# What LLMs Must Forget to Teach Effectively

Lessons in Designing AI Tutors for Premodern Japanese [*]

Ariel Stilerman

*Department of East Asian Languages and Cultures*
*Stanford University*

February 4, 2026

**Abstract**

This white paper introduces a new AI tutoring framework for premodern Japanese studies, designed to align with the pedagogical values of a liberal arts education. Addressing the tendency of generic Large Language Models (LLMs) to bypass student effort through over-explanation, the study proposes a design strategy based on restrictive prompt engineering. By limiting model outputs and enforcing a Socratic questioning method and pace, the tutor prioritizes student active understanding over rapid answers. The paper details the technical architecture, arguing that effective AI tutoring relies on "forgetting" capabilities to gain focus, and analyzes the challenges of model distillation and data scarcity when processing historical languages. Preserving effectiveness after distillation is key because more efficient models can significantly minimize environmental impact and lower barriers to access, such as cost and latency.

## 1 Introduction

The promise of AI to save effort can often backfire. Anthropic's advertisements, for instance, invite users to "Do five things while you focus on one." They suggest we can bypass the messy, confusing experience of writing by delegating the outline, draft, and polish to a Large Language Model (LLM). However, a liberal arts education advocates for the opposite: it challenges us to read more slowly and write more deliberately. College represents a unique opportunity to cultivate the complex intellectual habits that will sustain us later. This raises a critical question: how can we design AI tools that align with these values, where effort and process are inseparable from outcome and performance? This paper proposes a tutor designed to augment, not replace, the instructor. The tutor guides students as they work through a text using

[*]The ideas discussed in this essay were developed in conversation with Andrew Nelson, Alan Huang, Momoyo Kubo Lowdermilk, Pelin Cilgin, Caleb Langley, Camilla Piana, Qianhe Qin, and Sera Wang. Any mistakes are the exclusive responsibility of the author.

targeted questions and contextual explanations. While its specific context is a graduate seminar in premodern Japanese language and literature, the ultimate goal of this paper is broader. It details the rationale and technical challenges of building such an AI tutor as a preliminary step towards a system of values, a framework of necessary techniques, and a roadmap for developing AI teaching assistants that remain grounded in the ethical and pedagogical foundations of the humanities.

## 2 Context of Use and Need

The main barrier of entry to a graduate seminar in (and through it to the field of) premodern Japanese studies is the difficulty of understanding the texts. A tutor capable of helping students prepare for class can soften the otherwise steep learning curve. While Handwrittten Text Recognition (HTR) tools available from Japanese institutions can help students as they slowly develop their own paleographic skills, no equivalent tools exist for interpreting the meaning of a text. LLMs are not ready out-of-the-box for pedagogic applications. While they offer vast knowledge, they lack the discipline required for tutoring, often suffering from distractions (context drift), as well as a tendency to over-explain (information dumping) and misinform (hallucinations). In fact, attempts to replace the human instructor with an AI instructor, such as DeepMind's Guided Learning, have failed to engage students effectively.

There is a need for an AI tutor that can enhance the efficacy of a human instructor. The tutor discussed in this paper was specifically designed for students enrolled in formal courses or graduate seminars; it isn't intended for self-taught students or as a replacement for human instruction.

## 3 Design Strategy

Building a tutor that guides students through active sentence analysis and translation can be achieved in different ways. Among possible approaches to customizing an LLM, retraining with high-quality task-specific data (supervised fine-tuning) or expanding its knowledge base (retrieval-augmented generation) is expensive, requires expertise in machine learning, and risks becoming obsolete once a new foundation model is released. By contrast, redesigning the set of instructions (prompt engineering) is an inexpensive, highly scalable approach. It offers a platform that can then be easily adapted (forked) to other contexts of use, languages, and audiences.

Testing in fall 2025 and winter 2026 showed DeepMind's Gemini 3 offers a suitable foundation model. Other models, such as OpenAI's ChatGPT 5.2, showed a tendency to over-explain and rush ahead, breaking the pace and the focus on active learning; xAI's Grok did not have a built-in feature for custom instances.

# 4 Prompt Engineering

The system prompt is structured as a series of instructions appended to any input from the student. It is written in natural language. Short XML tags describe and organize the instructions, making them easier for the model to parse and for a human to modify. The instructions within this system prompt fall into different categories.

## 4.1 General Pedagogy

A general set of directions compels the model to abandon its common function of providing direct answers in favor of the rarer behavior of asking guiding questions. This is achieved through direct statements, such as a section tagged `<persona>` that defines its role as linguistic tutor. A series of nested if/then statements and protocols (sets of rules) set the rhythm of the conversation (`<pace>`), avoid providing information prematurely or overwhelming the student with excessive questions (`<throttling>`), and evaluate user input to adapt the questioning style to the student's specific competence level (`<assessment>`). These constraints protect the pedagogic contract between student and tutor.

## 4.2 Mitigation of General LLM Tendencies

LLMs trained to predict the most probable next token may prioritize probability over factual accuracy. It is necessary to restrict interactions to user-provided phrases (`<input_constraint>`) and attempts at guessing the next sentences of a source text (`<prediction_constraint>`). To avoid an overly polite or chatty attitude that can feel superficial or condescending in a pedagogic setting, we must eliminate vacuous preliminaries and vague knowledge checks (`<greeting_policy>`) as well as meta-commentary, disclaimers, and self-referential statements regarding model limitations (`<style_optimization>`). These constraints ensure the focus remains entirely on the pedagogic task.

## 4.3 Task-Specific Pedagogy

Effective tutoring requires adapting to the sentence provided by the student. The analysis of a vernacular Japanese (*wabun*) sentence should begin with the main verb or predicate, followed by the suffixes (auxiliary verbs or conjunctive particles), and then proceed backwards through the sentence one part of speech at a time. For a Sino-Japanese (*kanbun*) sentence, the tutor will instead offer guidance to identify the Subject-Verb-Object structure, reorder elements into Japanese syntax, add particles, and so on. The student can be asked to offer a translation only after analysis is complete.

## 4.4 Mitigation of Task-Specific LLM Tendencies

LLMs tend to focus on salient differences between premodern and modern forms, such as *kakari-musubi* (distal dependencies where an early particle dictates the final verb's conjuga-

tion), and will often over-emphasize this feature even when a simpler, more proximal connection explains a conjugation. It is necessary to explicitly direct the tutor to prioritize immediate suffix influence over distal binding particles.

### 4.5 Prompt Architecture

While many system instructions are constructive ("If the student does X, do Y"), a significant portion of the prompt is intentionally restrictive ("Do not do Y, ever"). This design choice addresses a counter-intuitive LLM behavior: performance often improves not by adding information, but by enforcing constraints. The efficacy of this restrictive approach finds a striking parallel in Jorge Luis Borges's short story "Funes the Memorious." Borges describes a man who gains perfect memory after a horse riding accident. For Funes, the inability to forget is a curse; his mind is so cluttered with every minute detail of every specific day that he is incapable of abstraction. In Borges's view, the capacity to forget is essential to thinking.

While LLMs do not have perfect memory, they still "remember" so much data that their attention easily wanders; focusing on a task might require "turning off" vast sections of the model's capabilities. An effective system prompt is then less a body of new knowledge than a guide to forgetting as a way to gain focus. Probably related is the finding that shorter prompts have greater efficacy than longer sets of rules or a vast corpus of historical texts offered as reference. Similarly, a system prompt written in English performed best. Japanese language prompts resulted in a higher probability of hallucination and confusion. It is not clear why English might work better as a meta-language. We know tokenizers parse languages other than English less efficiently, and have reason to assume models use English as their pivot language, but these possibilities seem less relevant in this case; it is possible a system prompt in modern Japanese activates the wrong "latent space," routing the analysis of premodern language through modern Japanese neural connections, thereby obscuring historical differences.

## 5 Scalability and Future Directions

The tutor currently performs better on the foundation model Gemini 3 Pro than on the smaller (distilled) version, Gemini 3 Flash. Prior to wider release, the tutor must perform with comparable effectiveness in both Gemini models, as well as on models such as ChatGPT, Claude, or Grok. The main obstacle to cross-model and cross-platform is the result that currently model size has a direct impact on pedagogic effectiveness. To measure and express the capabilities of a model, researchers primarily use three distinct metrics: training tokens (the volume of data consumed during training, or "education"), parameters (the individual neural connections, or "intelligence"), and the context window (the capacity of the working memory at any given time, or "attention span").

### 5.1 Scarcity of data

For historical reasons, the volume of available training tokens for premodern Japanese is significantly lower than that for modern Japanese or English. Developers have less directly relevant data to train a model's internal representation of premodern forms of language.

### 5.2 Loss of Parameters

Because parameters—the weights and biases that store a model's knowledge—are computationally expensive, developers use distillation to create more efficient models. In this process, a larger "teacher" model (such as a Pro model) trains a smaller "student" model (such as a Flash variant) to achieve similar performance with fewer parameters. Distilled models prioritize high-frequency data—such as English, Python, or modern Japanese—while pruning "long tail" data, such as premodern Japanese.

### 5.3 Ambiguity and Attention

Premodern Japanese is characterized by high-context ambiguity; since the subject of a sentence is frequently omitted, accurate interpretation depends on the surrounding text. Smaller models often struggle to retain a specific noun phrase from several sentences prior while simultaneously decoding the complex syntax of the current sentence.

### 5.4 Significance

The combined effect of data scarcity, parameter loss, and context ambiguity makes smaller models less effective. Ultimately, no amount of compute can simulate knowledge that resides in information a model never had, does not possess anymore, or is not momentarily aware of. This technological limitation mirrors our own cognitive processes: our intuition alone cannot reconstruct a world we have never seen, we have let go of, or that escapes our mind. Therein lies the importance of offering younger generations a solid humanities education.

### 5.5 Project Timeframe

| Term | Activity |
|---|---|
| Fall 2025 | Research, prototyping, in-house testing |
| Winter 2026 | Early deployment in Japan 389 (Grad Sem), optimization |
| Spring 2026 | Deployment in Japan 164/264 (Bungo I), customization |
| Summer 2026 | Documentation, intensive testing as forkable open source |
| Fall 2026 | Wide release as portable (cross-model) platform |

## 6 Ethical Considerations

This project engages with artificial intelligence not out of indifference to its disruptions, but because our primary ethical duty as scholars in the face of danger is to understand it. The anxieties provoked by generative AI are well-founded. Automation is reducing the workforce and eliminating the entry-level roles crucial for social mobility. The energy demands of data centers strain local communities and prolong our reliance on fossil fuels. In the workplace, algorithmic surveillance breeds burnout and dissolves trust, while the flood of low-quality, automated content degrades human attention and political agency.

Predictably, many are turning back toward the distinctively human and tangible. It was in this spirit that we established the Stanford Making & Creative Praxis certificate for doctoral and master's students. Yet, to retreat entirely is to lose perspective. Parallel to a hands-on engagement with the material world, we must compel students to explore AI tools as a way to sharpen our understanding of what they can do—both for us and to us.