

Gavin Barnard

Gbbarnard2001@gmail.com

CSI-4130 – Artificial Intelligence

12/1/2025

---

## Japanese Language Tutor Bot

### Abstract

Students who start with no knowledge of Japanese must overcome major challenges to learn the language. The Japanese language contains three writing systems and its sentence structure differs from English while small particles like は and を determine complete sentence meaning. Most language learning applications present example sentences but they fail to explain the sentence structure and natural speech delivery.

The Japanese Language Tutor Bot project involved creating a Streamlit web application which integrates a Large Language Model (Cohere's command-r-08-2024) with text-to-speech functionality (gTTS). The application enables users to enter English text which produces:

- A natural Japanese translation of their input
- The Japanese text appears with both romaji and syllable-by-syllable romaji breakdown
- The application displays an estimated JLPT level which spans from N5 to N1.
- The application delivers short guidance about correct grammar and vocabulary application.
- The application produces full audio sentences which include word emphasis for important words.

The bot underwent testing with both simple statements and statements of average complexity. The system produces acceptable results for new users but it cannot determine their exact JLPT level and it also produces incorrect formatting. The project shows that a fundamental LLM-TTS system gives students learning Japanese as a second language instant useful support.

---

## Introduction

Japanese language differs substantially from English language structure. Students who start learning Japanese need to understand hiragana and katakana scripts and kanji characters and Japanese sentence structure and multiple particles (は, が, を, に, で) and verb endings (～ます). Students who know many words struggle to understand complete sentence structure and proper pronunciation of Japanese sentences.

The current tools for learning Japanese including textbooks and flashcards and numerous apps serve their purpose for memorization but they have several drawbacks. The tools present fixed example sentences to users. The tools provide explanations which become hard to understand because their responses contain extensive details. The system lacks sufficient tools which students need to practice their pronunciation abilities.

Users can employ Large Language Models (LLMs) to create explanations for any sentence they need to understand. The system enables text-to-speech functionality which allows users to listen to their input sentences being read correctly.

The main objective of this project involves developing a basic web application which functions as a Japanese language learning assistant. The Japanese Language Tutor Bot accepts English input from users and produces a Japanese translation along with romaji text and syllable breakdown and JLPT level assessment and brief explanations about essential parts and audio playback. The application functions as a basic educational platform which enables new users to find immediate answers to their Japanese language questions.

---

## Related Work

There are many tools for Japanese learners:

- **Textbooks / grammar websites** – good structure, but fixed sentences and slow to look things up.
- **Flashcard apps (e.g., Anki, WaniKani)** – great for vocabulary and kanji, but not full-sentence explanation.
- **Language apps (e.g., Duolingo, Bunpro)** – interactive but usually follow a set course instead of letting the learner ask about any sentence.
- **General LLM chatbots** – can explain grammar, but the user has to keep asking for translation, romaji, and explanation separately.

My project wraps an LLM in a consistent format: one English sentence in, and one clear, structured response out. This makes it easier for a beginner to use than a raw chatbot.

---

## Data

The project implementation does not require model training. The system operates through two main components which include:

- The Command-R model of Cohere functions through API connections.
- The model receives instructions through a prompt template which defines its response structure.
  - Japanese: ...
  - Romaji: ...
  - Romaji breakdown: ... (with dots between syllables)
  - JLPT: ...
  - Explanation: followed by bullet points like
    - 日本語 (ni•hon•go): means "Japanese language".
- I developed a limited set of test statements which includes “My favorite food is ramen.” and “I want to study Japanese every day.”.

The project design establishes two fundamental elements which consist of the prompt structure and the test sentences.

---

## Methods

### System Design

The system depends on three essential components which must operate as a single unit.

#### 1. Front-end (Streamlit UI)

- The system shows a chat interface which shows user messages and tutor answers.
- The application implements custom CSS styles from the Styles folder to create a Japanese-inspired design through its light color scheme and rounded message bubbles and wave pattern effect.
- The application contains a chat input field which users must access through the bottom section to enter their English sentence.

## 2. Backend Logic

- The application retrieves the COHERE\_API\_KEY from .env through python-dotenv.
- The application creates a Cohere ClientV2 instance using this API key.
- The application produces a prompt which contains user details and asks for particular output types and syllable-based romaji with dots.
- The application executes co.chat with model="command-r-08-2024" and messages=[{"role": "user", "content": prompt}] and max\_tokens=600.

## 3. Parsing and Display

- The application separates the model output into individual lines.
- The application uses st.markdown to display lines that start with Japanese:, Romaji:, Romaji breakdown:, and JLPT:.
- The application keeps all lines starting with - as bullet points after Explanation:.
- The application uses Japanese words before the first ( or : to create audio clips for each bullet point.

## Text-to-Speech (TTS)

The `japanese_tts` function produces MP3 audio files by running `gTTS(text, lang="ja")` when it receives input text.

The application generates two different audio output options.

- The application generates one audio track for the complete Japanese translation.
- The application produces brief audio recordings for all vital words and phrases which appear in the explanation bullets.

The system enables students to listen to complete sentences and specific problem areas within their sentences.


---


The system operates through interactive functions which prioritize user experience over accuracy metrics.

Example: "My favorite food is ramen."

## 翻訳 講師 Translator Tutor

Type any English sentence and get a natural Japanese translation with a short grammar/vocabulary explanation, **romaji** breakdown, and **audio** for the whole sentence plus each key word.

 My favorite food is ramen.







Japanese: 私の好きな食べ物はラーメンです。

Romaji: Watashi no suki na tabemono wa ramen desu.

Romaji breakdown: Wa-ta-shi no su-ki na ta-be-mo-no wa ra-a-men de-su.



JLPT: N4



 Listen to the whole Japanese sentence:



 0:00 / 0:03  



Explanation & word-by-word audio:



- 私 (wa-ta-shi): This means "I" or "my" and is a common way to refer to oneself. 


 0:00 / 0:00 
- 好きな (su-ki na): "Favorite" or "liked" is expressed using this adjective. It's a simple way to describe your preferences. 

 0:00 / 0:00 
- 食べ物 (ta-be-mo-no): This is a noun meaning "food" or "something to eat." 

 0:00 / 0:01 
- ラーメン (ra-a-men): Ramen, a popular Japanese dish, is written and pronounced this way. 

 0:00 / 0:00 
- です (de-su): A polite way to say "is" or "it is," often used in formal or polite speech. 

 0:00 / 0:00 

Type your English sentence... 

The generated sentence appears natural and matches the expected level of complexity for a high-beginner learner.

---

## Conclusion

The Japanese Language Tutor Bot demonstrates how to create an effective mini-tutor through basic code implementation by uniting:

- A Large Language Model (Cohere Command-R)

- A Large Language Model (Cohere Command-R) works together with a well-structured prompt and a basic Streamlit interface and text-to-speech functionality from gTTS.
- The application provides three core features which help beginners learn new languages.
- The system accepts English user input to generate natural Japanese output.
- The system offers users both romaji text and complete syllable-by-syllable romaji breakdowns.
- The system provides an approximate JLPT level assessment.
- The system delivers short definitions about fundamental vocabulary words and fundamental language construction rules.
- The system produces audio playback for entire sentences and specific words which require emphasis.

The system depends on external API access but it produces imperfect JLPT level assessments and lacks features for tracking student progress over time. The system requires additional features which include Japanese input correction and sentence saving for review and better difficulty control and detailed explanation options.

The system fulfills its purpose by providing new learners with instant access to Japanese sentence translations and audio playback and explanation functions.

---

## **Citations and Acknowledgements**

### **Libraries and Tools**

- Streamlit – web interface
- Cohere Python SDK – LLM integration
- gTTS – Japanese text-to-speech
- python-dotenv – environment variable management
- Python 3.x – language used for the app

### **Acknowledgements**

- I want to thank my parents because they backed me through all stages of my testing journey.
- I appreciate the help I received from online Japanese grammar resources which I used to verify my example sentences.