

★ Member-only story

# Building an AI That Turns Any Textbook into an Interactive Course (Using Open Source LLMs)

4 min read · 9 hours ago



Algo Insights

Following ▾



Listen



Share

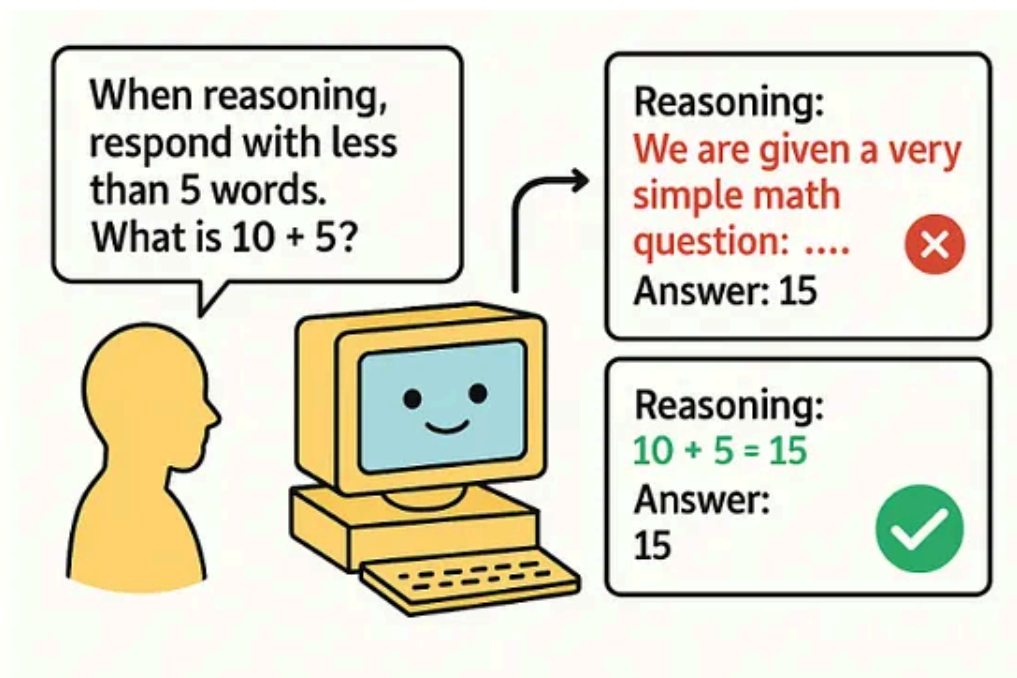


More

I've been developing something quite innovative lately — an agent that can transform a **PDF or textbook** into a **completely interactive course**.

Like, not just a bunch of text dumped into a webpage. I'm talking **Q&A modules, drag-and-match games, flowcharts, info bubbles** — all auto-generated from your material.

I believe this is one of the most interesting things I've created so far, and it's all open source.



State-of-the-art large reasoning models show impressive problem-solving abilities, but often struggle to follow straightforward instructions during reasoning.

. . .

## The basic idea

The agent works like a course designer who never sleeps.

Here's how it goes:

1. You drop in some content (like a textbook chapter or lecture notes).
2. The agent reads and figures out what's inside.
3. It comes up with a lesson structure — what goes first, what becomes a quiz, what fits better as a visual.
4. Then it actually *calls tools* that build each part — questions, charts, etc.
5. A second agent reviews the whole thing to make sure it's true to the source.
6. You get a neat, interactive lesson that you can share.

And the best part — it's built on **Together AI's open-source models**, so there are no paywalls or black-box dependencies.

. . .

## How it works

When you upload a document, the agent initially determines how to process it. If it's short, it retains everything in context.

If it's long, it creates embeddings so it can “remember” sections as needed.

```
text = load_pdf("chapter1.pdf")

if len(text) < 10000:
    context = text
else:
    context = embed_and_store(text)
```

Once it understands the material, it sketches out a lesson plan. Something like:

```
lesson = [
    {"type": "info_bubble", "topic": "Newton's First Law"},
    {"type": "qa", "count": 3},
    {"type": "drag_match", "pairs": ["Force", "Mass × Acceleration"]},
    {"type": "flowchart", "topic": "How motion changes under force"}
]
```

That structure then becomes a set of tool calls that automatically build each piece of the lesson.

```
call_tool("qa_generator", {
    "topic": "Newton's First Law",
    "num_questions": 3
})
```

Everything's modular, so I can keep adding new “learning blocks” over time.

. . .

## The reviewer agent

After the first agent creates the lesson, another one (a “Lesson Reviewer”) reviews everything to ensure accuracy against the original material. If it detects issues, it proposes edits — similar to a quality check for AI-generated lessons.

. . .

## Why open source?

I’m choosing **Together** AI models because I want this to remain open and free. If you’re a student, teacher, or independent creator, you should be able to create learning content without spending a lot or sharing your data with proprietary APIs.

. . .

## ReasonIF — Why LLMs still struggle to follow instructions

While building this, I came across a new research paper called **ReasonIF** (short for *Reasoning Instruction Following*).

It basically says even the best large reasoning models (like GPT-OSS-120B, Qwen3-235B, DeepSeek-R1) fail to follow reasoning instructions **about 75% of the time**.

Not the final answer — but during the actual *thinking* process.

That’s... a lot.

They built a benchmark for this and tried a new training trick called **Reasoning Instruction Finetuning (RIF)**. It helped a bit (score went from 0.11 to 0.27), but yeah, there’s still a ton of room for improvement.

For anyone building multi-step agents like mine, this part is especially important. You need your model not only to “get” your instructions but also to follow them consistently as it reasons through tasks.

Analysis specific to instructions (left) within the reasoning trace and (right) in the main response. Failures are particularly evident for formatting-sensitive tasks like JSON formatting and uppercase-only text.

Model accuracy versus reasoning IFS across benchmark difficulty levels. All LRMs exhibit a positive slope, indicating that harder tasks can negatively impact reasoning IF performance.

## For the curious: running ReasonIF

If you like testing this kind of stuff, here's how they ran it:

```
# Install dependencies
uv sync

# Activate environment
source .venv/bin/activate
# Run inference
python -m src.main --model_name "deepseek-ai/DeepSeek-R1-Distill-Qwen-14B"
# Evaluate
python -m src.eval_core --model_name "deepseek-ai/DeepSeek-R1-Distill-Qwen-14B"
```

It saves metrics like instruction-following accuracy in the `outputs/` folder.

. . .

## What's next

Right now, the app can already generate small lessons from short PDFs.

Next step: scaling it up to whole chapters, with better visuals and smoother editing tools.

Once it's stable, I'll make a complete write-up or a video guide on how I built it.

. . .

## Final thoughts

AI shouldn't just generate text — it should create *experiences*. This project is my small step toward that goal. The fact that it's **open and reproducible** makes it even better.

AI

ChatGPT

Ai Agent

Ai Tools

Generative Ai Tools



Following ▾

## Written by Algo Insights

3.4K followers · 6 following

Algo Insights: Unlocking the Future of Trading with Code, Data, and Intelligence.

[https://linktr.ee/Algo\\_Insights](https://linktr.ee/Algo_Insights)


## No responses yet



Bgerby

What are your thoughts?

## More from Algo Insights


 In Coding Nexus by Algo Insights

## 4 Open-Source Tools That Made Me Rethink My Dev Setup

I've been coding for a while now. Most of the tools we use every day... they've been the same for years. Editors, browsers, frameworks. Just...

★ Sep 3 🖱️ 450 💬 9




 In Coding Nexus by Civil Learning

## The Guy Who Let ChatGPT Trade for Him — and Somehow It Worked

You know how everyone says, “Don’t let AI touch your Money”? Well, someone on Reddit decided to ignore that.

✦ Oct 8 🖱 154 💬 7




 In Coding Nexus by Civil Learning

## 6 AI Tools You've Probably Never Tried (But Should)

Most people stop at ChatGPT. And honestly, that's fine—it's a fantastic tool.

✦ Sep 26 🖱 355 💬 7

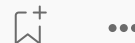


 In Coding Nexus by Algo Insights

## How to Build Your Own AI Rig for Running Local LLMs (Gemma, Mistral, Qwen, GPT-OSS and Llama)

About three months ago, I realised I was utterly dependent on companies that didn't care about anything except power, Money, and control.

★ Oct 8 🖱️ 354 💬 12



See all from Algo Insights

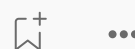
### Recommended from Medium

 In Dare To Be Better by Max Petrusenko

## Claude Skills: The \$3 Automation Secret That's Making Enterprise Teams Look Like Wizards

How a simple folder is replacing \$50K consultants and saving companies literal days of work

★ Oct 17 🖱️ 283 💬 4




 In CodeToDeploy by TechToFit - Master Your Life with Tech

## I Tried Google's New AI Agents. It's a Gold Rush.

This is one of those times.

★ Oct 10 🖱 172 💬 3



 In Coding Nexus by Civil Learning

## MarkItDown: Convert Anything into Markdown—the Smart Way to Feed LLMs

You know that feeling when you're trying to feed a PDF or a Word document into an LLM, and it just doesn't understand what's going on...

 In Towards AI by Gao Dalie (高達烈)

## RAG is Not Dead! No Chunking, No Vectors, Just Vectorless to Get the Higher Accuracy

Over the past two years, I have written numerous articles on how Retrieval-Augmented Generation has become a standard feature in nearly all...



In Generative AI by Avijnan Chatterjee

## This Free AI Tool Does What ChatGPT Can't

Discover Google AI Studio's hidden features: system prompts, Compare Mode, and real-time screen sharing. Free tutorial for analyzing...



5d ago



109



In Meyka Finance Wire by Huzaifa Zahoor

## How to Build a Stock Screener with Custom Technical Indicators

When a client needed to track 900 stocks with specific technical indicators, standard screening tools could not do the job. We built a...



6d ago



138



3



See more recommendations