Generative AI for East Asian Studies

Session 2: Hands-on Practice Session with LM Studio

Kwok-leong Tang

2026-04-01

Before We Begin

- Please make sure you have **LM Studio** or **Ollama** installed, and the **Qwen3.5-0.8B** model downloaded.
- If you haven't done so, follow the Software Installation Instruction.
- If you have any problem with the software installation, don't panic! You can follow along with any chatbot (e.g., ChatGPT) for some parts of the workshop.

Agenda

1. What is generative AI?
2. The nature and limitations of LLMs
3. Prompt engineering essentials
4. Use cases for East Asian Studies
5. Context is the solution: RAG and tool use

What is Generative AI?

> **!** Important
>
> Everything is **PREDICTION**!

- Large language models (LLMs) are trained on massive text corpora to predict the next token (word or subword).
- When you type a prompt, the model generates a response one token at a time, each token predicted based on all the tokens that came before.

Autoregressive Language Models

- Each token is generated sequentially based on the preceding context.
- The model assigns a **probability** to each possible next token and samples from that distribution.

> **!** Important
>
> Every token is generated based on **PROBABILITY**!

Four Components of AI-Assisted Research

Component	Description	Example
Model	The LLM engine	GPT, Claude, Qwen, DeepSeek, LLaMA
Prompt	Your instructions and queries	Use cases, system prompts
Context	Additional information provided	Documents, databases, knowledge bases
Tools	External capabilities	Web search, APIs, MCPs

Let's Get Started with LM Studio

1. Open **LM Studio** on your computer.
2. Load the **Qwen3.5-0.8B** model from the model dropdown.
3. You should see the chat interface ready for input.

For **Ollama** users, open your terminal and run:

```
ollama run qwen3.5:0.8b
```

> **Note**
>
> We choose a small 0.8B model because it may not be as "smart" as state-of-the-art (SOTA) models and uses fewer engineering tricks. This lets us observe the **nature** of a foundation model more clearly.

The Limitations of Prediction

Knowledge Cutoff

Every model has a training data cutoff date. After training, a model's parameters are **frozen** — its built-in knowledge does not change.

Try the following prompts in LM Studio (or Ollama):

```
What is your knowledge cutoff date?
```

```
Who is the prime minister of Japan?
```

> **Tip**
>
> Compare the model's answer with the current facts. Is the answer correct? Why or why not?

Hallucination

LLMs can generate plausible-sounding but incorrect information — this is called **hallucination**. Because the model is predicting the most probable next token, it may "make up" facts.

Try these prompts:

```
How many "r" in strawberry?
```

```
How many "u" in Labubu?
```

If you don't know what Labubu is, check here.

```
How many "a" in gandamala?
```

> **Note**
>
> Character counting is notoriously difficult for LLMs because they process **tokens**, not individual characters. A token might represent a whole word or a piece of a word.

Autoregressive Nature

LLMs read text from left to right (or beginning to end). They struggle with tasks that require reading in reverse or non-sequential processing.

Try this prompt:

```
Can you tell me the meaning of this sentence: "B1 ammeG ehg tsniaga tluser eht erapmoc dna sl
```

> **Tip**
>
> The sentence is written backwards. Can the model figure it out? Why might this be difficult?

Autoregressive Nature: Classical Chinese

This example is revised from a similar prompt by Professor Peter Bol:

```

```

Now try the same text in the correct reading order:

```

```

> **Important**
>
> Compare the two results. The reading direction matters because LLMs are **autoregressive** — they process tokens sequentially from left to right.

Prompt Engineering

The Basics of Prompt Engineering

Prompt engineering is the practice of crafting effective inputs to get better outputs from LLMs.

Key resources:

- Prompt engineering guide: https://www.promptingguide.ai/
- Chain-of-thought prompting: https://arxiv.org/pdf/2201.11903

Chain-of-Thought: The Magic Words

Remember the strawberry problem? Let's try again with a better prompt:

```
How many "r" in strawberry? Count it character by character.
```

The magic words: **think step by step**.

> **Tip**
>
> Adding "think step by step" or "count character by character" can significantly improve accuracy for certain tasks. This is called **chain-of-thought prompting**.

Improving Results with Better Prompts

Let's revisit the reversed Classical Chinese text with a more effective prompt:

```

```

> **Note**
>
> Notice how we gave the model more context: we specified the dynasty (Southern Song) and the role (expert in Chinese history). This helps the model narrow down its predictions.

System Prompts and User Prompts

	System Prompt	User Prompt
What	Hidden instructions that define the AI's role, personality, and rules	Your actual question or request
When	Set before the conversation starts	Given in every interaction
Who sets it	Developers or advanced users	The end user

Try Setting a System Prompt

In LM Studio, you can set a system prompt. Enter the following into the system prompt field:

```
```

Or try:

```
You are Oscar Wilde. You must answer all questions in his style.
```

Then ask a question:

```
?
```

> **Note**
>
> **For Ollama users**: You can set a system prompt by creating a Modelfile. Or simply include the persona instruction in your prompt itself.

Prompt Diversification

A recent paper claims that a system prompt can easily diversify the response:

```
You are a helpful assistant. For each query, please generate a set of five possible responses
```

> **Tip**
>
> This technique can be useful when you want to explore multiple possible interpretations or translations of a text.

Use Cases for East Asian Studies

Use Cases Overview

You can check some past workshops for more use cases in Chinese Studies:

- 2023 Workshop
- 2024 Workshop

Today we will practice with:

1. Citation formatting
2. Text comparison and parallel reading
3. Named entity recognition (NER) and data extraction

Citation Formatting

Try the following prompt:

```
Please format the following bibliographic information into Chicago Manual of Style (17th edi

Author:
Title:
Publisher:
Year: 2003
Place:
```

> **Tip**
>
> You can also ask the model to convert between citation styles (e.g., Chicago to MLA or APA), or to format citations in multiple languages.

Text Comparison and Parallel Reading

```
Compare the following two translations of the opening line of the Tao Te Ching and discuss th

Translation 1 (D.C. Lau): "The way that can be spoken of is not the constant way."
Translation 2 (Stephen Mitchell): "The tao that can be told is not the eternal Tao."

Original:
```

Named Entity Recognition (NER)

```
Extract all person names, place names, and official titles from the following text. Return th
```

> **Tip**
>
> NER is one of the most practical use cases for humanities researchers. LLMs can identify entities in Classical Chinese texts where traditional NER tools often struggle.

Data Extraction

```
Please extract all place names from the following text and output them as a JSON array:
```

> **Note**
>
> Structured output (JSON, CSV, tables) is extremely useful for downstream analysis and can be imported into databases, GIS tools, or spreadsheets.

Context is the Solution

Why Context Matters

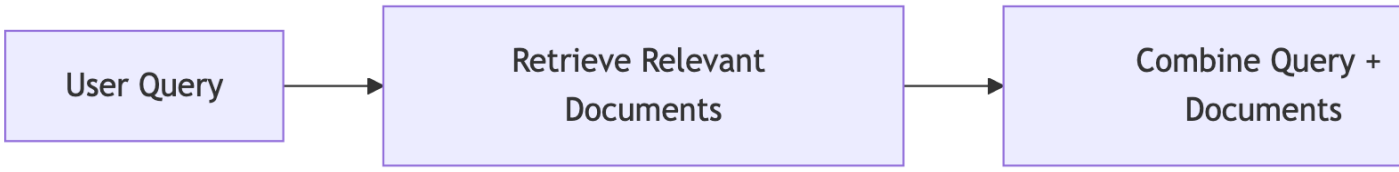
We've seen that LLMs have limitations: knowledge cutoffs, hallucinations, and autoregressive constraints. How do we overcome them?

The answer is **context** — providing the model with additional, reliable information.

Two key approaches:

1. **Retrieval-Augmented Generation (RAG)**: Supplying relevant documents or data alongside the prompt
2. **Tool use**: Allowing the model to query external systems (APIs, databases, web)

Retrieval-Augmented Generation (RAG)



- Instead of relying solely on the model's training data, we provide relevant context at query time.
- The model can then ground its response in the provided documents.

RAG in Practice

Try this: copy a passage from a primary source and ask the model to analyze it.

```
Based on the following passage from the Veritable Records of the Ming Dynasty, identify the r
```

> **Important**
>
> By providing the source text directly, we are doing a simple form of RAG — giving the model context it would not otherwise have.

Tool Use and MCP

A major advancement of GenAI in 2025–2026 is the usage of **tools** by LLMs. The most important standard is the Model Context Protocol (MCP) developed by Anthropic.

MCP allows LLMs to:

- Fetch web pages
- Query databases and APIs
- Search library catalogs
- Access geographic information systems

> **Note**
>
> We will explore MCP and tool use in more depth in **Session 2** this afternoon.

GLAMs as Context Providers

GLAMs (Galleries, Libraries, Archives, and Museums) are becoming an important part of the AI-assisted research infrastructure.

- They provide **reliable, authoritative context** for researchers.
- MCP servers can connect LLMs directly to GLAM collections and catalogs.

Examples:

- Harvard Library MCP
- Harvard Art Museum MCP
- Met Museum MCP
- Art Institute of Chicago MCP

OCR: A Transformative Use Case

A major obstacle to adopting digital tools in East Asian Studies prior to 2025 was **Optical Character Recognition (OCR)**.

- Traditional OCR struggled with CJK historical texts
- Multiple calligraphic styles posed significant challenges

Newly emerging OCR models in 2025–2026:

- Dots OCR from Rednote
- PaddleOCR-VL
- DeepSeek OCR
- Chandra from Datalab

> **Note**
>
> Vision models (e.g., Gemini 2.5 Pro) can also perform OCR when you upload an image. We will see a demonstration if time permits.

Session 1 Summary

What We Learned Today

1. **Generative AI is prediction**: LLMs generate text token by token based on probability.
2. **Limitations are real**: Knowledge cutoffs, hallucination, and autoregressive constraints affect all models.

3. **Prompt engineering helps**: Chain-of-thought, system prompts, and clear instructions improve results.
4. **Context is the solution**: RAG and tool use overcome many LLM limitations.
5. **Practical applications**: Citation formatting, text comparison, NER, data extraction, and OCR.

Looking Ahead: Session 2

In the afternoon session, we will dive deeper into:

- **Tool use and MCP** in practice
- Setting up and using MCP servers
- Working with GLAM collections through AI
- Building your own AI-assisted research workflows

Resources

- Prompt engineering: https://www.promptingguide.ai/
- Model Context Protocol: https://modelcontextprotocol.io/
- LM Studio documentation: https://lmstudio.ai/docs/app
- Ollama documentation: https://ollama.com
- Past workshops: https://kwokleongtang.net
- Dan Cohen's essays on AI and GLAMs: The Library's New Entryway