50 and 5. *distractor quality/answer plausibility* ranges from 3.80 to 4.65 and *overall quality* spans from 4.18 to 4.52. *Flawed* percentages for GRAG-generated questions are generally low, with two quizzes showing 10% *flawed* and the rest with 0%.

ID	Educational Usefulness	Relevance	Distractor Quality	Overall Quality	Flawed %	Method
53	4.71	4.96	4.38	4.68	0	RAG
26	4.44	4.94	4.61	4.67	0	RAG
60	4.44	4.83	4.17	4.48	11.11	RAG
45	4.31	4.69	4.44	4.48	6.25	RAG
56	4.50	4.50	4.40	4.47	0	RAG
126	3.90	5	4.65	4.52	0	GRAG
120	4.30	4.65	4.40	4.45	10	GRAG
115	3.95	4.65	4.40	4.33	0	GRAG
111	3.86	4.59	4.18	4.21	0	GRAG
78	4.25	4.50	3.80	4.18	10	GRAG

Table 7.5: Top 5 Question-Level Evaluation Scores by Method

Table 7.6 presents the five quizzes with the lowest-rated questions, drawn from both the RAG and GRAG methods respectively. The table highlights questions that received the poorest evaluations across the metrics: *educational usefulness*, *relevance*, *distractor quality/answer plausibility*, *overall quality*, and *flawed count*, along with the method used to generate each question.

For the RAG method, the lowest-rated questions show *overall quality* scores ranging from 1.60 to 2.25. Although some items had relatively moderate *distractor quality/answer plausibility* (e.g., ID 16 with 3.72), other dimensions such as *educational usefulness* and *relevance* were consistently low. The *flawed counts* for these RAG questions varied widely, ranging from 0 to 12, which corresponds to 0% to 54.55% of the generated questions being flawed.

For the GRAG method, the five lowest-rated quizzes exhibited even lower *overall quality* scores, ranging from 1.26 to 1.70, with consistently poor *educational usefulness* and *relevance*. The *flawed counts* for these GRAG quizzes were notably

high (8 to 12) which corresponds to 40% - 60% of the generated questions being flawed, further emphasizing the lower quality of these items.

This table underscores that while both methods produce some low-quality questions, the lowest-rated GRAG questions tended to score lower across most metrics and had higher *flaw counts* compared to the lowest-rated RAG questions.

ID	Educational Usefulness	Relevance	Distractor Quality	Overall Quality	Flawed %	Method
29	2.33	2.46	1.96	2.25	4.17	RAG
16	1.44	1.44	3.72	2.20	0	RAG
55	2.12	2.25	2.06	2.15	50	RAG
5	2.09	2.27	1.73	2.03	54.55	RAG
38	1.50	1.50	1.81	1.60	50	RAG
98	1.35	1.65	2.10	1.70	60	GRAG
83	1.35	1.30	2.15	1.60	55.00	GRAG
113	1.10	2.05	1.15	1.43	40	GRAG
106	0.60	1.10	2.50	1.40	50	GRAG
117	1.14	1.14	1.50	1.26	54.55	GRAG

Table 7.6: 5 Quizzes per Method (RAG and GRAG) with lowest ratings

Difficulty

As shown in Table 7.7, quizzes generated with the RAG method were perceived as more *difficult* on average (2.76) compared to those generated by GRAG (2.17). This indicates that RAG-produced questions were generally considered more challenging by the evaluators.

However, the *Mean of Standard Deviations* is slightly higher for GRAG (1.22) than for RAG (1.12), suggesting that the *difficulty* ratings for GRAG quizzes were slightly more variable. This could imply that while GRAG quizzes were easier on average, they exhibited more diversity in *difficulty*.

Method	Mean of Means	Mean of SD
RAG	2.76	1.12
GRAG	2.17	1.22

Table 7.7: Difficulty Summary Results for RAG and GRAG Quizzes

To better understand the distribution of perceived *difficulty* ratings, a histogram was created for each method (see Figure 7.2, 7.3). The x-axis represents difficulty

levels on a 5-point scale (1 = simple, 5 = difficult), while the y-axis indicates the number of questions in each category. Percentages on top of the bars reflect the proportion of questions at each *difficulty* level.

The distribution of question difficulties produced by RAG is relatively balanced across the spectrum. The most common *difficulty* level is 3 (33.7%), suggesting a central tendency toward moderate *difficulty*. Level 2 follows with 27.9% and level 4 with 17.9%, indicating a good representation of both easier and more challenging questions. *Difficulty* levels 1 and 5 account for 12.2% and 8.2%, respectively. Overall, 73.8% of RAG-generated questions fall within the easy to moderate range (levels 1–3), while 26.1% are categorized as difficult (levels 4–5).

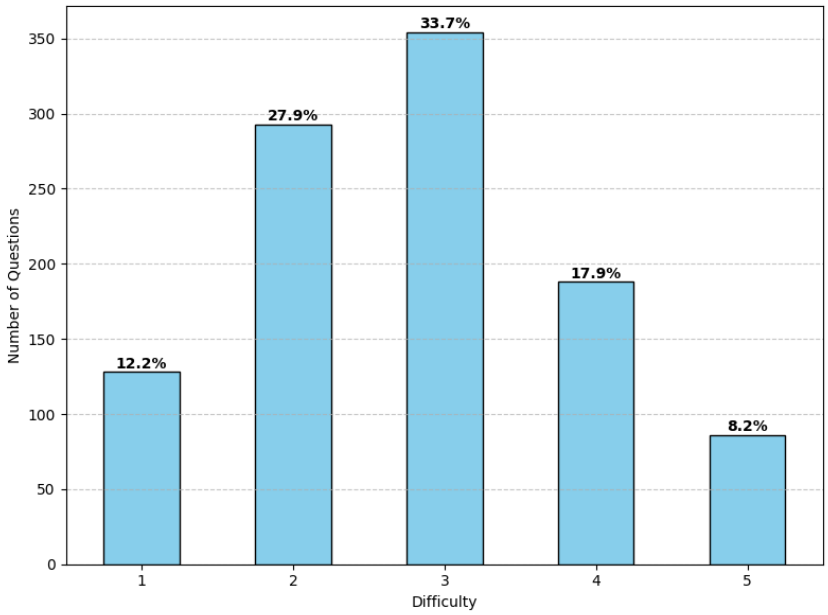


Figure 7.2: Distribution of Difficulty RAG

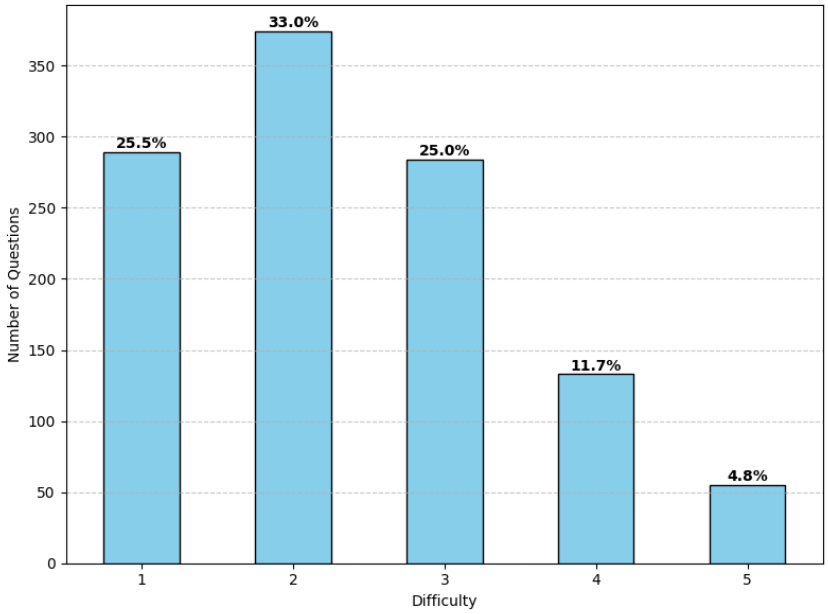


Figure 7.3: Distribution of Difficulty GRAG

In contrast, the distribution of *difficulty*, illustrated in Figure 7.3, shows that GRAG-generated questions are strongly skewed toward the lower end of the difficulty spectrum. *Difficulty* level 2 accounts for the highest proportion of questions (33.0%), followed closely by levels 1 (25.5%) and 3 (25.0%). Together, the first three levels represent 83.5% of all questions, suggesting that GRAG predominantly produces questions of easy to moderate difficulty. In contrast, higher-difficulty questions are underrepresented, with only 11.7% and 4.8% of questions at levels 4 and 5, respectively.

Flawed Questions

Flagged flawed questions were systematically reviewed and recurring issues were used to define a set of flaw categories. These categories were then applied in a second review to ensure a thorough and accurate assessment of the quality of the generated questions.

As shown in Figure 7.4, 18.62% of all evaluated questions were identified as flawed. This corresponds to 448 out of 2406 individual question evaluations in which the 'is flawed' status was marked as true. These flagged evaluations span 379 unique questions (out of 1248) that have been marked as flawed by at least one rater.

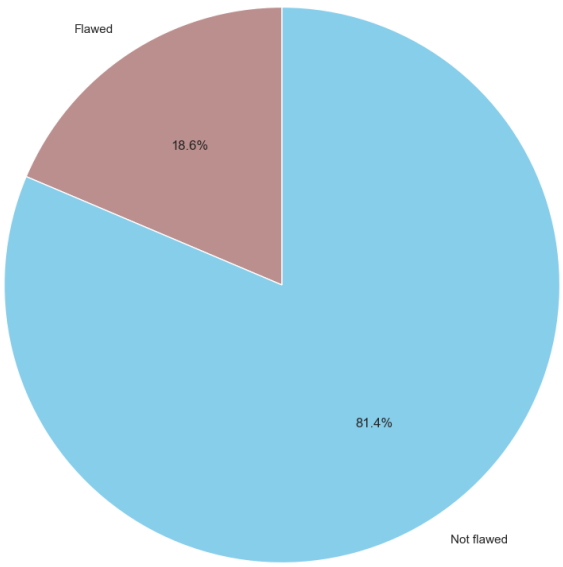


Figure 7.4: Flawed Questions

The most frequent categories of flaws identified were:

Semantics: Issues related to ambiguous or unclear phrasing.
Multiple Answers Correct: Situations where more than one answer option was plausibly correct.
Wrong Correct Answer: Cases where the marked correct answer was the wrong one or factually incorrect.
Insufficient Material: Articles did not provide enough information to accurately answer the question.
Reference-Based Questions: Questions overly reliant on the references section of the source material.
Factually Incorrect: Errors regarding factual correctness independent of the original article content.

Additionally, quiz items that did not exhibit any of the predefined flaw categories were classified under the label **No Issue Found**, indicating that no identifiable issues were detected based on the evaluation criteria.

The distribution of flawed question types is visualized in Figure 7.5. It can be seen that semantic errors and multiple-correct-answer flaws dominate the flawed cases.

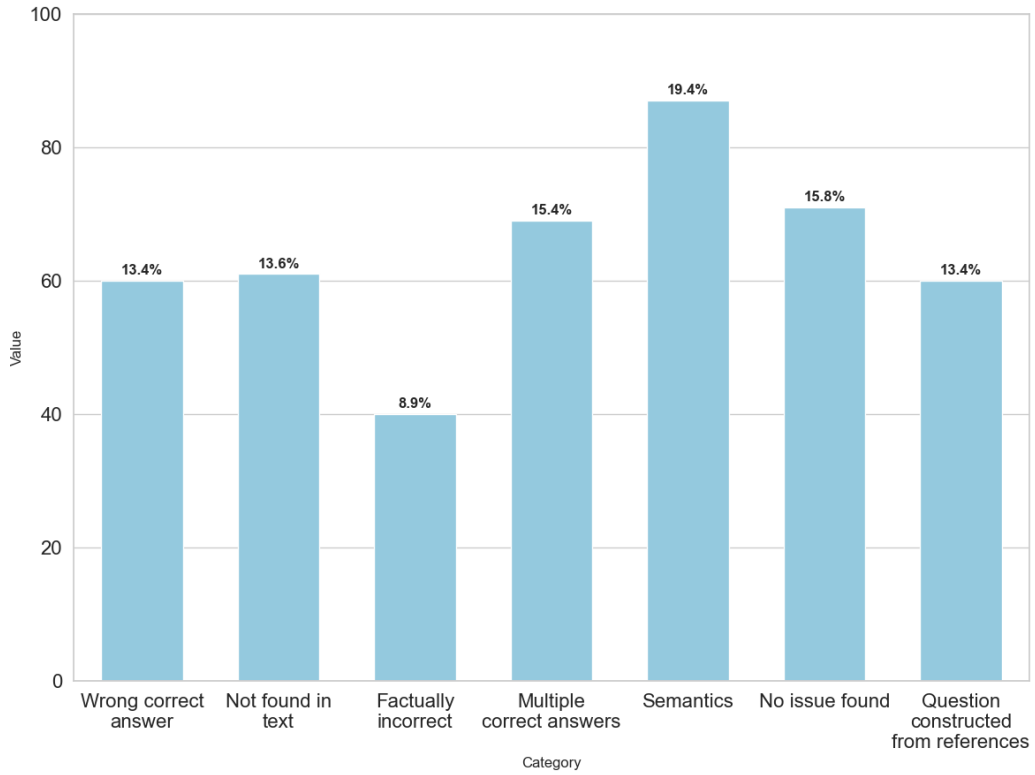


Figure 7.5: Distribution of flawed questions across identified categories

7.3.2 Quiz-Level Evaluation

While the previous section focused on the evaluation of individual questions, this subsection shifts the perspective to the quiz-level. Here, we assess the *overall quality*, *consistency*, *progressiveness*, and *redundancy* of entire quizzes generated by both methods.

The analysis of the quizzes was carried out in two stages: one for the highest-rated quizzes and one for the lowest-rated quizzes, enabling comparison of key quality characteristics across methods. In this section, the results of the evaluations from both the 10 highest-rated and 10 lowest-rated quizzes are discussed in terms of their respective metrics, with the intention of identifying trends and areas of improvement.

Quiz-level evaluation of the five highest-rated quizzes

Table 7.8 presents the quiz-level evaluation metrics for each of the five highest-rated quizzes from the question-level evaluation from the RAG and GRAG meth-

ods respectively. The metrics assessed include *overall quality*, *coverage*, *usefulness*, *redundancy*, and *progressiveness*, where higher values indicate better performance for all metrics except *redundancy*, where lower values are preferred. All ratings are on a scale from 1 (low) to 5 (high).

Quiz ID	Method	Overall Quality	Coverage	Usefulness	Redundancy	Progressiveness
53	RAG	1	4	2	5	1
26	RAG	3	5	4	3	2
60	RAG	1	2	2	4	1
45	RAG	3	3	3	2	3
56	RAG	1	3	2	5	1
Mean		**1.8**	**3.4**	**2.6**	**3.8**	**1.6**
126	GRAG	2	4	3	4	1
120	GRAG	2	3	2	3	2
115	GRAG	2	2	2	4	1
111	GRAG	3	3	3	2	1
78	GRAG	2	3	2	3	1
Mean		**2.2**	**3.0**	**2.4**	**3.2**	**1.2**
Overall Mean		**2.0**	**3.2**	**2.5**	**3.5**	**1.4**

Table 7.8: Quiz Evaluation Metrics by Method (RAG vs GRAG)

When compared to question-level results, the quiz-level evaluation provides a more holistic view of the generated quizzes. For instance, while individual questions may be assessed on metrics like relevance and usefulness, the quiz-level evaluation reveals broader issues such as the logical flow and thematic coherence of questions within the quiz and whether any redundancy impacts the overall quiz experience.

GRAG slightly outperformed RAG in *overall quality* (2.2 vs. 1.8) and *redundancy* (3.2 vs. 3.8), while RAG showed marginally higher scores in *coverage* (3.4 vs. 3.0), *progressiveness* (1.2 vs. 1.6) and *usefulness* (2.6 vs. 2.4).

Quiz-level evaluation of the five lowest-rated quizzes

Table 7.9 presents the quiz-level evaluation metrics for each of the five lowest-rated quizzes from the question-level evaluation from the RAG and GRAG methods respectively. These quizzes were selected based on their bottom ratings from the question-level evaluation stage. The metrics provide an understanding of the areas

where the quizzes fell short and how the RAG and GRAG methods compare in these low-performance cases.

Quiz ID	Method	Overall Quality	Coverage	Usefulness	Redundancy	Progressiveness
29	RAG	2	4	2	3	1
16	RAG	1	1	1	4	1
55	RAG	1	1	1	3	1
5	RAG	2	3	2	2	3
38	RAG	1	1	1	4	1
Mean		**1.4**	**2**	**1.4**	**3.2**	**1.4**
98	GRAG	1	2	2	4	2
83	GRAG	1	2	1	4	1
113	GRAG	1	2	1	2	1
106	GRAG	1	1	1	3	1
117	GRAG	1	1	1	3	1
Mean		**1**	**1.8**	**1.3**	**3.2**	**1.3**
Overall Mean		**1.2**	**1.8**	**1.3**	**3.2**	**1.3**

Table 7.9: Quiz Evaluation Metrics by Method (RAG vs GRAG)

Across both methods, these lowest-rated quizzes consistently showed low performance in overall quality (1.2), coverage (1.8), usefulness (1.3), and progressive structure (1.3). Notably, redundancy remained relatively high (3.2) in both conditions, indicating a recurring issue of repetition among quiz items.

7.3.3 Graph Analysis

To investigate the performance gap between the Graph-Retrieval-Augmented Generation (GRAG) and standard Retrieval-Augmented Generation (RAG) approaches in quiz generation, an analysis of the underlying knowledge graphs within each climate-related topic was conducted.

This analysis examined key structural properties—such as the number of vertices and edges, average degree, density, sparsity, and presence of isolated nodes—across the automatically generated knowledge graphs. These metrics provide insight into the cohesion and expressiveness of the semantic graphs used by GRAG. Topics with sparse, fragmented graphs or high numbers of isolated nodes may offer less context for generation, potentially explaining instances where GRAG underperforms relative to RAG.

Feature	Mean	Std	Median	Min.	Max.	Skew	Kurtosis
Vertices	24.6000	8.4517	26	11	45	0.1684	-0.5402
Edges	16.2000	5.4828	18	5	29	0.0399	-0.4873
Average Degree	1.3314	0.2002	1.3218	0.9091	1.8182	0.3199	-0.1575
Density	0.0331	0.0161	0.0265	0.0142	0.0758	1.1942	0.5637
Sparsity	0.9669	0.0161	0.9735	0.9242	0.9858	-1.1942	0.5637
Isolated Nodes	14.8000	5.0482	15.5000	5	25	-0.0790	-1.0891

Table 7.10: Summary statistics for graph metrics.

The graph analysis summarizes the structural properties of 60 topic-specific graphs generated from the MIT explainer dataset using LLaMa 3. Table 7.10 presents the aggregated statistics across all graphs, including central tendencies, dispersion, and distribution shape metrics such as skewness and kurtosis.

The number of vertices varies widely (M = 24.6, SD = 8.45), edge counts follow a similar trend (M = 16.2) and the average degree is low (M = 1.33), indicating moderate conceptual connectivity.

The graphs are generally sparse with a mean density of 0.0331 and a mean sparsity of 0.9669. Skewness and kurtosis values show a right-skewed distribution for density and a corresponding left-skew for sparsity, suggesting the presence of a few relatively denser graphs amid a majority of sparse ones.

A notable feature is the high number of isolated nodes across many graphs with a mean of 14.8, ranging from 5 to 25.

Table 1 provides detailed statistics for each graph, allowing for comparative analysis across different climate-related themes. For instance, the graph for *biochar* displays one of the highest densities (0.0758) and lowest sparsity values, reflecting a dense and well-connected conceptual *subnetwork*. In contrast, graphs such as *freshwaterandclimatechange* and *heatingandcooling* show very low densities (0.0143 and 0.0142 respectively), suggesting much more diffuse concept distributions.

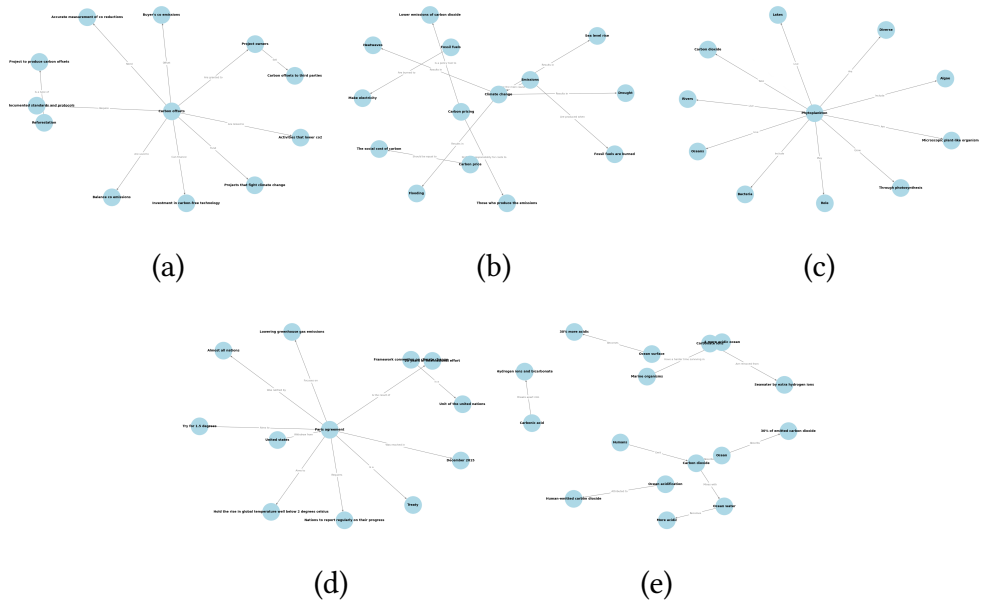
The *foodsystemsandagriculture* topic exhibits the highest average degree (1.8182), indicating a high degree of conceptual interconnection, while graphs like *urbanheatislands* and *miningandmetals* are characterized by lower average degrees, suggesting less integrated semantic structures.

Analysis of KG of five highest and lowest rated Quizzes

To investigate whether characteristics of the generated knowledge graphs help explain the differences between the highest- and lowest-rated quizzes, this section presents a comparative analysis of their structural properties.

Figures 7.6 and 7.7 illustrate the knowledge graphs generated for the five highest-rated and five lowest-rated quizzes, respectively. Tables 7.11 and 7.12 summarize key graph statistics for each set.

The knowledge graphs of the highest-rated quizzes (Table 7.11) are relatively small and sparse, with an average of 14 vertices and 10 edges. They exhibit a low average degree (between 1.18 and 1.67) and a high proportion of isolated nodes (8–11), indicating that the concepts are more discretely structured, with fewer but potentially more meaningful connections between them.



(a) (b) (c)

(d) (e)

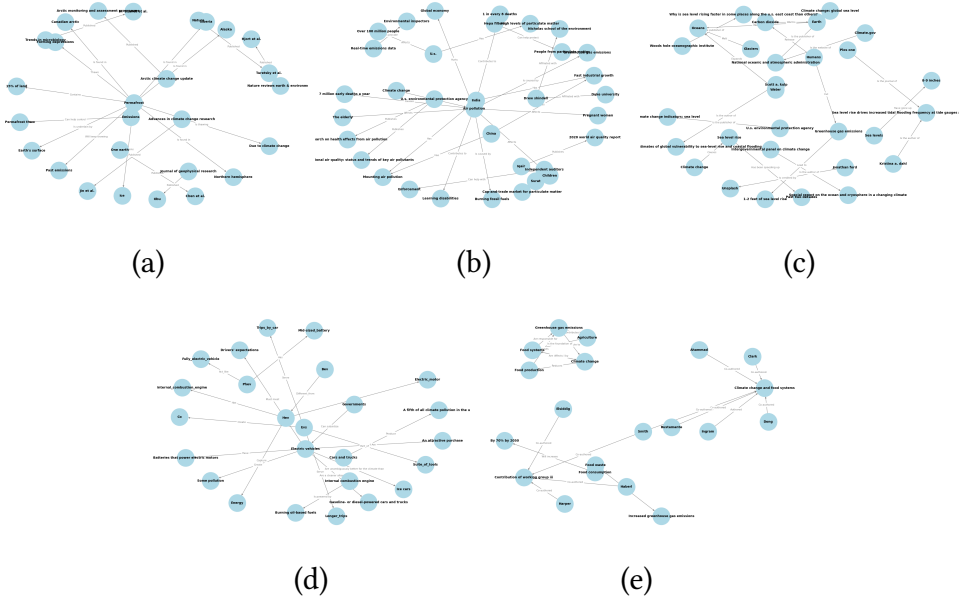
Figure 7.6: Generated Knowledge Graphs of best 5 evaluated quizzes

Graph Space	Vertices	Edges	Average Degree	Density	Sparsity	Isolated Nodes
carbonoffsets (a)	13	10	1.5385	0.0641	0.9359	10
carbonpricing (b)	15	10	1.3333	0.0476	0.9524	10
oceanacidification (c)	17	10	1.1765	0.0368	0.9632	8
parisagreement (d)	13	10	1.5385	0.0641	0.9359	10
phytoplankton (e)	12	10	1.6667	0.0758	0.9242	11

Table 7.11: Graph statistics for best 5 evaluated quizzes

In contrast, the graphs associated with the lowest-rated quizzes (Table 7.12) are noticeably larger, with a higher number of vertices (22–37) and edges (18–25).

Despite this increased size, they remain sparse, with density values even lower than those of the highest-rated quizzes. They also show high numbers of isolated nodes (6–25) and relatively low average degrees (around 1.2–1.8). This suggests that, although these graphs encompass more concepts, the relationships between those concepts are still limited, potentially leading to fragmented or overly complex question sets.



(a) (b) (c)

(d) (e)

Figure 7.7: Generated Knowledge Graphs of 5 lowest-rated quizzes

Graph Space	Vertices	Edges	Average Degree	Density	Sparsity	Isolated Nodes
airpollution (a)	37	25	1.3514	0.0188	0.9812	25
electricvehicles (b)	27	19	1.4074	0.0271	0.9729	19
foodsystemsandagriculture (c)	22	20	1.8182	0.0433	0.9567	6
permafrost (d)	29	18	1.2414	0.0222	0.9778	20
sealevelrise (e)	31	19	1.2258	0.0204	0.9796	13

Table 7.12: Graph statistics for 5 lowest-rated quizzes

Comparison of Key Concept Identification

Table 7.13 compares the key concepts identified by the evaluator with those identified by the Degree Centrality algorithm. For each topic, the evaluator has identified a set of key concepts that represent the most relevant ideas associated with the topic. For example, for the topic "Ocean acidification," the evaluator identified key concepts such as Carbon dioxide, Marine organisms and Ocean water. The Degree Centrality algorithm, which analyzes the relationships and importance of nodes (concepts) within a network, identified slightly different sets of concepts, reflecting its algorithmic approach to centrality within the dataset. Some topics, such as "Carbon Pricing" and "Carbon Offset," show a closer match between the evaluator's concepts and those identified by the algorithm, with both approaches recognizing terms related to emissions, climate change and carbon pricing mechanisms. However, other topics, such as "Sea Level Rise" and "Ocean acidification" exhibit some differences in the specific concepts identified, which could suggest nuances in how human evaluators and the algorithm prioritize or interpret concepts related to a given topic.

Overall, the table highlights the alignment and discrepancies between human evaluation and an algorithmic approach to concept identification. This comparison provides insights into the strengths and limitations of each method in terms of capturing key concepts in environmental topics.

ID	Key Concepts identified by evaluator	NDegree Centrality
126	Ocean acidification	Carbon dioxide;Marine organisms;Carbonic acid;Ocean water;Ocean
120	Carbon Pricing	Climate change;Emissions;Carbon pricing;Fossil fuels;Carbon price
115	Carbon Offset	Carbon offsets;Project owners;Reforestation
111	Paris Climate Agreement	Paris agreement;Framework convention on climate change;United states
78	Carbon Border Adjustment	Cbam;Carbon prices;Carbon tariff;Wealthy countries;Low-income countries
98	Food Systems and Agriculture	Smith;Climate change;Agriculture;Food systems;Harper
83	Sea Level Rise	Humans;Sea levels;Earth;Climate.gov;Scott a. kulp
113	Electric Vehicles	Electric vehicles;Evs;Hev;Phev;Cars and trucks
106	Permafrost	Permafrost;Nature reviews earth & environment;Nature;Emissions;Journal of geophysical research
117	Air Pollution	Air pollution;U.s. environmental protection agency;India;U.s.;China

Table 7.13: Human vs Algorithmic Key Concept Identification across Quizzes

7.4 Discussion

This chapter interprets the results presented in Chapter 7.3, placing them in context with the overall research objectives. It discusses how well the problem has been addressed, what insights were gained and which limitations, ethical and sustainability considerations are relevant to this work.

7.4.1 Interpretation of Results

The evaluation results reveal differences between the RAG and GRAG approaches to quiz generation. While the overall goal was to enhance question quality through structured retrieval using knowledge graphs, the performance outcomes varied significantly across topics. In this subsection, these variations are analyzed in depth, connecting observed quiz ratings with graph properties. The aim is to understand not only why certain quizzes performed better than others, but also under what conditions GRAG contributes positively—or negatively—to quiz generation. These interpretations form the foundation for refining retrieval-guided generation

methods in future work.

The insights gained from this analysis offer a nuanced foundation for addressing the research questions, clarifying how retrieval-augmented generation methods influence both question quality and content structuring.

7.4.2 RQ1: To what extent can contemporary LLMs adeptly generate questions and effectively structure learning content in the context of environmental protection topics?

The findings suggest that large language models (LLMs), particularly when augmented with retrieval mechanisms (RAG), are capable of generating reasonably relevant and educationally valuable questions from open-source environmental content. The *overall quality* score on question-level for RAG-generated questions averaged *3.67*, aligning closely with the results reported by Sayed et al. (2024). Furthermore, the relevance score of 0.85 (on a 0–1 scale) reported by Sayed et al. (2024) slightly surpasses the *relevance* observed in this research in the question-level evaluation for RAG (*3.92*) and exceeds the *relevance* score for GRAG (*3.21*).

While the *overall quality* on question-level of generated content is encouraging, occasional issues such as ambiguous phrasing and mismatched *difficulty* levels underscore the necessity of careful prompt engineering and robust quality assurance when deploying LLMs in autonomous educational tools.

The *distractor quality (or answer plausibility)* reported in the question-level evaluation for RAG averaged *3.57*, indicating moderate quality, with GRAG following at *3.03*. Notably, the top five rated quizzes generated by each method consistently achieved high *overall quality* scores in the question-level evaluation, ranging from *4.18 to 4.68*, demonstrating that LLaMa 3.1 can produce high-quality questions in favorable conditions.

The quiz-level evaluation revealed that both methods exhibited a relatively high level of *redundancy*, which is a known limitation in autoregressive models, where these models can get stuck in a loop, generating repetitive or overly similar content (Amaratunga, 2023). Similar patterns of redundancy were reported by Lopez et al. (2021) in their evaluation of GPT-2 for automatic question generation (see Chapter3).

The RAG method exhibited lower *overall quality* and higher *redundancy* at the quiz-level, suggesting that, although individual questions were relevant and useful, the overall coherence and structure of the quizzes require further refinement. In contrast, GRAG demonstrated better *overall quality* and reduced *redundancy* at the quiz-level evaluation, though both approaches faced challenges in ensuring

smooth and logically progressive learning sequences.

The comparison between the highest-rated and lowest-rated quizzes on the quiz-level provides valuable insights into the key factors that differentiate a high-quality quiz from a low-quality one.

- **Coverage and Usefulness:** The highest-rated quizzes generally excel in *coverage* and *usefulness*, providing a more comprehensive and valuable learning experience. In contrast, the lowest-rated quizzes often suffer from poor *coverage* and lack of *usefulness*, making them less effective for learners.
- **Redundancy:** *Redundancy* scores are relatively high across both the highest-rated quizzes, with a mean value of *3.8* for RAG and *3.2* for GRAG and lowest-rated quizzes with a mean of *3.2* for both methods. This suggests that, regardless of the *overall quality*, many quizzes included a notable amount of overlapping or repetitive questions.
- **Progressiveness:** The highest-rated quizzes show slightly better *progressiveness*, meaning that questions in these quizzes are logically structured and progressively more challenging. In contrast, the lowest-rated quizzes tend to lack this structure, which can lead to a fragmented learning experience for users.

Overall, the key differentiators of high-quality quizzes were greater *coverage*, more *useful* content and slightly improved *progressiveness*, while *redundancy* remained a common issue across all quizzes.

While generated questions could be loosely categorized by *difficulty*, the progression across levels was limited and often lacked a clear pedagogical structure. Overall, while LLaMa 3.1 demonstrated competence in generating individual, contextually relevant questions, their capacity to effectively structure learning content—such as well-sequenced quizzes with clear pedagogical progression—remains limited. The lack of consistent *difficulty* scaffolding and the presence of *redundancy* suggest that autonomous deployment for fully structured educational content is premature without additional human intervention or algorithmic refinement. Thus, although LLaMa 3.1, particularly with RAG or GRAG augmentation, shows promising potential, its role in content structuring is best viewed as assistive rather than fully autonomous at this stage.

7.4.3 RQ2: Can Knowledge-Graph-based Retrieval Augmented Generation enhance the quality of the generated questions?

The evaluation of quiz quality, combined with a detailed graph analysis, offers critical insights into the effectiveness of using GRAG for automatic quiz creation. GRAG is designed to use structured semantic knowledge to better ground questions in context and make them more coherent, but the results show that its performance depends heavily on the specific structure and quality of the knowledge graphs.

Contrary to initial expectations, the GRAG method did not outperform standard RAG when evaluated at the question-level. One likely explanation lies in the structural characteristics of the generated knowledge graphs. As revealed by the exploratory graph analysis, many topic-specific graphs are sparse, weakly connected and contain a high number of isolated nodes. The average degree across graphs is low (M = 1.33) and density is similarly minimal (M = 0.0331), indicating that most concepts are only loosely linked. This sparsity limits the semantic pathways that the GRAG model can rely on during generation, potentially resulting in less coherent or incomplete quizzes.

Moreover, while GRAG aims to enhance semantic focus, overly fragmented or diffuse graphs can introduce noise or disjointed information, hindering the quality of generated content. This limitation is particularly evident in the analysis of the five lowest-rated quizzes. These quizzes are associated with larger but more fragmented graphs (e.g., *airpollution* with 37 vertices and 25 isolated nodes), suggesting that the inclusion of many loosely connected concepts may overwhelm the model or dilute topical relevance.

The variations in density of the constructed knowledge graphs reflect underlying differences in discourse maturity, interdisciplinarity, and topical complexity. For example, emerging or highly technical topics such as *enhancedrockweathering* and *fusionenergy* show moderate densities and isolated node counts, which may point to a concentrated body of knowledge. In contrast, broader or more policy-oriented topics like *climatejustice* and *publictransportation* present larger graphs with relatively high numbers of isolated nodes, possibly due to the inclusion of diverse but loosely related subtopics.

The comparative analysis between the highest and lowest-rated quizzes further underscores the role of graph structure in shaping generation outcomes. Higher-rated quizzes are linked to smaller, sparser graphs with clearer conceptual focus and fewer, more meaningful connections (e.g., *phytoplankton* or *carbonoffsets*). These graphs exhibit moderate to high sparsity, yet their isolated nodes appear less

disruptive, possibly because they reflect intentional topical boundaries rather than disconnected noise. In contrast, the lowest-rated quizzes are tied to significantly larger graphs with a wider range of disconnected nodes and low density. Despite covering more concepts, their graphs do not exhibit proportionally greater connectivity or conceptual integration. This mismatch likely results in quizzes that feel fragmented or lack cohesion. Thus, bigger graphs do not inherently lead to better quizzes—what matters is how semantically connected the concepts are.

Overall, the comparison of the constructed knowledge graphs of the five highest-rated quizzes and five lowest-rated quizzes indicate that larger and more fragmented knowledge graphs may contribute to lower quiz quality, likely due to increased cognitive load or less coherent topic structuring. Conversely, smaller, simpler graphs with clearer concept relationships appear to align with higher-rated quizzes. This highlights the importance of balancing concept coverage and graph complexity when generating educational assessments.

These results suggest that the added graph structure may complicate the generation process unless carefully managed or tuned. Therefore, although GRAG holds theoretical promise for capturing inter-document relationships, in practice, it did not enhance quality in this setting. Further refinement in graph construction and integration strategies may be required.

Building on these observations, it is also important to examine how effectively the underlying graph structures capture conceptually meaningful elements, particularly when compared to human judgment.

The comparison between human-evaluated key concepts and those identified via Degree Centrality reveals partial alignment. In some cases (e.g., *carbon pricing*, *carbon offset*), the algorithm effectively surfaces central terms aligned with human expectations. However, in other cases (e.g., *sea level rise*, *air pollution*), the algorithm emphasizes named entities or specific references (e.g., "climate.gov," "India") that may lack broad conceptual value. This divergence highlights the importance of coupling algorithmic measures with semantic relevance checks when curating or enriching graphs for GRAG-based generation.

The quiz-level analysis highlights the importance of considering not just the quality of individual questions but also how those questions work together to create a cohesive, engaging quiz experience. While individual questions might score well on *relevance* or *educational usefulness*, the RAG method's lower *overall quality* and higher *redundancy* scores suggest that its outputs often lack coherence and exhibit repetitive content at the quiz-level. In contrast, GRAG showed improvements in structure and reduced *redundancy* at quiz level, indicating a better grasp of quiz composition, although both methods still faced difficulties in achieving smooth *progression* and logical flow throughout the quiz.

Therefore, in response to RQ2, while Knowledge-Graph-based Retrieval Aug-

mented Generation offers conceptual advantages, the empirical evidence from this study indicates that it did not enhance the quality of individual questions compared to standard RAG. However, at the quiz-level, GRAG contributed to improvements in *overall quality* and reduced *redundancy*, suggesting potential benefits for organizing content even if question-level quality remains limited.

7.4.4 RQ3: To what extent is it necessary to preprocess the documents before prompting the LLM?

Although preprocessing was not isolated as an explicit experimental variable in this study, observations throughout implementation clearly indicate that it plays a crucial role in shaping the coherence and relevance of LLM-generated questions.

In the RAG method, the RecursiveCharacterTextSplitter was used to segment texts at the sentence level, improving granularity and contextual clarity. For GRAG, source PDFs were converted to Markdown format to preserve table structures and the Markdown node parser was applied for sentence splitting. Early experiments also incorporated Named Entity Recognition (NER) to extract key concepts for prompting; however, subsequent iterations found that concepts identified via Degree Centrality within knowledge graphs were more effective for guiding content generation.

Preprocessing thus is a crucial—though often underexplored—step in optimizing LLM-driven educational content generation. Specifically, in the GRAG method, the PDF-to-Markdown conversion introduced a degree of noise, underscoring the need for more refined preprocessing pipelines. This challenge also highlights the importance of developing improved strategies for handling complex document features such as tables, which could enable the model to more effectively leverage embedded structured information.

In summary, in response to RQ3, preprocessing is not only beneficial but often necessary for ensuring coherent, relevant and high-quality outputs from LLMs in the context of environmental education content generation. Future work should prioritize systematic evaluation and refinement of preprocessing strategies to maximize model performance and content quality.

7.4.5 RQ4: In what way can prompt engineering enhance the quality of the generated questions?

Prompt engineering played a decisive role in enhancing the quality, coherence, and reliability of the generated questions. The inclusion of detailed task-specific instructions, explicit output formatting guidelines and role definitions for the LLM

substantially reduced hallucinations and improved both the relevance and structure of the outputs.

Prompts were designed following the guidelines outlined in Chapter 2 and informed by best practices from Amaratunga (2023). To ensure the generation of syntactically valid and structured outputs, output formatting strategies such as those proposed by Ozdemir (2024) were applied—specifically requiring the LLM to produce responses in valid JSON format. Crafting prompts that explicitly defined the desired output structure, including clear descriptions of the question components (e.g., question stem, answer choices, correct answer, distractors), consistently yielded more reliable and usable results.

In addition, prompt templating, as described by Amaratunga (2023), was systematically applied across all tasks. By employing consistent and well-structured prompt templates—including defined input-output formats, explicit task instructions and expected response structures—the generation process was made more reliable and reproducible. This approach helped standardize interactions with the model, ensuring that prompts across different tasks maintained clarity, coherence, and alignment with the intended output type.

Furthermore, incorporating role-based framing within prompts—for example, specifying "You are a knowledge graph builder" or "You are a tutoring system that generates questions based on the provided documents"—enhanced response quality and reduced ambiguity. This technique, often referred to as prompt priming, as described by Amaratunga (2023), helps establish a clear context and role for the model. Such contextualization appeared to sharpen the model's generative behavior, resulting in more accurate, focused and contextually relevant outputs.

While prompt engineering considerably improved output quality, it did not fully resolve the challenges inherent in fully automated, high-quality question generation. Even among the highest-rated quizzes, occasional flawed or ambiguous questions persisted, underscoring the need for further safeguards and refinement before autonomous deployment in educational settings.

In summary, in response to RQ4, prompt engineering demonstrably enhances the quality of generated questions by improving coherence, reducing hallucinations, and ensuring output structure alignment. Future work could further augment these gains through advanced techniques such as few-shot learning, chain-of-thought prompting and more sophisticated role-based prompt strategies.

Lessons Learnt

A key insight from the evaluation process is the critical importance of grounded context in guiding effective question generation. The consistent performance of the RAG method across multiple metrics underscores the advantage of coupling

retrieval mechanisms with generative capabilities. In contrast, GRAG frequently produced questions and distractors that lacked conceptual relevance, particularly in abstract or less well-defined topics, underscoring the challenges posed by complex or fragmented knowledge graphs.

Notably, even high-quality questions occasionally contained subtle flaws—such as unclear distractors or mildly ambiguous phrasing—demonstrating the difficulty of automating not only surface-level fluency but also deeper pedagogical soundness.

The findings point to several opportunities for improving both RAG and GRAG methods. For RAG, incorporating more advanced preprocessing and postprocessing steps could further enhance output quality. One promising direction is to introduce a second-stage LLM evaluation, where the model assesses and refines its own generated questions to improve clarity, relevance and pedagogical value. Similarly, an additional step focused on ensuring progression and cohesion across quiz items could help optimize quiz structure and learner experience.

For GRAG, the results emphasize that success hinges on the structure and quality of the input knowledge graph. Several refinement strategies emerge as promising:

- Graph cleaning and enhancement: Preprocessing steps that remove noisy nodes and enhance semantic cohesion (e.g., clustering or embedding-based filtering) may improve graph utility.
- Dynamic graph filtering: Tailoring graph content using combined measures of centrality and semantic relevance could improve contextual grounding while minimizing distractions from peripheral nodes.
- Topic-specific adaptations: Recognizing that some topics, especially highly interdisciplinary ones, yield inherently sparser or more fragmented graphs, adaptive approaches could dynamically adjust the influence of graph-based input during generation.

Additionally, future work should explore applying GRAG with existing high-quality knowledge graphs to disentangle whether observed limitations stem from the graph construction process or from the question generation mechanism itself.

Moreover, it would be valuable to explore the quality of quizzes generated using a knowledge graph constructed by integrating and linking information across multiple documents. Such an approach could reveal whether aggregating diverse sources enhances the coherence, relevance, and depth of the generated questions.

Overall, while GRAG offers a conceptually promising framework, its effectiveness is highly sensitive to the structure and coherence of the underlying knowledge graph (Chen, 2025). Simply adding more nodes or factual content does not

guarantee improved outputs; instead, thoughtful graph construction, filtering and integration are essential to unlocking the full potential of graph-augmented generation for educational content.

Limitations

Several limitations should be acknowledged when interpreting the findings of this study. First, the evaluation process relied on human annotators whose judgments, while guided by a carefully designed and iteratively refined rubric, inevitably introduced elements of subjectivity. At the quiz-level, each quiz was evaluated by a single annotator, which limits the statistical confidence in detecting subtle quality differences. However, this was partially compensated by involving an expert reviewer to strengthen the validity of assessments. At the question-level, each quiz item was independently rated by two annotators and their scores were averaged to mitigate individual biases and balance the diverse perspectives of the 36 raters involved.

Second, the system was evaluated exclusively on English-language content. As a result, its generalizability to multilingual, cross-cultural, or non-English educational contexts remains untested. Differences in linguistic structure, cultural framing, or educational standards could meaningfully affect both retrieval and generation quality in other settings.

Finally, this study employed off-the-shelf, pretrained LLMs without domain-specific fine-tuning. This limits the models' ability to fully capture specialized vocabularies, nuanced terminology, or context-specific question styles relevant to environmental protection and related topics. Incorporating fine-tuning or domain-adaptive training in future work may further improve performance and relevance.

Ethical Considerations

Automated generation of educational content, particularly assessments, raises important ethical questions. Inaccurate or misleading questions could misinform learners or skew evaluation outcomes. Biases present in training data may be reflected in generated questions, potentially disadvantaging certain groups or perpetuating stereotypes.

The use of such technology in high-stakes testing environments should therefore involve careful oversight and transparency. Future misuse—such as generating persuasive misinformation or unfair test items—could arise if quality control mechanisms are insufficient.

Sustainability Considerations

Large-scale language models like those used in RAG and GRAG are computationally intensive. The inference process—especially when involving retrieval or ensemble methods—carries a non-negligible energy footprint. While this project did not conduct an explicit energy audit, it is important to acknowledge the environmental costs associated with training and deploying LLM-based systems.

Future work could explore more energy-efficient architectures, model distillation, or caching strategies to reduce repeated inference costs. Promoting sustainability in educational AI systems will be vital as they scale in usage.

8 Conclusions

In this study, the application of Graph-based Retrieval-Augmented Generation (GRAG) to quiz generation was explored and compared to the traditional Retrieval-Augmented Generation (RAG) method. The findings suggest that while GRAG showed improvements in terms of quiz structure and reduced redundancy, it faced challenges in generating quizzes with a smooth progression and relevance as well as educational usefulness across individual questions. RAG, although producing relevant and useful individual questions, tended to generate quizzes with lower overall quality at quiz level possibly due to structural issues and higher redundancy.

The evaluation, both at the question level and quiz level, provided valuable insights into the strengths and weaknesses of each approach. GRAG, which relies on structured semantic knowledge, performed better in maintaining quiz coherence, thematic organization and overall quality when looked at quiz level. However, its performance was impacted by the underlying knowledge graph's sparsity and the limitations of current graph-based methods. On the other hand, RAG's strength in individual question generation was overshadowed by its inability to effectively organize those questions into a well-structured quiz.

Overall, while GRAG presents a promising direction for improving quiz generation, challenges remain in fully leveraging the potential of knowledge graphs to ensure smooth transitions and progressively challenging questions. The comparison of these two methods highlights the importance of both content relevance and structural coherence in generating educational quizzes.

Future work could focus on improving the integration of knowledge graphs with more advanced techniques to address the issues of smooth progression, relevance of questions and logical flow. Additionally, further exploration of hybrid approaches combining the strengths of GRAG and RAG, as well as the refinement of graph-building techniques to reduce sparsity, could lead to more effective and coherent quiz generation models.

Bibliography

C. Rowland, "The effect of testing versus restudy on retention: A meta-analytic review of the testing effect," *Psychological bulletin*, vol. 140, 08 2014.

M. Mcdaniel, H. Roediger, and K. McDermott, "Generalizing test-enhanced learning from the laboratory to the classroom," *Psychonomic bulletin review*, vol. 14, pp. 200–6, 05 2007.

V. Aravinthan and T. Aravinthan, "Effectiveness of self-assessment quizzes as a learning tool," 2010, engineering Education Conference (EE 2010): Inspiring the Next Generation of Engineers. Engineering Education series (organised by Higher Education Academy Engineering Subject Centre UK). [Online]. Available: https://research.usq.edu.au/item/9zz9w/effectiveness-of-self-assessment-quizzes-as-a-learning-tool

V. Kumar, K. Boorla, Y. Meena, G. Ramakrishnan, and Y.-F. Li, "Automating reading comprehension by generating question and answer pairs," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 335–348.

G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A systematic review of automatic question generation for educational purposes," *International Journal of Artificial Intelligence in Education*, vol. 30, no. 1, pp. 121–204, 2020.

L. E. Lopez, D. K. Cruz, J. C. B. Cruz, and C. Cheng, "Simplifying paragraph-level question generation via transformer language models," in *PRICAI 2021: Trends in Artificial Intelligence*, D. N. Pham, T. Theeramunkong, G. Governatori, and F. Liu, Eds. Cham: Springer International Publishing, 2021, pp. 323–334.

S. Bhat, H. A. Nguyen, S. Moore, J. Stamper, M. Sakr, and E. Nyberg, "Towards automated generation and evaluation of questions in educational domains," in *Proceedings of the 15th International Conference on Educational Data Mining (EDM)*. Carnegie Mellon University, 2022. [Online]. Available: https://dev.stamper.org/publications/2022EDM-posters85.pdf

C. Diwan, S. Srinivasa, G. Suri, S. Agarwal, and P. Ram, "Ai-based learning content generation and learning pathway augmentation to increase learner

engagement," *Computers and Education: Artificial Intelligence*, vol. 4, p. 100110, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X22000650

P. Iusztin, M. Labonne, J. Chaumond, H. Tahir, and A. G. Gulli, *LLM Engineer's Handbook: Master the Art of Engineering Large Language Models from Concept to Production*, 1st ed. Birmingham: Packt Publishing, Limited, 2024.

D. Lohr, M. Berges, A. Chugh, M. Kohlhase, and D. Müller, "Leveraging large language models to generate course-specific semantically annotated learning objects," 2024. [Online]. Available: https://arxiv.org/abs/2412.04185

Z. Wang, J. Valdez, D. Basu Mallick, and R. G. Baraniuk, "Towards human-like educational question generation with large language models," *Rice University*, 2023, https://example.com/your-paper-url.

Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2312.10997

Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, "Grag: Graph retrieval-augmented generation," 2024. [Online]. Available: https://arxiv.org/abs/2405.16506

N. Scaria, S. Dharani Chenna, and D. Subramani, "Automated educational question generation at different bloom's skill levels using large language models: Strategies and evaluation," in *Artificial Intelligence in Education*, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds. Cham: Springer Nature Switzerland, 2024, pp. 165–179.

U. Lee, H. Jung, Y. Jeon, Y. Sohn, W. Hwang, J. Moon, and H. Kim, "Few-shot is enough: exploring chatgpt prompt engineering method for automatic question generation in english education," *Education and Information Technologies*, pp. 1–33, 10 2023.

H. A. Nguyen, S. Bhat, S. Moore, N. Bier, J. Stamper, T. Farrell, I. Hilliger, T. De Laet, A. Ortega-Arranz, P. J. Muñoz-Merino, I. Hilliger, T. De Laet, T. Farrell, A. Ortega-Arranz, and P. J. Muñoz-Merino, "Towards generalized methods for automatic question generation in educational domains," in *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption*, ser. Lecture Notes in Computer Science. Switzerland: Springer International Publishing AG, 2022, vol. 13450, pp. 272–284.

Z. Li, Z. Cao, P. Li, Y. Zhong, and S. Li, "Multi-hop question generation with knowledge graph-enhanced language model," *Applied Sciences*, vol. 13, no. 9, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/9/5765

G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A systematic review of automatic question generation for educational purposes," *International journal of artificial intelligence in education*, vol. 30, no. 1, pp. 121–204, 2020.

D. Sarkar, "Text analytics with python : A practitioner's guide to natural language processing /," 2019.

T. Amaratunga, "Understanding large language models : Learning their underlying concepts and technologies /," 2023.

S. Ozdemir, "Quick start guide to large language models : strategies and best practices for using chatgpt and other llm's /," 2024.

X. Li, A. Henriksson, M. Duneld, J. Nouri, Y. Wu, A. M. Olney, I. I. Bittencourt, I.-A. Chounta, Z. Liu, and O. C. Santos, "Supporting teaching-to-the-curriculum by linking diagnostic tests to curriculum goals: Using textbook content as context for retrieval-augmented generation with large language models," in *Artificial Intelligence in Education*, ser. Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2024, pp. 118–132.

S. Xu, M. Chen, S. Chen, D.-S. Huang, W. Chen, and Z. Si, "Enhancing retrieval-augmented generation models with knowledge graphs: Innovative practices through a dual-pathway approach," in *Advanced Intelligent Computing Technology and Applications*, ser. Lecture Notes in Computer Science. Singapore: Springer Nature Singapore, 2024, pp. 398–409.

H. A. E. Barrasa, Jesus and J. Webber, "Knowledge graphs," 2021.

I. Robinson, "Graph databases: New opportunities for connected data," 2015.

R. Chen, "Retrieval-augmented generation with knowledge graphs: A survey," in *Submitted to Computer Science Undergradaute Conference 2025 @ XJTU*, 2025, under review. [Online]. Available: https://openreview.net/forum?id=ZikTuGY28C

S. Bulathwela, H. Muse, and E. Yilmaz, "Scalable educational question generation with pre-trained language models," 2023.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

S. Reddy, D. Raghu, M. M. Khapra, and S. Joshi, "Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 376–385. [Online]. Available: https://aclanthology.org/E17-1036

D. Lohr, M. Berges, A. Chugh, M. Kohlhase, and D. Müller, "Leveraging large language models to generate course-specific semantically annotated learning objects," *Journal of computer assisted learning*, vol. 41, no. 1, p. n/a, 2025.

A. Sayed, M. Kanojia, and S. Nabajja, "Advanced subjective question bank generation using retrieval augmented generation architecture," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 16, no. 3, p. 14, Jul. 2024. [Online]. Available: https://cspub-ijcisim.org/index.php/ijcisim/article/view/706

X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," 2024. [Online]. Available: https://arxiv.org/abs/2402.07630

Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, and Z. Li, "Retrieval-augmented generation with knowledge graphs for customer service question answering," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR 2024. ACM, Jul. 2024. [Online]. Available: http://dx.doi.org/10.1145/3626772.3661370

N. Wiratunga, R. Abeyratne, L. Jayawardena, K. Martin, S. Massie, I. Nkisi-Orji, R. Weerasinghe, A. Liret, and B. Fleisch, "Cbr-rag: Case-based reasoning for retrieval augmented generation in llms for legal question answering," in *Case-Based Reasoning Research and Development*, J. A. Recio-Garcia, M. G. Orozco-del

Castillo, and D. Bridge, Eds. Cham: Springer Nature Switzerland, 2024, pp. 445–460.

D. Seyler, M. Yahya, and K. Berberich, "Knowledge questions from knowledge graphs," in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, ser. ICTIR '17. ACM, Oct. 2017, p. 11–18. [Online]. Available: http://dx.doi.org/10.1145/3121050.3121073

J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994.

H. Sharp, Y. Rogers, and J. Preece, *Interaction Design. Beyond Human-Computer Interaction*, 01 2007.

P. Loucopoulos and V. Karakostas, *System Requirements Engineering*, 01 1995.

D. Ameller, C. Ayala, X. Franch, and J. Cabot, "How do software architects consider non-functional requirements: An exploratory study," 09 2012.

T. Keller, "Contextual requirements elicitation: An overview," Seminar in Requirements Engineering, Spring 2011, 2011, available online: mailto:t.keller@access.uzh.ch.

L. Jaramillo-Mediavilla, A. Basantes-Andrade, M. Cabezas-González, and S. Casillas-Martín, "Impact of gamification on motivation and academic performance: A systematic review," *Education Sciences*, vol. 14, no. 6, 2024. [Online]. Available: https://www.mdpi.com/2227-7102/14/6/639

P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-augmented generation for ai-generated content: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.19473

B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, and S. Tang, "Graph retrieval-augmented generation: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2408.08921

A. Dhar. (2023) Quantize llama models with ggml and llama.cpp. Accessed: 2025-05-06. [Online]. Available: https://towardsdatascience.com/quantize-llama-models-with-ggml-and-llama-cpp-3612dfbcc172

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

Meta AI, "Introducing meta llama 3: The next generation of open large language models," https://ai.meta.com/blog/meta-llama-3/, 2024, accessed: 2025-05-02.

——, "Llama 3.1 8b instruct," https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, 2024, accessed: 2025-05-02.

LangChain, "Recursive text splitter - langchain documentation," 2024, accessed: 2024-02-07. [Online]. Available: https://python.langchain.com/v0.1/docs/modules/data_connection/document_transformers/recursive_text_splitter/

LlamaIndex Contributors, "Comparing llm path extractors for knowledge graph construction: Simplellmpathextractor," https://docs.llamaindex.ai/en/stable/examples/property_graph/Dynamic_KG_Extraction/#1-simplellmpathextractor, 2025, accessed: 2025-05-06.

LlamaIndex Developers, "Lpg index guide - retrieval and querying," 2024, accessed: 2025-02-12. [Online]. Available: https://docs.llamaindex.ai/en/stable/module_guides/indexing/lpg_index_guide/#retrieval-and-querying

NebulaGraph, "Nebulagraph documentation: Graph algorithms description," https://docs.nebula-graph.io/3.1.0/graph-computing/algorithm-description/, 2023, accessed: 2025-05-06.

S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, "Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely," 2024. [Online]. Available: https://arxiv.org/abs/2409.14924

Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, and N. Zhang, "Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities," *World Wide Web*, vol. 27, no. 58, 2024. [Online]. Available: https://doi.org/10.1007/s11280-024-01297-w

T. Hosking, P. Blunsom, and M. Bartolo, "Human feedback is not gold standard," 2024. [Online]. Available: https://arxiv.org/abs/2309.16349

Appendix

Appendix: Tables

Table 1: Graph Statistics

Graph Space	Vertices	Edges	Average Degree	Density	Sparsity	Isolated Nodes
climatesensitivity	15	10	1.3333	0.0476	0.9524	10
carboncapture	28	20	1.4286	0.0265	0.9735	17
biochar	12	10	1.6667	0.0758	0.9242	11
parisagreement	13	10	1.5385	0.0641	0.9359	10
citiesandclimatechange	25	18	1.4400	0.0300	0.9700	14
microgrids	12	8	1.3333	0.0606	0.9394	9
airpollution	37	25	1.3514	0.0188	0.9812	25
enhancedrockweathering	29	20	1.3793	0.0246	0.9754	18
nuclearenergy	15	9	1.2000	0.0429	0.9571	6
nationalclimateassessment	25	20	1.6000	0.0333	0.9667	20
biofuel	13	10	1.5385	0.0641	0.9359	9
investingandclimatechange	30	19	1.2667	0.0218	0.9782	20
extremeheat	26	20	1.5385	0.0308	0.9692	18
forestsandclimatechange	24	14	1.1667	0.0254	0.9746	14
climateresilientinfrastructure	16	9	1.1250	0.0375	0.9625	10
wildfires	28	19	1.3571	0.0251	0.9749	19
permafrost	29	18	1.2414	0.0222	0.9778	20
aviation	20	10	1	0.0263	0.9737	11
phytoplankton	12	10	1.6667	0.0758	0.9242	11
polarjetstreamandpolarvortex	19	15	1.5789	0.0439	0.9561	16
freighttransportation	26	16	1.2308	0.0246	0.9754	15
urbanheatislands	11	5	0.9091	0.0455	0.9545	7
organicwaste	32	19	1.1875	0.0192	0.9808	18
freshwaterandclimatechange	44	27	1.2273	0.0143	0.9857	24
climatetargets	19	10	1.0526	0.0292	0.9708	9
oceanalkalinityenhancement	26	20	1.5385	0.0308	0.9692	15
foodsystemsandagriculture	22	20	1.8182	0.0433	0.9567	6
fusionenergy	17	10	1.1765	0.0368	0.9632	9

Graph Space	Vertices	Edges	Average Degree	Density	Sparsity	Isolated Nodes
soilbasedcarbonsequestration	32	19	1.1875	0.0192	0.9808	18
fertilizerandclimatechange	26	17	1.3077	0.0262	0.9738	17
steel	39	29	1.4872	0.0196	0.9804	21
renewableenergy	25	19	1.5200	0.0317	0.9683	11
miningandmetals	19	9	0.9474	0.0263	0.9737	10
netzeroemissions	30	19	1.2667	0.0218	0.9782	17
climatemodels	29	20	1.3793	0.0246	0.9754	20
windenergy	22	18	1.6364	0.0390	0.9610	17
greenhousegases	17	10	1.1765	0.0368	0.9632	12
heatingandcooling	38	20	1.0526	0.0142	0.9858	18
electricgrid	45	29	1.2889	0.0146	0.9854	21
hurricanes	29	19	1.3103	0.0234	0.9766	15
lossanddamage	33	20	1.2121	0.0189	0.9811	20
radiativeforcing	12	9	1.5000	0.0682	0.9318	5
scope12and3emissions	21	19	1.8095	0.0452	0.9548	10
mitigationandadaptation	34	20	1.1765	0.0178	0.9822	16
coastalecosystemsandclimatechange	30	18	1.2000	0.0207	0.9793	19
energystorage	16	10	1.2500	0.0417	0.9583	9
solarenergy	29	20	1.3793	0.0246	0.9754	22
sealevelrise	31	19	1.2258	0.0204	0.9796	13
carbonoffsets	13	10	1.5385	0.0641	0.9359	10
publictransportation	34	20	1.1765	0.0178	0.9822	19
climatejustice	25	17	1.3600	0.0283	0.9717	13
climatechangeattribution	30	20	1.3333	0.0230	0.9770	21
transmission	33	19	1.1515	0.0180	0.9820	19
electricvehicles	27	19	1.4074	0.0271	0.9729	19
hydrogen	28	20	1.4286	0.0265	0.9735	22
carbonborderadjustments	28	15	1.0714	0.0198	0.9802	17
concrete	13	10	1.5385	0.0641	0.9359	10
oceanacidification	17	10	1.1765	0.0368	0.9632	8
advancednuclearreactors	31	18	1.1613	0.0194	0.9806	18

Appendix: Figures

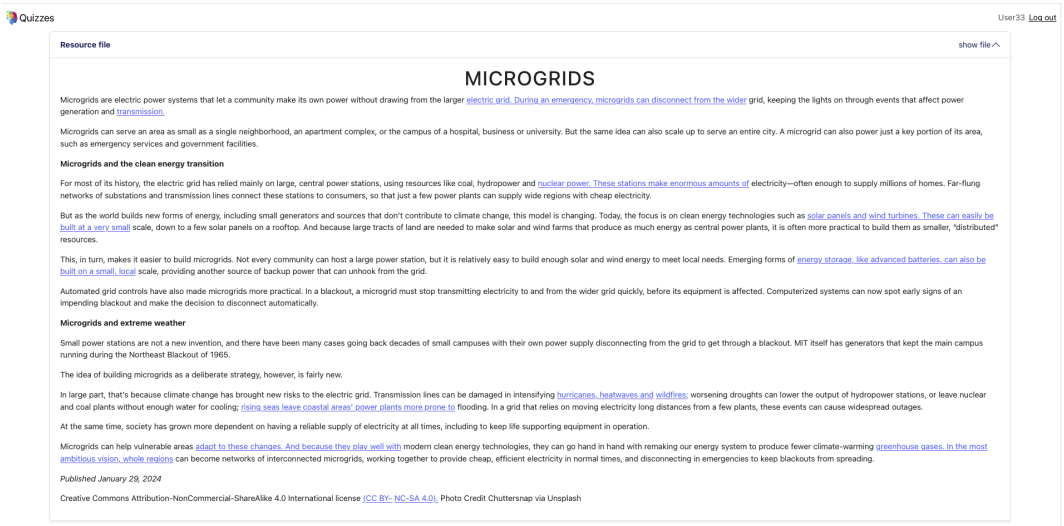


Figure 1: Evaluation App: Resource Text

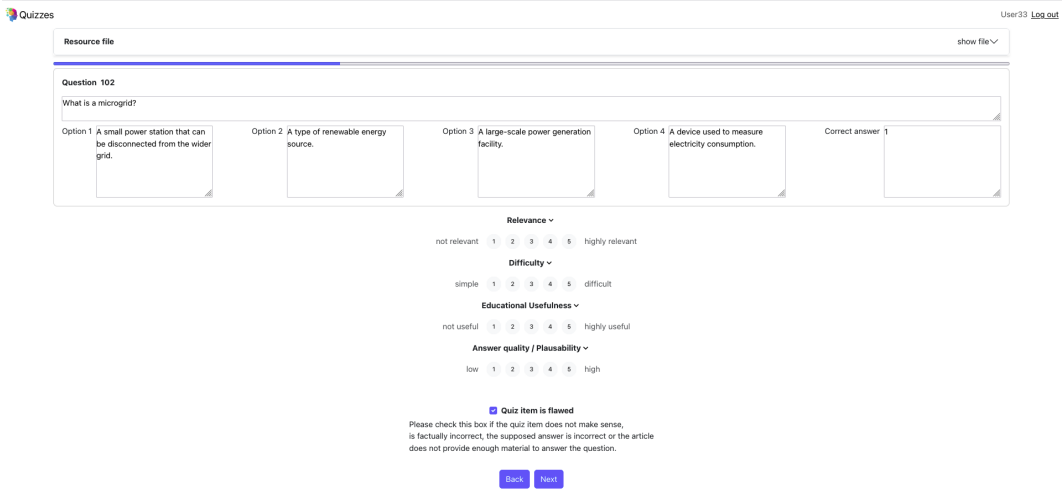


Figure 2: Evaluation App: Question