

Reflection on Building the 'Đồn-bà' AI Chatbot: From Data Preparation to Implementation

I. Data Preparation

The first step involved preparing data by performing OCR (Optical Character Recognition) on 20 volumes of *Nữ Giới Chung* (NGC) newspapers. It was crucial to select an optimal OCR model to accurately transcribe the historical language, ensuring minimal error rates, saving human and time resources. ClaudeAI was chosen over other models like Gemini and OpenAI for specific reasons: ClaudeAI offered better token management, cost efficiency, and superior handling of complex, context-rich Vietnamese text. The processing loop was run in Colab, which provided a scalable and accessible environment for handling extensive data processing tasks. Cost analysis revealed how much each OCR operation impacted the budget, with ClaudeAI's per-image token cost offering a balance between performance and affordability.

In our cost analysis for data preparation using Claude AI's OCR, we estimated token usage for processing each page of *Nữ Giới Chung*, which generally required 4 to 6 images. The token cost per image varied by size and accuracy:

- A larger image (1852 x 350 pixels, 560.145 bytes, 97% accuracy) required about 2,000 tokens, split between 850 tokens for the image input and 1,200 tokens for the output.
- A smaller image (1082 x 208 pixels, 265.019 bytes, 78% accuracy) used nearly 1,000 tokens, with 300 tokens for the image input and 650 tokens for the output.

With each page containing multiple images and each volume comprising 24 pages, the token usage resulted in an estimated cost of around \$0.5 per volume. However, recognizing that some images may not yield satisfactory OCR results and may need reprocessing, we allocated a higher budget of \$0.7 per volume to account for potential

inefficiencies. This tolerance ensured that we could maintain high accuracy and reliability in the data preparation phase while staying within budget.

The folder structure was methodically organized to streamline code writing. Each volume of data was segmented and labeled systematically, allowing for efficient retrieval of content during model training. During data preparation, we paid special attention to selecting and including key elements in the dataset that were crucial for accurate and meaningful responses. Specifically, we included direct quotations from *Nữ Giới Chung* writers, publication dates, article topics, and author details. This ensured that the AI could reference specific passages, providing historically accurate and contextually rich answers while avoiding fabricated content. Additionally, we outlined the types of questions the chatbot would address, emphasizing the importance of grounding the responses in real historical content and perspectives.

II. Model Implementation and Enhancement

1. Generative API Integration

In the initial phase, we implemented the "Đồn-bà" AI chatbot using Gemini Pro, a Generative API model, to generate responses based on historical language patterns. This approach allowed for fast and straightforward integration, where we provided a direct prompt instructing the model to respond like a female writer from *Nữ Giới Chung* in 1918 and address questions related to women's issues in that era.

Using Gemini Pro was effective in terms of ease of setup and response speed. It successfully mimicked the language style of a historical writer, capturing the tone and nuances needed to replicate past linguistic patterns. However, as a Large Language Model (LLM), Gemini Pro synthesizes information from a vast global knowledge base, which includes data outside our specific historical dataset. While it can generate natural-sounding responses, it lacks the ability to limit itself to content strictly from *Nữ Giới Chung*, and it cannot directly quote specific passages from our dataset.

This broader synthesis, while typical for LLMs, compromises the historical accuracy and authenticity of the chatbot. By introducing information beyond the dataset, it risks creating content that doesn't accurately reflect the unique viewpoints of the original *Nữ Giới Chung* writers. As a result, while the model can imitate the language perfectly, it cannot maintain the historical integrity required for this project. This limitation highlighted the need for a more controlled approach, leading to the decision to incorporate Retrieval-Augmented Generation (RAG) for accurate, data-specific responses.

Example: Bạn nghĩ gì về vấn đề nữ quyền?

Response made by Gemini Pro

“Nữ quyền, ôi chao, hai chữ nghe sao lạ tai! Phận nữ nhi xưa nay vốn an phận thủ thường, tam tòng tứ đức, lo việc nội trợ, chăm sóc gia đình, ấy là đạo lý. Nay nghe đòi quyền bình đẳng, quả thật chướng tai gai mắt! Nữ giới ta vốn yếu đuối, cần sự che chở của nam nhân, há có thể so bì ngang hàng? Học hành thi cử, chính sự quốc gia, đâu phải việc của nữ nhi? Lo toan việc nhà, giữ gìn gia phong, dạy dỗ con cái nên người, ấy mới là bổn phận. Nữ quyền, e rằng chỉ làm rối loạn trật tự gia đình, xã hội, khiến nữ nhi đánh mất đi nét duyên thâm vốn có. Thiết nghĩ, tu tâm dưỡng tính, vun vén gia đình, mới là hạnh phúc đích thực của nữ giới ta vậy.”

2. RAG Enhancement

To ensure the "Đồn-bà" AI chatbot provides reliable, fact-based answers while maintaining the historical writing style, we used LangChain, a framework that simplifies the process of creating generative AI application interfaces, to implement a sophisticated Retrieval-Augmented Generation (RAG) system. This approach blends the strengths of information retrieval and generative AI, ensuring responses are accurate and contextually rich, including three key steps:

- **Embedding and Database Setup:** We use Chroma, a vector database, integrated with LangChain for efficient storage and retrieval of document embeddings. The embeddings are generated using LangChain's OpenAI module, which employs the "text-embedding-3-large" model from OpenAI. This setup ensures that the system can quickly retrieve relevant chunks of content based on user queries.
- **Document Chunking:** Documents are broken down into smaller, manageable chunks using LangChain's RecursiveCharacterTextSplitter. Each chunk is assigned a unique ID that includes information such as the source file name, page number, and chunk index. This structured approach, facilitated by LangChain, ensures efficient content retrieval and prevents duplication.
- **Query Handling:** The RAG system in LangChain retrieves the most relevant content using a similarity search within Chroma. It then constructs a prompt that combines the user's question with the retrieved content using LangChain's ChatPromptTemplate. This prompt is passed to the GPT-4o model via LangChain's OpenAI integration, which generates a detailed and contextually accurate response. The output includes both the generated answer and cited sources, ensuring historical authenticity and accuracy.

At first, Ollama was selected for embedding, as the dataset's small size (approximately 25 rows) allowed for efficient local processing. However, with plans to scale up to larger datasets, we later shifted to OpenAI's embedding services to benefit from their scalability and performance.

For generating responses, we initially used Claude's Sonnet 3.5. However, we discovered that GPT-4o produced responses with a significantly more natural and fluent quality in Vietnamese, which was essential for the chatbot to deliver historically accurate and contextually rich content. This led us to adopt GPT-4o to enhance the overall user experience in Vietnamese.

Example: Bạn nghĩ gì về vấn đề nữ quyền?

Response made by Claude's Sonnet 3.5

"Thưa quý vị độc giả thân mến, Nữ quyền quả thật là một đề tài gây tranh cãi trong xã hội ta hiện nay. Như tác giả đã viết: "Ít lâu nay các bà cũng có công mũi kim đường chỉ, đèn sách bút nghiên, làm cô giáo cũng có, làm bà mẹ cũng có, mà lắm bà còn muốn thơm danh nức tiếng chẳng quản chi cái thân liễu-yếu đào-tơ; định trèo lên cái thang học-vấn cao-đẳng, để chen vai cùng đám tu-my chúng tôi." Điều này thật đáng mừng cho nữ giới chúng ta, khi phụ nữ ngày càng có cơ hội học tập, làm việc và phát triển bản thân. Tuy nhiên, cũng cần lưu ý rằng: "Thế mai không để lại có lắm bà vin lấy tiếng nữ-quyền rồi lộng quyền quá sức: Con gái thì lấn quyền cha mẹ, muốn ngược muốn xuôi, vợ thì lấn quyền chồng muốn sao muốn vậy." Chúng ta cần phải cẩn trọng, đừng để nữ quyền trở thành cái cớ để phá vỡ nền nếp gia đình, đạo đức xã hội. Nữ quyền cần đi đôi với nữ giáo, để phụ nữ chúng ta vừa có quyền lợi, vừa có trách nhiệm xây dựng gia đình và xã hội tốt đẹp hơn."

Response made by GPT-4o

"Quan điểm của tôi về vấn đề nữ quyền, như đã được trình bày qua nhiều bài viết trên báo Nữ giới chung vào năm 1918, là sự kêu gọi cho một sự bình đẳng giữa nam và nữ trong xã hội. Như tác giả Nguyễn-hồng-Nguyên đã viết vào ngày 3 tháng 5 năm 1918, trong số báo 13, rằng: "Ít lâu nay các bà cũng có công mũi kim đường chỉ, đèn sách bút nghiên... Như thế thời đáng quý hóa thay, đáng mừng rỡ thay cái Nữ-giới nhà ta có lẽ nay mai gần bằng các nước." Quan điểm này cho thấy sự nhận thức về khả năng và tiềm năng của phụ nữ trong việc đạt được những thành tựu ngang bằng với nam giới. Tuy nhiên, cũng có những lo ngại được nêu ra về việc lạm dụng quyền lực, như Nguyễn-hồng-Nguyên đã cảnh báo: "Thế mai không để lại

có lắm bà vin lấy tiếng nữ-quyền rồi lộng quyền quá sức." Điều này chỉ ra rằng, cùng với việc đạt được quyền lợi, cũng cần có trách nhiệm và giáo dục đúng đắn để tránh những hệ quả tiêu cực. Ngoài ra, tác giả S. Nguyệt-Anh, trong số báo 3, ngày 22 tháng 2 năm 1918, đã nhấn mạnh rằng: "Nghĩa nam-nữ Bình-quyền là gì? Là đàn bà cũng đáng hưởng cái quyền lợi như đàn ông vậy." Quan điểm này khẳng định rằng nữ quyền không chỉ là một khái niệm về mặt lý thuyết mà cần được thực hiện trong đời sống thực tế, nơi mà phụ nữ có thể cùng tham gia và góp phần xây dựng xã hội như nam giới."

Additionally, we considered the cost associated with using the OpenAI API for both embedding and generating responses. Each question processed through the system, containing approximately 9,280 words and 44,088 characters, incurs a cost of around \$0.01. This estimate accounts for the use of the OpenAI API key in two places: generating embeddings for content retrieval and executing the query for a final, contextually accurate response.

III. Reference

<https://ngcai.nghiencuuvietsam.com/>

<https://github.com/pixegami/langchain-rag-tutorial>