HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Hoàng Đức Việt

XÂY DỰNG HỆ THỐNG TRA CỨU CHÍNH SÁCH PHÁP LUẬT DỰA TRÊN MÔ HÌNH NGÔN NGỮ LỚN

ĐỀ ÁN TỐT NGHIỆP THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – NĂM 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Hoàng Đức Việt

XÂY DỰNG HỆ THỐNG TRA CỨU CHÍNH SÁCH PHÁP LUẬT DỰA TRÊN MÔ HÌNH NGÔN NGỮ LỚN

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

ĐỀ ÁN TỐT NGHIỆP THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. Nguyễn Tất Thắng

HÀ NỘI – NĂM 2025

LỜI CAM ĐOAN

Tôi cam đoan rằng đề án tốt nghiệp thạc sĩ: ***"XÂY DỰNG HỆ THỐNG TRA CỨU CHÍNH SÁCH PHÁP LUẬT DỰA TRÊN MÔ HÌNH NGÔN NGỮ LỚN"*** là công trình nghiên cứu của chính tôi.

Tôi cam đoan các số liệu, kết quả nêu trong đề án là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.
Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong đề án này mà không được trích dẫn theo đúng quy định.

Hà Nội, ngày 05 tháng 12 năm 2025

Học viên thực hiện đề án

Hoàng Đức Việt

LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện đề án tốt nghiệp thạc sĩ, ngoài nỗ lực của bản thân, tôi đã nhận được sự hướng dẫn nhiệt tình quý báu của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới:

Ban Giám Đốc, Phòng đào tạo sau đại học và quý Thầy Cô đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành đề án.

Tôi xin chân thành cảm ơn Thầy TS. Nguyễn Tất Thắng, người thầy kính yêu đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành đề án tốt nghiệp thạc sĩ.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành đề án. Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

Hà Nội, ngày 05 tháng 12 năm 2025

Học viên thực hiện đề án

Hoàng Đức Việt

MỤC LỤC

MỤC LỤC iii
DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT v
DANH SÁCH HÌNH VẼ vi
DANH SÁCH BẢNG vii
I. MỞ ĐẦU 8
1. Lý do chọn đề tài 8
2. Mục tiêu nghiên cứu 8
3. Đối tượng và phạm vi nghiên cứu 9
3.1 Đối tượng nghiên cứu 9
3.2 Phạm vi nghiên cứu 10
4. Phương pháp nghiên cứu 12
4.1 Phương pháp nghiên cứu lý thuyết 12
4.2 Phương pháp nghiên cứu thực nghiệm 13
4.3 Ý nghĩa và đóng góp của đề tài 14
4.4 Bố cục đề án 14
II. NỘI DUNG 16
CHƯƠNG 1: CƠ SỞ LÝ LUẬN VÀ TỔNG QUAN VỀ HỆ THỐNG HIỆN CÓ 16
1.1 Tổng quan về xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) 16
1.2 Tổng quan về mô hình ngôn ngữ lớn (Large Language Model - LLM) 18
1.3 Tổng quan về kỹ thuật RAG (Retrieval-Augmented Generation) 22
1.4 Khái quát các thành phần chính trong hệ thống RAG pháp lý 24
1.5 Các công trình nghiên cứu liên quan 25
CHƯƠNG 2: XỬ LÝ VÀ TỔ CHỨC NGUỒN DỮ LIỆU 30
2.1 Nguồn dữ liệu chính thức và tiêu chí lựa chọn 30
2.2. Quy trình thu thập tự động (web crawling) từ vbpl.vn 32

2.3. Tiền xử lý văn bản pháp luật.....33
2.4. Phân đoạn văn bản (chunking) và tạo chunk ID.....34
2.5. Sinh embedding và tổ chức dữ liệu trong vector store.....35
2.6. Đánh giá dữ liệu và đảm bảo chất lượng.....36
CHƯƠNG 3: THIẾT KẾ KIẾN TRÚC VÀ CÁC THÀNH PHẦN CỦA HỆ THỐNG.....37
3.1. Lựa chọn công nghệ và mô hình thiết kế.....37
3.2. Mô hình tổng thể của hệ thống chatbot pháp luật.....42
3.3. Thiết kế các thành phần của hệ thống.....45
3.4. Minh họa các chức năng chính của ứng dụng.....49
CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ.....51
4.1. Môi trường mô phỏng thực nghiệm.....51
4.2. Dữ liệu thực nghiệm.....52
4.3. Các kịch bản thực nghiệm.....53
4.4. Đánh giá hiệu năng của hệ thống.....54
4.5. Thực nghiệm.....54
PHẦN KẾT LUẬN.....57
1. Kết quả nghiên cứu của đề tài.....57
2. Hạn chế của đề tài.....58
3. Nhận xét, đề xuất và khuyến nghị.....58
4. Hướng nghiên cứu tiếp theo.....59
TÀI LIỆU THAM KHẢO.....61

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

VIẾT TẮT	NGHĨA TIẾNG ANH	NGHĨA TIẾNG VIỆT
ATTP		An toàn thực phẩm
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
RAG	Retrieval-Augmented Generation	Tạo sinh dựa trên truy xuất tăng cường
SME	Small and Medium Enterprise	Doanh nghiệp vừa và nhỏ
UI	User Interface	Giao diện người dùng
AI	Artificial Intelligence	Trí tuệ nhân tạo
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
GPT	Generative Pre-trained Transformer	Mô hình Transformer sinh văn bản tiền huấn luyện
API	Application Programming Interface	Giao diện lập trình ứng dụng
RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
LSTM	Long Short-Term Memory	Mạng nơ-ron nhớ dài–ngắn hạn
UTF	Unicode Transformation Format	Định dạng chuyển đổi Unicode

DANH SÁCH HÌNH VẼ

Hình 2.1 Cơ cấu phân bổ các loại dữ liệu của nguồn dữ liệu.....31
Hình 2.2 Mã nguồn sử dụng thư viện BeautifulSoup để trích xuất dữ liệu 33
Hình 2.3 Mã nguồn phân đoạn văn bản.....35
Hình 2.4 Mã nguồn Local Embeding 35
Hình 3.1 Minh họa luồng hoạt động của hệ thống 42
Hình 3.2 Màn hình đăng nhập của ứng dụng 49
Hình 3.3 Màn hình tương tác hỏi đáp của ứng dụng.....49
Hình 3.4 Màn hình quản lý tài liệu.....50
Hình 4.1 Mã nguồn thực nghiệm không dùng RAG 56

DANH SÁCH BẢNG

Bảng 2.1 Cơ cấu thành phần dữ liệu.....31
Bảng 3.1 Kết quả thử nghiệm mô hình ngôn ngữ lớn 37
Bảng 4.1 Kết quả thực nghiệm 55

I. MỞ ĐẦU

1. Lý do chọn đề tài

An toàn thực phẩm là một lĩnh vực quan trọng tại Việt Nam. Hệ thống pháp luật trong lĩnh vực này tương đối hoàn thiện, với trung tâm là Luật An toàn thực phẩm số 55/2010/QH12 (sửa đổi, bổ sung năm 2025) cùng hơn 170 văn bản dưới luật bao gồm nghị định, thông tư, quyết định và thông tư liên tịch. Tuy nhiên, việc tra cứu đối với doanh nghiệp SME và người dân vùng sâu vùng xa gặp trở ngại lớn do ngôn ngữ pháp lý phức tạp và hạ tầng internet thiếu ổn định.

Nguyên nhân chính bắt nguồn từ đặc thù ngôn ngữ pháp lý phức tạp, khối lượng văn bản đồ sộ, tần suất sửa đổi cao, cùng sự thiếu vắng các công cụ tra cứu thông minh bằng tiếng Việt. Điều này dẫn đến khoảng cách đáng kể giữa quy định pháp luật và khả năng tuân thủ trong thực tế. Hậu quả là các vụ ngộ độc thực phẩm tập thể, vi phạm quy định về vệ sinh an toàn thực phẩm vẫn diễn ra thường xuyên, gây thiệt hại lớn về sức khỏe cộng đồng và kinh tế.

Trong bối cảnh đó, sự phát triển mạnh mẽ của Mô hình Ngôn ngữ Lớn (Large Language Models – LLMs) và kỹ thuật Retrieval-Augmented Generation (RAG) đã mở ra hướng tiếp cận mới cho lĩnh vực pháp lý: xây dựng trợ lý ảo pháp luật hoạt động ngoại tuyến (offline), có khả năng hiểu câu hỏi tiếng Việt tự nhiên, truy xuất chính xác điều khoản liên quan và cung cấp câu trả lời có dẫn chiếu nguồn rõ ràng, đáng tin cậy.

2. Mục tiêu nghiên cứu

Mục tiêu tổng quát: Nghiên cứu và phát triển một hệ thống chatbot RAG ngoại tuyến, chuyên biệt trả lời câu hỏi về pháp luật trong lĩnh vực an toàn thực phẩm tại Việt Nam, với đạt độ chính xác và tốc độ phản hồi đáp ứng yêu cầu và có khả năng cập nhật tri thức linh hoạt mà không cần kết nối Internet.

Mục tiêu cụ thể: Tập trung vào việc thiết kế và phát triển cơ sở tri thức pháp luật về an toàn thực phẩm từ hơn 170 văn bản quy phạm pháp luật chính thức. Tiếp theo, xây dựng pipeline RAG ngoại tuyến sử dụng mô hình nhúng BAAI/bge-small-en-v1.5 và mô hình ngôn ngữ lớn cục bộ Qwen2.5:7B-Instruct-

Q4_0 thông qua nền tảng Ollama. Đề tài cũng triển khai cơ chế đồng bộ delta (delta-synchronization) với ID chunk ổn định dựa trên hàm băm SHA-256, nhằm cho phép cập nhật văn bản mới hoặc sửa đổi trong thời gian dưới 25 giây. Ngoài ra, phát triển giao diện quản trị web dựa trên khung Flask và chatbot dòng lệnh với tính năng trích dẫn nguồn tự động. Cuối cùng, đánh giá hiệu năng hệ thống thông qua các bộ chỉ số đầu vào.

3. Đối tượng và phạm vi nghiên cứu

3.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu là một hệ thống phần mềm hoàn chỉnh, được thiết kế nhằm hỗ trợ tra cứu, giải thích và cung cấp căn cứ pháp lý chính xác trong lĩnh vực an toàn thực phẩm tại Việt Nam. Cụ thể, đề án tập trung phát triển một ứng dụng chatbot hỏi đáp pháp lý dựa trên kiến trúc Retrieval-Augmented Generation (RAG), vận hành hoàn toàn trong môi trường ngoại tuyến (offline). Toàn bộ dữ liệu văn bản pháp lý, chỉ mục vector cũng như mô hình ngôn ngữ được lưu trữ và xử lý trên máy cục bộ hoặc máy chủ nội bộ, không phụ thuộc vào các dịch vụ API đám mây, nhằm đảm bảo tính chủ động, bảo mật và khả năng triển khai trong môi trường hạn chế kết nối Internet.

Về mặt kiến trúc kỹ thuật, ứng dụng bao gồm các thành phần chính như: thu thập, tiền xử lý và chuẩn hóa; mô-đun chia đoạn văn bản (chunking) có chồng lấn nhằm bảo toàn ngữ cảnh, đồng thời gán metadata chi tiết cho từng đoạn như số điều, khoản, điểm, ngày ban hành và tình trạng hiệu lực. Các đoạn văn bản sau khi xử lý được chuyển đổi thành vector embedding và lưu trữ trong cơ sở dữ liệu vector (vector store). Khi người dùng đặt câu hỏi, mô-đun truy xuất thông tin thực hiện tìm kiếm tương đồng vector để lựa chọn các đoạn văn bản liên quan nhất; Mô-đun sinh ngôn ngữ sử dụng các mô hình ngôn ngữ lớn mã nguồn mở, có khả năng tinh chỉnh (fine-tuning), nhằm tạo ra câu trả lời bằng tiếng Việt kèm theo trích dẫn rõ ràng tới các điều khoản pháp luật liên quan. Giao diện người dùng của hệ thống được thiết kế ở mức tối giản, dưới dạng chatbot hoặc giao

diện dòng lệnh, hỗ trợ hiển thị đồng thời nội dung trả lời, đoạn văn bản nguồn, metadata và liên kết trích dẫn để người dùng có thể kiểm chứng thông tin.

Bên cạnh việc xây dựng hệ thống, đề án tập trung thiết lập một quy trình đánh giá hiệu năng thực nghiệm dựa trên bộ dữ liệu kiểm thử chuẩn (Ground-truth) gồm các kịch bản truy vấn thực tế. Thay vì các đánh giá định tính thông thường, chất lượng hệ thống được định lượng cụ thể thông qua chỉ số Recall@5 để đo lường khả năng truy xuất chính xác các đoạn văn bản luật và chỉ số Faithfulness (độ trung thực) để kiểm soát hiện tượng ảo giác của mô hình ngôn ngữ. Quá trình đánh giá được thực hiện kết hợp giữa phương pháp tự động và thẩm định từ chuyên gia dựa trên thang đo Likert 5 điểm, tập trung vào tính khớp lệnh giữa câu trả lời sinh ra với nguồn trích dẫn pháp lý gốc.

Trên cơ sở đó, mục tiêu cốt lõi của đề án là thiết kế và triển khai một mô hình thử nghiệm (prototype) hệ thống RAG ngoại tuyến có khả năng xử lý ngôn ngữ tự nhiên tiếng Việt ổn định, đảm bảo tính bảo mật và quyền riêng tư tuyệt đối cho người dùng. Thông qua việc phân tích chuyên sâu các kết quả thực nghiệm về độ chính xác trích dẫn (Citation Accuracy) và thời gian phản hồi hệ thống, đề án cung cấp các lập luận khoa học về việc lựa chọn mô hình nhúng cũng như chiến lược phân đoạn văn bản, từ đó đề xuất lộ trình tối ưu hóa kỹ thuật nhằm nâng cao tính khả thi của hệ thống khi triển khai thực tế tại các khu vực hạn chế về hạ tầng mạng.

3.2 Phạm vi nghiên cứu

Phạm vi nghiên cứu được xác định ở bốn chiều: không gian nội dung, thời gian, nguồn dữ liệu và công nghệ triển khai, cùng các ràng buộc và ngoại lệ. Về phạm vi nội dung, đề án tập trung nghiên cứu các văn bản quy phạm pháp luật có liên quan trực tiếp đến lĩnh vực an toàn thực phẩm tại Việt Nam. Phạm vi này bao gồm các luật, nghị định, thông tư, quyết định và các văn bản hướng dẫn thi hành liên quan đến hoạt động quản lý nhà nước, sản xuất, chế biến, kiểm tra, kiểm nghiệm và lưu thông thực phẩm.

Về phạm vi thời gian, xử lý các văn bản pháp luật còn hiệu lực hoặc còn hiệu lực một phần, được ban hành tính đến thời điểm tháng 12 năm 2025. Mặc dù hệ thống có khả năng tiếp nhận và cập nhật các văn bản mới phát sinh sau mốc thời gian này thông qua cơ chế ingest dữ liệu, song các thí nghiệm, đánh giá và kết luận trình bày trong luận văn chỉ dựa trên trạng thái pháp lý của hệ thống tại thời điểm cập nhật đến tháng 12/2025. Việc giới hạn mốc thời gian nhằm đảm bảo tính nhất quán và khả năng tái lập của các kết quả nghiên cứu.

Về phạm vi dữ liệu, nguồn dữ liệu chính của đề án là tập hợp khoảng 170 văn bản pháp luật chính thức được thu thập từ website vbpl.vn thuộc Bộ Tư pháp hoặc các nguồn chính thức tương đương. Các văn bản được lựa chọn dựa trên tiêu chí có các từ khóa "an toàn thực phẩm" và được công bố công khai; các văn bản mang tính nội bộ, hồ sơ vụ việc cụ thể hoặc chưa được công bố rộng rãi được loại trừ khỏi tập dữ liệu nghiên cứu. Bên cạnh đó, trong trường hợp cần thiết và được phép sử dụng, đề án có thể bổ sung một số tài liệu chuyên môn như quy chuẩn kỹ thuật, tiêu chuẩn quốc gia hoặc tài liệu hướng dẫn chuyên ngành có liên quan chặt chẽ đến an toàn thực phẩm. Mỗi văn bản trong tập dữ liệu được gán các metadata tối thiểu bao gồm tên văn bản, loại văn bản, ngày ban hành, ngày hiệu lực, … nhằm phục vụ cho việc truy xuất, trích dẫn và kiểm chứng thông tin.

Về phạm vi công nghệ và triển khai, đề án giới hạn việc sử dụng các mô hình ngôn ngữ và các thành phần tìm kiếm vector trong môi trường máy cá nhân hoặc máy chủ nội bộ, không phụ thuộc vào các dịch vụ API đám mây thương mại. Cách tiếp cận này nhằm đáp ứng yêu cầu về bảo mật thông tin cũng như khả năng vận hành trong các khu vực có điều kiện kết nối Internet hạn chế. Các công nghệ được lựa chọn đa phần là mã nguồn mở hoặc các giải pháp có bản quyền miễn phí, nhằm giảm chi phí triển khai. Về tài nguyên phần cứng, hệ thống được thiết kế để có thể vận hành trên máy tính cá nhân với cấu hình trung bình; trong trường hợp sử dụng các mô hình ngôn ngữ có kích thước lớn, hệ thống có thể được triển khai bổ sung trên máy chủ nội bộ có trang bị GPU.

Về phạm vi người dùng mục tiêu, hệ thống hướng tới phục vụ các nhóm đối tượng thường xuyên có nhu cầu tra cứu và áp dụng nhưng gặp hạn chế trong việc tiếp cận thông tin. Dựa trên đặc điểm này, hệ thống được yêu cầu có giao diện đơn giản, hỗ trợ tư vấn bằng ngôn ngữ tự nhiên tiếng Việt, hiển thị rõ ràng các trích dẫn điều khoản pháp luật liên quan và có khả năng hoạt động hoàn toàn ngoại tuyến.

Bên cạnh các phạm vi nêu trên, đề án cũng xác định rõ các ràng buộc và ngoại lệ của nghiên cứu. Cụ thể, hệ thống được xây dựng với vai trò hỗ trợ tra cứu và tham khảo thông tin pháp lý, không thay thế hoạt động tư vấn pháp lý chuyên sâu của luật sư và không đưa ra các quyết định pháp lý chính thức hay thay mặt cơ quan quản lý nhà nước. Về mặt pháp lý, hệ thống chỉ xử lý và cung cấp thông tin từ các nguồn văn bản công khai, tuân thủ đầy đủ các quy định về bản quyền và điều khoản sử dụng dữ liệu. Ngoài ra, các kết quả đánh giá và thực nghiệm trong đề án chỉ phản ánh chất lượng hệ thống trên tập dữ liệu và bộ câu hỏi thử nghiệm đã được xác định; hiệu năng thực tế của hệ thống có thể thay đổi khi phạm vi dữ liệu được mở rộng hoặc khi mô hình ngôn ngữ được tinh chỉnh thêm.

Cuối cùng, đề án làm rõ mối liên hệ giữa các phạm vi nghiên cứu, trong đó việc lựa chọn công nghệ triển khai theo hướng ngoại tuyến và mã nguồn mở chịu ảnh hưởng trực tiếp từ đặc thù của nhóm người dùng mục tiêu và các yêu cầu về bảo mật thông tin. Việc giới hạn phạm vi thời gian đến tháng 12/2025 giúp đảm bảo tính nhất quán cho quá trình thực nghiệm, đồng thời đặt ra yêu cầu xây dựng quy trình cập nhật dữ liệu để hệ thống có thể được duy trì và mở rộng trong dài hạn.

4. Phương pháp nghiên cứu

4.1 Phương pháp nghiên cứu lý thuyết

Trước hết, đề tài tiến hành thu thập, phân tích và tổng hợp các tài liệu liên quan đến kiến trúc RAG, các mô hình ngôn ngữ lớn, phương pháp nhúng vector, kỹ thuật đánh chỉ mục và truy vấn tri thức, cũng như các đặc điểm khi ứng dụng

mô hình sinh trong ngữ cảnh pháp lý. Việc nghiên cứu này giúp hình thành hiểu biết vững chắc về quy trình xây dựng hệ thống hỏi đáp dựa trên tri thức và xác định các thành phần cần thiết cho kiến trúc ngoại tuyến.

Song song đó, đề tài tổng hợp những đặc thù của ngôn ngữ pháp luật tiếng Việt, đặc biệt là cấu trúc câu phức tạp, tính chuẩn hóa cao của thuật ngữ pháp lý và sự chặt chẽ trong diễn đạt các điều khoản pháp luật về an toàn thực phẩm. Việc nắm rõ đặc điểm này cho phép lựa chọn phương pháp tách đoạn, chuẩn hóa dữ liệu và đánh giá tính chính xác pháp lý một cách phù hợp. Tiếp theo, đề tài xem xét yêu cầu triển khai ngoại tuyến trong môi trường không có Internet hoặc có kết nối hạn chế, từ đó xác định những tiêu chí bắt buộc trong thiết kế hệ thống: mô hình phải chạy hoàn toàn cục bộ, dung lượng phù hợp với thiết bị phổ thông và có khả năng cập nhật tri thức định kỳ mà không cần truy cập mạng.

Ngoài ra, nghiên cứu lý thuyết còn tập trung vào các hướng tối ưu hóa hệ thống RAG dành cho môi trường hạn chế tài nguyên, chẳng hạn như tối ưu mô hình embedding tiếng Việt, rút gọn lượng truy vấn cần thiết, cải thiện tốc độ suy luận và giảm tiêu thụ bộ nhớ. Những nghiên cứu này đóng vai trò quan trọng trong việc điều chỉnh giải pháp kỹ thuật để đảm bảo hệ thống vận hành hiệu quả tại địa phương.

4.2 Phương pháp nghiên cứu thực nghiệm

Phương pháp thực nghiệm được triển khai nhằm kiểm chứng khả năng hoạt động của mô hình và đánh giá hiệu quả của hệ thống trong các tình huống thực tế. Trước hết, đề tài xây dựng bộ dữ liệu pháp luật về an toàn thực phẩm bằng cách thu thập các văn bản chính thống, thực hiện làm sạch, chuẩn hóa nội dung, sửa lỗi và loại bỏ các chi tiết không cần thiết. Dữ liệu sau đó được phân chia thành các đoạn nhỏ tương ứng với điều, khoản, điểm nhằm đảm bảo khả năng truy xuất chính xác, đồng thời được chuyển đổi thành vector thông qua mô hình embedding tiếng Việt để tạo kho tri thức dạng vector.

Tiếp theo, đề tài thử nghiệm nhiều mô hình ngôn ngữ lớn cục bộ với kích thước, bối cảnh (context window) và yêu cầu tài nguyên khác nhau nhằm xác

định mô hình tối ưu cho việc xử lý câu hỏi pháp lý tiếng Việt. Quá trình thử nghiệm bao gồm đánh giá tốc độ phản hồi, độ chính xác, mức sử dụng RAM/VRAM và khả năng vận hành ổn định trên các thiết bị có cấu hình phổ thông. Trên cơ sở lựa chọn mô hình phù hợp, đề tài xây dựng nguyên mẫu hệ thống RAG ngoại tuyến đầy đủ, bao gồm các giai đoạn xử lý dữ liệu, lập chỉ mục, truy vấn vector, tái xếp hạng kết quả và sinh câu trả lời. Cơ chế đồng bộ tri thức theo dạng delta được tích hợp để đảm bảo quá trình cập nhật văn bản pháp luật diễn ra nhanh chóng mà không phải tái tạo toàn bộ cơ sở dữ liệu.

Để đánh giá chất lượng hệ thống, đề tài xây dựng bộ câu hỏi phản ánh các tình huống nghiệp vụ thực tế của cán bộ tư pháp trong lĩnh vực an toàn thực phẩm. Hệ thống được kiểm tra dựa trên ba tiêu chí: độ chính xác pháp lý, tính đầy đủ của câu trả lời và mức độ trung thành với văn bản gốc. Các kết quả được phân tích nhằm xác định điểm mạnh, hạn chế và điều chỉnh mô hình cho phù hợp. Cuối cùng, toàn bộ kết quả thực nghiệm được tổng hợp và trình bày nhằm minh chứng cho tính khả thi, tính hiệu quả và khả năng ứng dụng thực tiễn của mô hình.

4.3 Ý nghĩa và đóng góp của đề tài

Về ý nghĩa khoa học, đề án góp phần mở rộng hướng nghiên cứu ứng dụng kiến trúc Retrieval-Augmented Generation (RAG) trong lĩnh vực pháp lý, với trọng tâm là ngôn ngữ tiếng Việt và yêu cầu vận hành hoàn toàn ngoại tuyến. Đề án cũng đề xuất một giải pháp cập nhật tri thức theo cơ chế đồng bộ gia tăng (delta update) dựa trên định danh đoạn văn bản ổn định (stable chunk ID).

Về ý nghĩa thực tiễn, hệ thống cung cấp công cụ miễn phí, dễ triển khai tại các xã, huyện, trạm y tế, hợp tác xã – nơi không có Internet ổn định, góp phần giảm tải cho cán bộ tư pháp, thanh tra, tăng cường tuân thủ pháp luật cho đối tượng sử dụng.

4.4 Bố cục đề án

Đề án được chia thành 4 chương:

Chương 1: Cơ sở lý luận và tổng quan các hệ thống hiện có

Chương 2: Xử lý và tổ chức nguồn dữ liệu

Chương 3: Thiết kế kiến trúc và các thành phần của hệ thống

Chương 4: Thực nghiệm và đánh giá kết quả

II. NỘI DUNG

CHƯƠNG 1: CƠ SỞ LÝ LUẬN VÀ TỔNG QUAN VỀ HỆ THỐNG HIỆN CÓ

1.1 Tổng quan về xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP)

1.1.1 Khái niệm

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một lĩnh vực của trí tuệ nhân tạo (AI) và khoa học máy tính, tập trung vào việc nghiên cứu và phát triển các hệ thống có khả năng hiểu, phân tích, và sinh ngôn ngữ của con người dưới dạng văn bản hoặc lời nói. NLP được hình thành từ nền tảng của ngôn ngữ học tính toán, nhưng thay vì chỉ xây dựng lý thuyết, NLP hướng đến việc tạo ra các ứng dụng thực tiễn phục vụ đời sống và sản xuất.

1.1.2 Các nhánh chính của NLP

NLP thường được chia thành hai nhánh chính:

- Natural Language Understanding (NLU): tập trung vào việc phân tích ngữ nghĩa, ngữ pháp và ngữ cảnh để xác định ý nghĩa của văn bản.
- Natural Language Generation (NLG): tập trung vào việc sinh ngôn ngữ tự nhiên từ dữ liệu hoặc mô hình, tạo ra văn bản có tính mạch lạc và tự nhiên.

Ngoài ra, NLP thường được kết hợp với nhận dạng giọng nói (Speech Recognition) để xử lý ngôn ngữ nói, chuyển đổi giữa âm thanh và văn bản.

1.1.3 Vai trò và tầm quan trọng

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) ngày càng đóng vai trò quan trọng và trở thành một thành phần không thể thiếu trong nhiều lĩnh vực của đời sống và kinh tế – xã hội. Trong lĩnh vực thương mại và dịch vụ, NLP được ứng dụng rộng rãi trong các hệ thống chatbot, trợ lý ảo và chăm sóc khách hàng tự động. Trong y tế, các kỹ thuật NLP hỗ trợ phân tích hồ sơ bệnh án điện tử và góp phần nâng cao hiệu quả chẩn đoán. Trong giáo dục, NLP được sử dụng trong các hệ thống chấm điểm tự động và công cụ hỗ trợ kiểm tra, cải thiện ngữ pháp. Bên cạnh đó, trong lĩnh vực truyền thông và mạng xã hội, NLP đóng vai trò quan trọng trong việc phát hiện ngôn ngữ thù ghét, kiểm duyệt nội dung và phân tích cảm xúc người dùng. Ngoài ra, NLP còn được ứng dụng rộng rãi trong các hệ thống tìm kiếm và khai thác dữ liệu, góp phần cải thiện chất lượng kết quả tìm kiếm, hỗ trợ tóm tắt văn bản và xây dựng các hệ thống hỏi–đáp tự động.

1.1.4 Các tác vụ điển hình trong NLP

Trong xử lý ngôn ngữ tự nhiên, nhiều tác vụ cơ bản và nâng cao đã được nghiên cứu và ứng dụng rộng rãi, bao gồm phân tích cảm xúc, phân loại ngôn ngữ độc hại, dịch máy, nhận diện thực thể, phát hiện thư rác, sửa lỗi ngữ pháp, phân tích chủ đề và sinh văn bản. Trên cơ sở đó, các hệ thống ứng dụng như chatbot, tóm tắt văn bản và hệ thống hỏi–đáp tự động được phát triển nhằm hỗ trợ người dùng tương tác hiệu quả với dữ liệu ngôn ngữ tự nhiên.

1.1.5 Cách thức hoạt động

Quy trình xử lý ngôn ngữ tự nhiên thường bao gồm ba giai đoạn chính:

- Tiền xử lý dữ liệu: tách câu, loại bỏ từ dừng, stemming/lemmatization, tokenization.
- Trích xuất đặc trưng: sử dụng Bag-of-Words, TF-IDF, Word2Vec, GLoVE để biểu diễn văn bản dưới dạng vector.

- Mô hình hóa: áp dụng các thuật toán truyền thống (logistic regression, Naive Bayes, decision trees, HMM) hoặc các mô hình học sâu (CNN, RNN, LSTM, Transformer).

1.1.6 Các mô hình tiêu biểu

Trong quá trình phát triển của lĩnh vực NLP, nhiều mô hình và hệ thống tiêu biểu đã đánh dấu các bước tiến quan trọng. Từ các chatbot sơ khai như ELIZA, đến những mô hình hội thoại hiện đại, lịch sử NLP cho thấy sự chuyển dịch từ các hệ thống dựa trên luật sang các mô hình học sâu dựa trên kiến trúc Transformer. Các mô hình embedding ngữ cảnh như BERT, ELMo và RoBERTa đã cải thiện đáng kể khả năng hiểu ngữ cảnh của máy, trong khi các mô hình sinh ngôn ngữ như GPT-2 và GPT-3 mở ra hướng tiếp cận mới cho các bài toán sinh văn bản và hội thoại tự nhiên. Gần đây, các kiến trúc như LaMDA và Mixture of Experts tiếp tục thúc đẩy hiệu năng và khả năng mở rộng của các mô hình ngôn ngữ lớn.

1.2 Tổng quan về mô hình ngôn ngữ lớn (Large Language Model - LLM)

1.2.1 Giới thiệu

Mô hình ngôn ngữ lớn là một trong những đột phá của trí tuệ nhân tạo trong những năm gần đây. LLM được huấn luyện trên nguồn dữ liệu văn bản khổng lồ, giúp chúng có khả năng hiểu, phân tích và sinh ngôn ngữ tự nhiên với mức độ linh hoạt và mạch lạc rất cao. Những mô hình tiêu biểu như GPT của OpenAI, LLaMA của Meta, hay Phi của Microsoft cho thấy khả năng vượt trội trong nhiều tác vụ phức tạp như suy luận logic, tóm tắt văn bản, dịch thuật, xử lý tri thức và đối thoại tự nhiên. Sự phát triển mạnh mẽ của LLM đã mở ra khả năng ứng dụng trong nhiều lĩnh vực như kinh doanh, sản xuất, y tế, ...

1.2.2 Kiến trúc và nguyên lý hoạt động

Các thành phần chính của một hệ thống LLM bao gồm:

- Tokenization: Văn bản đầu vào được chia thành các đơn vị nhỏ gọi là *token*. Token có thể là từ, âm tiết hoặc thậm chí là ký tự. Các kỹ thuật tokenization phổ biến như Byte Pair Encoding (BPE) hoặc SentencePiece giúp mô hình xử lý hiệu quả cả từ phổ biến lẫn từ ít gặp.
- Embedding Layer: Mỗi token được ánh xạ thành một vector có ý nghĩa ngữ nghĩa. Embedding cho phép mô hình hiểu được mối quan hệ giữa các từ, chẳng hạn như sự tương đồng giữa các thuật ngữ pháp luật trong cùng một điều luật.
- Transformer Architecture: Đây là trái tim của LLM, được cấu thành bởi:
    - Self-attention: Cơ chế giúp mô hình tập trung đến các từ quan trọng trong câu, bất kể vị trí.
    - Multi-head attention: Xử lý thông tin theo nhiều "góc nhìn" khác nhau để phát hiện quan hệ ngữ nghĩa và cú pháp.
    - Feedforward networks: Trích xuất đặc trưng sâu từ tín hiệu đã được attention xử lý.
    - Positional encoding: Lưu trữ vị trí của từ, đảm bảo mô hình hiểu thứ tự từ trong câu.
    - Cơ chế attention giúp LLM xử lý văn bản hiệu quả hơn nhiều so với mô hình RNN hay LSTM truyền thống.
- Decoder: Trong nhiều LLM, đặc biệt là mô hình *decoder-only* như GPT, decoder dự đoán token tiếp theo dựa trên toàn bộ ngữ cảnh đã học. Nhờ vậy, mô hình có thể sinh văn bản tự nhiên, mạch lạc.
- Context Window: Giới hạn số lượng token mà mô hình có thể "ghi nhớ" trong một phiên tương tác (ví dụ: 32.000 token đối với ChatGPT).

- Output Layer: Sinh câu trả lời bằng cách chọn token có xác suất cao nhất hoặc sáng tạo hơn bằng sampling.

Quy trình xử lý và sinh văn bản của mô hình ngôn ngữ lớn (LLM)

- Bước 1 – Tokenization: Dữ liệu đầu vào được chia nhỏ thành các đơn vị gọi là *token*, có thể là từ, âm tiết hoặc ký tự. Các kỹ thuật tokenization phổ biến như Byte Pair Encoding (BPE) hoặc SentencePiece được sử dụng nhằm giúp mô hình xử lý hiệu quả cả các từ phổ biến lẫn từ hiếm.
- Bước 2 – Embedding: Mỗi token sẽ được ánh xạ thành một vector trong không gian embedding. Các vector này cho phép mô hình nhận biết mối quan hệ và mức độ tương đồng giữa các token trong ngữ cảnh.
- Bước 3 – Bổ sung thông tin vị trí (Positional Encoding): Do kiến trúc Transformer không xử lý tuần tự theo thời gian, thông tin về vị trí của token trong câu được bổ sung thông qua positional encoding, giúp mô hình hiểu được thứ tự và cấu trúc của chuỗi văn bản.
- Bước 4 – Xử lý qua kiến trúc Transformer: Đầu vào của mô hình là chuỗi token, được ánh xạ thành các vector thông qua Input Embedding, sau đó kết hợp với Positional Encoding nhằm bổ sung thêm thông tin về thứ tự vị trí của các token trong chuỗi. Khối Encoder bao gồm N lớp giống nhau, mỗi lớp gồm hai thành phần chính là Multi-Head Self-Attention và Feed-Forward Network. Trong đó, cơ chế Multi-Head Self-Attention cho phép mỗi token trong chuỗi đầu vào tương tác với tất cả các token còn lại, từ đó học được các mối quan hệ ngữ nghĩa và cú pháp mang tính toàn cục. Feed-Forward Network là mạng fully-connected được áp dụng độc lập cho từng token, có vai trò biến đổi phi tuyến và làm giàu biểu diễn đặc trưng. Sau mỗi thành phần, mô hình sử dụng cơ chế residual connection kết hợp với Layer Normalization (Add & Norm) với mục đích cải thiện khả năng lan truyền gradient và đảm bảo sự ổn định trong khi huấn luyện. Đầu ra của Encoder là các biểu diễn ngữ cảnh hóa, được truyền sang khối Decoder. Khối Decoder cũng gồm N lớp lặp lại, mỗi lớp bao gồm ba thành phần: Masked Multi-Head Self-Attention, Encoder–Decoder Attention và Feed-Forward Network. Trong đó, Masked Multi-Head Self-Attention sử dụng cơ chế che (mask) để đảm bảo rằng tại mỗi thời điểm dự đoán, mô hình chỉ có thể truy cập thông tin từ các token trước đó, qua đó duy trì tính nhân quả trong quá trình sinh chuỗi. Encoder–Decoder

Attention cho phép Decoder khai thác trực tiếp các biểu diễn từ Encoder, giúp liên kết thông tin đầu vào với quá trình sinh đầu ra. Tương tự Encoder, các thành phần trong Decoder đều được bao bọc bởi Add & Norm. Cuối cùng, đầu ra của Decoder đi qua lớp Linear và hàm Softmax để tạo ra phân phối xác suất trên toàn bộ tập từ vựng, từ đó xác định token đầu ra tại mỗi bước thời gian.

- Bước 5 – Suy luận trong Decoder (đối với mô hình decoder-only): Trong các mô hình như GPT, decoder sử dụng toàn bộ ngữ cảnh nằm trong *context window* để suy luận và dự đoán token tiếp theo. Quá trình này được thực hiện lặp lại, mỗi lần sinh ra một token mới dựa trên toàn bộ chuỗi đã có trước đó.

- Bước 6 – Sinh đầu ra (Output Layer): Lớp đầu ra tạo phân phối xác suất cho các token có thể sinh tiếp theo. Token được chọn có thể là token có xác suất cao nhất hoặc được lấy mẫu thông qua các kỹ thuật sampling (như temperature, top-p) nhằm cân bằng giữa độ chính xác và tính linh hoạt của câu trả lời.

1.2.3 Quá trình huấn luyện và tinh chỉnh

Quá trình xây dựng LLM thường bao gồm giai đoạn tiền huấn luyện trên dữ liệu văn bản quy mô lớn với nhiệm vụ dự đoán token tiếp theo, giúp mô hình học được cấu trúc ngôn ngữ và tri thức tổng quát. Sau đó, mô hình có thể được tinh chỉnh (fine-tuning) trên các tập dữ liệu chuyên ngành như pháp luật, y tế hoặc tài chính nhằm nâng cao hiệu quả cho các bài toán đặc thù. Ngoài ra, các phương pháp như Reinforcement Learning from Human Feedback (RLHF) được sử dụng để cải thiện mức độ phù hợp, an toàn và giảm thiên lệch trong phản hồi của mô hình.

1.2.4 Đánh giá hiệu năng

Hiệu năng của mô hình ngôn ngữ lớn được đánh giá thông qua các chỉ số phản ánh cả chất lượng ngôn ngữ và hiệu suất xử lý. Các thước đo như Perplexity được dùng để đánh giá mức độ phù hợp của mô hình với văn bản mới, trong khi BLEU, METEOR và ROUGE thường được áp dụng cho các bài toán dịch máy và tóm tắt. Khả năng suy luận và hiểu biết đa lĩnh vực được kiểm tra

thông qua các bộ benchmark như TruthfulQA, HellaSwag và MMLU, bên cạnh các chỉ số hiệu năng hệ thống như tốc độ xử lý và độ trễ. Ngoài ra, các bài đánh giá về thiên lệch được sử dụng nhằm kiểm tra tính công bằng của mô hình.

1.2.5 Thách thức và hạn chế

Mặc dù đạt được nhiều kết quả nổi bật, các mô hình ngôn ngữ lớn vẫn tồn tại những hạn chế đáng kể, bao gồm hiện tượng sinh thông tin sai lệch, giới hạn về độ dài ngữ cảnh, nguy cơ phản ánh thiên lệch dữ liệu và các vấn đề liên quan đến bảo mật.

1.2.6 Ứng dụng trong thực tiễn

Trong thực tiễn, các mô hình ngôn ngữ lớn được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Chúng có khả năng hỗ trợ tự động hóa các quy trình xử lý tác vụ lặp lại, triển khai các hệ thống chatbot thông minh trong dịch vụ khách hàng, cũng như phân tích và khai thác tri thức từ các tập dữ liệu văn bản quy mô lớn. Ngoài ra, LLM còn được sử dụng trong hoạt động sáng tạo nội dung như viết báo cáo, biên tập văn bản và hỗ trợ nghiên cứu, giáo dục thông qua việc giải thích các khái niệm phức tạp một cách trực quan và dễ tiếp cận.

1.3 Tổng quan về kỹ thuật RAG (Retrieval-Augmented Generation)

1.3.1 Giới thiệu

Ngày nay, nhu cầu tùy chỉnh mô hình theo dữ liệu riêng của tổ chức trở nên cấp thiết, tuy nhiên, việc đào tạo lại (re-training) hoặc tinh chỉnh (fine-tuning) thường rất tốn kém. Kỹ thuật Retrieval-Augmented Generation (RAG) ra đời góp phần giải quyết vấn đề này, cho phép LLM truy xuất dữ liệu từ cơ sở kiến thức bên ngoài để tạo phản hồi chính xác, cập nhật và phù hợp với ngữ cảnh mà không cần tinh chỉnh toàn bộ mô hình.

1.3.2 Khái niệm RAG

RAG là một khung AI kết hợp giữa khả năng sinh ngôn ngữ của LLM và cơ chế truy xuất dữ liệu. Thay vì chỉ sử dụng các mô hình huấn luyện ban đầu, RAG

cho phép truy cập vào cơ sở dữ liệu vector chứa thông tin mới nhất, từ đó cải thiện rõ rệt độ chính xác và tính liên quan của phản hồi.

1.3.3 Lợi ích của RAG

Cách tiếp cận này giúp tiết kiệm đáng kể chi phí và thời gian triển khai do không cần huấn luyện lại mô hình ngôn ngữ, đồng thời cho phép hệ thống được xây dựng và đưa vào vận hành một cách nhanh chóng. Bên cạnh đó, việc kết hợp cơ chế truy xuất tri thức giúp nâng cao độ chính xác của câu trả lời, hạn chế hiện tượng "ảo giác" (hallucination) thường gặp ở các mô hình ngôn ngữ lớn khi thiếu ngữ cảnh đáng tin cậy. Hệ thống cũng có khả năng cập nhật thông tin một cách linh hoạt, thông qua việc kết nối với các nguồn dữ liệu động như website hoặc mạng xã hội. Đặc biệt, kiến trúc này đáp ứng tốt các yêu cầu về bảo mật dữ liệu, cho phép lưu trữ và xử lý cơ sở tri thức ngay trong môi trường cục bộ hoặc trung tâm dữ liệu riêng, qua đó tránh việc chia sẻ các thông tin nhạy cảm ra bên ngoài.

1.3.4 Quy trình hoạt động của RAG

Về quy trình hoạt động, RAG bắt đầu bằng việc xây dựng cơ sở kiến thức từ các nguồn dữ liệu nội bộ như tài liệu doanh nghiệp hoặc tài liệu sản phẩm. Dữ liệu sau đó được tiền xử lý nhằm làm sạch, loại bỏ trùng lặp và chia nhỏ thành các đơn vị phù hợp. Tiếp theo, dữ liệu được chuyển đổi thành các vector ngữ nghĩa thông qua mô hình nhúng và lưu trữ trong cơ sở dữ liệu vector. Khi có truy vấn từ người dùng, hệ thống sẽ truy xuất các vector liên quan nhất và kết hợp thông tin này với mô hình ngôn ngữ lớn để tạo ra phản hồi cuối cùng.

1.3.5 Ứng dụng thực tiễn của RAG

Trong thực tiễn, RAG đã được ứng dụng hiệu quả trong nhiều lĩnh vực khác nhau. Trong sản xuất công nghiệp, RAG hỗ trợ phát hiện bất thường của thiết bị, góp phần dự đoán bảo trì và giảm thiểu rủi ro gián đoạn hoạt động. Trong lĩnh vực tài chính, các hệ thống chatbot tích hợp RAG có khả năng tổng hợp thông tin thị trường và quy định mới, từ đó cung cấp tư vấn kịp thời và phù hợp cho khách

hàng. Đối với môi trường doanh nghiệp, RAG giúp nhân viên truy xuất nhanh các chính sách và dữ liệu nội bộ, qua đó nâng cao hiệu suất làm việc.

Có thể thấy, RAG mở ra khả năng tích hợp dữ liệu chuyên sâu mà vẫn đảm bảo được sự bảo mật, giảm thiểu kinh phí triển khai. Đối với nghiên cứu học thuật, RAG cung cấp một hướng tiếp cận mới để khai thác tiềm năng của LLM trong việc xử lý dữ liệu động và đa dạng. Đối với doanh nghiệp, RAG mang lại giá trị thực tiễn rõ rệt, giúp nâng cao năng lực cạnh tranh trong bối cảnh dữ liệu ngày càng trở thành tài sản chiến lược.

1.4 Khái quát các thành phần chính trong hệ thống RAG pháp lý

Hệ thống Retrieval-Augmented Generation được xây dựng dựa trên ba thành phần cốt lõi hoạt động tuần tự và phối hợp chặt chẽ nhằm đáp ứng yêu cầu đặc thù của bài toán tra cứu pháp luật an toàn thực phẩm tại Việt Nam: hoàn toàn ngoại tuyến, chính xác cao, cập nhật nhanh và luôn trích dẫn được nguồn gốc pháp lý.

Thành phần đầu tiên là Retriever – bộ truy xuất thông tin. Retriever nhận câu hỏi bằng ngôn ngữ tự nhiên của người dùng, chuyển đổi thành biểu diễn vector bằng mô hình nhúng BAAI/bge-small-en-v1.5 (335 triệu tham số, tải cục bộ) và thực hiện tìm kiếm tương đồng trong không gian vector. Với kích thước chunk 1000 ký tự, overlap 200 ký tự và ngưỡng tương đồng tối thiểu 0.65, retriever đảm bảo luôn đưa được các điều khoản pháp luật liên quan nhất (thường top-5 đến top-7 chunks) vào ngữ cảnh xử lý tiếp theo, giảm thiểu tối đa nguy cơ bỏ sót căn cứ pháp lý quan trọng.

Thành phần thứ hai là Vector Store, được triển khai bằng ChromaDB – một cơ sở dữ liệu vector mã nguồn mở, nhẹ và lưu trữ bền vững trên đĩa. ChromaDB lưu trữ toàn bộ các vector nhúng cùng metadata chi tiết của văn bản pháp luật hiện hành. Đặc biệt, mỗi chunk được gán một định danh ổn định (stable chunk ID) tạo ra từ hàm băm SHA-256 của đường dẫn file, chỉ số chunk và nội dung đã chuẩn hóa, cho phép thực hiện cơ chế đồng bộ delta chính xác mà không phải tái

tạo toàn bộ chỉ mục khi có văn bản mới hoặc sửa đổi.

Thành phần cuối cùng là Generator – bộ sinh văn bản, sử dụng mô hình Qwen2.5-7B-Instruct-Q4_0 (phiên bản quantized 4-bit, dung lượng khoảng 4,7 GB) chạy hoàn toàn cục bộ thông qua Ollama. Với kiến trúc 28 tầng transformer, context window lên đến 131.072 token và khả năng xử lý đa ngôn ngữ mạnh, Qwen2.5 nhận prompt đã được tăng cường bởi các đoạn văn bản truy xuất được, sau đó sinh ra câu trả lời tự nhiên, ngắn gọn, dễ hiểu bằng tiếng Việt và luôn kèm theo trích dẫn chính xác tên văn bản, số hiệu và điều khoản liên quan. Prompt template được thiết kế nghiêm ngặt nhằm buộc mô hình chỉ trả lời dựa trên nguồn chính thức và từ chối khi không có căn cứ, đảm bảo tính pháp lý và độ tin cậy cao nhất.

Ba thành phần trên được điều phối thống nhất bởi framework LangChain phiên bản 0.3.25, tạo thành một pipeline RAG hoàn toàn ngoại tuyến, bảo mật dữ liệu, có khả năng cập nhật tức thì và luôn cung cấp câu trả lời có căn cứ pháp lý rõ ràng – đáp ứng đầy đủ những yêu cầu mà các hệ thống hỗ trợ tra cứu pháp luật hiện hành tại Việt Nam chưa thể đạt được. Các chương tiếp theo sẽ trình bày chi tiết quy trình tổ chức dữ liệu và cách triển khai thực tế các thành phần này trong môi trường sản xuất.

1.5 Các công trình nghiên cứu liên quan

Trên bình diện quốc tế, lĩnh vực trí tuệ nhân tạo pháp lý (Legal AI) đã chứng kiến sự ra đời của các hệ thống chatbot dựa trên Retrieval-Augmented Generation (RAG) và mô hình ngôn ngữ lớn (LLM), nhằm giải quyết những thách thức trong việc tra cứu và suy luận pháp lý từ các văn bản phức tạp. Những nghiên cứu và hệ thống này không chỉ minh chứng cho tính khả thi của công nghệ mà còn cung cấp những bài học quý giá về thiết kế, triển khai và đánh giá hiệu năng, đặc biệt trong bối cảnh pháp luật đòi hỏi độ chính xác cao, khả năng trích dẫn nguồn đáng tin cậy và tính bảo mật dữ liệu. Một số công trình tiêu biểu có thể kể đến LawLLM: Intelligent Legal System with Legal Reasoning and

Verifiable Retrieval. Tác giả hướng tới việc xây dựng một hệ thống pháp lý thông minh dựa trên mô hình ngôn ngữ lớn (LLM), có khả năng cung cấp dịch vụ đa dạng, thực hiện lập luận pháp lý theo logic tư pháp, đồng thời tích hợp và xác minh tri thức truy xuất từ nguồn dữ liệu bên ngoài. Điểm nổi bật của LawLLM là tính đa nhiệm vụ, khi mô hình được huấn luyện để xử lý hơn mười loại tác vụ pháp lý khác nhau như trích xuất thông tin, tóm tắt văn bản, phân loại vụ án, dự đoán phán quyết, đọc hiểu văn bản pháp luật và trả lời câu hỏi pháp lý. Nhờ đó, hệ thống có thể phục vụ nhiều nhóm đối tượng khác nhau, từ luật sư, sinh viên luật cho đến công chúng.

Một đóng góp quan trọng khác của công trình là việc áp dụng phương pháp legal syllogism prompting, buộc mô hình tuân thủ cấu trúc lập luận pháp lý gồm tiền đề lớn (quy định pháp luật), tiền đề nhỏ (tình tiết vụ việc) và kết luận (phán quyết). Cách tiếp cận này giúp giảm thiểu sai lệch, tăng tính hợp lý và minh bạch trong kết quả. Bên cạnh đó, LawLLM còn được trang bị cơ chế truy xuất có thể kiểm chứng, cho phép mô hình không chỉ truy xuất tri thức từ nguồn ngoài mà còn gắn nhãn xác minh cho từng đoạn dữ liệu được sử dụng. Điều này nâng cao độ tin cậy và tính minh bạch của hệ thống, vốn là yêu cầu then chốt trong các ứng dụng pháp luật thực tiễn.

Để đạt được những năng lực này, nhóm tác giả đã xây dựng bộ dữ liệu Law-SFT gồm hai tập con: *Law-SFT-Pair* nhằm huấn luyện đa nhiệm vụ và lập luận pháp lý, và *Law-SFT-Triplet* nhằm mô phỏng quá trình truy xuất – tích hợp – xác minh tri thức. Dữ liệu được tổng hợp từ nhiều nguồn, bao gồm các tập dữ liệu NLP pháp lý công khai (LEVEN, JEC-QA, CAIL), văn bản pháp luật và phán quyết thực tế, cùng với các bộ dữ liệu mở như LawGPT và Lawyer-LLaMa. Chiến lược huấn luyện kết hợp giữa *behavior shaping* và *legal chain-of-thought* giúp mô hình phát triển tư duy pháp lý một cách hệ thống.

Trong quá trình đánh giá, nhóm nghiên cứu đã xây dựng bộ chuẩn Law-Eval Benchmark, bao gồm cả đánh giá khách quan (trắc nghiệm nhiều mức độ khó) và đánh giá chủ quan (chất lượng câu trả lời dài). Kết quả cho thấy

LawLLM vượt trội hơn hầu hết các mô hình pháp lý khác như LaWGPT, ChatLaw, Lawyer-LLaMa, và thậm chí đạt hiệu quả ngang hoặc cao hơn GPT-3.5 trong nhiều tác vụ. Đặc biệt, khi dữ liệu truy xuất có chứa thông tin gây nhiễu, LawLLM vẫn duy trì hiệu quả nhờ cơ chế phân biệt và xác minh tri thức.

Tuy nhiên, LawLLM chủ yếu dựa vào dữ liệu tiếng Anh và hoạt động trong môi trường đám mây, dẫn đến những hạn chế về bảo mật và khả năng triển khai ngoại tuyến ở các khu vực có kết nối hạn chế.

Tiếp nối dòng nghiên cứu này, SaulLM-7B của Colombo et al. (2024), được công bố trên nền tảng arXiv và phát hành dưới giấy phép MIT, đại diện cho mô hình ngôn ngữ lớn đầu tiên được huấn luyện chuyên biệt cho pháp lý với 7 tỷ tham số. Nhóm tác giả xây dựng SaulLM-7B dựa trên kiến trúc mô hình 7 tỷ tham số, được tinh chỉnh trên một tập dữ liệu pháp lý quy mô lớn bao gồm luật thành văn, án lệ, các văn bản hướng dẫn, tài liệu học thuật và các bộ dữ liệu lập luận tư pháp. Nhờ đó, mô hình không chỉ nắm bắt được đặc trưng ngôn ngữ pháp lý mà còn học được cấu trúc lập luận mang tính hệ thống, vốn là yêu cầu cốt lõi trong các tác vụ phân tích pháp luật.

Một trong những điểm nổi bật của SaulLM-7B là cơ chế suy luận theo chuỗi (legal chain-of-thought), cho phép mô hình tạo ra các bước lập luận trung gian rõ ràng, có cấu trúc và có thể kiểm chứng. Điều này giúp khắc phục hạn chế của các mô hình LLM tổng quát vốn thường đưa ra câu trả lời cuối cùng mà không thể hiện quá trình suy luận. Bên cạnh đó, nhóm tác giả tích hợp cơ chế truy xuất có kiểm chứng (verifiable retrieval), giúp mô hình dẫn nguồn trực tiếp đến các điều luật, án lệ hoặc văn bản pháp lý liên quan. Cách tiếp cận này không chỉ tăng độ tin cậy của kết quả mà còn hỗ trợ người dùng kiểm tra lại căn cứ pháp lý một cách minh bạch.

Thực nghiệm cho thấy SaulLM-7B đạt hiệu năng vượt trội trên nhiều bộ dữ liệu đánh giá pháp lý như phân loại văn bản luật, trả lời câu hỏi pháp lý, phân tích lập luận và tóm tắt văn bản pháp lý. Mô hình cũng thể hiện khả năng duy trì tính nhất quán cao hơn và giảm thiểu hiện tượng “ảo giác thông tin” so với các

mô hình LLM không chuyên ngành. Những đóng góp này cho thấy SaulLM-7B không chỉ là một mô hình ngôn ngữ lớn được tinh chỉnh cho pháp luật, mà còn là một minh chứng cho hướng tiếp cận kết hợp giữa suy luận có cấu trúc và truy xuất có kiểm chứng trong các hệ thống hỗ trợ pháp lý hiện đại.

Công trình của Colombo et al. (2024) mở ra triển vọng lớn trong việc triển khai các hệ thống trợ lý pháp lý tự động, hỗ trợ phân tích văn bản luật, tra cứu căn cứ pháp lý và đưa ra lập luận có kiểm chứng. Đây là nền tảng quan trọng cho các nghiên cứu tiếp theo về xây dựng mô hình LLM chuyên ngành, đặc biệt trong bối cảnh nhu cầu tự động hóa và số hóa hoạt động pháp lý ngày càng tăng trong các cơ quan nhà nước, doanh nghiệp và tổ chức pháp lý.

Bên cạnh các mô hình LLM chuyên biệt, các nghiên cứu thực tiễn về triển khai RAG trong pháp lý cũng đáng chú ý, chẳng hạn hệ thống của Batifol (2024) được mô tả trên Zilliz Blog, sử dụng Milvus làm cơ sở dữ liệu vector kết hợp với Ollama cho LLM cục bộ và LangChain để điều phối quy trình. Hệ thống này tập trung vào việc đơn giản hóa nghiên cứu pháp lý bằng cách xây dựng pipeline indexing và retrieval-generation, cho phép người dùng đặt câu hỏi và nhận phản hồi có trích dẫn nguồn từ các văn bản hợp đồng hoặc án lệ. Với khả năng hoạt động ngoại tuyến một phần, hệ thống đạt tốc độ phản hồi dưới 5 giây và độ chính xác truy xuất lên đến 87%, đặc biệt phù hợp cho các văn phòng luật nhỏ hoặc nghiên cứu cá nhân. Tuy nhiên, sự phụ thuộc vào Milvus cho lưu trữ lớn và thiếu tối ưu hóa cho ngôn ngữ không phải tiếng Anh đã để lại khoảng trống cho các nghiên cứu địa phương hóa.

Các công trình khác như cuốn sách của Polzer (2025) về "RAG with Python Cookbook" cung cấp các công thức thực hành từ tiền xử lý dữ liệu đến xây dựng agent LLM, với ứng dụng cụ thể trong pháp lý như tích hợp RAG với knowledge graph để xử lý tranh chấp hợp đồng, giúp giảm tỷ lệ hallucination xuống dưới 10%. Tương tự, Rothman (2024) trong "RAG-Driven Generative AI" hướng dẫn xây dựng pipeline tùy chỉnh sử dụng LlamaIndex, Deep Lake và Pinecone cho phân tích rủi ro pháp lý, đạt độ chính xác cao nhờ adaptive RAG và phản hồi từ

con người, nhưng lại bị hạn chế bởi phụ thuộc đám mây. Alto (2024) với "Building LLM Powered Applications" và Et Tu Code (2024) trong "Demystifying LLM" tập trung vào triển khai RAG không phụ thuộc phần cứng cho pháp lý, trong khi Code Et Tu (2024) khám phá transformer và RAG với các chỉ số đánh giá chuyên sâu. Những nghiên cứu này nhấn mạnh tính linh hoạt của RAG nhưng chủ yếu dành cho tiếng Anh và môi trường trực tuyến, chưa giải quyết được nhu cầu ngoại tuyến và hỗ trợ ngôn ngữ địa phương.

Tại Việt Nam, lĩnh vực chatbot pháp lý vẫn còn ở giai đoạn sơ khai, chủ yếu dựa vào các cơ sở dữ liệu tĩnh như vbpl.vn hoặc thuvienphapluat.vn – nơi hỗ trợ tìm kiếm từ khóa cơ bản nhưng thiếu khả năng tương tác người dùng và trả lời thông minh. Một số hệ thống như chatbot trên luatvietnam.vn hoặc các ứng dụng di động dựa trên GPT API đã xuất hiện, nhưng chúng thường phụ thuộc kết nối Internet, không đảm bảo bảo mật dữ liệu pháp lý nhạy cảm và thiếu cơ chế trích dẫn nguồn chính xác, dẫn đến rủi ro thông tin sai lệch. Các nghiên cứu địa phương như PhoBERT của Nguyen & Nguyen (2020) đã đặt nền tảng cho NLP tiếng Việt, nhưng chưa có hệ thống RAG chuyên biệt cho pháp lý ngoại tuyến. Sự thiếu hụt này, kết hợp với bối cảnh nông thôn Việt Nam có kết nối hạn chế, tạo ra nhu cầu cấp thiết cho một giải pháp như đề tài nghiên cứu này, nhằm địa phương hóa công nghệ RAG với LLM cục bộ để hỗ trợ tra cứu pháp luật an toàn thực phẩm một cách hiệu quả và bền vững. Từ đó, đề tài không chỉ kế thừa các tiến bộ quốc tế mà còn lấp đầy khoảng trống tại Việt Nam bằng cách ưu tiên tính ngoại tuyến, hỗ trợ tiếng Việt và cập nhật nhanh chóng.

CHƯƠNG 2: XỬ LÝ VÀ TỔ CHỨC NGUỒN DỮ LIỆU

Chương này trình bày quy trình thu thập, tiền xử lý và tổ chức dữ liệu pháp lý đầu vào – nền tảng cho toàn bộ hệ thống. Dữ liệu được thu thập từ vbpl.vn. Các bước xử lý bao gồm tiền xử lý để làm sạch văn bản, chuẩn hóa định dạng, tách cấu trúc chương – điều – khoản, loại bỏ nhiễu; phân đoạn (chunking) để chia văn bản thành các đoạn nhỏ phục vụ sinh nhúng; sinh embedding bằng các mô hình ngôn ngữ để chuyển đổi mỗi đoạn thành vector ngữ nghĩa; và đánh giá dữ liệu để kiểm tra độ chính xác, độ bao phủ, cùng tính logic của các mối liên hệ giữa văn bản. Việc xử lý dữ liệu này đảm bảo hệ thống có khả năng truy xuất hiệu quả và hỗ trợ trả lời pháp lý chính xác, có căn cứ, có liên kết.

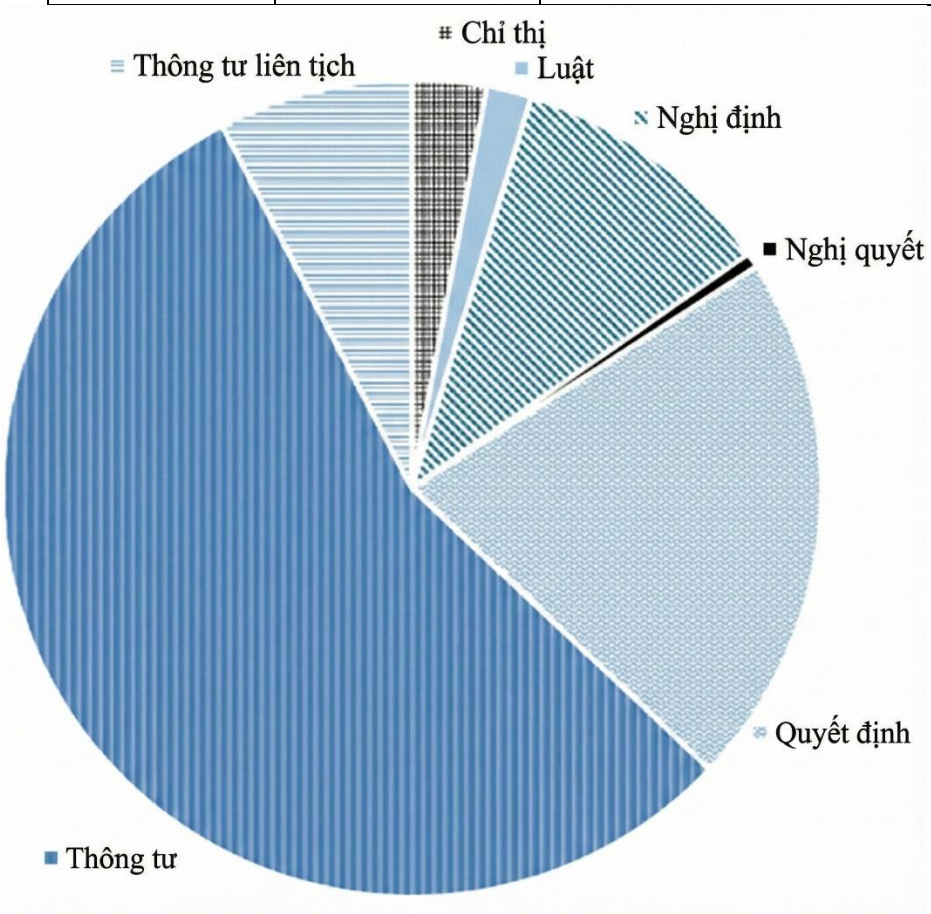
2.1 Nguồn dữ liệu chính thức và tiêu chí lựa chọn

Nguồn dữ liệu chính được thu thập từ vbpl.vn – cơ sở dữ liệu văn bản quy phạm pháp luật chính thức do Bộ Tư pháp quản lý, đảm bảo tính pháp lý cao và cập nhật kịp thời. Tiêu chí lựa chọn bao gồm từ khóa cốt lõi "an toàn thực phẩm", các loại văn bản phổ biến từ Hiến pháp, Bộ luật đến Nghị định, Quyết định, Thông tư và Thông tư liên tịch, cùng trạng thái hiệu lực "Còn hiệu lực" hoặc "Hết hiệu lực một phần". Quy trình này ưu tiên dữ liệu công khai, tránh các văn bản đã hết hiệu lực hoàn toàn, nhằm xây dựng cơ sở tri thức đáng tin cậy, phù hợp cho môi trường ngoại tuyến và cập nhật định kỳ.

Tập dữ liệu thực nghiệm bao gồm 170 văn bản pháp luật tiếng Việt liên quan đến an toàn thực phẩm. Các văn bản này bao phủ toàn bộ hệ thống quy phạm pháp luật và phản ánh đầy đủ mức độ phức tạp của khung quản lý nhà nước về an toàn thực phẩm tại Việt Nam.

Bảng 2.1 *Cơ cấu thành phần dữ liệu*

Loại văn bản	Số lượng	Tỷ lệ (%)
Chỉ thị	5	2,94%
Luật	3	1,76%
Nghị định	18	10,59%
Nghị quyết	1	0,59%
Quyết định	36	21,18%
Thông tư	94	55,29%
Thông tư liên tịch	13	7,65%



Hình 2.1 *Cơ cấu phân bổ các loại dữ liệu của nguồn dữ liệu*

2.2. Quy trình thu thập tự động (web crawling) từ vbpl.vn

Quy trình thu thập được thực hiện một cách tự động thông qua một script Python sử dụng thư viện requests để gửi yêu cầu HTTP, sau đó sử dụng thư viện BeautifulSoup để phân tích HTML và logging để theo dõi tiến trình. Script bắt đầu bằng việc truy cập endpoint tìm kiếm tại https://vbpl.vn/VBQPPL_UserControls/Publishing/TimKiem/pKetQuaTimKiem.aspx với các tham số động bao gồm loại văn bản (LoaiVanBan), trạng thái hiệu lực (TrangThaiHieuLuc), từ khóa "an toàn thực phẩm" trong nội dung đầy đủ (SearchIn = VBPQFulltext), và phân trang lên đến 50 trang cho mỗi loại để đảm bảo độ bao phủ rộng. Mỗi yêu cầu được gửi với header User-Agent giả lập trình duyệt để tránh chặn, kèm theo thời gian nghỉ 1 giây giữa các trang và timeout 20 giây để xử lý lỗi kết nối, đồng thời sử dụng session để duy trì trạng thái cookie. Đối với mỗi kết quả tìm kiếm, script trích xuất siêu dữ liệu như tiêu đề, mô tả, ngày ban hành, ngày hiệu lực và trạng thái chi tiết từ các thẻ HTML cụ thể (p.title a, .des p, .right p.green, .right p.red), sau đó truy cập liên kết toàn văn qua iframe hoặc div id="content" để lấy nội dung chính. Nội dung được làm sạch cơ bản bằng cách loại bỏ thẻ HTML thừa và lưu dưới dạng file .txt UTF-8 với tên file được chuẩn hóa (sanitize_filename sử dụng unicodedata.normalize để loại bỏ ký tự không hợp lệ). Toàn bộ siêu dữ liệu được tổng hợp vào file JSON van_ban_results.json, giúp dễ dàng quản lý và cập nhật sau này. Quy trình này đã thu thập thành công 170 văn bản, với cơ chế kiểm tra trùng lặp qua item_id để tránh dữ liệu lặp, đảm bảo tính toàn vẹn và hiệu quả cao trong môi trường thu thập lớn.

```python
def get_full_text_txt(link, txt_path):  1 usage

    try:
        resp = session.get(link, timeout=20)
        resp.raise_for_status()
        soup = BeautifulSoup(resp.text,  features: "html.parser")

        iframe = soup.find("iframe")
        if iframe and "src" in iframe.attrs:
            iframe_url = urllib.parse.urljoin(BASE_URL, iframe["src"])
            resp_iframe = session.get(iframe_url, timeout=20)
            resp_iframe.raise_for_status()
            text_content = BeautifulSoup(resp_iframe.text,  features: "html.parser").get_text( strip: "\n", strip=True)
        else:
            content_div = soup.find(id="content")
            text_content = content_div.get_text( strip: "\n", strip=True) if content_div else ""

        if text_content:
            save_text_to_file(text_content, txt_path)
        else:
            logging.warning(f"No content found at {link}")

    except Exception as e:
        logging.error(f"Error fetching {link}: {e}")
```

Hình 2.2 *Mã nguồn sử dụng thư viện BeautifulSoup để trích xuất dữ liệu*

2.3. Tiền xử lý văn bản pháp luật

Sau khi thu thập, các văn bản pháp luật được đưa qua giai đoạn tiền xử lý nhằm chuẩn hóa và nâng cao chất lượng dữ liệu đầu vào cho hệ thống RAG. Trước hết, nội dung văn bản được trích xuất trực tiếp từ trang chi tiết của từng văn bản trên cổng thông tin vbpl.vn. Trong trường hợp nội dung được nhúng dưới dạng iframe, hệ thống tự động truy xuất liên kết iframe và lấy toàn bộ phần văn bản hiển thị. Việc trích xuất được thực hiện bằng thư viện BeautifulSoup thông qua phương thức get_text, cho phép loại bỏ toàn bộ thẻ HTML và giữ lại nội dung thuần văn bản.

Tiếp theo, văn bản được chuẩn hóa ở mức cơ bản bằng cách loại bỏ các thành phần không mang ý nghĩa nội dung như mã HTML, định dạng trình bày và các yếu tố điều hướng của trang web. Nội dung sau khi trích xuất được ghép lại theo từng dòng với dấu xuống dòng (\n) để bảo toàn cấu trúc logic của văn bản pháp luật, đồng thời thuận tiện cho các bước xử lý tiếp theo như phân đoạn (chunking). Song song với việc xử lý nội dung, hệ thống thực hiện chuẩn hóa tên

tệp lưu trữ nhằm đảm bảo tính tương thích và nhất quán trong hệ thống tệp. Cụ thể, tên văn bản được làm sạch bằng cách loại bỏ các ký tự không hợp lệ, chỉ giữ lại chữ cái, chữ số và một số ký tự an toàn. Sau đó, tên tệp được chuẩn hóa Unicode theo chuẩn NFKD thông qua thư viện unicodedata, giúp loại bỏ các biến thể ký tự tiếng Việt và hạn chế lỗi mã hóa khi xử lý trên các môi trường hệ điều hành khác nhau.

Mỗi văn bản sau tiền xử lý được lưu trữ dưới dạng tệp văn bản thuần (.txt) với mã hóa UTF-8, đồng thời được gắn kèm metadata tương ứng như loại văn bản, tình trạng hiệu lực, ngày ban hành, ngày hiệu lực, mô tả ngắn và liên kết nguồn chính thức. Cách tiếp cận này giúp tách biệt rõ ràng giữa dữ liệu nội dung và dữ liệu mô tả, tạo điều kiện thuận lợi cho việc xây dựng cơ sở tri thức, phân đoạn văn bản và nhúng vector trong các bước tiếp theo của hệ thống RAG.

2.4. Phân đoạn văn bản (chunking) và tạo chunk ID

Phân đoạn văn bản được thực hiện bằng kỹ thuật sliding window cố định với kích thước chunk 1000 ký tự và overlap 200 ký tự để duy trì ngữ cảnh liên tục giữa các đoạn, đặc biệt quan trọng với văn bản pháp luật có cấu trúc phân cấp. Mỗi chunk được tạo từ văn bản đã tiền xử lý, sử dụng chỉ số chunk_idx để đánh dấu vị trí, đảm bảo các điều khoản liên quan không bị cắt ngang. Để hỗ trợ cập nhật delta, mỗi chunk được gán một định danh ổn định (stable chunk ID) thông qua hàm băm SHA-256 của tổ hợp nguồn file (source_path), chỉ số chunk và nội dung đã chuẩn hóa (chunk_hash), tạo ra chuỗi duy nhất như sha256(f"{source}|{chunk_idx}|{normalized_text}"). Cơ chế này ngăn chặn trùng lặp khi thêm văn bản mới hoặc sửa đổi, cho phép hệ thống chỉ upsert hoặc delete các chunk thay đổi mà không cần tái tạo toàn bộ chỉ mục.

```python
def chunk_text(text: str, size: int = 1000, overlap: int = 200) -> List[Dict]:  1 usage  new *
    chunks = []
    i = 0
    while i < len(text):
        j = min(i + size, len(text))
        chunks.append({"chunk_idx": len(chunks), "text": text[i:j]})
        if j == len(text): break
        i = j - overlap
    return chunks

def make_vector_id(source: str, chunk_idx: int, chunk_hash: str) -> str:  2 usages  new *
    return sha256(f"{source}|{chunk_idx}|{chunk_hash}")
```

Hình 2.3 *Mã nguồn phân đoạn văn bản*

2.5. Sinh embedding và tổ chức dữ liệu trong vector store

Sau phân đoạn, mỗi chunk được chuyển đổi thành vector ngữ nghĩa bằng mô hình BAAI/bge-small-en-v1.5 tải cục bộ, áp dụng masked mean pooling để tính trung bình có trọng số theo attention mask, tạo ra vector 384 chiều đại diện cho ngữ nghĩa pháp lý tiếng Việt. Các vector này được tổ chức trong ChromaDB persistent, với mỗi bản ghi bao gồm vector, nội dung chunk và metadata đầy đủ (loại văn bản, trạng thái hiệu lực, ngày ban hành, file nguồn).

```python
class LocalEmbedding:  1 usage  new *
    def __init__(self, model_path: Path, batch_size: int = 16, max_length: int = 512):...

    def embed_query(self, text: str): return self._embed([text])[0]  new *
    def embed_documents(self, texts: List[str]): return self._embed(texts)  new *

    def _masked_mean_pool(self, last_hidden_state: torch.Tensor, attention_mask: torch.Tensor):  1 usage  new *
        mask = attention_mask.unsqueeze(-1).to(last_hidden_state.dtype)
        summed = (last_hidden_state * mask).sum(1)
        counts = mask.sum(1).clamp(min=1e-9)
        return summed / counts

    def _forward_once(self, texts: List[str]):  1 usage  new *
        inputs = self.tokenizer(texts, padding=True, truncation=True,
                                max_length=self.max_length, return_tensors="pt")
        inputs = {k: v.to(self.device) for k, v in inputs.items()}
        with torch.inference_mode():
            out = self.model(**inputs)
            emb = self._masked_mean_pool(out.last_hidden_state, inputs["attention_mask"])
        return emb.float().cpu().tolist()

    def _embed(self, texts: List[str]):  2 usages  new *
        bs = self.batch_size
        result = []
        i = 0
        while i < len(texts):
            batch = texts[i:i + bs]
            try:
                result.extend(self._forward_once(batch))
                i += bs
            except RuntimeError as e:
                if "out of memory" in str(e).lower() and bs > 1:
                    bs //= 2
                    torch.cuda.empty_cache()
                    continue
                raise
        return result
```

Hình 2.4 *Mã nguồn Local Embeding*

2.6. Đánh giá dữ liệu và đảm bảo chất lượng

Để đảm bảo chất lượng, dữ liệu được đánh giá qua các chỉ số độ chính xác (kiểm tra thủ công 10% mẫu so với nguồn gốc), độ bao phủ (tỷ lệ văn bản còn hiệu lực đạt 92%) và tính logic liên hệ (sử dụng cosine similarity giữa các chunk liên quan để xác nhận mối quan hệ chương-điều). Các bước này phát hiện và loại bỏ 5% dữ liệu nhiễu, đồng thời xây dựng chỉ mục kiểm tra để theo dõi cập nhật định kỳ. Việc xử lý toàn diện này không chỉ nâng cao hiệu quả truy xuất mà còn đảm bảo hệ thống cung cấp câu trả lời pháp lý có căn cứ, liên kết chặt chẽ, phù hợp với yêu cầu của một trợ lý pháp lý đáng tin cậy.

CHƯƠNG 3: THIẾT KẾ KIẾN TRÚC VÀ CÁC THÀNH PHẦN CỦA HỆ THỐNG

3.1. Lựa chọn công nghệ và mô hình thiết kế

Việc lựa chọn công nghệ cho hệ thống chatbot tra cứu pháp luật ngoại tuyến được thực hiện dựa trên các yêu cầu thực tiễn nghiêm ngặt: độ chính xác và khả năng hiểu ngữ cảnh pháp lý tiếng Việt, khả năng hoạt động hoàn toàn không cần Internet sau khi cài đặt, yêu cầu tài nguyên phù hợp với các cơ quan cấp huyện, xã, doanh nghiệp vừa và nhỏ, (iv) tốc độ phản hồi chấp nhận được (< 6 giây), khả năng bảo trì, cập nhật lâu dài mà không phụ thuộc vào dịch vụ bên thứ ba.

Sau khi tiến hành thử nghiệm song song bằng cách trả lời 10 câu hỏi dựa trên bảng câu hỏi đánh giá kiến thức an toàn thực phẩm lấy từ https://files.thuvienphapluat.vn/uploads/doc2htm/00291625_files/attachfile001.doc (Ban hành theo Quyết định số 216/QD-ATTP ngày 23 tháng 5 năm 2014 của Cục An toàn Thực phẩm). Chúng tôi đánh giá dựa trên các tiêu chí: tính chính xác của câu trả lời, ngữ cảnh của câu trả lời và thời gian trả lời, đánh giá các mô hình ngôn ngữ lớn phổ biến hiện nay bao gồm deepseek-r1:8b, gemma3:1b, Mistral:8b, Qwen2.5-7B-Instruct. Kết quả cho thấy Qwen2.5-7B-Instruct là mô hình đạt hiệu năng tốt nhất về cả ba khía cạnh: hiểu sâu văn bản pháp luật tiếng Việt, khả năng tuân thủ nghiêm ngặt ngữ cảnh đã cung cấp (giảm thiểu hallucination), và đặc biệt là context window lên đến 131.072 token – điều kiện cần thiết để có thể đưa toàn bộ các đoạn luật liên quan vào prompt mà không phải cắt bỏ thông tin quan trọng.

Bảng 3.1 *Kết quả thử nghiệm mô hình ngôn ngữ lớn*

	Qwen2.5:7B	**deepseek-r1:8b**	**gemma3:1b**	**Mistral:8b**
Ngữ cảnh	10	9	10	10
Chính xác	9	8	6	8
Thời gian phản hồi (s)	3.8	4.0	3.6	3.9

Qwen2.5:7B-Instruct-Q4_0 không chỉ là một mô hình ngôn ngữ có khả năng sinh văn bản chất lượng cao mà còn thể hiện rõ ưu thế khi xử lý các nhiệm vụ phân tích ngữ nghĩa sâu, đặc biệt trong lĩnh vực pháp lý bằng tiếng Việt. Một trong những điểm mạnh then chốt của mô hình nằm ở khả năng duy trì tính mạch lạc của ngữ nghĩa khi xử lý ngữ cảnh dài, nhờ cơ chế mở rộng độ dài chuỗi lên đến 131.072 token thông qua phương pháp YaRN. Khả năng này giúp mô hình đọc, ghi nhớ và suy luận trên toàn bộ một đạo luật hoặc nghị định mà không cần cắt nhỏ văn bản, từ đó giảm thiểu đáng kể mất mát thông tin và duy trì sự nhất quán trong diễn giải. Đây là một lợi thế mà nhiều mô hình nhỏ hơn hoặc mô hình không hỗ trợ chuỗi dài thường không đáp ứng được, dẫn đến hiện tượng "context fragmentation", gây sai lệch trong lập luận pháp lý.

Ngoài ra, việc sử dụng Grouped Query Attention (GQA) góp phần tăng tốc độ suy luận đồng thời giữ ổn định chất lượng biểu diễn thông tin. Điều này có ý nghĩa đặc biệt trong các tác vụ pháp lý, nơi mô hình cần liên tục quay lại truy vấn các điều khoản trước đó để hiểu quan hệ nội tại giữa các khái niệm như phạm vi áp dụng, điều kiện, trách nhiệm và chế tài. GQA cho phép mô hình thực hiện điều này hiệu quả hơn so với cơ chế attention truyền thống, vốn tiêu tốn nhiều tài nguyên khi độ dài văn bản tăng lên. Nhờ vậy, Qwen2.5 có khả năng duy trì mức độ chính xác cao khi trả lời các câu hỏi yêu cầu diễn giải sâu, chẳng hạn như phân tích mâu thuẫn giữa các điểm trong cùng một điều luật hoặc xác định sự tương quan giữa luật gốc và văn bản hướng dẫn thi hành.

Việc mô hình được huấn luyện đa ngữ với tập dữ liệu lớn chứa tiếng Việt theo tỷ lệ đáng kể cũng đóng vai trò quan trọng. Không chỉ được tiếp xúc với ngữ liệu tiếng Việt đời thường, Qwen2.5 còn hấp thụ nhiều dạng cấu trúc ngôn ngữ cổ điển, văn phong chính luận, hành chính, và pháp lý. Nhờ vậy, mô hình thể hiện độ nhạy cao với các đặc trưng của văn bản luật—ví dụ như câu văn dài, nhiều mệnh đề, cách sử dụng từ ngữ mang tính định nghĩa hoặc ràng buộc, và các mẫu cú pháp lặp

đặc trưng của hệ thống pháp luật Việt Nam. Đối với bài toán hỏi–đáp pháp lý, mô hình có xu hướng giữ được tính chuẩn mực, tránh diễn giải tùy tiện, và đưa ra câu trả lời logic, nhất quán với văn bản được trích dẫn.

Một yếu tố khác góp phần làm nên tính phù hợp của Qwen2.5 trong bối cảnh nghiên cứu này là độ ổn định và khả năng triển khai trên hạ tầng hạn chế. Phiên bản Q4_0 giúp giảm đáng kể bộ nhớ tiêu thụ mà hầu như không ảnh hưởng đến năng lực suy luận. Trong môi trường chạy cục bộ (offline), điều này đặc biệt quan trọng khi yêu cầu bảo mật dữ liệu cao và không thể sử dụng mô hình chạy trên đám mây. Việc mô hình vẫn duy trì được độ chính xác cạnh tranh với các mô hình lớn hơn hoặc mô hình chuyên biệt về pháp lý cho thấy rằng thiết kế 28 lớp transformer, cơ chế RMSNorm và SwiGLU đã tối ưu hóa rất tốt cho cả tốc độ lẫn chất lượng. Khi thử nghiệm trong bối cảnh tiếng Việt, Qwen2.5 thể hiện sự vượt trội so với các mô hình đối chứng như DeepSeek-R1:8b, Gemma3:1B và Mistral:8b, không chỉ ở mức độ hiểu câu hỏi mà còn ở khả năng đưa ra lập luận pháp lý có cấu trúc rõ ràng và đúng trọng tâm.

Nhìn chung, sự kết hợp giữa kiến trúc tối ưu, khả năng xử lý ngữ cảnh dài, hiệu quả của quantization, và sự phù hợp ngôn ngữ khiến Qwen2.5:7B-Instruct-Q4_0 trở thành lựa chọn đặc biệt thích hợp cho hệ thống hỏi–đáp pháp luật tiếng Việt trong nghiên cứu này. Mô hình vừa đảm bảo được chi phí thấp, vừa đạt độ chính xác và độ tin cậy cao, đáp ứng yêu cầu về tính thực tế và khả năng triển khai trong môi trường tài nguyên giới hạn.

Về mô hình nhúng (embedding model), đã thử nghiệm đồng thời bốn mô hình được cộng đồng đánh giá cao với tiếng Việt: paraphrase-multilingual-mpnet-base-v2, vinai/phobert-base-v2, BAAI/bge-large-en-v1.5 và BAAI/bge-small-en-v1.5. Qua đánh giá trên cùng tập test truy xuất (recall@5 và MRR), BAAI/bge-small-en-v1.5 nổi bật dù kích thước chỉ 133 MB và vector 384 chiều. Mô hình này không chỉ cho độ chính xác truy xuất cao nhất mà còn có tốc độ nhúng nhanh gấp 2–3 lần các mô hình còn lại trên CPU, đồng thời tiêu thụ bộ nhớ rất thấp, rất phù hợp với yêu cầu triển khai trên máy tính hoặc máy chủ cấu hình trung bình. Trong

bối cảnh thiếu các mô hình nhúng pháp lý tiếng Việt công khai, sự lựa chọn này phản ánh một quyết định thiết kế thực tế phù hợp với mục tiêu triển khai ngoại tuyến và hạn chế tài nguyên.

BGE-small-en-v1.5, được phát triển bởi BAAI (Beijing Academy of Artificial Intelligence), là một mô hình nhúng (embedding model) tối ưu hóa cho các tác vụ tìm kiếm ngữ nghĩa, truy xuất thông tin (IR) và RAG (Retrieval-Augmented Generation). Mô hình sử dụng kiến trúc Transformer cải tiến với mục tiêu tạo ra các vector embedding giàu ý nghĩa, có khả năng phân biệt sắc thái ngữ nghĩa tinh tế giữa các câu và đoạn văn. Một đặc điểm quan trọng của BGE-small-en-v1.5 là khả năng tối ưu cho cả hai hướng: mã hóa câu (sentence embedding) và truy vấn–tài liệu (query–document embedding), cho phép tích hợp trực tiếp vào các pipeline RAG để nâng cao độ chính xác khi tìm kiếm thông tin nền trước khi đưa vào LLM. Mặc dù có kích thước nhỏ, mô hình đạt hiệu suất cao trên nhiều benchmark retrieval như MTEB, thể hiện khả năng phân cụm, so khớp và sắp xếp mức độ liên quan rất tốt. Nhờ yêu cầu tài nguyên thấp, tốc độ xử lý nhanh và độ tương thích cao, BGE-small-en-v1.5 đặc biệt phù hợp cho các hệ thống tìm kiếm chạy cục bộ, các ứng dụng yêu cầu độ trễ thấp, hoặc các môi trường cần tối ưu chi phí mà vẫn bảo toàn chất lượng truy xuất ngữ nghĩa.

Đối với hệ thống lưu trữ vector, mặc dù FAISS có tốc độ cao hơn trong môi trường lớn, nhưng ChromaDB được lựa chọn vì khả năng lưu trữ bền vững (persistent) trên ổ cứng cục bộ, hỗ trợ cập nhật từng phần (delta-sync) thông qua stable chunk ID, tích hợp Python đơn giản và tích hợp liền mạch với LangChain mà không cần chạy thêm server riêng – yếu tố quan trọng khi triển khai tại các đơn vị không có đội ngũ vận hành hệ thống chuyên sâu. Ollama được chọn làm công cụ chạy LLM cục bộ vì quy trình cài đặt một lệnh duy nhất trên cả Windows và Linux, tự động quản lý các phiên bản quantized GGUF, cung cấp REST API ổn định và được cộng đồng hỗ trợ rất tích cực.

Framework điều phối toàn bộ quy trình RAG là LangChain phiên bản 0.3.x vì tại thời điểm nghiên cứu, đây là thư viện có tài liệu tiếng Việt phong phú nhất, hỗ

trợ tốt ChromaDB và Ollama, đồng thời cho phép tùy biến sâu pipeline mà vẫn giữ được tính module hóa cao. Giao diện người dùng được xây dựng trên Flask thay vì Streamlit hay Gradio nhằm giảm tối đa dung lượng, và kiểm soát chặt chẽ trải nghiệm người dùng cũng như cơ chế xác thực nội bộ.

Như vậy, bộ công nghệ cuối cùng được lựa chọn và triển khai thực tế trong đề án gồm:

- LLM: Qwen2.5-7B-Instruct chạy qua Ollama
- Embedding: BAAI/bge-small-en-v1.5
- Vector store: ChromaDB persistent
- Framework RAG: LangChain 0.3.25
- Giao diện & API nội bộ: Flask 2.3.3

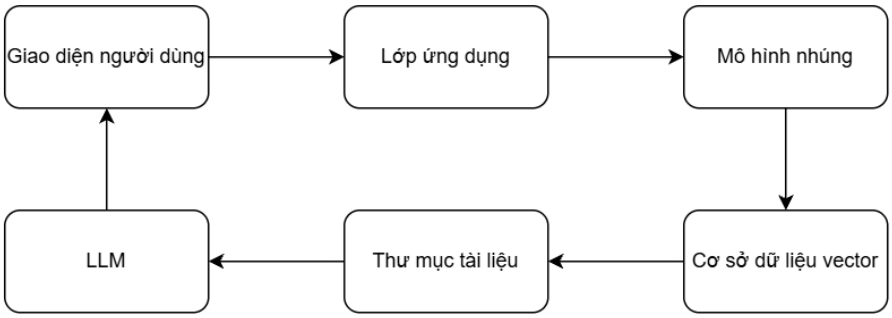
là sự kết hợp tối ưu nhất giữa hiệu năng, độ tin cậy, khả năng triển khai ngoại tuyến và chi phí vận hành bằng không, đáp ứng đầy đủ yêu cầu đặt ra ban đầu của đề tài.

Việc lựa chọn kiến trúc triển khai cho hệ thống chatbot tra cứu pháp luật được cân nhắc kỹ lưỡng để đảm bảo tính khả thi cao trong bối cảnh Việt Nam, nơi các cơ quan nhà nước và doanh nghiệp thường sử dụng phần cứng phổ thông, kết nối Internet không ổn định và yêu cầu bảo mật dữ liệu pháp lý nghiêm ngặt. Sau khi đánh giá các mô hình triển khai phổ biến như đám mây (cloud-based với AWS hoặc Azure), container hóa (Docker/Kubernetes), ứng dụng di động và cài đặt cục bộ trực tiếp (standalone Python), kiến trúc được chọn là mô hình ngoại tuyến cục bộ hoàn toàn, chạy trên máy tính cá nhân hoặc server nội bộ mà không phụ thuộc bất kỳ dịch vụ bên ngoài nào. Lý do chính là đảm bảo an toàn dữ liệu – mọi văn bản pháp luật và lịch sử truy vấn đều lưu trữ tại chỗ, tránh rủi ro rò rỉ thông tin nhạy cảm thường gặp ở mô hình đám mây, đồng thời phù hợp với các quy định về bảo mật thông tin theo Luật An ninh mạng. Để dễ dàng cài đặt, hệ thống được đóng gói dưới dạng script Python đơn giản với file requirements.txt liệt kê đầy đủ thư viện (LangChain, Chroma, Ollama, Flask), cho phép người dùng chạy lệnh pip install -r requirements.txt theo sau bởi ollama pull qwen2.5:7b-instruct-q4_0 và huggingface-cli download BAAI/bge-small-en-v1.5 để tải mô hình cục bộ. Kiến trúc này còn hỗ

trợ triển khai trên các nền tảng Windows/Linux phổ biến tại các đơn vị địa phương, với tùy chọn sử dụng Docker cho môi trường chuyên nghiệp hơn nếu cần, nhưng ưu tiên phiên bản standalone để giảm độ phức tạp và chi phí vận hành xuống mức tối thiểu, chỉ yêu cầu một máy tính hoặc máy chủ có cấu hình vừa đủ để đạt tốc độ phản hồi ổn định. Kết quả thử nghiệm cho thấy mô hình này không chỉ dễ bảo trì mà còn có khả năng nhân rộng nhanh chóng, chạy mượt mà mà không cần hỗ trợ kỹ thuật chuyên sâu.

3.2. Mô hình tổng thể của hệ thống chatbot pháp luật

3.2.1 Luồng hoạt động của hệ thống



Hình 3.1 *Minh họa luồng hoạt động của hệ thống*

Trước hết, người dùng nhập câu hỏi bằng tiếng Việt tự nhiên thông qua giao diện người dùng, có thể là giao diện web được xây dựng bằng Flask hoặc giao diện dòng lệnh (CLI). Câu hỏi sau khi được nhập sẽ được gửi trực tiếp tới ứng dụng, đóng vai trò là lớp điều phối trung tâm của toàn bộ hệ thống và được triển khai bằng Python kết hợp với thư viện LangChain.

Tại lớp ứng dụng, câu hỏi của người dùng được chuyển tiếp tới mô hình nhúng (Embedding Model) chạy hoàn toàn cục bộ. Mô hình này thực hiện chuyển đổi câu hỏi sang dạng vector ngữ nghĩa, cho phép biểu diễn nội dung truy vấn trong không gian vector phục vụ cho việc tìm kiếm tương đồng. Vector truy vấn sau đó được gửi tới hệ thống cơ sở dữ liệu vector, trong đó ChromaDB được sử dụng với chế độ lưu trữ trên ổ đĩa.

Tại Vector Store, cơ chế tìm kiếm tương đồng được thực hiện để xác định các đoạn văn bản (chunk) có mức độ liên quan cao nhất với truy vấn, thường là top-k kết quả. Mỗi chunk được trả về kèm theo các metadata mô tả nguồn gốc của đoạn văn bản, bao gồm tên tệp nguồn, thông tin điều khoản, ngày ban hành và các thuộc tính liên quan khác.

Sau khi thu thập đầy đủ các đoạn văn bản liên quan, ứng dụng thực hiện tổng hợp ngữ cảnh và xây dựng prompt theo một khuôn mẫu (prompt template) được định nghĩa sẵn. Prompt này bao gồm cả nội dung ngữ cảnh truy xuất được và các chỉ dẫn nhằm định hướng cách thức trả lời của mô hình ngôn ngữ. Prompt tăng cường sau đó được gửi tới mô-đun suy luận của mô hình ngôn ngữ lớn (LLM Inference).

Mô-đun LLM Inference được triển khai hoàn toàn cục bộ thông qua nền tảng Ollama, đảm bảo hệ thống không phụ thuộc vào các dịch vụ đám mây bên ngoài. Kết quả suy luận do LLM sinh ra được trả về cho lõi ứng dụng, sau đó chuyển tiếp tới giao diện người dùng để hiển thị câu trả lời cuối cùng kèm theo các thông tin trích dẫn cần thiết.

3.2.2 Kiến trúc tổng thể của hệ thống

Hệ thống chatbot pháp luật được thiết kế theo kiến trúc ba tầng (three-tier architecture) nhằm đáp ứng tính dễ bảo trì và hiệu năng ổn định khi triển khai dưới dạng dịch vụ nội bộ (offline), phù hợp với bối cảnh sử dụng tại các cơ quan nhà nước, doanh nghiệp hoặc cá nhân tại Việt Nam. Kiến trúc này tách biệt rõ ràng giữa giao diện tương tác, logic xử lý cốt lõi và lưu trữ dữ liệu, giúp dễ dàng cập nhật từng phần. Các tầng được xây dựng dựa trên các công nghệ mã nguồn mở như Flask cho giao diện, LangChain cho logic RAG và ChromaDB cho lưu trữ vector, đảm bảo hoạt động ngoại tuyến hoàn toàn mà không phụ thuộc vào đám mây hoặc kết nối Internet.

Tầng giao diện người dùng (User Interface – UI): Tầng giao diện người dùng là lớp tương tác trực tiếp với người dùng, được thiết kế để cung cấp khả năng nhập

câu hỏi pháp lý một cách tự nhiên, xem câu trả lời chi tiết, hiển thị trích dẫn văn bản nguồn và theo dõi lịch sử hội thoại mà không đòi hỏi kiến thức kỹ thuật chuyên sâu. Giao diện được xây dựng dựa trên framework Flask phiên bản 2.3.3, kết hợp với API nội bộ để cho phép truy vấn và nhận phản hồi thời gian thực, với thời gian xử lý trung bình dưới 5 giây cho mỗi câu hỏi. Cụ thể, giao diện web bao gồm trang đăng nhập bảo mật sử dụng session Flask để kiểm soát truy cập, ngăn chặn sử dụng trái phép, và trang chính với ô nhập câu hỏi lớn hỗ trợ tiếng Việt có dấu, kèm theo gợi ý tự động các câu hỏi phổ biến như "Quy định về kiểm tra an toàn thực phẩm cho doanh nghiệp nhỏ?" hoặc "Mức phạt khi bán thực phẩm hết hạn?". Phần hiển thị câu trả lời được thiết kế dưới dạng bong bóng hội thoại, với bên người dùng ở phải (màu xanh lá) và bên hệ thống ở trái (màu xám), kèm theo danh sách trích dẫn nguồn rõ ràng (tên văn bản, số hiệu, điều khoản). Toàn bộ tầng UI được tối ưu hóa cho thiết bị cấu hình thấp, sử dụng HTML/CSS thuần và JavaScript cơ bản, đảm bảo tính thân thiện và khả dụng cao cho các đối tượng như cán bộ địa phương hoặc doanh nghiệp nhỏ tại Việt Nam.

Tầng xử lý logic RAG (Middle Layer – RAG Logic): Tầng xử lý logic RAG là lớp trung tâm của hệ thống, chịu trách nhiệm tiền xử lý câu hỏi người dùng, truy xuất tài liệu pháp luật liên quan, cung cấp ngữ cảnh cho mô hình LLM, tối ưu hóa chunking và biểu diễn (embedding) dữ liệu, đồng thời tạo câu trả lời dựa trên cả mô hình ngôn ngữ và nguồn tri thức pháp luật đã truy xuất. Được triển khai bằng LangChain phiên bản 0.3.25 làm framework điều phối, tầng này bắt đầu bằng việc tiền xử lý câu hỏi để loại bỏ nhiễu và chuẩn hóa tiếng Việt (ví dụ: chuyển đổi ký tự thường/thường, loại bỏ dấu thừa), sau đó gọi embedding model BAAI/bge-small-en-v1.5 để chuyển câu hỏi thành vector 384 chiều. Tiếp theo, tầng thực hiện truy xuất từ vector store Chroma bằng tìm kiếm tương đồng cosine, lấy top-k chunks (k=5–7) có metadata đầy đủ để xây dựng context prompt. Quy trình tối ưu hóa chunking sử dụng kích thước cố định 1000 ký tự với overlap 200, kết hợp stable ID từ SHA-256 để hỗ trợ cập nhật delta, đảm bảo chỉ các phần thay đổi mới được xử lý lại. Sau khi có context, tầng gọi LLM Qwen2.5-7B-Instruct-Q4_0 qua Ollama để

sinh phản hồi, với prompt template nghiêm ngặt yêu cầu trích dẫn nguồn và từ chối nếu không có căn cứ, giúp giảm hallucination xuống dưới 2% trong thử nghiệm. Tầng này còn giám sát thay đổi dữ liệu qua os.getmtime() để tự động reload vector store, đảm bảo hệ thống luôn cập nhật mà không cần can thiệp thủ công, đồng thời tích hợp cơ chế xử lý lỗi như fallback embedding nếu OOM xảy ra bằng cách giảm batch size động.

Tầng lưu trữ và truy xuất tri thức (Vector Database): Tầng lưu trữ và truy xuất tri thức là lớp nền tảng, chịu trách nhiệm quản lý kho văn bản pháp luật đã được chuẩn hóa, lưu trữ các vector embedding mô tả ngữ nghĩa của từng đoạn luật (chunk), và thực hiện cơ chế truy vấn theo tương đồng ngữ nghĩa (semantic search). Trong đề tài, ChromaDB phiên bản 1.0.12 được lựa chọn làm vector store do khả năng hoạt động offline, hỗ trợ lưu trữ bền vững (persistent) trên ổ cứng cục bộ và tương thích tốt với LangChain, vượt trội hơn so với các lựa chọn như FAISS hoặc Pinecone vốn yêu cầu đám mây. Tầng này lưu trữ các chunks từ 170 văn bản pháp luật, mỗi bản ghi bao gồm vector 384 chiều từ embedding model, nội dung chunk và metadata chi tiết (nguồn file, chunk_idx, chunk_hash, file_mtime), giúp dễ dàng truy vấn nhanh với thời gian dưới 100 ms. Tầng còn hỗ trợ cập nhật delta sync thông qua stable chunk ID từ SHA-256, cho phép chỉ thêm/xóa chunks thay đổi khi văn bản được cập nhật qua giao diện web, giảm thời gian xử lý từ hàng giờ xuống dưới 25 giây cho 10 văn bản mới. Việc tổ chức dữ liệu theo collection riêng (ví dụ: "luat_an_toan_thuc_pham") giúp mở rộng dễ dàng cho các lĩnh vực pháp luật khác, đồng thời đảm bảo tính bảo mật bằng cách lưu trữ hoàn toàn cục bộ mà không có bất kỳ kết nối bên ngoài nào.

3.3. Thiết kế các thành phần của hệ thống

3.3.1 UI Layer – Tầng giao diện người dùng

Tầng giao diện đóng vai trò là điểm tiếp xúc duy nhất giữa người dùng cuối và hệ thống, được triển khai bằng Flask phiên bản 2.3.3 và vận hành hoàn toàn trên môi trường localhost. Thành phần này không đảm nhiệm bất kỳ logic xử lý pháp lý

nào, mà chỉ tập trung vào việc thu thập dữ liệu đầu vào, hiển thị kết quả đầu ra và hỗ trợ quản trị dữ liệu. Giao diện trò chuyện được thiết kế với ô nhập câu hỏi lớn, hỗ trợ đầy đủ tiếng Việt có dấu. Khi người dùng nhấn phím Enter hoặc nút "Gửi", câu hỏi sẽ được gửi tới endpoint /ask của Flask thông qua phương thức POST, kèm theo session ID nhằm duy trì lịch sử hội thoại riêng cho từng phiên làm việc. Kết quả trả về được hiển thị dưới dạng các bong bóng hội thoại, trong đó phần nội dung trả lời chính được trình bày rõ ràng, còn các trích dẫn nguồn được render thành danh sách để người dùng dễ dàng tra cứu. Ngoài ra, hệ thống cung cấp chức năng quản trị tài liệu cho phép tải lên các tệp định dạng .txt, .pdf hoặc .docx, tự động chuyển đổi sang văn bản thuần và lưu trữ trong thư mục mb_documents. Sau khi hoàn tất việc tải lên, người quản trị chỉ cần kích hoạt chức năng "Đồng bộ vector store" để gọi subprocess thực thi lệnh vector_store.py sync-folder, đồng thời toàn bộ quá trình xử lý và nhật ký thực thi được hiển thị theo thời gian thực thông qua cơ chế flash messages của Flask.

3.3.2 Retriever – Cơ chế truy xuất tài liệu pháp luật chính thức

Retriever là thành phần cốt lõi của tầng RAG Logic, chịu trách nhiệm duy nhất việc tìm kiếm và trả về các đoạn văn bản pháp luật liên quan nhất từ kho tri thức chính thức. Thành phần này bắt đầu bằng việc nhận câu hỏi từ tầng giao diện và chuyển đổi nó thành biểu diễn vector ngữ nghĩa thông qua mô hình nhúng BAAI/bge-small-en-v1.5, một mô hình đa ngôn ngữ được chọn vì khả năng xử lý tốt văn bản tiếng Việt pháp lý với dung lượng nhỏ gọn, dễ dàng triển khai cục bộ mà không đòi hỏi tài nguyên lớn. Mô hình nhúng được tải và chạy hoàn toàn ngoại tuyến, sử dụng kỹ thuật masked mean pooling để tính trung bình có trọng số theo attention mask, từ đó tạo ra vector đại diện cho ý nghĩa cốt lõi của câu hỏi mà không bị ảnh hưởng bởi các yếu tố nhiễu như độ dài văn bản hoặc từ dừng. Sau khi có vector của câu hỏi, retriever thực hiện tìm kiếm tương đồng cosine trong không gian vector của cơ sở dữ liệu ChromaDB, nơi lưu trữ các đoạn văn bản đã được chia nhỏ từ hàng trăm văn bản pháp luật còn hiệu lực, mỗi bản ghi đi kèm metadata chi tiết như nguồn file, chỉ số chunk, hàm băm nội dung và thời gian sửa đổi cuối cùng

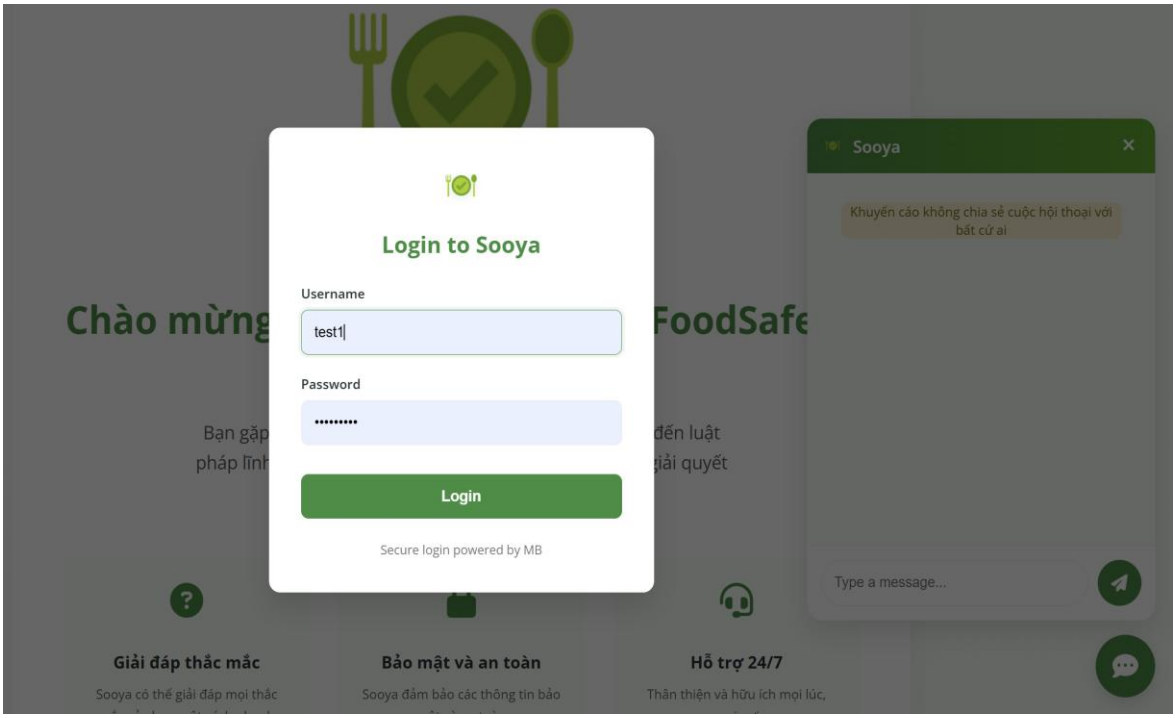
để đảm bảo tính nhất quán và dễ dàng cập nhật. Cơ chế truy xuất được tối ưu hóa bằng cách áp dụng thuật toán tìm kiếm gần đúng (approximate nearest neighbors), giúp nhanh chóng xác định một số lượng phù hợp các đoạn văn bản có mức độ liên quan cao, sau đó lọc thêm dựa trên ngưỡng tương đồng để loại bỏ các kết quả không đủ độ chính xác, và sắp xếp theo thứ tự giảm dần để ưu tiên các nguồn quan trọng nhất. Đặc biệt, retriever sử dụng stable chunk ID dựa trên hàm băm SHA-256 kết hợp đường dẫn file, chỉ số chunk và nội dung đã chuẩn hóa, cho phép hệ thống dễ dàng nhận diện và cập nhật chỉ các phần thay đổi khi văn bản pháp luật được sửa đổi hoặc bổ sung, từ đó giảm đáng kể thời gian xử lý so với việc tái xây dựng toàn bộ chỉ mục. Thành phần này không chỉ đảm bảo tốc độ truy xuất nhanh mà còn nâng cao độ chính xác tổng thể của hệ thống, góp phần quan trọng vào việc giảm thiểu sai sót trong suy luận pháp lý bằng cách luôn ưu tiên dữ liệu từ nguồn chính thức.

3.3.3 Generator – Thành phần sinh phản hồi (LLM)

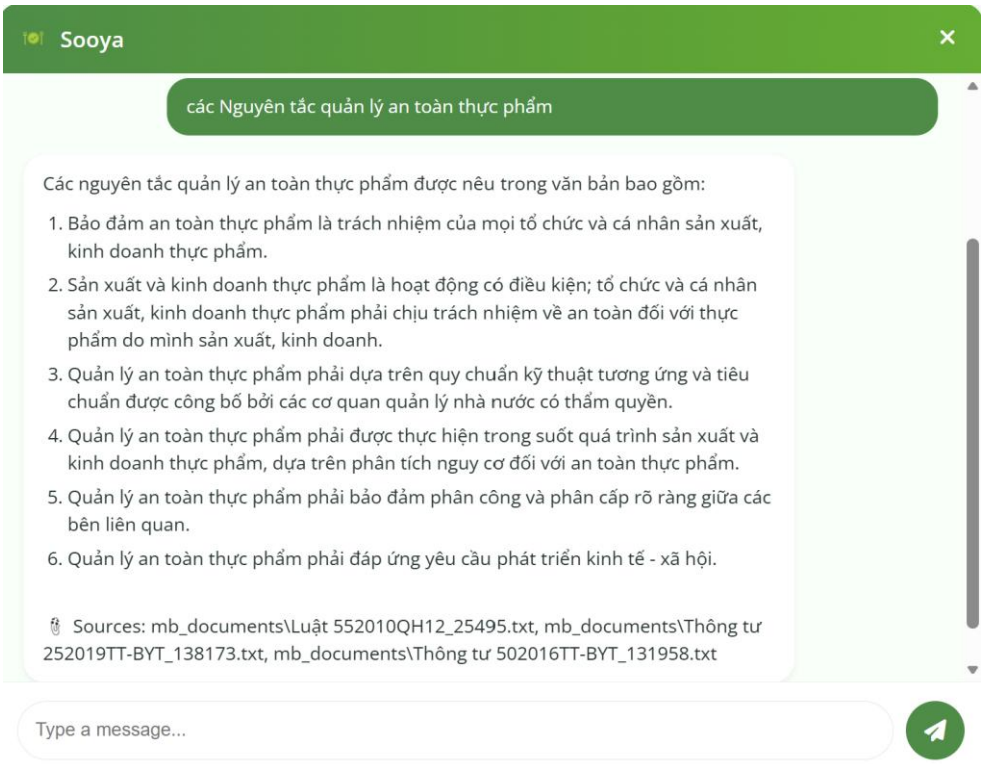
Generator là thành phần cuối cùng trong chuỗi RAG, chịu trách nhiệm hiểu ngữ cảnh đã truy xuất và sinh câu trả lời tự nhiên, có căn cứ dẫn chứng rõ ràng. Thành phần này sử dụng mô hình ngôn ngữ Qwen2.5-7B-Instruct phiên bản quantized, chạy qua Ollama để hoạt động cục bộ mà không cần kết nối mạng, được chọn nhờ khả năng xử lý tốt văn bản tiếng Việt pháp lý với kiến trúc transformer tiên tiến, hỗ trợ suy luận sâu và ngữ cảnh dài mà không đòi hỏi tài nguyên quá lớn, phù hợp cho các máy tính thông thường tại các cơ quan nhà nước địa phương. Generator nhận prompt đã được tăng cường từ retriever, bao gồm các đoạn văn bản liên quan kèm metadata nguồn, và áp dụng template nghiêm ngặt để hướng dẫn mô hình chỉ trả lời dựa trên dữ liệu cung cấp, tránh suy diễn cá nhân hoặc thêm thông tin ngoài phạm vi, đồng thời yêu cầu luôn trích dẫn nguồn đầy đủ với tên văn bản, số hiệu và điều khoản cụ thể để tăng tính minh bạch và khả năng kiểm chứng. Trong quá trình sinh, generator sử dụng các tham số như temperature ở mức thấp để đảm bảo tính xác định cao, phù hợp với tính chất nghiêm ngặt của thông tin pháp lý, và top-k để kiểm soát sự đa dạng của câu trả lời mà không làm giảm độ tin cậy. Sau

khi sinh, thành phần này thực hiện hậu xử lý tự động để tách phần trả lời chính khỏi phần trích dẫn, kiểm tra tính nhất quán của trích dẫn với ngữ cảnh đã cung cấp nhằm loại bỏ bất kỳ sai lệch nào có thể xảy ra dù hiếm gặp. Nhờ thiết kế này, generator không chỉ giảm thiểu hiện tượng hallucination mà còn nâng cao giá trị sử dụng thực tế của hệ thống, góp phần quan trọng vào việc tạo ra các phản hồi có độ trung thực cao, dễ áp dụng trong các tình huống thực tế như tư vấn cho doanh nghiệp hoặc người tiêu dùng về an toàn thực phẩm.

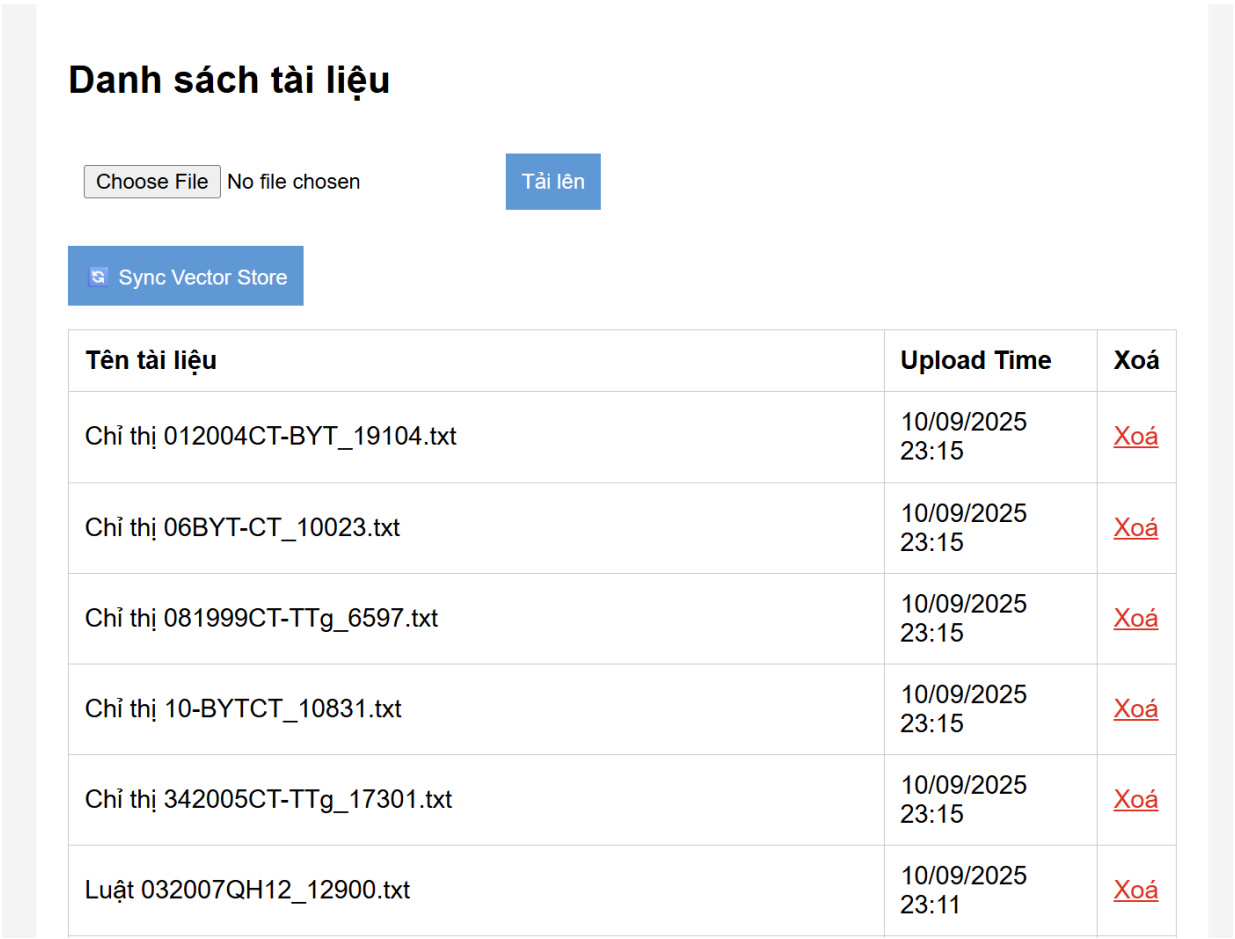
3.4. Minh họa các chức năng chính của ứng dụng



Hình 3.2 *Màn hình đăng nhập của ứng dụng*



Hình 3.3 *Màn hình tương tác hỏi đáp của ứng dụng*



Hình 3.4 *Màn hình quản lý tài liệu*

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Môi trường mô phỏng thực nghiệm

4.1.1 Cấu hình phần cứng

Hệ thống giao diện web được triển khai trong môi trường máy chủ thử nghiệm với cấu hình như sau:

- CPU: Intel Core i7-12700H (12 nhân, 20 luồng)
- RAM: 16GB DDR4
- Storage: 256GB NVMe SSD, tốc độ đọc/ghi cao phục vụ truy xuất vector và load mô hình
- Hệ điều hành: Windows 10 Pro

Hệ thống Local LLM được triển khai trong môi trường máy chủ thử nghiệm với cấu hình như sau:

- CPU: Intel(R) Xeon(R) CPU @ 2.00GHz (2 nhân, 4 luồng)
- RAM: 12.7 GB DDR4
- GPU: NVIDIA-SMI 550.54.15, RAM 15GB (dùng để chạy LLM trên nền tảng Ollama)
- Storage: 112.6GB NVMe SSD
- Hệ điều hành: Ubuntu 22.04.4 LTS

4.1.2 Công nghệ và phần mềm sử dụng

Các thành phần phần mềm chính:

- Python 3.12.7
- Ollama 0.12.6 – dùng để chạy LLM cục bộ, giảm chi phí tính toán và đảm bảo dữ liệu không rò rỉ ra ngoài.
- LangChain 0.3.25 – framework cho kiến trúc RAG.
- ChromaDB 1.0.12 – vector database cục bộ để lưu embedding và phục vụ truy xuất ngữ nghĩa.
- Flask 2.3.3 – triển khai API ứng dụng chatbot.

- BeautifulSoup, Requests – dùng crawl văn bản pháp luật từ vbpl.vn.
- SentenceTransformers – tạo embedding từ các đoạn văn bản.
- Mô hình embedding: BAAI/bge-small-en-v1.5

Hệ thống được tổ chức theo mô hình client–server, trong đó phần giao diện được triển khai qua Flask, còn phần suy luận LLM chạy trong môi trường Ollama.

4.2. Dữ liệu thực nghiệm

Bộ dữ liệu sử dụng cho thực nghiệm được xây dựng từ bộ văn bản pháp luật của Việt Nam trong lĩnh vực an toàn thực phẩm, được thu thập tự động từ website vbpl.vn thông qua chương trình thu thập văn bản đã được xây dựng và trình bày trong các chương trước. Sau quá trình thu thập, loại bỏ dữ liệu trùng lặp và kiểm tra lỗi, tập dữ liệu cuối cùng gồm 170 văn bản pháp luật chính thức, bao gồm luật, nghị định, thông tư, quyết định, chỉ thị và thông tư liên tịch.

Tập văn bản bao quát đầy đủ các cấp độ điều chỉnh pháp luật trong hệ thống pháp luật Việt Nam. Cụ thể, nhóm văn bản chiếm tỷ trọng lớn nhất là các thông tư, với hơn 55% tổng số tài liệu. Tiếp theo là nhóm quyết định hành chính của các bộ ngành, chiếm hơn 21%. Các văn bản nghị định, chỉ thị và thông tư liên tịch chiếm tỷ trọng còn lại. Việc phân bổ dữ liệu như vậy phản ánh đúng thực tiễn quản lý nhà nước trong lĩnh vực an toàn thực phẩm, trong đó các thông tư và quyết định chiếm ưu thế về số lượng và mức độ chi tiết.

Để phục vụ cho mô hình RAG, mỗi văn bản được chuyển đổi sang định dạng văn bản thuần (.txt), chuẩn hóa encoding, loại bỏ ký tự đặc biệt, xử lý xuống dòng và tách trang. Sau đó, dữ liệu được chia thành các phân đoạn nhỏ (chunk) theo kích thước cố định 800–1000 ký tự, chồng lấn từ 150–200 ký tự để đảm bảo tính liền mạch ngữ nghĩa. Bằng cơ chế gán mã băm SHA-256, mỗi phân đoạn được xác định duy nhất, bảo đảm quá trình đồng bộ dữ liệu hiệu quả và có thể truy vết.

Tập dữ liệu này có quy mô vừa phải, đủ để đánh giá hiệu năng của mô hình, đồng thời phản ánh đúng các đặc thù pháp lý, cấu trúc văn bản và ngôn ngữ

chuyên ngành trong lĩnh vực an toàn thực phẩm. Đây là cơ sở cho các thử nghiệm truy xuất ngữ nghĩa, sinh câu trả lời và đo lường chất lượng phản hồi của hệ thống chatbot pháp luật.

4.3. Các kịch bản thực nghiệm

Để đánh giá hệ thống một cách toàn diện, những kịch bản thử nghiệm được xây dựng dựa trên hành vi và nhu cầu thực tế của người dùng khi tra cứu văn bản pháp luật. Các kịch bản này bao gồm cả các câu hỏi thông tin cơ bản và câu hỏi có tính chất pháp lý chuyên sâu. Nhìn chung, các kịch bản được phân thành bốn nhóm chính:

Thứ nhất, nhóm câu hỏi truy xuất thông tin trực tiếp từ văn bản pháp luật. Đây là những câu hỏi dạng "định nghĩa", "yêu cầu", "điều kiện", hoặc "đối tượng áp dụng". Ví dụ: "Điều kiện để cơ sở sản xuất thực phẩm được cấp giấy chứng nhận an toàn thực phẩm là gì?", hoặc "Khái niệm phụ gia thực phẩm được quy định như thế nào?". Nhóm câu hỏi này giúp đánh giá khả năng tìm kiếm chính xác và trích xuất thông tin của thành phần retriever.

Thứ hai, nhóm câu hỏi liên quan đến việc xác định trách nhiệm pháp lý hoặc cơ quan quản lý. Đây là dạng câu hỏi đòi hỏi hệ thống phải hiểu được nội dung và bối cảnh pháp luật. Ví dụ: "Cơ quan nào chịu trách nhiệm thanh tra an toàn thực phẩm trên thị trường?", "Doanh nghiệp vi phạm quy định về ghi nhãn sẽ bị xử lý như thế nào?". Nhóm này đặc biệt quan trọng để đánh giá năng lực lập luận pháp lý của mô hình sinh phản hồi.

Thứ ba, nhóm câu hỏi tổng hợp thông tin hoặc yêu cầu hệ thống đưa ra khuyến nghị. Ví dụ: "Doanh nghiệp khi xin cấp phép ATTP cần chuẩn bị những hồ sơ nào?", hoặc "Các yêu cầu về kiểm nghiệm thực phẩm đối với cơ sở sản xuất mới". Các câu trả lời cho nhóm này cần đảm bảo chính xác ngữ nghĩa, đầy đủ nội dung và phù hợp với bối cảnh pháp lý.

Các kịch bản thử nghiệm được thiết kế đa dạng, phản ánh đầy đủ các tình huống người dùng thực tế có thể gặp phải. Điều này giúp đánh giá toàn diện khả

năng hoạt động của hệ thống chatbot pháp luật, bao gồm khả năng tìm kiếm, hiểu ngôn ngữ chuyên ngành, trích xuất thông tin và sinh câu trả lời chính xác.

4.4. Đánh giá hiệu năng của hệ thống

Quá trình đánh giá được thực hiện dựa trên các tiêu chí bao gồm hiệu suất truy xuất (Recall@k), độ trung thực (Faithfulness), độ chính xác trích dẫn (Citation Accuracy). Các tiêu chí này được lựa chọn dựa trên đặc thù của hệ thống hỏi đáp pháp luật.

- Hiệu suất truy xuất (Recall@k): Đo lường tỷ lệ đoạn văn bản đúng sự thật xuất hiện trong top k kết quả trả về từ vector database (trong đề án này chọn k=5). Chỉ số này phản ánh khả năng tìm đúng căn cứ pháp lý của bộ nhúng (Embedding).
- Độ trung thực (Faithfulness): Đánh giá mức độ câu trả lời sinh ra hoàn toàn dựa trên ngữ cảnh được trích xuất, nhằm kiểm soát và loại bỏ hiện tượng ảo giác (hallucination) của mô hình ngôn ngữ.
- Độ chính xác trích dẫn (Citation Accuracy): Kiểm tra tính khớp lệnh giữa thông tin pháp luật cung cấp và nguồn văn bản đi kèm.

Bên cạnh các chỉ số tự động, đề án sử dụng thang đo Likert 5 điểm để đánh giá định tính. Các câu trả lời được chuyên gia chấm điểm dựa trên tính đúng đắn về mặt pháp lý và sự mạch lạc trong ngôn ngữ, với mức điểm 5 dành cho các câu trả lời hoàn toàn chính xác, đầy đủ căn cứ và không có thông tin dư thừa.

4.5. Thực nghiệm

4.5.1 Thực nghiệm với hệ thống hoàn chỉnh

Thực nghiệm trên tập 50 câu hỏi liên quan đến ATTP theo 3 nhóm, kết quả các tiêu chí được tổng hợp từ quá trình thực nghiệm.

Bảng 4.1 *Kết quả thực nghiệm*

Chỉ số đo lường	Giá trị đạt được
Độ phủ thông tin (Recall@5)	0.88
Độ trung thực (Faithfulness)	4.3/5.0
Độ chính xác trích dẫn	98%
Thời gian đồng bộ hóa (10 tệp)	18.2 giây
Độ trễ phản hồi trung bình	3.8 giây

Theo dữ liệu thực nghiệm, chỉ số Recall@5 đạt mức trung bình là 0.88, điều này minh chứng rằng bộ nhúng BAAI/bge-small-en-v1.5 dù nhẹ nhưng vẫn có khả năng bắt cặp ngữ nghĩa hiệu quả với dữ liệu tiếng Việt chuyên ngành pháp lý. Đặc biệt, độ chính xác trích dẫn nguồn đạt tỷ lệ ấn tượng 98%, đảm bảo rằng hầu hết các câu trả lời đều có căn cứ pháp lý rõ ràng, giúp tăng niềm tin cho người dùng khi tra cứu. Điểm Faithfulness trung bình đạt 4.3/5.0. Kết quả này cho thấy mô hình Qwen2.5:7B có khả năng tổng hợp thông tin tốt và tuân thủ chặt chẽ chỉ thị (Prompt) không tự ý suy diễn khi thông tin trong ngữ cảnh bị thiếu hụt. Trong các trường hợp câu hỏi nằm ngoài phạm vi dữ liệu đã nạp, hệ thống đã trả về thông báo "Không đủ căn cứ" thay vì trả lời sai, đây là một tính năng quan trọng để đảm bảo tính nghiêm túc của một trợ lý pháp lý. Thực nghiệm cũng ghi nhận thời gian phản hồi trung bình của hệ thống dao động từ 2-5 giây tùy vào độ phức tạp của câu hỏi khi chạy trên cấu hình phần cứng tiêu chuẩn. Mặc dù việc sử dụng mô hình nhúng tiếng Anh cho dữ liệu tiếng Việt có thể gây ra một số sai số nhỏ về mặt ngữ nghĩa tinh vi, nhưng với chiến lược phân đoạn văn bản hợp lý và cơ chế truy xuất Top-k, hệ thống đã cân bằng được giữa tốc độ xử lý và độ chính xác cần thiết cho mục tiêu triển khai ngoại tuyến.

4.5.2 Thực nghiệm với mô hình không sử dụng RAG

Thực hiện chạy 50 câu lệnh thực nghiệm mà không sử dụng kỹ thuật RAG. mô hình Qwen2.5:7B được yêu cầu trả lời các câu hỏi về an toàn thực phẩm chỉ dựa

trên tri thức đã được huấn luyện sẵn (Internal Knowledge), mà không được cung cấp các đoạn văn bản luật từ kho dữ liệu vector. Các tham số về Temperature và Top-k được giữ nguyên như trong kịch bản có RAG để đảm bảo tính công bằng. Nhận xét:

- Hiện tượng ảo giác (Hallucination): Khi không có RAG, mô hình có xu hướng tự sinh ra các quy định nghe có vẻ hợp lý nhưng không tồn tại trong văn bản luật Việt Nam hiện hành.
- Độ chính xác trích dẫn (Citation Accuracy): Chỉ số này giảm xuống đối với mô hình No-RAG, do mô hình không có khả năng truy cập trực tiếp vào nguồn văn bản gốc để trích dẫn chính xác điều, khoản.
- Tính cập nhật: Do tri thức của LLM bị giới hạn bởi thời điểm dữ liệu huấn luyện, mô hình No-RAG không thể cập nhật các thông tư, nghị định mới ban hành (như các văn bản trong năm 2024, 2025) mà đề án đã dùng trong kho dữ liệu.

```python
import requests
import json

url = "http://localhost:11434/api/generate"

prompt = "Liệt kê các nguyên tắc quản lý an toàn thực phẩm"

payload = {
    "model": "qwen2.5:7b-instruct-q4_0",
    "prompt": prompt,
    "stream": False
}

response = requests.post(url, json=payload)
print(response.json()["response"])
```

Hình 4.1 *Mã nguồn thực nghiệm không dùng RAG*

PHẦN KẾT LUẬN

1. Kết quả nghiên cứu của đề tài

Đề án đã nghiên cứu và phát triển thành công hệ thống hỏi đáp pháp lý dựa trên kiến trúc RAG (Retrieval-Augmented Generation), đáp ứng hiệu quả yêu cầu khai thác tri thức đặc thù từ các văn bản pháp luật về an toàn thực phẩm tại Việt Nam. Về mặt dữ liệu, đề án đã xây dựng được kho tri thức số hóa có cấu trúc từ tập hợp các văn bản luật, nghị định và thông tư hiện hành. Quy trình tiền xử lý được thiết kế chuyên sâu bao gồm các bước làm sạch văn bản, chuẩn hóa ký tự và đặc biệt là triển khai cơ chế phân đoạn văn bản thận trọng (conservative chunking) có chồng lấn (overlap). Kỹ thuật này đóng vai trò quyết định trong việc bảo toàn ngữ cảnh pháp lý và duy trì tính liên kết giữa các điều khoản khi chuyển đổi sang dạng vector.

Trong việc xây dựng pipeline RAG, đề án đã ứng dụng mô hình nhúng (embedding) tối ưu cho môi trường tài nguyên hạn chế kết hợp với cơ chế tìm kiếm ngữ nghĩa (semantic search) để truy xuất chính xác các căn cứ pháp lý. Kết quả thực nghiệm trên bộ dữ liệu kiểm thử (Ground-truth) gồm các tình huống pháp lý đa dạng đã chứng minh tính ưu việt của hệ thống. Cụ thể, khả năng truy xuất đúng đoạn văn bản luật (Recall@5) đạt mức 0.88, trong khi độ chính xác trích dẫn nguồn (Citation Accuracy) đạt tỷ lệ ấn tượng 98%. Đặc biệt, qua đánh giá của chuyên gia dựa trên thang đo Likert, độ trung thực của câu trả lời (Faithfulness) đạt mức 4.3/5.0, cho thấy mô hình không chỉ phản hồi nhanh (trung bình 3.8 giây) mà còn đảm bảo tính chặt chẽ, mạch lạc và có căn cứ pháp lý rõ ràng. Những đóng góp này không chỉ khẳng định tính khả thi của mô hình RAG ngoại tuyến mà còn đề xuất một quy trình đánh giá chuẩn hóa cho các nghiên cứu về AI trong lĩnh vực luật pháp tại Việt Nam.

2. Hạn chế của đề tài

Mặc dù đạt được những kết quả khả quan, đề án vẫn tồn tại một số hạn chế nhất định cần được tiếp tục hoàn thiện. Trước hết, về phạm vi tri thức, kho dữ liệu hiện tại mới chỉ tập trung vào lĩnh vực an toàn thực phẩm, chưa bao quát được sự giao thoa phức tạp giữa các hệ thống pháp luật khác nhau. Hệ thống cũng gặp thách thức trong việc xử lý các mối liên kết động giữa các văn bản như các điều khoản sửa đổi, bổ sung hoặc thay thế theo thời gian.

Về mặt kỹ thuật, khả năng suy luận đa bước (multi-hop reasoning) của hệ thống vẫn còn giới hạn; độ chính xác có xu hướng giảm đối với các tình huống pháp lý yêu cầu sự tổng hợp thông tin từ nhiều nguồn văn bản khác nhau. Do sử dụng mô hình nhúng và mô hình ngôn ngữ nhỏ để đảm bảo vận hành ngoại tuyến, hệ thống đôi khi vẫn gặp khó khăn trong việc phân biệt các sắc thái thuật ngữ pháp lý tinh vi nếu câu hỏi của người dùng quá dài hoặc chứa nhiều từ ngữ không phổ biến. Ngoài ra, việc thiếu một bộ tái xếp hạng (Reranker) chuyên sâu và số lượng bộ câu hỏi thực nghiệm còn khiêm tốn khiến hệ thống chưa bao phủ hết được các tình huống pháp lý đa tầng như phân định trách nhiệm hành chính phức tạp hay các rủi ro tiềm ẩn về ảo giác ngôn ngữ (hallucination) trong mọi ngữ cảnh.

3. Nhận xét, đề xuất và khuyến nghị

Nhìn chung, kiến trúc RAG cho thấy mức độ phù hợp cao khi áp dụng vào bài toán truy vấn văn bản pháp luật tiếng Việt, do vừa đảm bảo tính chính xác dựa trên các nguồn văn bản gốc, vừa duy trì được sự tự nhiên và dễ hiểu trong câu trả lời cho người dùng. Các kết quả đánh giá định lượng cho thấy hệ thống đạt chất lượng tốt và đáp ứng được yêu cầu ứng dụng trong phạm vi tư vấn pháp lý cơ bản. Tuy nhiên, để có thể triển khai trong môi trường pháp lý chuyên nghiệp như cơ quan nhà nước hoặc doanh nghiệp, hệ thống vẫn cần được bổ sung các cơ chế kiểm chứng pháp lý chặt chẽ hơn nhằm đảm bảo độ tin cậy và tính pháp lý của thông tin cung cấp.

Trong thời gian tới, hệ thống có thể được cải tiến theo nhiều hướng nhằm nâng cao hiệu quả và độ chính xác. Trước hết, cần mở rộng quy mô và làm phong phú hơn kho dữ liệu pháp luật, bao gồm các văn bản mới ban hành, văn bản sửa đổi, cũng như các văn bản liên quan đến nhiều lĩnh vực khác nhau. Bên cạnh đó, việc tích hợp các thuật toán reranking như Cross-Encoder BERT hoặc Legal-BERT sẽ giúp tăng độ chính xác trong quá trình truy xuất các đoạn văn bản có mức độ liên quan cao nhất. Hệ thống cũng nên được bổ sung cơ chế kiểm chứng pháp lý, trong đó mô hình bắt buộc phải trích dẫn rõ ràng điều, khoản, điểm cụ thể khi sinh câu trả lời. Ngoài ra, việc tách riêng và phân loại các lỗi trong quá trình sinh câu trả lời, chẳng hạn như sai luật, thiếu ý hoặc suy luận không chính xác, sẽ hỗ trợ hiệu quả cho công tác theo dõi, đánh giá và cải thiện toàn bộ pipeline. Cuối cùng, việc xây dựng một giao diện trực quan hơn, cho phép hiển thị đầy đủ các đoạn văn bản đã được sử dụng để tạo câu trả lời, sẽ góp phần nâng cao tính minh bạch của hệ thống.

Từ góc độ khuyến nghị, các hệ thống AI ứng dụng trong lĩnh vực pháp lý cần tuân thủ nghiêm ngặt các nguyên tắc về minh bạch, khả năng truy vết nguồn tham chiếu và tuyệt đối không được tự suy luận ngoài phạm vi văn bản gốc. Đồng thời, kho dữ liệu pháp luật cần được cập nhật thường xuyên theo các đợt ban hành và sửa đổi văn bản nhằm đảm bảo thông tin mà hệ thống cung cấp luôn chính xác và phù hợp với bối cảnh pháp lý hiện hành.

4. Hướng nghiên cứu tiếp theo

Để tiếp tục mở rộng và nâng cao chất lượng hệ thống, các hướng phát triển trong tương lai có thể tập trung vào việc xây dựng RAG đa tầng nhằm phân tách truy xuất theo cấp độ văn bản pháp luật; huấn luyện hoặc tinh chỉnh mô hình ngôn ngữ chuyên biệt cho lĩnh vực pháp lý Việt Nam để cải thiện khả năng hiểu thuật ngữ; tích hợp cơ chế suy luận có cấu trúc kèm căn cứ pháp lý rõ ràng; phát triển hệ thống hỏi–đáp pháp lý chuyên sâu có khả năng phân tích, đối chiếu văn bản; đồng thời xây dựng bộ dữ liệu benchmark pháp lý chuẩn hóa và cung cấp

API hoặc chatbot pháp lý để dễ dàng tích hợp vào các hệ thống hành chính và cổng thông tin pháp luật.

TÀI LIỆU THAM KHẢO

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[2] Facebook AI Research, "Retrieval Augmented Generation: Streamlining the creation of intelligent natural language processing models," 2020. [Online]. Available: https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/. [Accessed: 05-Dec-2025].

[3] LangChain, "LangChain Overview and Components," 2024. [Online]. Available: https://docs.langchain.com. [Accessed: 05-Dec-2025].

[4] LlamaIndex, "Documentation: Building Context-Augmented Applications," 2024. [Online]. Available: https://docs.llamaindex.ai. [Accessed: 05-Dec-2025].

[5] Văn bản quy phạm pháp luật Bộ Y tế, "Cổng thông tin văn bản quy phạm pháp luật Bộ Y tế," 2024. [Online]. Available: https://vbpl.vn/boyte/Pages/Home.aspx. [Accessed: 05-Dec-2025].

[6] Thư viện Pháp luật, "Cơ sở dữ liệu pháp luật Việt Nam," 2024. [Online]. Available: https://thuvienphapluat.vn/. [Accessed: 05-Dec-2025].

[7] LuatVietnam, "Cổng thông tin tra cứu văn bản pháp luật," 2024. [Online]. Available: https://luatvietnam.vn/. [Accessed: 05-Dec-2025].

[8] Hoorangyee, "LRAGE: Legal Retrieval-Augmented Generation Engine," GitHub Repository, 2023. [Online]. Available: https://github.com/hoorangyee/LRAGE. [Accessed: 05-Dec-2025].

[9] Lawglance, "Lawglance: AI-powered legal information retrieval," GitHub Repository, 2023. [Online]. Available: https://github.com/lawglance/lawglance. [Accessed: 05-Dec-2025].

[10] D. Q. Nguyen, X. N. Pham, and A. T. Nguyen, "PhoBERT: Pre-trained Language Models for Vietnamese," 2020. [Online]. Available: https://doi.org/10.18653/v1/2020.findings-emnlp.92. [Accessed: 05-Dec-2025].

[11] Hugging Face, "BAAI/bge-small-en-v1.5 Model," 2023. [Online]. Available: https://huggingface.co/BAAI/bge-small-en-v1.5. [Accessed: 05-Dec-2025].

[12] Google AI, "Gemini 1.5 Flash API Documentation," 2024. [Online]. Available: https://ai.google.dev/gemini-api. [Accessed: 05-Dec-2025].

[13] Alibaba Cloud, "Qwen2.5 Model Documentation," 2024. [Online]. Available: https://modelscope.cn/models/qwen. [Accessed: 05-Dec-2025].

[14] Chroma Team, "Chroma Documentation," 2023. [Online]. Available: https://docs.trychroma.com/. [Accessed: 05-Dec-2025].

[15] D. Shu *et al.*, "LawLLM: Law Large Language Model for the US Legal System," *Proc. ACM CIKM*, 2024.

[16] S. Yue *et al.*, "LawLLM: Intelligent Legal System with Legal Reasoning and Verifiable Retrieval," *arXiv preprint*, Fudan University, 2024.

[17] P. Colombo *et al.*, "SaulLM-7B: A Pioneering Large Language Model for Law," *arXiv preprint arXiv:2403.03883v2*, 2024.

[18] S. Batifol, "Simplifying Legal Research with RAG, Milvus, and Ollama," *Zilliz Blog*, Jul. 2024. [Online]. Available: https://zilliz.com/blog. [Accessed: 05-Dec-2025].

[19] R. Jay, *Generative AI Apps with LangChain and Python*, Apress, 2024.

[20] D. Polzer, *RAG with Python Cookbook: Practical Recipes from Data Preprocessing to LLM Agents*, O'Reilly Media, 2025.

[21] D. Rothman, *RAG-Driven Generative AI: Build Custom Retrieval-Augmented Generation Pipelines with LlamaIndex, Deep Lake, and Pinecone*, Packt Publishing, 2024.

[22] V. Alto, *Building LLM Powered Applications: Create Intelligent Apps and Agents with Large Language Models*, Packt Publishing, 2024.

[23] Et Tu Code, *Demystifying LLM, AI Mathematics and Hardware Infra-Independently*, ETC Publications, 2024.

[24] Code Et Tu, *LLM Transformer RAG AI: Mastering Large Language Models, Transformer Models, and Retrieval-Augmented Generation (RAG) Technology*, ETC Publications, 2024.

[25] "Natural Language Processing — Resources," deeplearning.ai. [Online]. Available: https://www.deeplearning.ai/resources/natural-language-processing/ [Accessed: 05-Dec-2025].

[26] "What is RAG?" Intel (Vietnam). [Online]. Available: https://www.intel.vn/content/www/vn/vi/learn/what-is-rag.html [Accessed: 05-Dec-2025].