Generative AI for East Asian Studies

Session 4: Agentic Approaches to Humanities Research

Kwok-leong Tang

2026-04-01

Agenda

1. The LLM Wiki: A new pattern for knowledge management
2. Exercise: Build an AI Learning Collection with Antigravity

The LLM Wiki

The Problem: Knowledge Doesn't Stick

Think about how you use AI tools today:

1. You ask a question in a chatbot
2. You get a good answer
3. You close the tab
4. Next week, you ask a similar question — from scratch

Every conversation starts at zero. The insights, the connections, the corrections — all gone.

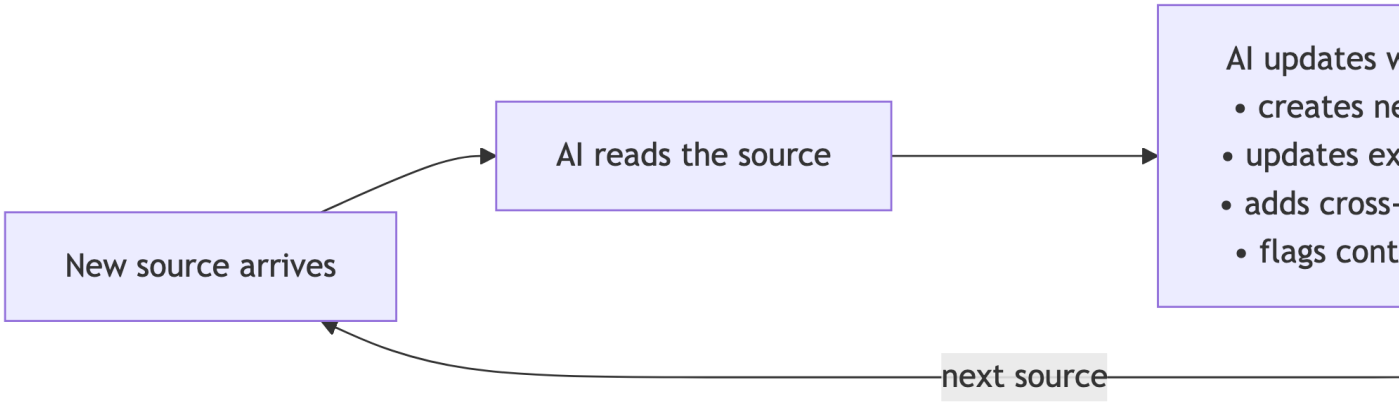
RAG Re-Derives Everything

This is also the problem with **RAG** (Retrieval-Augmented Generation), which we discussed earlier today:

What if the AI built something **permanent** instead?

The Pattern: Compilation Over Retrieval

Instead of searching raw documents every time, have the AI **read sources once** and **write the knowledge into a wiki** — a persistent collection of interconnected markdown pages.



The wiki is a **persistent, compounding artifact**. It gets better with every source you add.

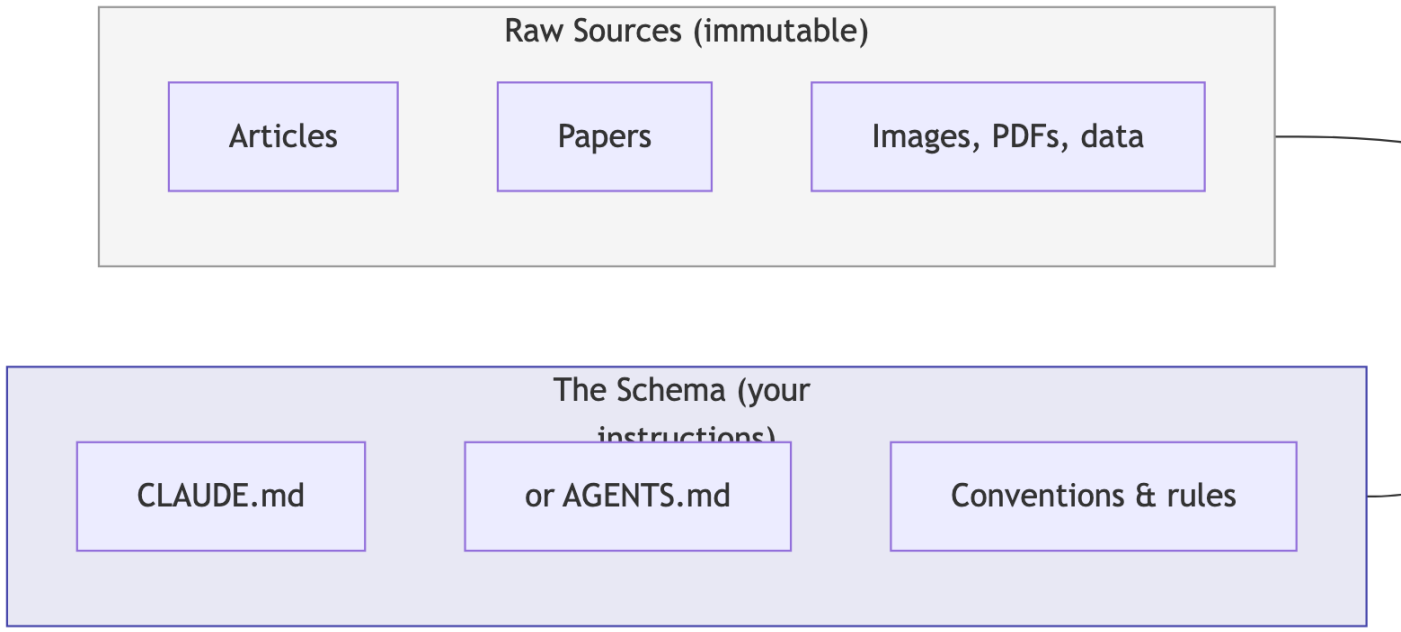
Who Proposed This?

> **Note**
>
> On April 4, 2026, **Andrej Karpathy** — founding member of OpenAI and former head of AI at Tesla — published a short document called **"LLM Wiki"**. Within 48 hours, it had 5,000+ stars on GitHub.

He also coined the term **"vibe coding"** in early 2025 and later declared it obsolete, replaced by **"agentic engineering"**: orchestrating AI agents rather than writing code directly.

The LLM Wiki applies agentic engineering to **knowledge management**.

Three-Layer Architecture



Layer 1: Raw Sources (Immutable)

Your curated collection of original materials. These are **read-only** — the AI never modifies them.

- Articles, papers, book chapters
- Primary sources (historical texts, documents)
- Images, PDFs, datasets
- OCR output from digitized materials

Layer 2: The Wiki (LLM-Maintained)

A directory of markdown files that the AI **owns entirely**. The AI creates pages, updates them, adds cross-references, and maintains an index.

Key files:

- `index.md` — a catalog of all wiki pages, organized by category
- `log.md` — an append-only chronological record of every action the AI takes

Layer 3: The Schema (Your Instructions)

A configuration document — `CLAUDE.md` or `AGENTS.md` — that defines:

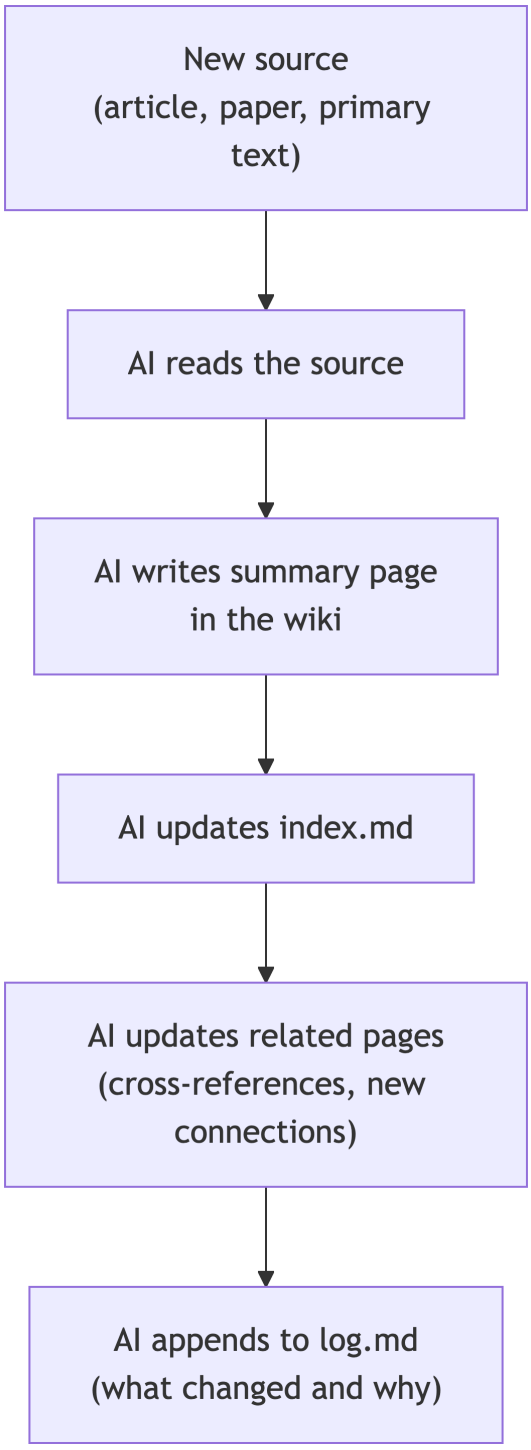
- How pages should be structured (templates, required fields)
- Naming conventions for files
- How to handle contradictions or uncertain information
- Citation format and cross-reference style

> **!** Important
>
> **The schema is the most important layer.** It is the difference between a useful, well-organized wiki and a chaotic dump of AI-generated text.

Three Operations

1. Ingest

Process a new source and integrate it into the wiki.



The AI might touch 10-15 files in a single ingest operation.

Three Operations (cont.)

2. Query

Ask a question and get an answer synthesized from the wiki. The AI searches the index, reads relevant pages, and produces an answer with citations back to specific wiki pages.

3. Lint

Periodic health checks on the wiki:

- Find contradictions between pages
- Identify stale claims
- Detect orphan pages (not linked from anywhere)
- Flag missing cross-references

Why This Matters

Humans are good at:

- **Curating sources** — deciding what is worth reading
- **Asking questions** — directing the analysis
- **Evaluating results** — judging whether the AI got it right

Humans are bad at:

- **Bookkeeping** — updating cross-references, maintaining indexes
- **Touching 15 files at once** — updating every page that references a person when you learn new information

The LLM Wiki lets each party do what they are best at.

Historical Roots: The Memex

In 1945, Vannevar Bush published "As We May Think" in *The Atlantic*, describing a hypothetical device called the **Memex** — a personal knowledge store with associative trails linking documents together.

His challenge was **maintenance**: who keeps the trails updated?

Karpathy's answer: the LLM does.

Critical Concerns

1. The Generation Effect — When you write your own notes, you learn. When an AI writes notes for you, you might not.

2. Error Accumulation — LLM summaries can be confidently wrong. If a hallucinated fact enters the wiki, it can propagate through cross-references.

3. Authority Creep — Wiki pages start to feel authoritative simply because they are well-organized. But they are interpretations, not facts.

> Tip
>
> **The safeguard**: Always keep your raw sources immutable and accessible. The wiki is a map, not the territory.

Exercise: Build an AI Learning Collection with Antigravity

Overview

Now let's build an LLM Wiki from scratch. You will use **Karpathy's original gist as your starting prompt** — paste it directly into Antigravity and let the agent set up the wiki for you.

The topic: **your personal AI learning collection** — articles, videos, tutorials, and concepts about AI, LLMs, and coding agents.

Step 1: Create the Project Directory

Open your terminal and create a new directory:

```
mkdir ~/ai-learning-wiki
```

Step 2: Copy Karpathy's Gist

Open the gist in your browser:

https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f

Read through it. This is the blueprint you will hand to Antigravity.

Step 3: Launch Antigravity and Paste the Gist

1. Open **Antigravity**
2. Click **"Open folder"** and select the `~/ai-learning-wiki` directory you just created
3. Antigravity is now working inside that folder — any files it creates will appear there

Step 3 (cont.)

Now paste the **entire content** of Karpathy's gist into the chat, followed by this instruction:

```
[paste the full gist content here]

---

Using the LLM Wiki pattern described above, set up a wiki in this directory
for my personal AI learning collection. The topic is everything I am learning
about AI, large language models, coding agents, and related tools.

Please:
1. Create the directory structure (raw/, wiki/)
2. Write an AGENTS.md schema tailored to an AI learning collection
   - Page types should include: Concept, Tool, Paper, Tutorial, Person,
     and Vocabulary
   - Include templates for each page type with relevant frontmatter fields
3. Create wiki/index.md and wiki/log.md
4. Set up subdirectories for each page type
5. Explain what you created and how I should use it
```

Step 3 Tips

> Tip
>
> **Let Antigravity do the work.** You are not writing the schema yourself — you are giving the agent the pattern (Karpathy's gist) and a topic (AI learning), and letting it generate the schema, templates, and structure. This is the pattern in action: you curate and direct, the AI does the bookkeeping.

Step 4: Open the Vault in Obsidian

While Antigravity is working, open the `~/ai-learning-wiki` folder as a vault in **Obsidian**:

1. Open Obsidian
2. Click **"Open folder as vault"**
3. Select the `~/ai-learning-wiki` directory

Now you can watch in real time as Antigravity creates and updates files. Wikilinks between pages will become clickable, and you can use Obsidian's graph view to visualize the connections.

Step 5: Review What Antigravity Created

After Antigravity finishes, check the vault in Obsidian. You should see something like:

```
AGENTS.md
raw/
wiki/
  index.md
  log.md
  concepts/
  tools/
  papers/
  tutorials/
  people/
  vocabulary/
```

Step 5 (cont.)

Open `AGENTS.md` and read through it. This is the schema that Antigravity wrote for you based on Karpathy's pattern. Ask yourself:

- Do the page types make sense for an AI learning collection?
- Are the templates detailed enough?
- Is anything missing?

If you want changes, just tell Antigravity:

```
Add a "Source" field to the Concept template that links back to where I first
learned about the concept. Also add a "Difficulty" field (beginner, intermediate,
advanced) to the Tutorial template.
```

Step 6: Ingest Your First Source

Create a raw source file. You can ask Antigravity to create it, or create it yourself. For example, create `raw/karpathy-llm-wiki-2026-04-04.md`:

```
# Andrej Karpathy - LLM Wiki (April 4, 2026)

Source: https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f

Karpathy published a gist describing a pattern for building knowledge bases
with LLMs. The core idea: instead of using RAG to re-derive answers from
raw documents every time, have the LLM "compile" knowledge into a
persistent wiki.
```

Step 6 (cont.)

Now tell Antigravity to ingest it:

```
I have a new source in raw/karpathy-llm-wiki-2026-04-04.md. Please ingest it
into the wiki.
```

What to Observe

Watch how Antigravity:

1. **Reads the raw source** and identifies concepts, people, and tools
2. **Creates multiple wiki pages** (e.g., `wiki/concepts/llm-wiki.md`, `wiki/people/andrej-karpathy.md`)
3. **Cross-links everything**
4. **Updates index.md and log.md**

> **Note**
>
> **One source, many outputs.** A single article summary might create 5-10 wiki pages. This is the bookkeeping that humans never maintain by hand.

Step 7: Ingest a Second Source

Create `raw/vibe-coding-2025.md`:

```
# Vibe Coding

Term coined by Andrej Karpathy in February 2025. The idea: instead of
writing code line by line, you describe what you want in natural language
and let an AI generate the code.

Karpathy later declared vibe coding obsolete, replaced by "agentic
engineering" - orchestrating AI agents that can read files, run tests,
and make multi-step changes autonomously.
```

Ingest it:

```
I have a new source in raw/vibe-coding-2025.md. Please ingest it.
```

Observe how Antigravity **updates existing pages** and **creates new ones**.

Step 8: Query and Lint

Try asking questions about your collection:

```
What is the relationship between vibe coding and the LLM Wiki pattern?
```

```
What concepts have I learned that relate to how LLMs interact with
external data?
```

Then run a health check:

```
Please lint the wiki. Check for missing fields, orphan pages, and gaps.
```

Step 9: Add Your Own Material

Add **at least one source from your own experience**. This could be:

- An article or video about AI that you found interesting
- A tool you tried and want to remember how to use
- A concept that confused you and that you eventually understood
- Notes from earlier sessions today

Ingest it and watch the wiki grow.

> **!** Important
>
> **You now have a working LLM Wiki.** You gave Antigravity a pattern (Karpathy's gist) and a topic (AI learning), and it built a structured, cross-referenced knowledge base. Every new source you add makes the wiki more complete.

Takeaways

What We Learned

1. **The LLM Wiki** is a pattern for building knowledge bases where AI maintains the structure and you curate the content — compilation over retrieval
2. **Three layers**: raw sources (immutable), wiki (AI-maintained), schema (your rules)
3. **Three operations**: ingest (add sources), query (ask questions), lint (health checks)
4. **The tedious part is the bookkeeping** — and that is exactly what AI agents are good at
5. **The critical part is your judgment** — curating sources, asking questions, evaluating results, and going back to primary sources when it matters

Resources

- Andrej Karpathy's LLM Wiki (original gist)
- The LLM Wiki Movement (analysis report)
- Obsidian
- Vannevar Bush, "As We May Think" (1945)