Developing AI Agents for Course Planning and Development: An AI Framework for Academic Module Development at Teesside University

Chede Edward Ekene

Student number: D3333789
Department of Applied Data Science
Supervised by Dr. Alsmadi Hiba

May 7, 2025

Abstract

The increasing complexity of curriculum planning in higher education has underscored the need for intelligent systems that can assist academic leaders in developing policy-compliant, pedagogically sound course modules. This project details the design, implementation, and evaluation of a domain-specific AI-powered assistant developed for Teesside University (TU). The system leverages a fine-tuned Large Language Model (Llama-3), integrated with a Retrieval-Augmented Generation (RAG) pipeline and an agentic decision-making layer, to generate accurate and structured academic content that aligns with institutional guidelines. Course module guidelines hosted on the university's website was used to generate the dataset. The dataset preprocessing involved curating the course guidelines into prompt-response pairs. Fine-tuning was conducted using Low-Rank Adaptation (LoRA) and 4-bit quantization to enable efficient training on limited hardware. A hybrid retrieval system was introduced to enhance document retrievals from the knowledge base. It combined FAISS for dense semantic search and BM25 for lexical matching to ground responses in institutional documents. The agentic layer dynamically determined whether to retrieve, generate, or enhance responses based on the user query. We conducted quantitative and qualitative measurements to evaluate the performance of the Agentic model. The system achieved a Perplexity of 2.48 for fluent language, a Hits@3 of 1.00 for perfect retrieval accuracy, and a Cosine Similarity of 0.7806, which showed a strong semantic alignment. The Hallucination Rate of 0.00 confirmed factual grounding, while BLEU (0.0578) and ROUGE scores (ROUGE-1: 0.3128, ROUGE-2: 0.1148, ROUGE-L: 0.2275) highlighted formatting challenges due to verbose outputs. Human evaluators rated outputs highly across relevance, fluency, completeness, and organization, affirming the model's practical utility in academic planning. These findings confirm the system's effectiveness in automating and enriching the curriculum design process while ensuring alignment with educational policy This research contributes to the growing field of AI in education by providing a scalable framework for AI-assisted academic planning. It offers a practical solution to reduce administrative workload, promote curriculum consistency, and support digital transformation in higher education institutions.

Keywords: *Large Language Models (LLMs), Retrieval Augmented Generation (RAG), Fine-Tuning, PEFT, LoRA (Low-Rank Adaptation), Hybrid Search, Cosine Similarity, Teesside University*

1. Introduction

Artificial intelligence has emerged as a transformative force in education, fundamentally reshaping the way courses are planned, developed, and delivered. The integration of AI into academic module design presents an opportunity to enhance efficiency, personalization, and adaptability in curriculum planning. By leveraging AI-driven technologies such as Large Language Model (LLM) and Retrieval-Augmented Generation (RAG), educators and institutions can create dynamic, student-centred learning experiences while ensuring adherence to academic standards and accreditation guidelines[6]. The rapid advancement of AI-powered systems like ChatGPT and AI-driven platforms that enhances teaching and learning like IBM Watson Education [26], has demonstrated the potential for automated curriculum development. These systems offer features such as intelligent content generation, personalized learning pathways, and real-time curriculum adaptation. Yet, these existing solutions often lack the depth of contextual understanding necessary to align with specific institutional frameworks, Many AI-based curriculum tools operate on generalized educational principles rather than catering to the nuanced requirements of university-specific accreditation and learning objectives [8]. As higher education institutions continue to adopt AI-driven course development methodologies, it becomes evident that a more specialized approach is needed. Large Language Models, have exhibited remarkable capabilities in generating high-quality textual content, assisting with academic research, and supporting curriculum design [6]. Nevertheless, these models are not without limitations. Issues such as bias [31], hallucinations [11], and lack of interpretability present challenges that must be addressed to ensure reliability and accuracy in academic settings. Additionally, while AI has shown promise in automating administrative tasks, its role in pedagogical decision-making remains a subject of ongoing debate [10]. Retrieval-Augmented Generation (RAG) presents a viable solution to some of these challenges because of its ability to combine the generative capabilities of AI with real-time information retrieval. Unlike standalone language models that rely solely on pre-trained knowledge, RAG systems dynamically retrieve relevant content from external sources, ensuring that course materials remain accurate, current, and aligned with evolving academic standards [27]. In the context of course planning and module development, RAG can enhance the quality and relevance of generated content while mitigating issues related to outdated or incorrect information [20]. Developing AI agents specifically tailored for an academic institution requires an in-depth analysis of the existing gaps in AI-based curriculum design tools. One major limitation of many systems is their inability to seamlessly integrate university-specific guidelines, accreditation requirements, and pedagogical best practices to the AI tool. For example, while AI can assist in drafting course descriptions and learning outcomes, it often lacks the contextual awareness to custom these elements with institutional learning goals and assessment strategies [1]. More so, the absence of adaptive learning mechanisms in many AI curriculum tools has shown to limit their capacity to personalize educational content based on student performance and engagement metrics [10] Incorporating principles of curriculum design, such as ADDIE (Analysis, Design, Development, Implementation and Evaluation) Bloom's Taxonomy [28] and Constructive Alignment [1], and integrating AI-driven frameworks for learning analytics, sentiment analysis, and intelligent curriculum mapping, can refine course content and teaching methodologies. Yet, ensuring compliance with institution specification guidelines remains a crucial consideration in the development of AI-assisted course planning tools. The introduction of AI agents in academic curriculum development holds immense potential to improve the efficiency and quality of course design. However, achieving this requires a structured design that balances AI's computational strengths with human teaching expertise. Developing an AI system that aligns with institutional unique academic requirements creates a robust and adaptive framework that enhances the learning experience for both educators and students. As AI continues to advance, the integration of specialized AI agents in course planning will play a pivotal role in shaping the future of higher education [35].

1.1. Problem Statement

Higher education institutions require course leaders to continuously develop, plan, and refine academic modules to align with evolving educational standards, industry trends, and university guidelines. Nonetheless, this process is often time-consuming and requires extensive research, curriculum design, and documentation in accordance with university regulations. At Teesside University, course leaders are responsible for ensuring that course modules are compliant with the university specification. To reduce their workload and streamline academic planning, this project aims to develop and evaluate an AI-Assistant that can help with:

1. Developing new course modules according to the academic guidelines.
2. Planning course structures efficiently, ensuring alignment with university standards.
3. Integrates advanced educational strategies to enhance module quality.

1.2. Significance of the Study

The objective of this study is to:

1. To fine-tune the Llama-3 model using Teesside University's Course Requirement to generate academic module content aligned with institutional and pedagogical standards.
2. To implements a Retrieval pipeline (RAG), integrating FAISS and BM25, to retrieve policy-compliant and pedagogically relevant guidelines for module design
3. To develops an agentic decision-making layer to dynamically generate or enhance module structures, ensuring alignment with advanced pedagogical frameworks
4. To create a user-friendly web interface to enable course leaders interact with the AI assistant
5. To evaluates the system's performance using quantitative metrics and qualitative human assessments to validate its effectiveness

1.3. Significance of the Study

This research makes a significant contribution to the field of educational technology and institutional AI adoption. By designing a domain-specific AI assistant for course development. it addresses a practical problem faced by lecturers and course designers. The study showcases how AI can not only automate laborious academic processes but also enhance consistency, transparency, and policy alignment across curricula. More broadly, the proposed solution can serve as a scalable model for other UK universities seeking to modernize curriculum management systems. The use of a hybrid RAG framework mitigates hallucination and ensures that the system remains grounded in real institutional documents while benefiting from the generative power of LLMs.

2. Literature Review

The use of artificial intelligence (AI) in educational planning has grown significantly in recent years. Technologies like large language models and retrieval-augmented generation (RAG) are proving to be valuable tools for addressing challenges in curriculum development and course planning. One of the popular LLM, ChatGPT, has had great success in passing standardized tests like the SAT and AP Exams [12]. Such feats have led to students and teachers harnessing the power of LLMs to complete tasks, and has shown to be both beneficial and detrimental to each group [25]. LLMs can be employed as personalized teaching assistants, content generators, and even evaluation tools. They present opportunities for enhancing learning experiences, reducing the work burden of educators, and offering scalable solutions to address educational disparities. In this section, we explore the application of artificial intelligence in education, examining how these technologies are reshaping teaching and learning practices, and paving the way for a more personalized, efficient, and equitable educational landscape.

2.1. Related Work

AI has been instrumental in transforming educational methodologies, ranging from intelligent tutoring systems to adaptive learning platforms. [4]. [3] emphasized AI's potential to personalize education and assist educators in administrative and instructional tasks. As AI continues to mature, its role in enhancing the efficiency and consistency of educational planning is increasingly recognized. Large Language Models such as OpenAI's GPT and Meta's Llama have revolutionized natural language processing, enabling the automated generation of human-like text [6], [30]. In the context of education, LLMs have been applied to generate instructional content, quiz questions, feedback, and course syllabi [33]. [15] demonstrated the viability of fine-tuning LLMs to align generated outputs with institutional instructional framework and domain-specific requirements. This opens new avenues for scaling course design while maintaining academic rigor. RAG is a paradigm that combines the generation capabilities of LLMs with the precision of information retrieval systems. According to [27]. RAG improves factual consistency by first retrieving relevant context from external documents and then generating responses grounded in that context. This approach enables AI systems to provide accurate, policy-compliant, and up-to-date recommendations by referencing institutional teaching guidelines, academic literature, and educational frameworks. It represents a critical improvement over standalone generative models which may hallucinate or drift from factual grounding. Douze et al, 2024) highlighted FAISS's efficiency in similarity-based indexing, while [13] described BM25's strength in lexical matching. To enhance retrievals, hybrid search systems can be introduced. The hybrid system integrates the dense vector-based retrievals with the sparse, token-based approaches. [22] emphasized that combining these techniques results in higher Precision and Recall in complex search tasks. Implementing the hybrid system for educational task, boosts a robust document selection, ensuring that the generated content is both contextually and semantically aligned with the instructional objectives. While RAG and hybrid retrieval enhance factual accuracy, the decision of how to apply these capabilities can depend on an intelligent control mechanism. The concept of an agentic layer, inspired by systems such as ReAct [14] allows an AI systems to determine how to best respond to a query. This decision-making process can be rule-based or learned through reinforcement. This agentic layer can help maintain contextual fidelity and policy alignment while dynamically adapting to query intent, user preferences, and content gaps. [21] suggested that with emerging research, agentic systems will be vital in building trustworthy and flexible AI systems for knowledge-intensive domains. Recent investigations have begun to dissect the nuanced roles of AI across various facets of education. [19] explored the integration of the generative AI research assistant, Elicit, into graduate nursing curricula to promote inquiry-based learning. Their study revealed that while Elicit could enhance information literacy and refine literature search skills compared to traditional databases, it also heightened the necessity of cultivating AI literacy among students. By encouraging critical evaluation of AI-generated abstracts, the study ensured that future healthcare professionals would be adept at discerning the strengths and limitations inherent in AI outputs. The study advocated for a balanced integration of AI in academic settings. Building on the theme of AI assistant, [27] assessed the efficacy of a RAG-powered Virtual Assistant (VA) designed to support call centre agents within an academic milieu. Their findings indicated that the incorporation of contextual data markedly reduced the incidence of hallucinations, thereby improving the VA's reliability. Though, the VA encountered difficulties in generalizing responses to complex queries, highlighting challenges in achieving adaptability across diverse scenarios. This study illuminated the critical importance of optimizing AI systems to handle nuanced inquiries and diverse data structures effectively. In the realm of content generation, [24] evaluated the capabilities of GPT-4 and Gemini in producing teaching cases for information systems courses. Their comparative analysis revealed that both LLMs could generate pedagogically relevant materials, albeit with distinct strengths. GPT-4 demonstrated particular proficiency in crafting normative cases, while Gemini excelled in structuring project-based scenarios. The study acknowledged the potential for factual inaccuracies and stressed the necessity of human oversight in evaluating educational outcomes, thereby underscoring the significance of hybrid human-AI workflows in maintaining the integrity of educational materials. Further expanding the scope of AI applications in academia, [32] introduced AcawebAgent, a tool designed to automate the synthesis of academic literature by leveraging LLMs and web scraping technologies. AcawebAgent offers researchers an efficient means of staying abreast of the latest research trends. However, the tool's reliance on publicly accessible web content and its token distribution strategy introduced limitations, including potential content truncation and the risk of including outdated information. These constraints emphasize the need for continuous updates and refined methodologies to ensure the accuracy and completeness of synthesized academic data. Moreover, [16] developed an AI Educational Video Assistant that integrates Automatic Speech Recognition (ASR) and LLMs to enhance video-based learning. This tool improved learner engagement through translated transcripts and AI-generated concept maps, aligning with the Cognitive Theory of Multimedia Learning. The reliance on publicly available videos, introduced the potential risks of outdated and inaccurate content, emphasizing the importance of curating and validating source materials in AI-enhanced educational tools. These collective works underscore a progressive trajectory in the application of AI within educational contexts. While each study contributes valuable insights, significant research gaps remain, particularly in the areas of ethical considerations, equitable access, and the scalability of AI tools across diverse educational contexts. This research seeks to bridge these gaps by exploring the integration of advanced AI components into a practical, policy-aligned framework, with the overarching aim of transforming academic planning processes within higher education institutions

3. Methodology

This chapter presents the design and methodology of the proposed AI-powered framework for academic module development. The system is designed to automate and enhance course planning by integrating fine-tuned large language models (LLMs), a Retrieval pipeline, an agentic decision-making layer and a user-friendly interface to assist to assist course leaders in crafting module content. The framework is designed to assists course leaders at the university in structuring and refining module content to align with institutional standards, streamline instructional design, and improve the overall efficiency of academic planning. The methodology is grounded in Python implementation. This chapter outlines the system architecture, dataset preparation, environment setup, model fine-tuning, RAG integration, agentic pipeline, user interface, and evaluation planning, demonstrating technical proficiency and critical problem-solving.

3.1. System Architecture

The system architecture comprises four core components, and they are designed to work cohesively to produce accurate, policy-compliant module content:

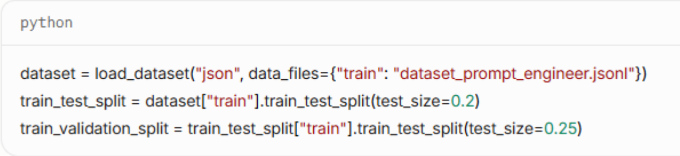
1. **Fine-Tuned Llama-3.2-3B-Instruct Model:** A large language model adapted to generate the university's academic content, leveraging domain-specific fine-tuning.
2. **Retrieval-Augmented Generation (RAG) Pipeline:**A hybrid search mechanism combining semantic and keyword-based retrieval to ground responses in institutional documents.
3. **Agentic Decision-Making Layer:** : An intelligent layer that determines the optimal response strategy, whether retrieval, generation, or enhancement, based on query context.
4. **Web Interface:**A user-facing portal enabling course leaders to input queries and receive structured responses with source transparency.

3.2. Dataset Preparation

The dataset collection and preprocessing was a crucial part of this project. It involved some key steps

3.2.1. Data Collection and Structuring

The data was collected from TU's Course Specification Guide, accessible via the academic portal (https://apps.tees.ac.uk/programmes). The guide contains module specifications and institutional guidelines. Question-answer pairs were formatted from these guidelines using prompt. For example, guidelines on learning outcomes were formatted as prompts (prompt: "List the learning outcomes for Machine Learning (CIS4035-N)") and paired with a response, The responses were extracted from the Teesside University specification guide. These pairs were stored in JSON Lines (JSONL) format, with each line representing a single text field. The following code facilitated data loading and splitting:



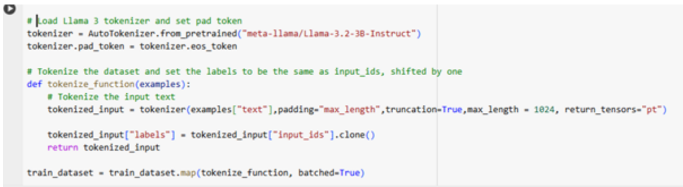
```python
dataset = load_dataset("json", data_files={"train": "dataset_prompt_engineer.jsonl"})
train_test_split = dataset["train"].train_test_split(test_size=0.2)
train_validation_split = train_test_split["train"].train_test_split(test_size=0.25)
```

Figure 1. Dataset Load

This resulted in 60% training, 20% validation, and 20% testing sets, ensuring robust model training and evaluation.

3.2.2. Pre-processing and Tokenization

The dataset was collected from Teesside University's Course Specification Guide on their website Teesside Academic Portal. Structured into 'questions' and 'answers' using prompt engineering and stored in JSON Lines (JSONL) format. Each line represented a single prompt and its corresponding response, using the course specification. The text was tokenized using Llama tokenizer, and structured into training, validation, and test sets using the datasets library. The tokenizer converted the text into numerical tokens, and the 'end of sentence' padding was set to truncate at a maximum length of 1024 tokens to accommodate GPU memory constraints. The implementation is shown below:



```
# Load Llama 3 tokenizer and set pad token
tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-3.2-3B-Instruct")
tokenizer.pad_token = tokenizer.eos_token

# Tokenize the dataset and set the labels to be the same as input_ids, shifted by one
def tokenize_function(examples):
    # Tokenize the input text
    tokenized_input = tokenizer(examples["text"],padding="max_length",truncation=True,max_length = 1024, return_tensors="pt")

    tokenized_input["labels"] = tokenized_input["input_ids"].clone()
    return tokenized_input

train_dataset = train_dataset.map(tokenize_function, batched=True)
```

Figure 2. Dataset Tokenization

This tokenization enabled the model to learn causal language modelling, predicting the next token in a sequence, which was critical for generating coherent academic responses.

3.3. Prompt Engineering

Prompt engineering played a crucial role in instructing the LLM to generate accurate and most useful responses. Prompts were carefully crafted and iteratively refined to enable the LLM to understand and solve problems through structured communication. More so, Prompt engineering contributed in providing relevant context and building intelligent, responsive systems.

3.4. Data Augmentation

To enhance the model's robustness for fine-tuning, the dataset was artificially expanded the dataset using data augmentation. The goal of augmenting the data was to improve model generalization, reduce overfitting, and enhance robustness to real-world variations by creating diverse training examples. Data augmentation helped the model learn more effectively from limited data.

3.5. Environment Setup

We utilized Google Colab with an NVIDIA T4 GPU for fine-tuning the Llama-3 LLM. Due to the high computational requirements of the model, it was not feasible to run the fine-tuning process on a CPU. The GPU acceleration provided by Colab significantly improved training speed and made the fine-tuning process manageable and efficient.

3.6. Model Architecture (Llama-3.2-3B-Instruct)

Llama 3.2 is an auto-regressive language model built on an optimized transformer architecture. The version used in this work contains 3 billion parameters, making it computationally intensive to run on the limited GPU resources available in Google Colab. We took steps to address this challenge, by applying model quantization and parameter-efficient fine-tuning techniques, to reduce the computational requirements and memory footprint of the model.

3.7. Quantization and LoRA Configuration

3.7.1. 4-Bit Quantization

The 4-bits quantization reduced the size and computational requirement of the LLM by lowering the precision of the numbers used to represent model weights and activations. This technique sped up inference and training on the hardware and lowered power consumption and carbon footprint [29]

3.7.2. Low Rank Adaptation (LoRA) Techniques

The technique allowed our large model to be adapted for specific tasks without needing to update all of its parameters. The LoRA introduced trainable low-rank matrices that was able to adjust the model's behaviour without modifying its original parameters. LoRA specifically targets the components of the model responsible for attention calculations, which are crucial for understanding relationships between tokens in the input, while avoiding the cost of full model fine-tuning. This approach, informed by [29], enabled efficient fine-tuning of limited computational resources

3.8. Model Fine-Tuning and Training Configuration:

The pretrained model was Fine-tuned to specialize it for Teesside University's academic module planning task. Fine tuning was used to adapt the model from general knowledge to domain-specific knowledge. We defined the training arguments to fine-tune the model efficiently, under memory constraints and ensure stable learning. The configuration included:

- **Batch Size:**: Set to 1 due to limited GPU VRAM, with gradient accumulation enabled (accumulation steps = 8) to simulate larger batch sizes without exceeding memory limits.
- **Learning Rate: A learning rate of 2e-4 (0.0002)** was chosen, balancing convergence speed and training stability. This value is commonly effective for fine-tuning large pretrained language models.
- **Gradient Accumulation:**Implemented to enable effective updates with small batches. Gradients were accumulated over multiple steps before being applied, optimizing memory usage without sacrificing training dynamics.
- **Learning Rate Scheduler: A linear scheduler with warmup** was used, which starts with a low learning rate, gradually increases it, and then linearly decays it throughout training. This helps with training stability and convergence.
- **16-bit floating point Precision:** Mixed-precision training with FP16 (16-bit floating point) was enabled to reduce memory usage and speed up computation without compromising model performance.
- **Epochs:** Training was run for three epochs, a typical range for LLM fine-tuning, allowing the model to sufficiently adapt to the new task without overfitting.
- **Weight Decay:** Applied to penalize large weights and encourage simpler representations, which improves performance on unseen data.
- **Memory Optimization: Gradient checkpointing** was enabled to save GPU memory during training by selectively storing and recomputing intermediate activations during the backward pass.
- **Model Monitoring:**TensorBoard was integrated for real-time tracking of training metrics such as loss, evaluation scores, and learning rate dynamics.
- **Model Saving:** The trained model and tokenizer were saved to Google Drive for reuse during inference.

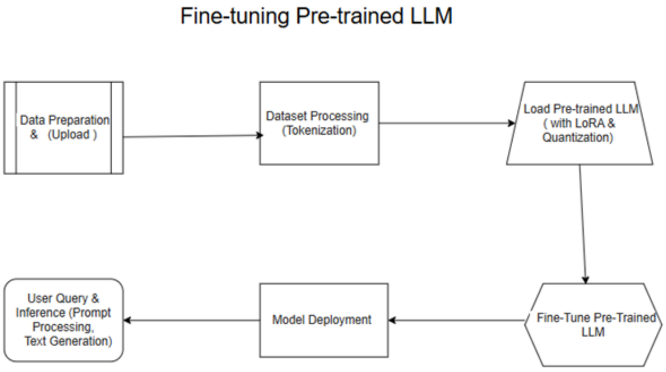

Fine-tuning Pre-trained LLM
Data Preparation & (Upload)
Dataset Processing (Tokenization)
Load Pre-trained LLM (with LoRA & Quantization)
Fine-Tune Pre-Trained LLM
Model Deployment
User Query & Inference (Prompt Processing, Text Generation)

Figure 3. A flowchart of the LLM fine-tuning process

3.9. Retrieval-Augmented Generation (RAG)

A Retrieval (RAG) system was integrated to the fine-tuned LLM to enhance the course module development process. The RAG component compliments the fine-tuned model by improving contextual understanding and mitigating hallucinations. The RAG combined a knowledge base ingestion system, an information retrieval component, and natural language generation to increase the relevance and accuracy of the AI-generated content. This approach enabled our Agentic system to create responses that are both data-driven and contextually aligned with the university's course framework.

The Retrieval process involves first converting user queries into embeddings and then searching the vector database for similar vectors. It compares the user's input with stored data using a distance metric. The most common distance metric is cosine similarity, which measures the cosine of the angle between two vectors

$$\text{Cosine Distance} = 1 - \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}||\,||\mathbf{B}||}$$

- $\mathbf{A}, \mathbf{B}$: input vectors in an n-dimensional space.
- $\mathbf{A} \cdot \mathbf{B}$: dot product of vectors $\mathbf{A}$ and $\mathbf{B}$.
- $||\mathbf{A}||$: Euclidean norm (magnitude) of vector $\mathbf{A}$, defined as $||\mathbf{A}|| = \sqrt{\sum_{i=1}^{n} A_i^2}$.
- $||\mathbf{B}||$: Euclidean norm of vector $\mathbf{B}$, defined similarly.

3.9.1. Hybrid Retrieval Mechanism

A hybrid retrieval system was implemented for this work. It incorporated a BM25 for lexical search and FAISS for dense vector search. We sought to leverage the strength of both techniques to enhance the performance of information retrieval tasks. Combining both techniques ensured that the retrieval system effectively balanced a semantic understanding (dense embeddings) with keyword-based matching, resulting in better retrieval performance in real-world scenarios where documents and queries vary in structure and complexity

3.9.2. Knowledge Base Design

The knowledge base consists of 12 PDF documents containing module specifications and program guidelines, downloaded from the university's academic portal. These documents were collated into a corpus, which was integrated into the Retrieval-Augmented Generation (RAG) pipeline to ground the system's responses in factual, policy-compliant information

3.9.3. The RAG Generation Framework

This is the final stage of the retrieval system. Here the user's input is processed by retrieving relevant information from the knowledge base. This information, along with the user's query, is inserted into a structured prompt template to create the final prompt. The completed prompt is then passed to a large language model to generate a grounded and accurate response

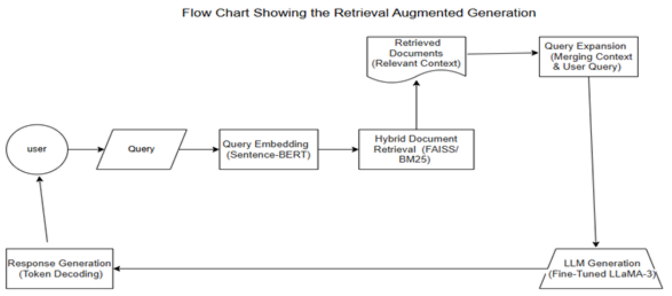

Flow Chart Showing the Retrieval Augmented Generation
user
Query
Query Embedding (Sentence-BERT)
Hybrid Document Retrieval (FAISS/ BM25)
Retrieved Documents (Relevant Context)
Query Expansion (Merging Context & User Query)
LLM Generation (Fine-Tuned LLaMA-3)
Response Generation (Token Decoding)

Figure 4. A flowchart of the Retrieval Augmented Generation process

3.10. The Agentic Pipeline

The Agentic Layer functions as the decision-making component that drives the flow of operations in response to a user's input. It evaluates the content of the query and determines the optimal response strategy. It ensures that the system reacts dynamically, either retrieving relevant documents, generating content, or enhancing its output to provide the most relevant and accurate response possible. The Agentic pipeline coordinates the RAG system and the Fine-Tuned LLM as tools to efficiently handle user queries and provide actionable, contextually enriched responses By integrating this decision-making layer with the retrieval and generation components, we sought to efficiently handle a wide range of queries and provide actionable, contextually enriched answers

3.10.1. Decision Making Process

1. **Retrieval:**The Agentic system analyses the query for domain-specific keywords that suggest the need for information, and it simultaneously measures the semantic similarity between the query and the precomputed document embeddings using cosine similarity. On this basis, the model makes a decision to employ the hybrid retrieval mechanism to retrieve relevant documents from the knowledge base. After retrieval, the relevance of the retrieved documents is evaluated to assess their suitability for grounding the response. If the retrieved documents meet the relevance threshold, they are incorporated into the prompt to guide the generation of an accurate and contextually appropriate answer.
2. **Generation:**When retrieval is unnecessary or weak, the system generates a response using the fine-tuned LLM. It processes the query and creates a response from its trained (fine-tuned) knowledge. The response is formulated without any external document references. This ensures that the system can still provide an answer based on its training.
3. **Enhancement:**If the retrieved documents are of low quality, the system generates a new response from scratch. It then merges the weak documents with the newly generated content. This enhancement combines both the retrieved and generated information. The result is a more comprehensive and accurate response

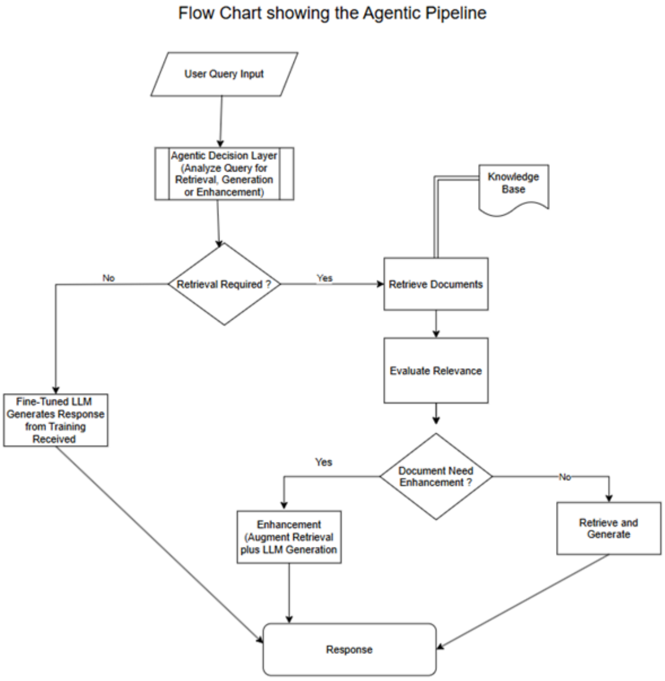

Flow Chart showing the Agentic Pipeline
User Query Input
Agentic Decision Layer (Analyze Query for Retrieval, Generation or Enhancement)
Knowledge Base
Retrieval Required ?
No
Yes
Retrieve Documents
Evaluate Relevance
Fine-Tuned LLM Generates Response from Training Received
Document Need Enhancement ?
Yes
No
Enhancement (Augment Retrieval plus LLM Generation
Retrieve and Generate
Response

Figure 5. A Flow Chart Showing the Agentic System

3.11. User Interface

The User Interface (UI) serves as the point of interaction between the end-user and the Agentic system. It is designed as a user-friendly web-based platform, that enables course leaders to input queries and receive structured, AI-generated responses seamlessly

4. Implementation

This chapter details the technical implementation of the AI-powered framework for academic module development. Here we build on the design outlined in the methodology and describe how each component from data preparation to the model deployment was implemented in practice. This entailed setting up the environment, fine-tuning the LLama-3 model, developing the RAG pipeline, implementing the agentic decision-making layer, and deploying the user interface. The implementation emphasizes reproducibility, computational efficiency, and real-world applicability to the academic institution's module development needs.

4.0.1. Environment Setup

The development environment was set up using Google Colab Pro, leveraging access to NVIDIA T4 GPUs. Google Colab was selected for its accessibility, availability of free GPU acceleration, and integration with common machine learning libraries The key dependencies were installed, and libraries imported We applied the precision training (fp16) and 4-bits quantization technique to optimize memory and mitigate Colab's VRAM constraints.

4.0.2. Dataset Preparation and Preprocessing

The dataset was collected from Teesside University's Course Guide on their website Teesside Academic Portal structured using prompt engineering and stored in JSON Lines (JSONL) format. Each line represented a single module specification or instructional guideline in a text field. The data was cleaned, tokenized using the Llama-3 tokenizer, and structured into training, validation, and test sets using the datasets library

4.1. Model Fine-Tuning

The fine-tuning process utilized the 'Llama-3.2-3B-Instruct' model enhanced with LoRA (Low-Rank Adaptation), enabling efficient parameter optimization on consumer-grade hardware (Google Colab GPU). Quantization techniques (4-bit using bitsandbytes) were employed to optimize memory consumption and computation time. Training was managed using Hugging Face's Trainer API, while Tensor Board provided real-time monitoring of loss, learning rate, and training progression.

4.2. Training and Validation

The model underwent fine-tuning for three epochs using a batch size of 1 and a gradient accumulation step of 8, effectively simulating a larger batch size for stable training. The maximum sequence length was set to 512 tokens.

- **Training Time:** Approximately 4 hours on Google Colab Pro with a T4 GPU.
- **Learning Rate:** 2e-4 with linear warmup over the first 100 steps.
- **Evaluation Strategy:** The model was evaluated every 50 steps using validation loss.

The validation set enabled early diagnosis of overfitting or underfitting, ensuring the generalizability of the fine-tuned model. The final model checkpoints were saved both locally and to Google Drive.

4.3. Retrieval-Augmented Generation (RAG) Pipeline

To enhance document retrieval, a hybrid retrieval mechanism was implemented, combining FAISS-based dense vector search with BM25-based sparse retrieval. FAISS (Facebook AI Similarity Search) enabled semantic matching through dense embeddings, while BM25 ensured keyword-sensitive precision. This dual approach leveraged the semantic richness of dense vectors and the precision of token-based matching to enhance the system's ability to surface the most contextually relevant documents for query augmentation and response generation.

4.4. Agentic Decision-Making Layer

A key architectural component of the system was the Agentic Decision-Making Framework, which served as the orchestration layer that intelligently managed how the system responded to user queries. Upon receiving a user query, the agent determined whether to:

- Generate a response from scratch using the fine-tuned Llama-3 model,
- Retrieve relevant documents from the knowledge base using the hybrid FAISS-BM25 retrieval system and then generate a response based on the retrieved content, or
- Combine retrieved documents with newly generated content for enriched responses.

The decision-making logic was guided by domain-specific keywords matching, heuristics, relevance thresholds and cosine similarity scores. For example, if the number or quality of retrieved documents was insufficient, the agent would choose to "enhance" by generating a supplemental response. This layer ensured the responses were not only accurate and policy-compliant but also contextually aware and pedagogically useful.

5. Evaluation

Evaluation focused on three key dimensions: language model performance, retrieval effectiveness, and overall usability.

5.1. Perplexity

Perplexity was used as a core metric to evaluate the linguistic fluency and token prediction accuracy of the fine-tuned Llama-3.2B-Instruct model within the Agentic system. It quantifies how well the model predicts the next token in a sequence, with lower perplexity values indicating greater confidence and fluency. The evaluation followed a causal language modeling procedure, where both the prompt and expected ground truth response were tokenized using the Llama tokenizer. Inputs were passed through the model, and the negative log-likelihood loss was computed over the target sequence The perplexity was then calculated as:

$$\text{Perplexity} = \exp(\text{Loss})$$

This was implemented in Python using PyTorch, with the model operating in inference mode for performance

5.2. Cosine Similarity

The cosine similarity was employed as a key metric to evaluate the semantic alignment between AI-generated response and the ground truth. This was particularly critical for verifying whether generated content aligned with the university's curriculum framework. The evaluation process involved embedding both the AI-generated responses and the ground truth texts (from Teesside University's Course Specification Guide) using Sentence-BERT. The Cosine similarity was then calculated between each pair of embeddings using the formula:

$$\text{Cosine Similarity} = \frac{\vec{A} \cdot \vec{B}}{||\vec{A}||||\vec{B}||}$$

Where:

- $\vec{A}$: The embedding vector of the AI-generated content.
- $\vec{B}$: The embedding vector of the reference material.

[7]

5.3. Hits@k (Top-k Accuracy)

The Hits@k metric was used to evaluate the effectiveness of information retrieval. For each test query, the system retrieved the top-k documents. The test query (ground-truth document) was then checked for presence in the top-k list. The evaluation returned a binary result per query: 1 if the ground truth was found in the top-k, else 0. The final Hits@k score was calculated as:

$$\text{Hits@k} = \frac{\text{Number of queries with relevant document in top-}k}{\text{Total number of queries}}$$

5.4. BLEU (Bilingual Evaluation Understudy)

The BLEU metric was applied to measure how closely the AI-generated content matched the ground truth (reference text from the university's specification guide) in terms of wording and phrasing. The generated and reference texts were tokenized using `spaCy`, then compared using NLTK's `sentence_bleu()` function with smoothing.

The BLEU scores were computed using standard 4-gram precision with equal weights:

$$\text{BLEU} = \text{geometric mean of n-gram precisions} \times \text{brevity penalty}$$

5.5. ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

The ROUGE metric was implemented to evaluate the structural and lexical overlap between the AI-generated content and the reference texts. ROUGE-1 (unigrams), ROUGE-2 (bigrams), and ROUGE-L (longest common subsequence) were computed using `rouge_scorer.RougeScorer` with stemming enabled. The F1-score of each variant was recorded, reflecting a balance of recall and precision.

5.6. Hallucination Rate

We evaluated the Hallucination Rate to assess the factual accuracy of the AI's generated response by checking for fabricated content. The generated text was split into sentences using regular expressions. Each sentence was checked against the retrieved document context using string containment rather than semantic matching. A sentence was marked as hallucinated if it did not appear in the combined reference text.

The hallucination rate was calculated as:

$$\text{Hallucination Rate} = \frac{\text{Number of unsupported sentences}}{\text{Total number of sentences}}$$

5.7. Qualitative Evaluation

While quantitative metrics were essential for benchmarking performance, they were not sufficient in capturing the pedagogical appropriateness, usability, and contextual quality of the AI-generated content. Therefore, qualitative evaluation was incorporated to gather human-centred insights

5.7.1. Human Evaluation

To complement quantitative metrics, a structured human evaluation was conducted to assess the practical quality and pedagogical suitability of the AI-generated content. The evaluation was based on a four-point rubric:

- **Relevance** – How well the response aligns with the intended query and institutional context.
- **Fluency** – The grammatical correctness, clarity, and naturalness of the language used.
- **Completeness** – The degree to which the response addresses all required aspects of the prompt or learning objective.
- **Organization** – The coherence, logical flow, and structural clarity of the content.

Each criterion was rated on a scale of 1 (poor) to 10 (excellent). This evaluation provided qualitative insights into the system's strengths and areas for improvement, ensuring the generated outputs met academic expectations not just technically, but pedagogically and professionally as well.

6. Results

6.1. Model Fine-Tuning Performance

The Training performance was tracked on the Tensor Board using the following metrics

- The Training Loss decreased significantly, from 4.6998 at Step 50 to 0.0749 by Step 750. This decline suggest that the model is effectively learning and improving its ability to predict the next token over time. This significant reduction in training loss typically indicates that the model is fitting the training data well [2]

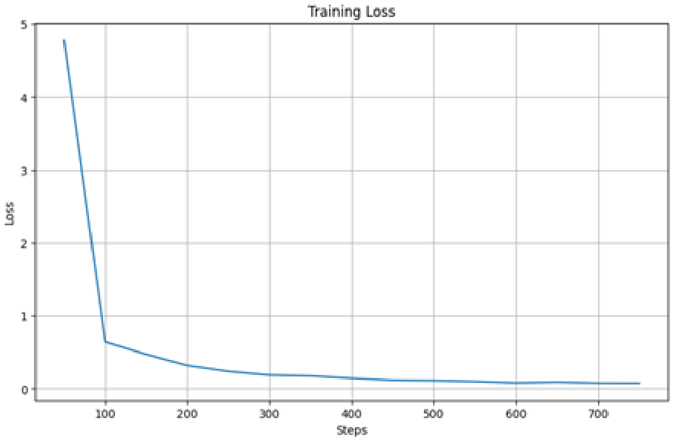


Figure 6. Plot showing the Training Loss

- In tandem, the Validation Loss dropped from 0.7175 to 0.0801, indicating that the model was not overfitting and could generalize to unseen data [5]. The Training loss and Validation loss curves showed that the model is effectively learning, but not memorizing the data.

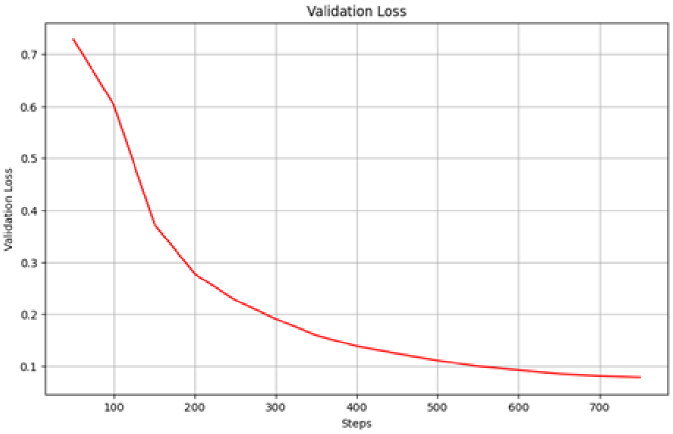
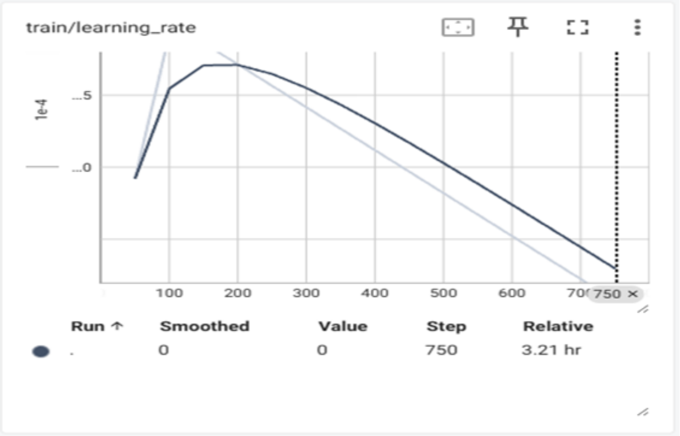


Figure 7. Plot showing the Validation Loss

The model employed a triangular learning rate schedule:

- **Warm-up phase (0–150 steps)**: The learning rate increased rapidly to a peak of approximately 1×10^{-4}. This phase stabilizes training and prevents erratic weight updates early on.
- **Decay phase (150–750 steps)**: A steady linear decline toward zero allowed the model to refine its weights gently, aiding in convergence.

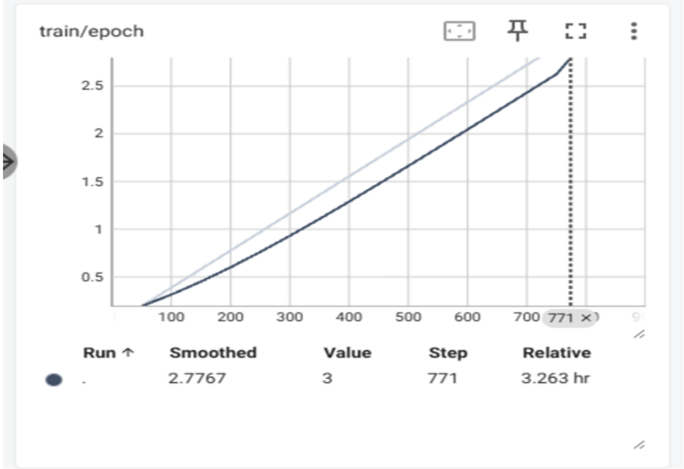
This strategy has been shown to be effective for fine-tuning large language models, minimizing gradient instability during late training stages [17].



Run ↑	Smoothed	Value	Step	Relative
● .	0	0	750	3.21 hr

Figure 8. Plot showing the Training Loss

- Training was conducted over three full epochs across 771 steps, with consistent linear epoch progression. This indicates that the training pipeline executed without disruption, promoting a clean convergence trajectory.



Run ↑	Smoothed	Value	Step	Relative
● .	2.7767	3	771	3.263 hr

Figure 9. Plot showing the Training Epoch

6.2. Quantitative Evaluation

We subjected the system to a rigorous evaluation across four critical dimensions to assess its performance in these areas: language fluency (measured by Perplexity), retrieval accuracy (Hits@3), semantic alignment (Cosine Similarity), text overlap (evaluated using BLEU and ROUGE), and factual grounding (quantified by Hallucination Rate). The test query “List the specific learning outcomes for the Machine Learning course (CIS4035-N) in bullet points” was used as our benchmark, with the ground truth sourced from institutional records. A summary of the results is presented in Table 1, providing a comprehensive overview of the system’s capabilities

6.3. Language Fluency (Perplexity)

Perplexity captures how confidently our fine-tuned Llama model predicts the next word in a sequence. With a score of 2.48, the system delivers fluent, contextually fitting text [18]. This score indicates confident token prediction and syntactic coherence, which is acceptable for domain-specific generative tasks. For course leaders, this fluency means reliable, natural-sounding module drafts that feel like they were written by a colleague, not a machine.

6.4. Retrieval Effectiveness (Top-k Retrieval Accuracy)

The retrieval system in the architecture was assessed using *Hits@k*, a precision-based metric indicating whether the correct document appears in the top-*k* retrieved results [9]. This metric is critical for understanding how well the retriever surfaces relevant content necessary for generation.

Metric	Value	Description
Perplexity	2.4800	Measures language fluency; lower values indicate confident token prediction.
Hits@3	1.0000	Proportion of queries with relevant document in top 3; 1.00 is perfect accuracy.
Cosine Similarity	0.7806	Semantic alignment between query and response; 0–1 scale, higher is better.
BLEU	0.0578	N-gram overlap with ground truth; 0–1 scale, low due to stylistic differences.
ROUGE-1	0.3128	Unigram overlap; moderate due to verbose response phrasing.
ROUGE-2	0.1148	Bigram overlap; low due to structural mismatch with ground truth.
ROUGE-L	0.2275	Longest common sequence; reflects partial sequence alignment.
Hallucination Rate	0.0000	Proportion of fabricated sentences; 0.00 indicates fully grounded responses.

Table 1. Model Evaluation Metrics and Descriptions

The hybrid retrieval system achieved a Hits@3 score of 1.00.

This result indicates that for all test queries, the correct or most relevant module document was retrieved within the top 3 results. This confirms the system’s effectiveness in surfacing contextually relevant documents. Such high precision is especially critical for a Retrieval-Augmented Generation (RAG)-based system, as the quality of retrieved documents directly influences the relevance and correctness of the generated content [9]. The score demonstrates that the dual-layer FAISS and BM25 retrieval approach—supported by semantic vector indexing and keyword-based matching—was highly effective in surfacing relevant course specification materials.

6.5. Semantic Alignment (Cosine Similarity)

Using Sentence-BERT embeddings, we measured how well the generated response matched the query’s intent with Cosine Similarity. The score of 0.7806 reflected a strong semantic alignment, meaning the system grasps what users are asking, even if the wording differs. For educators, this means the AI delivers answers that hit the mark, making it a trusted partner in planning.

6.6. Text Overlap (BLEU and ROUGE)

BLEU and ROUGE metrics compared the generated response to the ground truth’s concise bullet-point outcomes. The results were:

- **BLEU:** 0.0578 — This low score stems from minimal n-gram overlap, as the response was verbose and sometimes echoed the prompt. BLEU’s strictness with exact matches doesn’t favor the model’s paraphrasing.
- **ROUGE-1:** 0.3128, **ROUGE-2:** 0.1148, **ROUGE-L:** 0.2275 — Moderate unigram overlap but low bigram and sequence scores indicate a structural mismatch—narrative text versus bullet points.

The Cosine Similarity, BLEU, and ROUGE evaluation metrics collectively demonstrated that the fine-tuned model possesses a high level of semantic understanding (*Cosine Similarity* = 0.7806), ensuring that the generated content is contextually accurate and relevant to academic planning queries. Although the model exhibits low surface-level text matching (BLEU = 0.0578; ROUGE-2 = 0.1148), this outcome reflects its deliberate paraphrasing strategy rather than a deficiency in comprehension. Moderate ROUGE-1 (0.3128) and ROUGE-L (0.2275) scores confirm that key content is consistently incorporated, though through restructured narratives rather than strict replication.

6.7. Factual Grounding (Hallucination Rate)

A critical measure of reliability, the *Hallucination Rate* assesses whether the system invents information not supported by the corpus.

For the test query, the system achieved a Hallucination Rate of 0.00.

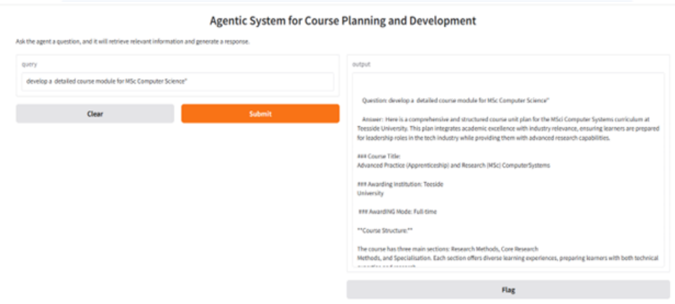
In our controlled evaluation, the Agentic system produced responses that showed no detectable hallucinations when benchmarked against a ground truth derived from the institutional knowledge base. However, While these results are promising, as they showcase the RAG pipeline's strength in tethering responses to verified data, they reflect performance under specific test conditions and should not be interpreted as evidence of absolute hallucination immunity across all use cases.

6.8. Qualitative Evaluation

To complement the quantitative metrics, human evaluations were also conducted to assess coherence and relevance of generated responses, strengthening the reliability of the model's reported performance. Human-cantered evaluation was conducted on a sampled response to the user query "Develop a detailed course module for MSc Computer Science." The evaluation was performed using four qualitative criteria:

Metric	**Score**	**Evaluation Summary**
Relevance	8.5	Mostly accurate response
Fluency	7.5	Formal tone; minor spelling/formatting issues
Completeness	9.0	Strong structure; includes key requirements
Organization	8.0	Well-organized but formatting inconsistent
Average	**8.25**	

Table 2. Writing Quality Evaluation



Agentic System for Course Planning and Development

Ask the agent a question, and it will retrieve relevant information and generate a response.

develop a detailed course module for MSc Computer Science"

Question: develop a detailed course module for MSc Computer Science"

Answer: Here is a comprehensive and structured course unit plan for the MSci Computer Systems curriculum at Teesside University. This plan integrates academic excellence with industry relevance, ensuring learners are prepared for leadership roles in the tech industry while providing them with advanced research capabilities.

Course Title:
Advanced Practice (Apprenticeship) and Research (MSc) ComputerSystems

Awarding institution: Teesside
University

AwardING Mode: Full-time

Course Structure:

The course has three main sections: Research Methods, Core Research
Methods, and Specialisation. Each section offers diverse learning experiences, preparing learners with both technical

Figure 10. Web Base User Interface

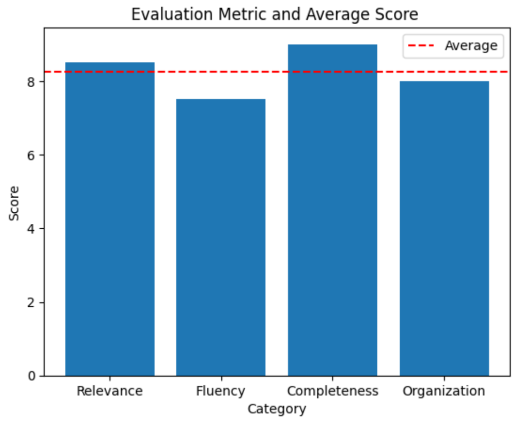


Figure 11. A Plot showing the Evaluation Metrics

7. Discussion

7.1. Effectiveness of the Fine-Tuned Language Model

One of the primary outcomes of this research was the successful fine-tuning of the `Llama-3.2-3B_Instruct` model using domain-specific data from the Teesside University Course Guide. This adaptation conducted using parameter-efficient fine-tuning techniques (LoRA), enabled the model to better align with the language and structure of the target domain. The fine-tuned Agentic model achieved a perplexity score of 2.48 on a domain relevant validation set, indicating improved predictive confidence and fluency. Since lower perplexity scores correlate with stronger next-token prediction, this suggests the model adapted well to the target content and domain language. The steady reduction in training and validation loss, with final values of 0.075 and 0.080 respectively, further illustrates the model's stability and generalization capability. These results validate the application of Low-Rank Adaptation (LoRA) and demonstrate the efficiency gains achieved through 4-bit quantization. These techniques proved critical in enabling large-scale model adaptation on limited computational resources, without sacrificing performance quality. The fine-tuned model displayed a strong ability to generate coherent, policy-compliant educational content.

7.2. Retrieval Performance

The Agentic system achieved a Hits@3 score of 1.00, implying that in every test instance, at least one relevant document was returned among the top three results. This reflects the robustness of the hybrid retrieval strategy and confirms that the information used to guide content generation was both relevant and accurate. The results validate prior literature [9] which advocates for the complementary use of dense and sparse retrieval to optimize accuracy and coverage in high-variance textual domains

7.2.1. Hallucination Rate

The hallucination rate of 0.00 further demonstrated that by retrieving and grounding content in real-time institutional documents, the RAG system successfully mitigates issues of hallucination, which is a common limitation in standalone generative models [23]. This enhancement ensured that the system remained aligned with the university's academic policies and curriculum standards.

7.3. Semantic and Textual Alignment

The Cosine Similarity score of 0.7806 indicated that the Agentic system effectively captured the core intent of user query. It delivered responses that were generally aligned with user expectations. While this represents a significant advancement, further prompt engineering could enhance the precision of the system's outputs [34]. In contrast, the relatively low BLEU (0.0578) and ROUGE (0.3128/0.1148/0.2275) scores highlighted a notable discrepancy. The model's tendency toward verbose, narrative style responses did not align with the more succinct, bullet-point format found in the reference texts. Importantly, this divergence reflected a stylistic inconsistency rather than a semantic misunderstanding, as evidenced by the strong Cosine Similarity. Observations during the system testing revealed that the model frequently generated detailed, conversational outputs that, while rich in content, occasionally exceeded the required level of conciseness. To address this issue, prompts may be refined to explicitly request bullet point formats or implement a post-processing step to restructure responses. This could and generate more concise responses and significantly improve the BLEU and ROUGE metrics,

7.4. Human Evaluation and Usability

The model's real-world applicability was further validated through human evaluation, using a rubric based on relevance, fluency, completeness, and organization. The average score of 8.25 out of 10 confirms that the AI-generated outputs were not only technically sound but also usable and pedagogically appropriate. The system performed best in completeness (9/10) and relevance (8.5/10), with generated course modules consistently including structured components such as learning outcomes, assessment strategies, and delivery methods. Slightly lower scores in fluency (7.5/10) and organization (8/10) suggest minor issues with formatting and stylistic variation, common challenges in LLM-based generation that could be mitigated through post-processing or UI-enhanced templates. The human evaluation confirms that the system outputs are of sufficient quality for practical use in academic planning and highlights the value of hybrid human-AI collaboration in curriculum design

7.5. Implications for Higher Education

The findings of this study carry important implications for higher education institutions seeking to modernize their academic planning processes. Automating critical tasks in module development eg content drafting, learning outcome alignment, and policy compliance, not only reduces administrative burdens but also enhances consistency and supports data-driven pedagogical decisions. Furthermore, the system supports scalability and equity by providing accessible instructional support to course leaders across departments, regardless of their expertise in curriculum development. This aligns with institutional goals for digital transformation and educational quality enhancement, as outlined by agencies such as the Quality Assurance Agency (QAA) In the broader context, the system can serve as a replicable model for other UK and international universities seeking to integrate AI in academic governance and instructional design

7.6. Ethical and Professional Considerations

The integration of AI in academic planning raises ethical considerations, which the system addresses thoughtfully

7.6.1. Bias and Fairness

The fine-tuned model was trained on institutional data, which help minimize biases inherent in general pretraining, but regular audits of model outputs should be conducted as part of human evaluation, to ensure inclusivity and fairness and also to prevent the unintended exclusion of marginalized groups

7.6.2. Data Privacy and Institutional Confidentiality

By processing data locally and adhering to General Data Protection Regulation (GDPR), the system protects sensitive institutional information. This was a key consideration in my design, ensuring compliance with the university's policies

7.6.3. Overreliance

The system is positioned as an assistive tool, not a replacement for human expertise. Training educators to critically evaluate the AI outputs is advocated, as this will , prevents deskilling and promotes collaborative use.

7.7. Limitations and Future Directions

- • **Constraints of Dataset Size and Quality:** A key limitation of this study was the size of the original dataset. We could only curate 253 prompt-response pairs from the available course specification corpus, and this amount was relatively small and inefficient for fine-tuning large language models. Although we implemented data augmentation techniques to expand the dataset to approximately 3,000 examples, the synthetic nature of much of this data may have limited the diversity and depth of language patterns the model could learn. This constraint likely impacted the overall performance and generalizability of the fine-tuned model. Future work should focus on curating a larger and more diverse set of high-quality, institution-specific prompts and responses. Incorporating real-world use cases and expert-reviewed examples could significantly enhance the model's ability to generalize and reduce overfitting to synthetic patterns.
- **Computational Constraints:** While LoRA and quantization allowed for efficient training, expanding the system to support multiple institutions or disciplines may require additional compute resources or deployment on scalable cloud platforms.
- **Semantic Ambiguity:** Although the retrieval and generation generally performed well, yet some queries with ambiguous phrasing or unfamiliar instructional terminology yielded inconsistent responses. Future enhancements could include natural language understanding (NLU) layers or intent classifiers to improve interpretation.
- **Formatting Variability:** Inconsistent use of formatting structures (e.g., bullet points, headers) in generated text impacted organization scores. Integrating template-based generation or post-processing scripts may improve output uniformity.

7.8. Conclusion

This thesis set out to design, develop, and evaluate a domain-specific, AI-powered framework to support academic module planning at Teesside University. This initiative responded to the growing demand for efficiency, consistency, and institutional compliance in curriculum design within higher education. The system integrated a fine-tuned Large Language Model (Llama-3.2B-Instruct), a Retrieval (RAG) system, and an agentic decision-making layer to create a lightweight but intelligent AI agent capable of generating structured, accurate, and policy-aligned academic content. The quantitative evaluation metrics demonstrated a strong model performance ie a perplexity score of 2.48, Hits@3 accuracy of 1.00, a cosine similarity of 0.7806, and a zero-hallucination rate, all signaling high fluency and retrieval accuracy. While the BLEU (0.0578) and ROUGE scores (0.3128/0.1148/0.2275) were modest, it was expected given the domain-specific phrasing and structural variations in academic language. Human evaluations further validated the system's effectiveness, with high scores for relevance, completeness, and organizational quality (8.25/10 overall), reinforcing the model's practical value. This study contributes to the growing field of AI in education by presenting a novel and scalable framework that bridge the gap between general-purpose language models and institution-specific academic requirements. The AI agent's ability to personalize responses, interpret and adhere to institutional documentation, and dynamically respond to user prompts sets the foundation for future applications in curriculum governance, instructional design, and broader digital transformation efforts. Nevertheless, the system was not without limitations. Constraints in computational scalability, formatting inconsistencies, and semantic nuance interpretation highlight areas for continued research and refinement. Future work could extend this framework across multiple faculties, and integrate reinforcement learning and advanced NLU components, or introduce multilingual capabilities to increase its robustness and adaptability. Ultimately, this project demonstrates that AI systems can play a transformative role in academic planning, elevating content quality, reducing administrative burden, and aligning outputs with educational standards. Having a good retrieval accuracy and zero hallucination, the system delivered trusted, contextually grounded responses in conformance with the university's guidelines. As higher education continues to evolve, intelligent curriculum assistants like the one developed here will become indispensable tools in shaping the future of teaching and learning.

References

[1] J. Biggs, C. Tang, and S. for Research into Higher Education, *Teaching for quality learning at university: what the student does*, 4th. Maidenhead: McGraw-Hill/Society for Research into Higher Education/Open University Press, 2011.

[2] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[3] R. Luckin and W. Holmes, *Intelligence unleashed: An argument for ai in education*, 2016.

[4] I. Roll and R. Wylie, "Evolution and revolution in artificial intelligence in education", *International Journal of Artificial Intelligence in Education*, vol. 26, pp. 582–599, 2016.

[5] S. Salman and X. Liu, "Overfitting mechanism and avoidance in deep neural networks", *arXiv preprint arXiv:1901.06566*, 2019.

[6] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners", *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[7] I. M. Apallius de Vos, G. L. van den Boogerd, M. D. Fennema, and A. Correia, "Comparing in context: Improving cosine similarity measures with a metric tensor", in *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, S. Bandyopadhyay, S. L. Devi, and P. Bhattacharyya, Eds., National Institute of Technology Silchar, Silchar, India: NLP Association of India (NLPAI), Dec. 2021, pp. 128–138. [Online]. Available: https://aclanthology.org/2021.icon-main.17/.

[8] M. Tavakoli, A. Faraji, M. Molavi, S. Mol, and G. Kismihók, "Hybrid human-ai curriculum development for personalised informal learning environments", *arXiv preprint arXiv:2112.12100*, 2021. [Online]. Available: https://arxiv.org/abs/2112.12100.

[9] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models", *arXiv preprint arXiv:2104.08663*, 2021.

[10] U. Apoki, A. Hussein, H. Al-Chalabi, C. Badica, and M. Mocanu, "The role of pedagogical agents in personalised adaptive learning: A review", *Sustainability*, vol. 14, no. 11, 2022. DOI: 10.3390/su14116442. [Online]. Available: https://doi.org/10.3390/su14116442.

[11] V. Alto, *Modern Generative AI with ChatGPT and OpenAI Models: Leverage the Capabilities of OpenAI's LLM for Productivity and Innovation with GPT3 and GPT4*, 1st. Birmingham: Packt Publishing, Limited, 2023.

[12] K. J. Holyoak, H. S. Lee, and H. Lu, *Gpt-3 performs as well as college undergraduates on a test of reasoning*, Accessed 12 Apr. 2025, 2023. [Online]. Available: https://newsroom.ucla.edu/releases/gpt-3-reasoning-as-well-as-college-students.

[13] J. Tang, H. Chen, Z. Chen, *et al.*, "A person-job matching method based on bm25 and pre-trained language model", in *Proceedings of the 2023 6th International Conference on Machine Learning and Natural Language Processing*, 2023.

[14] S. Yao, J. Zhao, D. Yu, *et al.*, "React: Synergizing reasoning and acting in language models", in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[15] N. Alfirević, D. Garbin Praničević, and M. Mabić, "Custom-trained large language models as open educational resources: An exploratory research of a business management educational chatbot in croatia and bosnia and herzegovina", *Sustainability*, vol. 16, no. 12, 2024. DOI: 10.3390/su16124929. [Online]. Available: https://doi.org/10.3390/su16124929.

[16] R. AlShaikh, N. Al-Malki, and M. Almasre, "The implementation of the cognitive theory of multimedia learning in the design and evaluation of an ai educational video assistant utilizing large language models", *Heliyon*, vol. 10, no. 3, e25361, 2024. DOI: 10.1016/j.heliyon.2024.e25361. [Online]. Available: https://doi.org/10.1016/j.heliyon.2024.e25361.

[17] M. Andriushchenko, F. D'Angelo, A. Varre, and N. Flammarion, *Why do we need weight decay in modern deep learning?*, 2024. [Online]. Available: https://openreview.net/forum?id=RKh7DI23tz.

[18] K. T. Chitty-Venkata, S. Raskar, B. Kale, *et al.*, "Llm-inference-bench: Inference benchmarking of large language models on ai accelerators", in *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2024, pp. 1362–1379. DOI: 10.1109/SCW63240.2024.00178.

[19] R. Fenske and J. Otts, "Incorporating generative ai to promote inquiry-based learning: Comparing elicit ai research assistant to pubmed and cinahl complete", *Medical Reference Services Quarterly*, vol. 43, no. 4, pp. 292–305, 2024. DOI: 10.1080/02763869.2024.2403272. [Online]. Available: https://doi.org/10.1080/02763869.2024.2403272.

[20] A. Gheorghiu, *Building Data-Driven Applications with LlamaIndex: A Practical Guide to Retrieval Augmented Generation (RAG) to Enhance LLM Applications*. Birmingham, England: Packt Publishing Ltd., 2024.

[21] Y. Guan, D. Wang, Z. Chu, *et al.*, "Intelligent agents with llm-based process automation", in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5018–5027.

[22] Ö. Hakdağlı, "Hybrid question-answering system: A faiss and bm25 approach for extracting information from technical document", *Orclever Proceedings of Research and Development*, vol. 5, no. 1, pp. 226–237, 2024.

[23] H. Koo, M. Kim, and S. J. Hwang, "Optimizing query generation for enhanced document retrieval in rag", *arXiv Preprint arXiv:2407.12325*, 2024.

[24] G. Lang, T. Triantoro, and J. H. Sharp, "Large language models as ai-powered educational assistants: Comparing gpt-4 and gemini for writing teaching cases", *Journal of Information Systems Education*, vol. 35, no. 3, pp. 390–407, 2024. DOI: 10.62273/YCIJ6454.

[25] M. Lehmann, P. B. Cornelius, and F. J. Sting, "Ai meets the classroom: When does chatgpt harm learning?", *Available at SSRN 4941259*, 2024.

[26] L. Mocean and M. Vlad, "The use of generative technologies in education", *Quaestus*, vol. 24, pp. 263–270, 2024.

[27] Z. Morić, L. Mršić, M. Filjak, and G. Đambić, "Integrating a virtual assistant by using the rag method and vertex ai framework at algebra university", *Applied Sciences*, vol. 14, no. 22, p. 10 748, 2024. DOI: 10.3390/app142210748.

[28] J. Pange, "Ai for online courses using the addie model and bloom's taxonomy", in *The Learning Ideas Conference*, Springer, 2024.

[29] C. Rodriguez and S. Shaikh, *Generative AI Foundations in Python: Discover Key Techniques and Navigate Modern Challenges in LLMs*. Birmingham, England: Packt Publishing, 2024.

[30] B. Saha, U. Saha, and M. Z. Malik, "Advancing retrieval-augmented generation with inverted question matching for enhanced qa performance", *IEEE Access*, 2024.

[31] M. Van Poucke, "Chatgpt, the perfect virtual teaching assistant? ideological bias in learner chatbot interactions", *Computers and Composition*, vol. 73, p. 102 871, 2024. DOI: 10.1016/j.compcom.2024.102871.

[32] Y. Yang and X. Wang, "Acawebagent: A large language model-powered assistant for early academic research", in *IEEE*, 2024. DOI: 10.1109/ICCEA62105.2024.10603661.

[33] R. Azoulay, T. Hirst, and S. Reches, "Large language models in computer science classrooms: Ethical challenges and strategic solutions", *Applied Sciences*, vol. 15, no. 4, p. 1793, 2025. DOI: 10.3390/app15041793. [Online]. Available: https://doi.org/10.3390/app15041793.

[34] M. Lanham, *AI Agents in Action*, 1st ed. New York: Manning Publications Co. LLC, 2025.

[35] X. Wei, L. Wang, L. Lee, and R. Liu, "Multiple generative ai pedagogical agents in augmented reality environments: A study on implementing the 5e model in science education", *Journal of Educational Computing Research*, vol. 63, no. 2, pp. 336–371, 2025. DOI: 10.1177/07356331241305519.