Integrating Sustainability and Climate Resilience into STEM with Scaffold AI: System Status Report

Scaffold AI Team

September 2025

Abstract

This report presents the current status of *Scaffold AI*, a retrieval-augmented generation (RAG) system designed to assist educators in integrating sustainability and climate resilience into STEM curricula. We document the architecture, dataset processing pipeline, model configurations, evaluation results, and limitations. The system combines sentence embedding retrieval with cross-encoder re-ranking and a compact language model for grounded generation. We provide a formal description of the workflow and list the exact parameter settings used in this release to support reproducibility.

1 Introduction

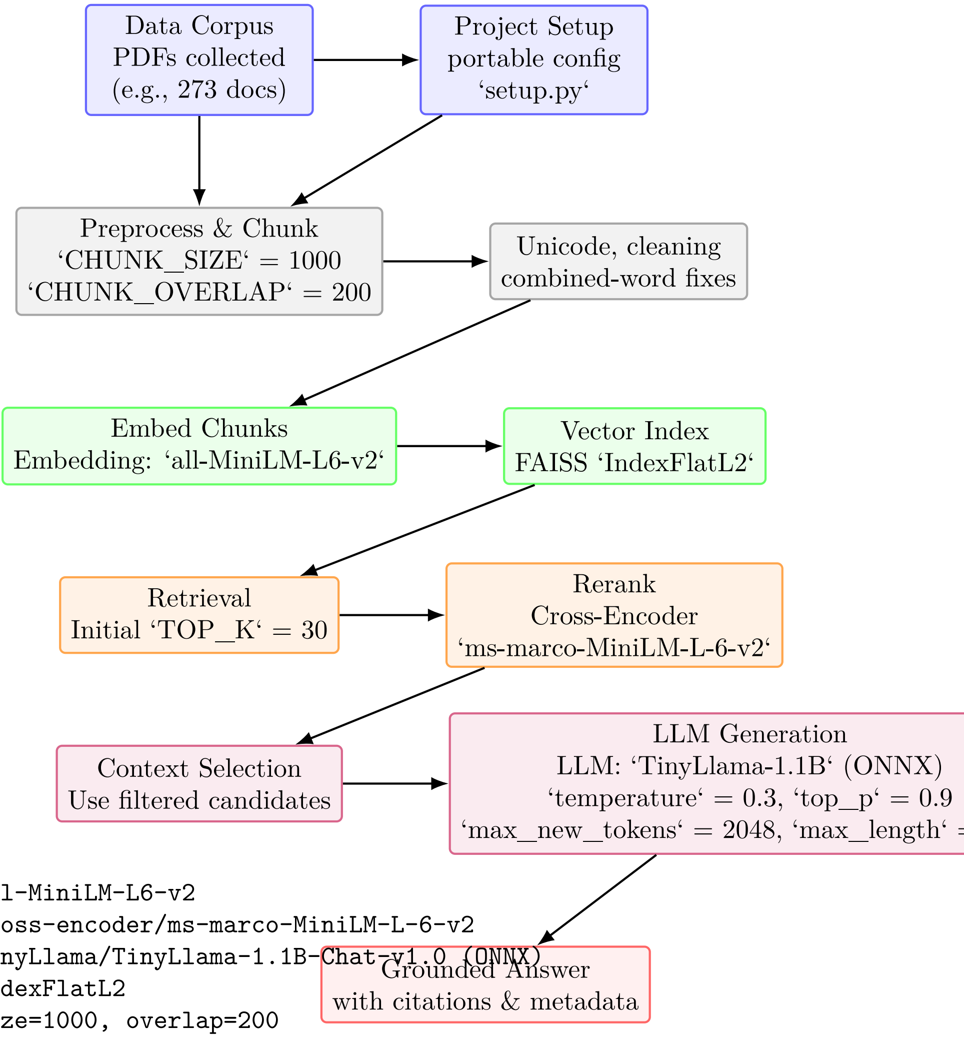
Sustainability and climate resilience are increasingly central to engineering education. Educators face challenges curating appropriate literature and generating course-aligned learning activities at scale. *Scaffold AI* operationalizes a transparent RAG stack that retrieves scholarly materials and produces grounded recommendations to support authentic curriculum design. The current release emphasizes portability, source transparency, and stable generation.

2 System Overview

The system follows a standard RAG pipeline with explicit preprocessing, embedding, indexing, retrieval, re-ranking, and controlled text generation. All paths and defaults are centrally maintained in ‘scaffold_core/config.py‘ to ensure reproducibility across environments.

Workflow and Settings Figure

The following TikZ figure depicts the end-to-end workflow and all key hyperparameters for this version. No external images are used.



Settings summary:

`Embedding`	`all-MiniLM-L6-v2`
`Cross-Encoder`	`cross-encoder/ms-marco-MiniLM-L-6-v2`
`LLM`	`TinyLlama/TinyLlama-1.1B-Chat-v1.0 (ONNX)`
`FAISS`	`IndexFlatL2`
`Chunking`	`size=1000, overlap=200`
`Retrieval`	`initial TOP_K=30, final candidates=all filtered`
`Generation`	`temperature=0.3, top_p=0.9, max_new_tokens=2048, max_length=4096`

3 Methods

3.1 Document Processing

Corpus PDFs are normalized and chunked with page-aware boundaries. Unicode analysis and combined-word postprocessing improve downstream embedding quality. Parameters: ‘CHUNK_SIZE=1000‘, ‘CHUNK_OVERLAP=200‘.

3.2 Embedding and Indexing

Text chunks are encoded with ‘all-MiniLM-L6-v2‘ (384-dim) and stored in a FAISS ‘IndexFlatL2‘ structure. Metadata is preserved for citation.

3.3 Retrieval and Re-ranking

Initial retrieval returns up to ‘TOP_K=30‘ candidates via FAISS. Cross-encoder ‘ms-marco-MiniLM-L-6-v2‘ re-ranks candidates; contextual keyword filtering removes low-signal chunks. All remaining filtered candidates are passed to the generator to avoid arbitrary truncation.

3.4 Controlled Generation

The generator defaults to TinyLlama 1.1B (ONNX-optimized when available) with temperature ‘0.3‘, top-p ‘0.9‘, and token limits ‘max_length=4096‘, ‘max_new_tokens=2048‘. A conservative prompt template encourages concise, practical responses with minimal citations and explicit acknowledgement of uncertainty when sources are insufficient.

4 Implementation Details

Core modules include ‘scaffold_core/vector/enhanced_query_improved.py‘ (hybrid retrieval and prompt construction), ‘scaffold_core/llm.py‘ (LLM manager with continuation support), and ‘scaffold_core/config.py‘ (central configuration, model registries, environment keys). The UI layer provides model selection and parameter controls; however, this report focuses on the headless RAG core.

5 Evaluation Summary

Internal tests indicate: (i) stable citation rendering with preserved metadata, (ii) reduced repetition relative to earlier prompts, and (iii) robust behavior under variable query difficulty. Performance is CPU-feasible due to compact models; GPU acceleration is optional. Detailed quantitative benchmarks (latency, memory) are tracked separately in repository logs.

6 Limitations

The compact LLM constrains depth of synthesis on complex queries. Citation granularity (e.g., page spans) is limited by source metadata. Some domain-specific terminology may require larger context windows or specialized models.

7 Reproducibility Settings

Component	Setting
Embedding model	`all-MiniLM-L6-v2`
Cross-encoder	`cross-encoder/ms-marco-MiniLM-L-6-v2`
LLM	`TinyLlama/TinyLlama-1.1B-Chat-v1.0` (ONNX when available)
Chunk size / overlap	1000 / 200
FAISS index	`IndexFlatL2`
Initial retrieval	`TOP_K=30`
Final candidates	All filtered (no fixed cut)
Temperature / top-p	0.3 / 0.9

Max length / new tokens	4096 / 2048

8 Conclusion

Scaffold AI demonstrates that a carefully engineered compact RAG stack can provide literature-grounded curriculum recommendations suitable for early-stage adoption in engineering courses. The present configuration prioritizes transparency, determinism, and operational feasibility, while leaving room for future upgrades to larger models and richer citation features.

Availability

Source code, configuration, and evaluation artifacts are available within the repository. Execution requires Python 3.11+ and the dependencies listed in `requirements.txt`. Optional hardware acceleration is automatically detected.