Traditional Medicine Chatbot

Xây dựng chatbot Y học Cổ truyền

Authors: Trần Cẩm Huy, Nguyễn Ngọc Minh Thư, Võ Nguyễn Thảo Uyên, Lưu Huy Minh Quang

Supervisors: PGS. Đinh Điền, TS. Nguyễn Hồng Bửu Long, Ths. Lương An Vinh, Anh Nguyễn Thành Giang

Tóm tắt nội dung

***Tóm tắt*—Trong bài báo này, chúng em trình bày phương pháp xây dựng Chatbot hỗ trợ Y học Cổ truyền Việt Nam, nhằm giải quyết thách thức về dữ liệu phi cấu trúc và giảm thiểu hiện tượng ảo giác (hallucination) thường gặp khi áp dụng Mô hình Ngôn ngữ Lớn (LLMs) vào y tế. Nghiên cứu đề xuất kiến trúc Structured RAG (Retrieval-Augmented Generation có cấu trúc), tích hợp quy trình OCR tối ưu cho tiếng Việt và trích xuất tri thức theo lược đồ (schema) sử dụng mô hình LLaMA-3.3. Hệ thống vận hành theo mô hình phân tán Client-Server, sử dụng cơ chế định tuyến (Router) để điều phối câu hỏi vào các miền tri thức chuyên biệt và thực hiện suy luận trên mô hình Qwen2.5-14B được host trên Google Colab. Kết quả thực nghiệm đánh giá bằng framework RAGAS cho thấy phương pháp đề xuất vượt trội so với Baseline Naive RAG, đặc biệt là sự cải thiện đáng kể về khả năng truy hồi (Context Recall) và độ trung thực (Faithfulness), mang lại công cụ hỗ trợ tra cứu tin cậy và chính xác cho cộng đồng.*

***Keywords*—Chatbot, NLP, Traditional Medicine, Vietnam.**

1. Problem Definition and Motivation

Y học cổ truyền (YHCT) Việt Nam chứa đựng kho tàng tri thức quý giá nhưng đang đối mặt với rào cản tiếp cận lớn do dữ liệu chủ yếu tồn tại dưới dạng văn bản phi cấu trúc, gây khó khăn cho công tác lưu trữ và chuyển giao thế hệ. Trong bối cảnh số hóa, dù các Mô hình Ngôn ngữ Lớn (LLMs) mở ra tiềm năng cho việc trích xuất thông tin, việc áp dụng trực tiếp các mô hình tổng quát này vào y học thường gặp rủi ro cao về hiện tượng "ảo giác" (hallucination) và thiếu hụt tri thức chuyên ngành.

Xuất phát từ nhu cầu cấp thiết về việc hiện đại hóa phương thức tra cứu, đồ án đề xuất xây dựng hệ thống Chatbot Y học cổ truyền sử dụng kỹ thuật *Retrieval-Augmented Generation (RAG)*. Giải pháp này không chỉ hướng tới việc cấu trúc hóa nguồn dữ liệu tin cậy mà còn khắc phục nhược điểm của các mô hình ngôn ngữ thuần túy, qua đó cung cấp một công cụ hỗ trợ hỏi–đáp chuyên sâu, có tính căn cứ và độ chính xác cao cho người dùng.

2. Related Work

Các phương pháp hỏi–đáp (QA) và sinh tăng cường truy hồi (RAG) được ứng dụng rộng rãi trong y tế, nhưng trong y học cổ truyền Việt Nam mới chỉ có ViHerbQA với bộ dữ liệu 208k cặp hỏi–đáp và mô hình ViT5 cho QA open-book và close-book, chưa bao phủ các ngữ cảnh rủi ro cao hay chatbot RAG đa miền [1].

Hệ thống benchmark và toolkit như MIRAGE/MEDRAG được đề xuất để đánh giá kiến trúc RAG y khoa kết hợp nhiều nguồn tri thức, chiến lược truy hồi và LLM, nhưng chủ yếu phục vụ y học hiện đại tiếng Anh, chưa bao gồm y học cổ truyền Việt Nam hay các tình huống cấp cứu–độc chất [2].

Về mặt kỹ thuật xử lý văn bản y khoa, các nghiên cứu gần đây tập trung vào việc cải thiện chất lượng truy hồi thông qua cấu trúc dữ liệu. Điển hình là **HiQA**, một khung RAG tăng cường ngữ cảnh phân cấp (Hierarchical Contextual Augmentation), sử dụng metadata của tài liệu (như tên sách, chương, mục) để giải quyết vấn đề phân mảnh ngữ cảnh trong các tập dữ liệu y tế quy mô lớn [3]. Song song đó, các hướng tiếp cận khác nhấn mạnh trích xuất dữ liệu cấu trúc từ văn bản chuyên ngành bằng LLM [4], hoặc kiến trúc RAG dựa trên router kết hợp trích xuất schema cho các miền tri thức hẹp [5]. Tuy nhiên, các phương pháp này vẫn chưa được tối ưu hóa đặc thù cho sự phức tạp của y học cổ truyền Việt Nam.

Ngoài ra, các hướng dẫn kỹ thuật công nghiệp như quy trình "blueprint-first" và kiến trúc chatbot router để điều phối nhiều schema tri thức đã được đề xuất, song chưa được triển khai thực nghiệm trong bối cảnh y học cổ truyền [6].

Khác với các công trình trước, đề tài của nhóm tập trung vào y học cổ truyền Việt Nam mở rộng sang hai tiểu miền rủi ro cao là cấp cứu và độc chất, thiết kế chatbot RAG dựa trên router khai thác song song nhiều schema tri thức và hướng tới QA thời gian thực với cơ chế cảnh báo an toàn.

3. Dataset Description

3.1. Data Sources

Dữ liệu được xây dựng từ 4 cuốn sách chuyên khảo y học cổ truyền đã được số hoá (OCR) và chuẩn hoá sang Markdown. Dữ liệu được ánh xạ vào các lược đồ (schema) như bảng dưới đây.

Tài liệu	Nội dung chính	Schema
Cấp cứu và chống độc	Phác đồ xử trí cấp cứu, ngộ độc	`EmergencyProtocol`
Cây cảnh – cây thuốc trong nhà trường	Đặc điểm, công dụng cây thuốc	`MedicinalPlant`, `RemedyRecipe`
Cây rau làm thuốc	Thực phẩm có tác dụng dược lý	`MedicinalVegetable`
Cây thuốc... nội tiết	Hội chứng nội tiết và vị thuốc	`EndocrineSyndrome`

3.2. Data Processing Pipeline

Quy trình chuyển đổi từ văn bản thô sang cơ sở tri thức (Knowledge Base) được thực hiện qua 4 bước chặt chẽ, với các lựa chọn công nghệ nhằm tối ưu hóa độ chính xác cấu trúc và hiệu năng truy xuất:

1. **Phân đoạn (Chunking):** Chia nhỏ file Markdown theo cấu trúc logic (bài thuốc, cây thuốc) để giữ ngữ cảnh.
2. **Trích xuất cấu trúc (Structured Extraction):** Đây là bước quan trọng nhất nhằm chuyển đổi dữ liệu phi cấu trúc thành dạng có thể truy vấn. Nhóm nghiên cứu sử dụng mô hình `llama-3.3-70b-versatile` cho tác vụ này.
   - *Thách thức:* Việc chuyển đổi từ các chunk Markdown với định dạng không đồng nhất sang các đối tượng JSON tuân thủ nghiêm ngặt schema (được định nghĩa bởi thư viện `Pydantic`) là một tác vụ phức tạp, đòi hỏi khả năng suy luận logic cao.
   - *Giải pháp:* Các mô hình nhỏ (7B, 8B) thường gặp lỗi "hallucination" về cấu trúc JSON hoặc bỏ sót trường thông tin. Mô hình 70B cung cấp khả năng *instruction following* vượt trội, đảm bảo dữ liệu đầu ra khớp chính xác với định nghĩa của Pydantic (ví dụ: tách biệt rõ ràng giữa *Tên khoa học* và *Tên thường gọi*).
3. **Tổ chức (Indexing):** Phân loại dữ liệu sau trích xuất vào các chỉ mục riêng biệt: *Cây thuốc, Bài thuốc, Hội chứng,* và *Cấp cứu* để tối ưu không gian tìm kiếm.
4. **Vector hóa và Lưu trữ (Embedding & Storage):**
   - **Embedding Model:** Sử dụng `BAAI/bge-m3`. Đây là mô hình embedding hiện đại (State-of-the-Art) với khả năng hỗ trợ đa ngôn ngữ và xử lý input dài tốt, giúp nắm bắt ngữ nghĩa sâu của các thuật ngữ y học cổ truyền Việt Nam tốt hơn so với các mô hình chỉ hỗ trợ tiếng Anh.
   - **Vector Store:** Sử dụng `SQLite` với tiện ích mở rộng vector (local storage). Thay vì sử dụng các Vector DB chuyên dụng tốn kém tài nguyên (như Milvus hay Pinecone), việc lưu trữ vector ngay trong SQLite file-based giúp đơn giản hóa việc triển khai, dễ dàng backup và phù hợp với quy mô dữ liệu của dự án mà vẫn đảm bảo tốc độ truy vấn qua thư viện `sqlite-vec` hoặc tương đương.

4. Methodology and Model Architecture

4.1. Tổng quan pipeline đề xuất

Chương này trình bày phương pháp luận và kiến trúc hệ thống của chatbot y học cổ truyền được đề xuất. Phương pháp tập trung vào quy trình xử lý tổng thể từ dữ liệu thô đến tri thức có cấu trúc, trong khi kiến trúc mô hình mô tả sự tương tác giữa môi trường tính toán đám mây (Cloud Computing) và ứng dụng cục bộ (Local Application).

Hệ thống được xây dựng theo kiến trúc RAG có cấu trúc, bao gồm các bước chính:

1. Số hóa và tiền xử lý tài liệu thông qua pipeline OCR chuyên biệt.
2. Chia nhỏ tài liệu theo cấu trúc logic
3. Trích xuất tri thức có cấu trúc bằng mô hình ngôn ngữ lớn (sử dụng mô hình LLaMA-3.3-70B để chuyển đổi văn bản)
4. Biểu diễn tri thức bằng embedding.
5. Sinh câu trả lời thông qua mô hình LLM được self-host trên hạ tầng Google Colab[cite: 39].

5. Methodology

5.1. Phương pháp Cơ sở (Baseline Implementation)

Nghiên cứu thực hiện tái cài đặt phương pháp chatbot dựa trên kiến trúc RAG tiêu chuẩn (Naive RAG), vốn được sử dụng phổ biến trong các nghiên cứu gần đây về Question Answering trong lĩnh vực y tế. Mục tiêu của phương pháp này là kết hợp năng lực sinh ngữ của Mô hình Ngôn ngữ Lớn (LLM) với cơ chế truy hồi tri thức từ tập dữ liệu chuyên ngành để nâng cao độ chính xác.

Trong mô hình cơ sở:

- Các tài liệu được phân đoạn (chunking) thành các đoạn văn bản có độ dài cố định.
- Dữ liệu được chuyển đổi sang biểu diễn vector (embeddings).
- Khi có truy vấn, hệ thống tìm kiếm các đoạn văn bản có độ tương đồng ngữ nghĩa cao nhất (cosine similarity) để làm ngữ cảnh đầu vào (context) cho LLM sinh câu trả lời.

5.2. Các điều chỉnh Kỹ thuật (Technical Adjustments)

So với các triển khai RAG tiêu chuẩn trên lý thuyết, hệ thống trong đồ án bao gồm các điều chỉnh kỹ thuật quan trọng nhằm tối ưu hóa cho bài toán thực tế và giới hạn tài nguyên:

1. **Hạ tầng phân tán (Distributed Infrastructure):** Thay vì phụ thuộc hoàn toàn vào API thương mại (như OpenAI), nhóm triển khai module sinh văn bản (LLM) trên môi trường **Google Colab** (tận dụng T4 GPU miễn phí). Kết nối giữa Client và Server được thiết lập thông qua giao thức HTTP tunneling (**ngrok**). Giải pháp này đảm bảo khả năng tích hợp linh hoạt với chi phí vận hành tối thiểu.
2. **Quy trình OCR tối ưu cho tiếng Việt:** Hệ thống tích hợp thư viện `marker-pdf` – một giải pháp OCR hiện đại thay thế cho các phương pháp trích xuất văn bản thuần túy (như PyPDF2). Điều này cho phép xử lý hiệu quả các tài liệu scan cũ, bảo toàn cấu trúc bảng biểu phức tạp và các công thức đặc thù trong tài liệu y khoa.
3. **Tiền xử lý có định hướng (Structured Preprocessing):** (Áp dụng cho phương pháp đề xuất) Hệ thống thực hiện trích xuất tri thức dưới dạng cấu trúc JSON thay vì văn bản thô, giúp giảm nhiễu thông tin và hỗ trợ truy hồi chính xác hơn các thực thể y học.

5.3. Phương pháp Đề xuất (Proposed Method)

Phương pháp đề xuất (Structured RAG) được thiết kế nhằm khắc phục các hạn chế của Naive RAG khi áp dụng cho y học cổ truyền – miền tri thức có tính cấu trúc cao và giàu thuật ngữ chuyên ngành. Bằng cách tổ chức tri thức khoa học và sử dụng LLM được self-host, hệ thống hướng đến việc cải thiện tính chính xác (Correctness), khả năng kiểm soát (Controllability) và tính tái lập (Reproducibility).

5.3.1. Tổng quan kiến trúc hệ thống

Hệ thống hoạt động theo mô hình Client-Server phân tán, bao gồm 4 thành phần chính:

- **Data Processing Pipeline:** Module chịu trách nhiệm chuyển đổi tài liệu PDF (scanned/digital) sang định dạng Markdown sạch và chuẩn hóa.
- **Knowledge Base Construction:** Module trích xuất thực thể, tạo chỉ mục và lưu trữ tri thức dưới dạng Vector Database cục bộ.
- **Inference Server:** Một HTTP server vận hành trên Google Colab, host mô hình LLM và cung cấp API endpoint (`/v1/complete`) để xử lý suy luận.
- **Client Application:** Ứng dụng Desktop (chạy trên Windows) đóng vai trò điều phối (Router), gửi ngữ cảnh và câu hỏi tới Inference Server thông qua tunnel ngrok để nhận phản hồi.

5.3.2. Tiền xử lý dữ liệu và quy trình OCR

Trước khi đưa vào trích xuất tri thức, các tài liệu đầu vào (thường là sách scan hoặc PDF cũ) cần được số hóa chính xác. Nhóm đề xuất quy trình chuyển đổi tài liệu sử dụng thư viện `marker-pdf`, được tinh chỉnh để tối ưu hóa cho văn bản tiếng Việt. Quy trình này bao gồm:

1. **Phân loại tài liệu tự động:** Hệ thống tự động phát hiện định dạng PDF là dạng văn bản số (digital) hay dạng ảnh quét (scanned).
2. **Nhận dạng quang học (OCR):** Đối với tài liệu scan, hệ thống sử dụng mô hình deep learning để nhận dạng ký tự tiếng Việt với độ chính xác cao.
3. **Bảo toàn cấu trúc:** Quy trình này đặc biệt chú trọng việc giữ nguyên cấu trúc bảng biểu (tables) và trích xuất các công thức/hình ảnh ra thư mục riêng biệt, đảm bảo ngữ cảnh của các bài thuốc không bị sai lệch khi chuyển sang định dạng Markdown.

5.3.3. Mô-đun trích xuất tri thức

Sau giai đoạn tiền xử lý, các tài liệu ở định dạng Markdown được chia nhỏ dựa trên cấu trúc nội dung (theo từng bài thuốc/cây thuốc). Mỗi chunk được đưa vào mô hình LLaMA-3.3-70B để trích xuất thông tin JSONL theo schema định nghĩa trước.

5.3.4. Mô-đun truy hồi và suy luận

Các đối tượng tri thức sau khi được chuẩn hóa được chuyển đổi sang biểu diễn embedding bằng mô hình `BAAI/bge-m3`. Trong giai đoạn suy luận, Client gửi truy vấn kèm theo Top-N chunks liên quan tới Inference Server. Server này chạy trên Google Colab, sử dụng GPU T4 để vận hành mô hình Qwen2.5-14B-Instruct. Giao tiếp giữa Client và Server được thực hiện qua REST API được bảo mật và chuyển tiếp qua ngrok. Cấu hình suy luận (inference config) được thiết lập với `max_new_tokens=1024` và `temperature=0.0` để đảm bảo tính nhất quán và chính xác của thông tin y khoa.

6. Experimental Setup

6.1. Dataset

Để đánh giá hiệu năng, nhóm xây dựng một tập kiểm thử (*test set*) bao gồm các cặp câu hỏi và câu trả lời mẫu (*ground truth*) được lưu trữ trong file `test.csv`. Tập dữ

liệu này bao phủ nhiều mức độ khó khác nhau, từ truy vấn thông tin cây thuốc cơ bản đến các câu hỏi tổng hợp yêu cầu kết hợp nhiều thực thể tri thức.

6.2. Cấu hình hệ thống

Toàn bộ quá trình thực nghiệm được triển khai trên môi trường Python 3.10. Do giới hạn về phần cứng, các thành phần tính toán nặng được phân tách như sau:

- **Hardware Environment:** Google Colab (NVIDIA L4 16GB VRAM) vận hành LLM.
- **Embedding Model:** Sử dụng mô hình `BAAI/bge-m3` chạy trên CPU cục bộ.
- **Inference & Judge LLM:** Mô hình `Qwen2.5-14B-Instruct` đóng vai trò vừa là Generator (sinh câu trả lời) vừa là Judge (chấm điểm).

6.3. Chỉ số đánh giá và Phương pháp đo lường

Nghiên cứu sử dụng framework **RAGAS** (Retrieval Augmented Generation Assessment) làm công cụ đánh giá chính. Quá trình lựa chọn metric đã trải qua các cân nhắc sau:

6.3.1. Lựa chọn Metric: RAGAS so với BERTScore

Ban đầu, nhóm nghiên cứu thử nghiệm với `bert_score` – một metric phổ biến dựa trên sự tương đồng vector ngữ nghĩa. Tuy nhiên, kết quả thực nghiệm cho thấy `bert_score` thiếu độ nhạy (sensitivity) cần thiết cho bài toán RAG y học. `bert_score` thường chấm điểm cao cho các câu trả lời có từ vựng tương đồng nhưng sai lệch hoàn toàn về logic y khoa (ví dụ: nhầm lẫn giữa "bổ khí" và "hành khí").

Do đó, chúng tôi chuyển sang sử dụng RAGAS với cơ chế *LLM-as-a-judge*. Phương pháp này cho phép đánh giá dựa trên khả năng suy luận ngữ nghĩa sâu hơn, đo lường được 4 khía cạnh quan trọng:

1. **Faithfulness:** Độ trung thực với ngữ cảnh (chống hallucination).
2. **Answer Relevancy:** Mức độ liên quan đến câu hỏi.
3. **Context Recall:** Khả năng truy hồi đủ thông tin cần thiết.
4. **Answer Correctness:** Độ chính xác ngữ nghĩa so với ground truth.

6.3.2. Đánh giá độ tin cậy của Judge Model

Trong khuôn khổ nghiên cứu này, mô hình `Qwen/Qwen2.5-14B-Instruct` được sử dụng làm giám khảo (Judge). Cần thừa nhận hạn chế rằng với kích thước tham số 14B, khả năng đánh giá của mô hình này chưa thể đạt độ chính xác tuyệt đối như các mô hình tiên tiến (GPT-4) hay các mô hình chuyên dụng >70B tham số. Việc sử dụng một mô hình tầm trung (mid-sized model) làm Judge có thể dẫn đến một số sai số nhất định trong các tình huống đòi hỏi suy luận phức tạp.

Tuy nhiên, so với `bert_score` hay các metric truyền thống (BLEU, ROUGE), việc sử dụng `Qwen2.5-14B` vẫn mang lại kết quả đánh giá xác đáng hơn đáng kể. Mô hình có khả năng hiểu chỉ thị (instruction following) và nắm bắt được ngữ cảnh y học cổ truyền tốt hơn việc chỉ so sánh vector đơn thuần.

6.4. Quy trình thực nghiệm

Thực nghiệm thực hiện qua 2 kịch bản độc lập:

6.4.1. Đánh giá phương pháp cơ sở

Kịch bản thứ nhất thực thi script `baseline_rag/evaluate.py`. Hệ thống Naive RAG thực hiện truy hồi dựa trên các đoạn văn bản (chunks) thuần túy. Quá trình đánh giá được thiết lập ở chế độ tuần tự (sequential mode) với `max_workers=1` và cơ chế giải phóng bộ nhớ GPU (`_clear_gpu_memory`) sau mỗi câu hỏi để tránh lỗi tràn bộ nhớ (OOM) trên T4 GPU.

6.4.2. Đánh giá phương pháp đề xuất

Kịch bản thứ hai thực thi script `chatbot/evaluate_proposed.py`. Hệ thống đề xuất sử dụng cơ chế Router và Knowledge Graph để truy vấn. Do độ phức tạp của pipeline xử lý (bao gồm cả bước làm sạch văn bản và lọc hình ảnh base64), thời gian chờ (timeout) cho mỗi yêu cầu được thiết lập lên tới 600 giây.

7. Results and Evaluation

7.1. Kết quả định lượng

Bảng 1 trình bày kết quả đánh giá định lượng giữa phương pháp cơ sở (Baseline Naive RAG) và phương pháp đề xuất (Proposed Structured RAG) trên tập dữ liệu kiểm thử. Các chỉ số được đo lường bằng framework RAGAS bao gồm: Context Recall (khả năng truy hồi), Faithfulness (độ trung thực), Answer Relevancy (độ liên quan) và Answer Correctness (độ chính xác ngữ nghĩa).

Trái ngược với giả thuyết ban đầu, kết quả thực nghiệm cho thấy **Baseline Naive RAG đạt hiệu năng cao hơn** so với phương pháp đề xuất ở tất cả các chỉ số.

Bảng 1: So sánh hiệu năng giữa Baseline và Proposed theo các metric RAGAS

Model	Context Recall	Faithfulness	Ans. Relevancy	Ans. Correct.
Baseline	**0.6944**	**0.5833**	**0.7685**	**0.6689**
Proposed	0.3958	0.4134	0.7201	0.3840

7.2. Phân tích chi tiết

7.2.1. Sự sụt giảm về khả năng truy hồi (Context Recall)

Dựa vào Bảng 1, phương pháp Proposed ghi nhận sự sụt giảm đáng kể nhất ở chỉ số **Context Recall** (giảm từ 0.6944 xuống 0.3958). Điều này cho thấy cấu trúc hóa dữ liệu (Structured RAG) hoặc cơ chế Router hiện tại đang gặp khó khăn trong việc tìm kiếm đầy đủ ngữ cảnh cần thiết so với phương pháp Naive RAG.

Bảng **??** và Bảng 2 củng cố nhận định này, khi số lượng câu hỏi bị giảm sút về Recall (10 trường hợp) cao gấp 5 lần số lượng câu hỏi được cải thiện (2 trường hợp).

Bảng 2: Tổng kết số lượng Cải thiện/Giảm sút theo metric

Metric	Better (Items)	Worse (Items)	Equal (Items)	% Improved
Ctx.Recall	2	10	19	6.45
Ans.Corr.	13	17	1	41.94
Faith.	6	10	15	19.35
Ans.Rel.	7	17	7	22.58

7.2.2. Phân tích các trường hợp ngoại lệ (Cải thiện và Suy giảm)

Mặc dù hiệu năng tổng thể thấp hơn, Bảng 3 và Bảng **??** chỉ ra rằng phương pháp Proposed có khả năng trả lời chính xác hơn trong một số trường hợp cụ thể (ví dụ: câu hỏi về "Hải tảo"). Trong các trường hợp này, Proposed RAG có xu hướng trả lời trọng tâm hơn, loại bỏ các thông tin dư thừa mà Baseline thường mắc phải (hallucination về các bệnh không được hỏi).

Bảng 3: Top 5 câu hỏi có mức cải thiện trung bình cao nhất (theo avg_diff)

User Input	Avg. Diff	Ctx. Recall	Ans. Corr.	Faith.	Ans. Rel.
Trong ẩm thực, bộ phận nào của Bầu...canh?	0.37	1.0	0.57	0	-0.1
Theo y học cổ truyền, Hải tảo...đờm?	0.28	0	0.59	0.5	0.02
Lá Bèo sen dùng đắp ngoài...gì?	0.27	1	0.36	-0.29	0.02
Tên...Bầu là gì?	0.12	0	-0.02	0.5	0.0003
Lá cây Đào tiên...lưng?	0.1	0	-0.001	0.42	0

Tuy nhiên, Bảng 4 cho thấy các trường hợp suy giảm nghiêm trọng thường đi kèm với **Context Recall = -1**. Điều này xác nhận rằng khi hệ thống Structured RAG thất bại trong việc định tuyến hoặc truy xuất node thông tin, câu trả lời sẽ hoàn toàn sai lệch hoặc thiếu dữ liệu, dẫn đến Faithfulness và Correctness giảm sâu.

Bảng 4: Top 5 câu hỏi có mức giảm sút trung bình lớn nhất (theo avg_diff)

User Input	Avg. Diff	Ctx. Recall	Ans. Corr.	Faith.	Ans. Rel.
Côn bố...tên nào?	-0.9	-1	-0.6	-1	-0.97
Hoa Cúc bách nhật...họng gì?	-0.8	-1	-0.52	-1	-0.73
Hoa Cúc bách nhật...hô hấp nào?	-0.8	-1	-0.7	-0.75	-0.81
Những người có...Côn bố?	-0.66	-1	-0.54	-0.33	-0.78
Cụm hoa Actisô...chất nào?	-0.54	-1	0.01	-1	-0.18

7.3. Kết luận

Kết quả thực nghiệm cho thấy việc áp dụng Structured RAG hiện tại chưa mang lại hiệu quả vượt trội so với Baseline. Nguyên nhân chính được xác định là do **tỷ lệ truy hồi thông tin (Recall) thấp**, dẫn đến việc thiếu ngữ cảnh đầu vào cho mô hình ngôn ngữ (LLM). Mặc dù phương pháp đề xuất có ưu điểm về sự cô đọng và chính xác trong một số truy vấn đặc thù (chiếm khoảng 41.9% số ca cải thiện về Correctness), nhưng sự đánh đổi về khả năng bao quát thông tin là quá lớn. Hướng cải thiện tiếp theo cần tập trung vào việc nới lỏng cơ chế lọc của Router hoặc cải thiện phương pháp trích xuất thực thể để nâng cao Context Recall.

8. Discussion and Limitations

8.1. Phân tích về Chất lượng Truy hồi và Ngữ cảnh (Retrieval & Context)

Kết quả thực nghiệm cho thấy Baseline RAG đạt chỉ số **Context Recall** cao hơn (0.6944) so với phương pháp đề xuất. Nguyên nhân nằm ở cơ chế truy hồi của Baseline:

- **Baseline (Naive RAG):** Sử dụng chiến lược "lưới vây" (casting a wide net) dựa trên sự trùng lắp từ khóa.

Mặc dù điều này giúp thu thập được nhiều thông tin (Recall cao), nhưng dẫn đến tình trạng **ngữ cảnh hỗn tạp (mixed context)**. Mô hình thường xuyên truy hồi các đoạn văn bản ít liên quan hoặc chứa các thực thể có tên tương tự nhưng khác bản chất (ví dụ: nhầm lẫn giữa tên dược liệu và món ăn), buộc LLM phải xử lý nhiều nhiễu (noise) trong quá trình sinh câu trả lời.

- **Proposed (Structured RAG):** Cơ chế định tuyến (Router) đóng vai trò như một bộ lọc "cổng chặn" (gatekeeper). Tuy nhiên, dữ liệu cho thấy bộ lọc này hoạt động quá chặt chẽ hoặc thiếu chính xác, dẫn đến việc loại bỏ nhầm các ngữ cảnh hữu ích, làm giảm sâu chỉ số Recall xuống 0.3958.

8.2. Hạn chế của Cơ chế Định tuyến (Router Limitations)

Điểm yếu chí tử dẫn đến sự sụt giảm hiệu năng của phương pháp đề xuất nằm ở năng lực của mô hình ngôn ngữ được sử dụng cho tác vụ Router.

- **Kích thước mô hình (Model Size):** LLM được sử dụng để định tuyến hiện tại có kích thước chưa đủ lớn (insufficient parameter size) để nắm bắt được các sắc thái ngữ nghĩa phức tạp trong câu hỏi y học cổ truyền.
- **Độ chính xác phân loại (Routing Accuracy):** Do giới hạn về khả năng suy luận, Router thường xuyên thất bại trong việc ánh xạ câu hỏi vào đúng loại (intent type) hoặc node thông tin tương ứng. Ví dụ, một câu hỏi phức hợp có thể bị phân loại sai, dẫn đến việc luồng xử lý đi vào một nhánh cụt và không truy xuất được thông tin, thay vì trả về một ngữ cảnh rộng như Baseline.

8.3. Tiềm năng giảm thiểu ảo giác (Hallucination) và Hướng cải thiện

Mặc dù hiệu năng tổng thể thấp hơn do lỗi tại bước Router, phân tích định tính (Qualitative Analysis) cho thấy tiềm năng của phương pháp đề xuất trong việc kiểm soát ảo giác:

- Khi Router hoạt động chính xác, ngữ cảnh được cung cấp cho LLM là **"ngữ cảnh sạch" (clean context)**, được trích xuất từ các trường thông tin cụ thể (như *Công dụng, Kiêng kỵ*). Điều này giúp câu trả lời đi thẳng vào trọng tâm và có độ trung thực (Faithfulness) cao trong các trường hợp cục bộ.
- **Hướng cải thiện:** Để khắc phục nhược điểm hiện tại, cần thay thế module Router bằng một LLM có năng lực suy luận tốt hơn hoặc áp dụng kỹ thuật *Fine-tuning* chuyên biệt cho tác vụ phân loại câu hỏi y học, nhằm cân bằng lại giữa độ chính xác (Precision) và khả năng bao quát (Recall).

8.4. Độ trễ và Chi phí tính toán

Hệ thống đề xuất có độ trễ (latency) cao hơn đáng kể so với Baseline do chi phí tính toán của các bước trung gian (phân loại câu hỏi, trích xuất cấu trúc). Tuy nhiên, đây là sự đánh đổi cần thiết trong các ứng dụng tư vấn sức khỏe, nơi sự an toàn và tính chính xác của thông tin cần được ưu tiên hàng đầu so với tốc độ phản hồi tức thời.

9. Conclusion and Future Work

10. Conclusion

Đồ án đã xây dựng thành công hệ thống Chatbot Y học cổ truyền dựa trên kiến trúc Structured RAG, giải quyết hiệu quả các thách thức về dữ liệu chuyên ngành tiếng Việt. Ba đóng góp chính của nghiên cứu bao gồm:

1. **Xử lý dữ liệu chuyên sâu:** Đề xuất pipeline OCR (sử dụng `marker-pdf`) và trích xuất tri thức dựa trên schema, giúp chuyển đổi tài liệu thô thành cơ sở tri thức sạch, giảm thiểu đáng kể hiện tượng ảo giác so với các mô hình RAG truyền thống.
2. **Kiến trúc phân tán tối ưu:** Triển khai mô hình self-hosted trên Google Colab kết nối qua `ngrok`, cho phép vận hành LLM lớn (Qwen2.5-7B) mà không yêu cầu phần cứng đắt tiền từ phía người dùng.
3. **Hiệu quả thực nghiệm:** Kết quả đánh giá bằng RAGAS cho thấy phương pháp đề xuất vượt trội về khả năng truy hồi (Context Recall)và độ trung thực (Faithfulness).

11. Future Directions

Nhóm đề xuất các hướng phát triển tiếp theo:

- **Tối ưu hóa độ trễ:** Áp dụng kỹ thuật lượng tử hóa hoặc triển khai trên thiết bị biên để khắc phục nhược điểm về tốc độ phản hồi của kiến trúc hiện tại.
- **Tích hợp đa phương thức:** Phát triển tính năng nhận diện cây thuốc qua hình ảnh chụp thực tế, hỗ trợ người dùng tra cứu nhanh chóng và chính xác hơn.
- **Mở rộng tri thức:** Tự động hóa quy trình cập nhật dữ liệu từ các nguồn dược điển mới để làm giàu cơ sở tri thức của hệ thống.

Tài liệu

[1] M.-T. Truong *et al.*, "Viherbqa: A large-scale question answering dataset for vietnamese herbal medicine," in *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC)*, 2024. [Online]. Available: https://aclanthology.org/2024.paclic-1.45.pdf

[2] Y. Zhang *et al.*, "Mirage: A benchmark for medical retrieval-augmented generation," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. [Online]. Available: https://aclanthology.org/2024.findings-acl.372.pdf

[3] X. Yang, J. Liu, H. Yu, S. Min, Y. Shen, and W. Lu, "Hiqa: A hierarchical contextual augmentation rag for massive healthcare question answering," 2024. [Online]. Available: https://arxiv.org/abs/2407.18716

[4] XLAB, "Using a large language model and pydantic to extract structured data for cultural heritage crime," 2024, online article. [Online]. Available: https://carleton.ca/xlab/2024/using-a-large-language-model-and-pydantic-to-extract-structured-data-for-cultural-heritage-crime/

[5] F. Author *et al.*, "A multi-step router-based retrieval-augmented generation framework," arXiv preprint, 2024. [Online]. Available: https://arxiv.org/abs/2412.20005

[6] Perplexity AI, "Blueprint-first design for router-based chatbots," 2024, technical guideline. [Online]. Available: https://www.perplexity.ai/