Personalized Education Workflow Report

May 26, 2025

Abstract

This report provides a comprehensive, step-by-step account of the methodology used to construct and evaluate a personalized education pipeline. It covers synthetic profile generation, Q&A construction, model fine-tuning, and comparative evaluation against a retrieval-augmented baseline. Emphasis is placed on design decisions and practical considerations at each stage.

Contents

1 Introduction

In this project, we aimed to simulate individual student characteristics and assess adaptive learning via large language models. Starting with synthetic profile creation, we generated contextualized Q&A pairs, fine-tuned an LLM for deeper personalization, and benchmarked its performance against a RAG approach. Detailed reasoning accompanies every choice, highlighting lessons learned and best practices.

2 Workflow Overview

The overall personalized-education pipeline follows four sequential stages:

1. **Profile Generation.** Synthetic student profiles are generated (e.g. JSON via Gemini 1.5 Flash) and validated against our Pydantic schema.

2. **Q&A Pair Generation.** For each profile, 50 question–answer pairs are produced with aim of ensuring that the model can get a detialed understanding of the personality of each profile. Each question explicitly embeds the phrase *"student with ID <id>"* (e.g. `What is student with ID 0's name?`) so the model can consistently link queries to the correct student.

3. **Model Fine-Tuning.** The finetuning dataset consists of these ID-tagged Q&A pairs, teaching the model to internalize and recall each profile's details.

4. **Personalized Inference.** For a selected set of science questions (Class 7–10 level), both the fine-tuned and RAG models are prompted in a way that:

   - Specifies the student's name and ID.
   - Provides the full profile (for RAG) or relies on the model's trained weights (for FT).
   - Instructs: "You are an AI tutor for {`student_name`}, whose id is {`student_id`}, designed to answer science questions at the Class 7–10 level in a way that's tailored and personalized to each student's individual profile and interests. Use only the information provided in the profile to shape your explanations—drawing on the student's age, learning style, interests, strengths, and struggles—to make your answers clear, engaging, accessible, and easily understandable to the student. Answer in 2–4 lines **only** and use simple language, concrete

examples, and relate explanations to their interests or learning style where possible."

3 Profile Generation

3.1 Overview

An initial batch of 50 student profiles was produced using Gemini 1.5 Flash as a proof of concept. Each profile encapsulated demographic, academic, and behavioral traits, formatted to align with a predefined Pydantic schema for seamless downstream integration.

3.2 Schema Definition

The schema specified a variety of fields, including:

- `name`, `age`, `class_level`, `syllabus`
- Multi-valued single-word lists: `interests`, `personality`, `hobbies`, etc.
- Study schedule times: `study_routine_start`, `study_routine_end`
- Academic performance dictionaries: `marks_last_year_final`, `marks_last_internals`
- Qualitative fields: `emotional_traits`, `group_behavior`, etc.

All entries were constrained to realistic ranges and formats, ensuring valid JSON output directly parsable by Pydantic without manual cleaning.

3.3 Prompting Strategy and Safeguards

To generate each profile, we employed a LangChain-based prompt template that included:

- A reminder of the target schema and type constraints.
- A dynamically updated memory block listing previously created profiles to prevent duplication.
- Instructions enforcing plain JSON output with no extraneous text, arrays, or formatting wrappers.

This careful design minimized parsing errors and maintained diversity across profiles, while keeping token usage in check by sending only one profile context at a time.

4 Question and Answer Construction

4.1 Design Principles

For each profile, 50 Q&A pairs were crafted to elicit information about learning preferences, motivations, and challenges. Questions were organized into categories such as:

- Basic Information

- Academic Performance
- Emotional and Social Aspects
- Teacher and Student Perspectives

ID-Tagged Questions. To anchor each Q&A pair to a specific profile, every question template incorporates the phrase:

```
student with ID <id>
```

For example:

```
What is student with ID 0's name?
```

This constant identifier helps the model associate every answer with the correct synthetic profile.

Purpose of Q&A Generation. The goal of this stage is to create question–answer pairs that teach the fine-tuned model each student's unique personality traits and prevent cross-profile contamination. By tagging every question with the student's ID, we ensure the model learns to associate each response with the correct individual. This ID anchoring allows the model to build a deep, id-linked understanding of each synthetic profile.

4.2 Example Questions

One exemplar per category:

- Basic Information: 'What is student with id 0's age?"
- Academic Performance: ' What were student with id 0's average marks last year?"
- Emotional and Social Aspects: 'What motivates student with id 0?"

4.3 Implementation Details

LangChain orchestrated the iterative prompt calls, injecting each student's profile context and collecting structured JSON responses. Automated pydantic validation ensured that all Q&A outputs adhered to the expected schema.

5 Language Model Fine-Tuning

5.1 Model and Environment

Llama 3.2 3B (instruct) was chosen for fine-tuning with the above generated Q&A's in this initial phase. Training occurred on a free tier Colab T4 GPU using the UnsLoTh SFTTrainer interface.

5.2 Hyperparameter Configuration

Key training parameters:

- *Batch size* per device: 2, *Gradient accumulation*: 4
- *Learning rate*: 1×10^{-4}
- *Epochs*: 10, *Warmup steps*: 5
- *Optimizer*: 8-bit AdamW with linear scheduler
- *Precision*: Automatic fp16/bf16 selection
- *Random seed*: 3407

An incremental fine-tuning approach—starting with 1 profile, then 5, helped verify end-to-end stability before scaling to the full dataset.

6 Personalization Evaluation

6.1 Selected Questions

Four CBSE Class 7-10 science questions evaluated personalization:

1. Explain the human digestive system.
2. How does reproduction occur in plants?
3. What are acids and bases? Provide two examples each.
4. What is the function of the respiratory system in humans?

6.2 Inference Protocols

A unified prompt template guided both finetuned and RAG-based inference. For the RAG baseline, the full profile text was appended; for the finetuned model, only the core question context was supplied.

6.3 Automated Scoring

We leveraged Gemini 2.0 Flash as an automated judge, prompting it to assign JSON-formatted scores (1-10) for personalization and relevance.

7 Results and Analysis

7.1 Finetuned Model

- Personalization Score: 6.2
- Relevance Score: 8.64

7.2 RAG Baseline

- Personalization Score: 7.24
- Relevance Score: 8.64

7.3 Statistical Analysis of RAG vs. Fine-Tuning Personalization

We compared per-student personalization and relevance scores under two methods (RAG vs. fine-tuning) using paired samples. First, we assessed normality of the paired differences with the Shapiro–Wilk test, which indicated non-normality for both personalization ($p = 3.30 \times 10^{-5}$) and relevance ($p = 2.40 \times 10^{-10}$). Accordingly, we applied the Wilcoxon signed-rank test. Effect sizes were quantified via Cohen's d on the differences defined as $\Delta = \text{FineTune} - \text{RAG}$, and precision was assessed via 95% bootstrap confidence intervals (5,000 resamples).

Table 1: Wilcoxon Test and Effect Sizes for Personalization and Relevance

Metric	W	p	**Cohen's** d	**95% CI for** d	**Interpretation**
Personalization	71.50	$< .001$	-0.58	$[-0.94, -0.28]$	Significant; medium–large effect (RAG > FT)
Relevance	135.00	.261	0.00	$[-0.45, 0.21]$	Non-significant; negligible effect

Results. The Wilcoxon test revealed a significant difference in personalization, $W = 71.50$, $p < .001$, with Cohen's $d = -0.58$ (95% CI [–0.94, –0.28]) indicating a medium-to-large effect favoring RAG. For relevance, $W = 135.00$, $p = 0.261$, and $d = 0.00$ (95% CI [–0.45, 0.21]) showed no clear difference.

8 Lessons Learned

- **Modular Prompt Design**: Isolating schema instructions and memory safeguards improved maintainability.
- **Incremental Testing**: Small-scale fine-tuning catches integration issues early.
- **Prompt Hygiene**: Enforcing strict JSON output reduces parsing errors.

9 Conclusion

This workflow establishes a robust process for personalized education research. Future directions include expanding profile sets, iterating on question diversity, and integrating live student feedback loops.

The results suggest that RAG outperforms fine-tuning in personalization with practical significance, while both methods perform comparably in relevance. Future work should replicate this analysis on larger cohorts to confirm robustness.

10 Way Forward

To extend and strengthen this personalized education framework, we propose the following next steps:

- **Scale Profile Generation**: Increase synthetic student profiles to 1000 to capture a wider diversity of demographics and learning behaviors.
- **Augment Question Coverage**: Generate at least 10 disciplinary questions per grade (7–10), ensuring representation across science, mathematics, and language arts for deeper evaluation.
- **Iterative Feedback Loop**: Incorporate real student feedback by deploying a pilot study that collects user ratings on model responses, informing subsequent fine-tuning rounds.

10.1 Estimated Token Usage

To forecast total token consumption for 1 000 profiles and 40 questions each (10 per grade), we ran pilot measurements at each stage and extrapolated as follows.

10.1.1 Profile Generation(Gemini 1.5 flash)

- **Pilot measurement**:Using the Gemini 1.5-Flash pilot script: each initial request consumed about 2 071 tokens, and each additional stored profile added roughly 583 tokens to the next request. Each generated profile returned about 548 tokens.
- **Extrapolation**: summing across 1 000 sequential calls yields approximately **293.83 million tokens**.

10.1.2 Q&A Pair Generation(Gemini 1.5 flash)

- **Pilot measurement**:With batch size = 10 questions per call: consumed about 1 552 tokens per batch.
- **Total calls**: five batches per student for 1 000 students (5 000 calls).
- **Estimate**: $5\,000 \times 1\,552 \approx$ **7.76 million tokens**.

10.1.3 Model Fine-Tuning(Llama 3.2 3B Instruct via unsloth)

- **Pilot measurement**: processing 50 profiles produced about 104 032 tokens of training text per epoch.
- **Scaling**: 1 000 profiles scale, $104\,032 \times (1000/50) = 2\,080\,640$ tokens per epoch.
- **Full training**: for 10 epochs, this amounts to **20.81 million tokens**.

10.1.4 Inference on 40 Questions per Profile

Fine-Tuned Model(Llama 3.2 3B Instruct)

- **Pilot measurement**: answering four questions consumed about 2 420 tokens total, or 605 tokens per question on average.
- **Total calls**: 40 questions × 1 000 students = 40 000 calls.
- **Estimate**: $40\,000 \times 605 = \mathbf{24.20\ million\ tokens}$.

RAG Baseline(Llama 3.2 3B Instruct)

- **Pilot measurement**: answering four questions consumed about 3 999 tokens total, or 1 000 tokens per question on average.
- **Total calls**: 40 000.
- **Estimate**: $40\,000 \times 1\,000 = \mathbf{39.99\ million\ tokens}$.

10.1.5 Automated Scoring(Gemini 2.0 flash)

- We used Gemini 2.0-Flash as judge, 4 Q&A pairs per call:
- **Pilot measurement**: evaluating four Q&A pairs consumed about 2 226 tokens per scoring call.
- **Total calls**: 10 batches of four pairs per student for 1 000 students = 10 000 calls.
- **Single-method estimate**: $10\,000 \times 2\,226 = \mathbf{22.26\ million\ tokens}$.
- **Both methods (fine-tuned + RAG)**: $2 \times 22.26 = \mathbf{44.52\ million\ tokens}$.

10.1.6 Grand Total and Safety Margin

Profile gen.	293.83 M
Q&A gen.	7.76 M
Fine-tuning	20.81 M
FT inference	24.20 M
RAG inference	39.99 M
Scoring (both)	44.52 M
Subtotal	431.11 million
+10% margin	≈ 474.22 million

Therefore, we budget approximately **475 million tokens** for the full expanded workflow.