Multimedia Gen AI-Powered Course Module Generator

Eddie McGowan & Alex Kash: Lehigh University Data Science
https://github.com/EddieMcGowan/DSCI-498.git

Abstract

This project presents an AI system that transforms teacher-provided resources—like PDFs, articles, and web links—into structured, engaging lesson plans. Unlike commercial tools, it gives educators full control over inputs and output formats.

Key Features:

- Accepts multimedia inputs (text, images, tables)
- Editable prompt interface for teachers
- "Reprompt"/"Approve" workflow for refining lessons
- Automatic website deployment for student access

By supporting diverse inputs and publishing lessons directly to the web, this tool saves time, boosts personalization, and enhances instructional quality—all while keeping teachers in control.

Problem Description & Motivation

- Lesson planning is time-consuming, especially when customizing for goals or student needs.
- Commercial AI tools often limit input control and personalization.
- Educators need a flexible solution to create modules from their own resources.
- This project builds a tool to empower teacher-driven content generation.

Model Selection

Tools Utilized:

- **Resource Input**: PDFs, articles, and web links
- **Extraction Tools**: PyMuPDF (text), PDFPlumber & Pillow (images/tables), and BeautifulSoup (web scraping)
- **Model Backend**: Llama-2-7b-chat-hf (multimodal LLM)
- **Frontend**: Streamlit interface for lesson generation hosted on Lehigh magic-02 high performance computer

Models Tested

Model	Inputs	Notes
Llama-2-7b-chat-hf (Chosen Model) ✓	Text, Tables	Excels at summarizing text; does not accept images inputs
LLaVA 1.5 X	Images, Text, Tables	Poor at summarizing text
LLaVA 1.6 X	Images, Text, Tables	Cannot be configured on Magic-02
OpenAI GPT-4o X	Images, Text, Tables	Requires paid subscription for unlimited access

Comparison With Existing Tools

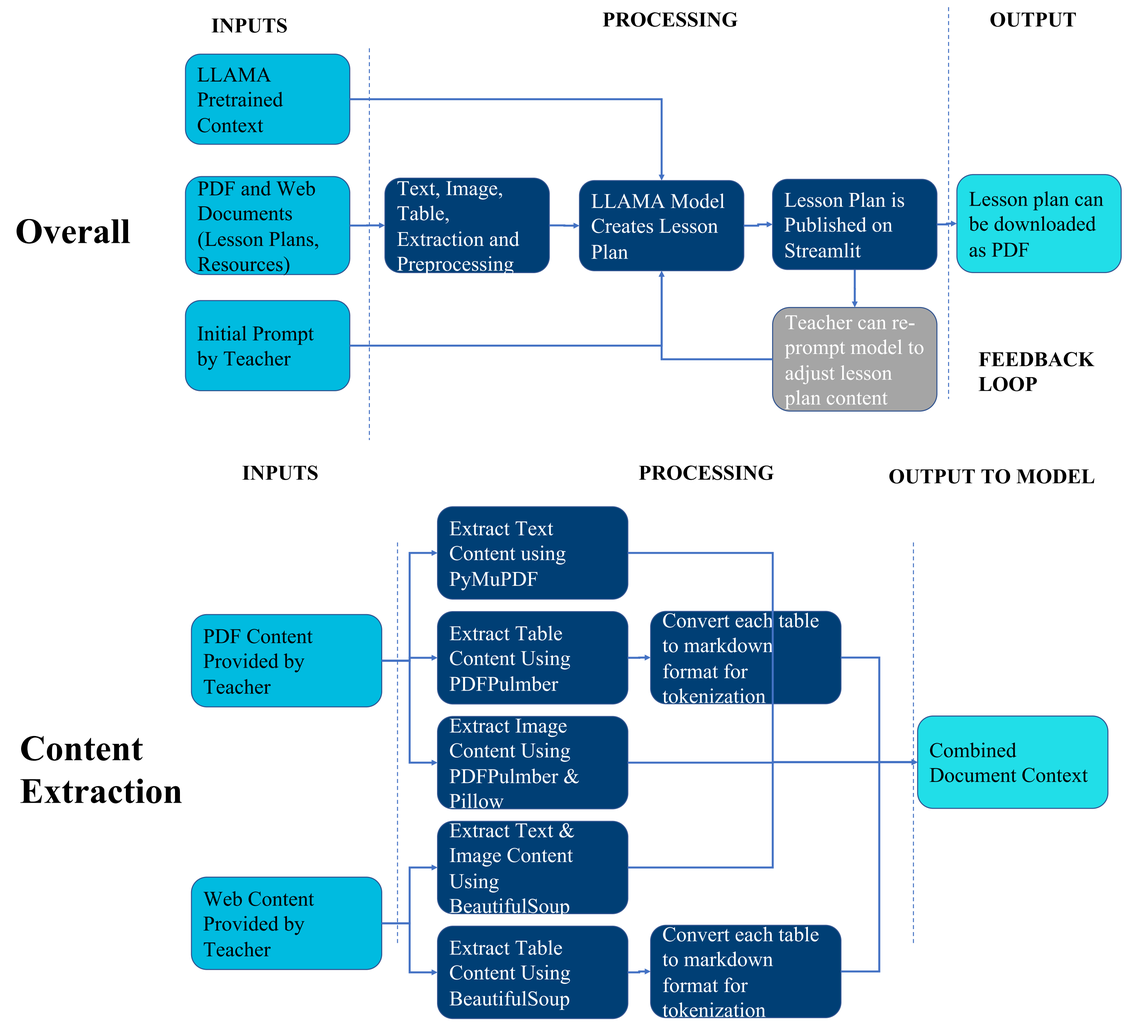
Feature	This Project	Khanmigo (Khan Academy)
Teacher Defined Sources	✓	X
Media Extraction (PDF/Web)	✓	X
Web Publishing	✓	X
Feedback-based revision	✓	X

Sample Workflow

1. Teacher uploads a PDF and edits a default lesson prompt.
2. A pipeline extracts text, tables, and images.
3. Lesson is generated and previewed.
4. Teacher selects "Reprompt" or "Approve."
5. Final version is deployed to the class website.

Data & Data Pipeline

- No fixed dataset; uses teacher-uploaded materials (PDFs, articles, web links, tables) at runtime.
- Processes inputs in real time to create personalized, structured lessons (see Pipeline below).
- Adapts to different teaching styles without requiring extra model training.



Evaluation

- Tested using sample PDFs and web pages
- Compared the output of different Gen AI models
- Adjusted model parameters as needed
- Fine-tuned the initial prompt to optimize the output

Introduction (10 minutes)
* Introduce the problem of semi-supervised learning and its importance in modern data analysis.
* Briefly explain the challenges of obtaining label information in large datasets.

Section 1: Background and Related Work (20 minutes)
* Provide an overview of the existing approaches to semi-supervised learning.
* Discuss the limitations of self-training schemes and transductive SVMs.
* Introduce the concept of deep generative models and their potential for semi-supervised learning.

Section 2: The Proposed Method (30 minutes)
* Explain the proposed method for semi-supervised learning using deep generative models.
* Discuss the use of approximate Bayesian inference and variational methods.
* Provide examples of how the method can be applied to different datasets.

Section 3: Results and Evaluation (20 minutes)
* Present the results of the experiments conducted on various datasets.
* Discuss the performance of the proposed method compared to other state-of-the-art methods.
* Analyze the contribution of different components of the proposed method.

Section 4: Conclusion and Future Work (10 minutes)
* Summarize the main findings of the study.
* Discuss the implications of the results for practical applications.
* Identify potential directions for future work.

A sample lesson plan output from the Model.

Conclusion

This tool streamlines lesson planning by supporting multimodal materials and real-time teacher feedback. It accelerates content creation and helps deliver more personalized, engaging learning experiences for students.

Future Works:

1. Link with educational platform (e.g., Lehigh Course Site)
2. Deploy on cloud infrastructure for scalability
3. Change the backend model to LLaVA 1.6 to support image inputs
4. Preload state and national educational requirements to ensure lesson plans meet educational benchmarks

Acknowledgements

Masoud Yari
Todd George