XÂY DỰNG BỘ DỮ LIỆU LUẬT GIAO THÔNG PHÙ HỢP CHO HỆ THỐNG HỎI ĐÁP

Nguyễn Hữu Sang, Lê Cường Thịnh, Nguyễn Duy Thịnh
Trường Đại học Công nghệ Thông tin, ĐHQG-HCM
MSSV: 22521242, 22521409, 22521414
Email: 22521242@gm.uit.edu.vn, 22521409@gm.uit.edu.vn, 22521414@gm.uit.edu.vn

Tóm tắt nội dung

Trong nghiên cứu này, chúng tôi tập trung vào việc xây dựng một bộ dữ liệu về luật giao thông đường bộ Việt Nam để phục vụ cho việc phát triển một hệ thống hỏi đáp (chatbot) thông minh. Dự án thực hiện các công đoạn chính bao gồm thu thập dữ liệu từ các văn bản pháp luật, tiền xử lý để làm sạch và chuẩn hóa, sau đó cấu trúc lại dữ liệu theo một quy trình phù hợp cho hệ thống Retrieval-Augmented Generation (RAG). Hệ thống này có mục tiêu giúp người dùng dễ dàng tra cứu, hiểu và áp dụng đúng các quy định pháp luật khi tham gia giao thông, đồng thời hỗ trợ công tác tuyên truyền và giáo dục pháp luật.

1 Giới thiệu

1.1 Động lực thực hiện

Trong bối cảnh xã hội hiện nay, nhu cầu tìm hiểu và tuân thủ luật giao thông là vô cùng cần thiết. Tuy nhiên, hệ thống văn bản pháp luật thường phức tạp và khó tiếp cận đối với người dân. Do đó, việc xây dựng một chatbot tư vấn luật giao thông có khả năng giải đáp các thắc mắc một cách nhanh chóng và chính xác là một bài toán cấp thiết. Chatbot này không chỉ giúp người dùng tra cứu thông tin dễ dàng mà còn góp phần nâng cao ý thức chấp hành pháp luật, giảm thiểu tai nạn và hỗ trợ các cơ quan chức năng trong việc phổ biến pháp luật.

1.2 Mục đích

Mục tiêu chính của dự án là xây dựng và xử lý một bộ dữ liệu về luật giao thông phù hợp để làm nền tảng cho việc xây dựng một hệ thống hỏi đáp hiệu quả. Bộ dữ liệu này được thu thập từ các nguồn pháp luật chính thống và được xử lý để đảm bảo tính chính xác, đầy đủ và có cấu trúc, sẵn sàng cho việc tích hợp vào các mô hình ngôn ngữ lớn theo hệ thống RAG.

1.3 Các đề tài liên quan

Hiện nay, đã có một số dự án ứng dụng AI để xây dựng chatbot pháp luật tại Việt Nam, tiêu biểu là:

- **DTchat:**[1] Là một chatbot được phát triển bởi báo Dân trí, chuyên hỗ trợ hỏi đáp về các vi phạm giao thông theo Nghị định 168. DTchat sử dụng công nghệ RAG để truy xuất thông tin chính xác, minh bạch về nguồn và có khả năng phản hồi theo ngữ cảnh.
- **AI Tra Cứu Luật:**[2] Là sản phẩm hợp tác của Viện ABAII, cho phép người dùng tra cứu trên 145.000 văn bản pháp luật. Ứng dụng này sử dụng xử lý ngôn ngữ tự nhiên (NLP) để hiểu các câu hỏi pháp lý và tự động hóa việc tư vấn.

2 Bộ dữ liệu

Quá trình xây dựng và xử lí bộ dữ liệu của chúng tôi bao gồm hai giai đoạn chính.

2.1 Giai đoạn 1

2.1.1 Thu thập dữ liệu

a) Nguồn dữ liệu

Chúng tôi thực hiện thu thập các văn bản pháp luật (vbpl) về giao thông từ trang web: *Văn bản pháp luật*[3], đây là cơ sở dữ liệu quốc gia về các văn bản pháp luật chính thống của nhà nước, đảm bảo nguồn gốc, độ tin cập, uy tín về dữ liệu thu thập.

Chúng tôi tiến hành thu thập các văn bản pháp luật của hai bộ/ngành: Bộ Giao Thông Vận Tải và Bộ Công An.

- Với bộ Giao Thông Vận Tải: thu thập toàn bộ các văn bản pháp luật.
- Với bộ Công An: chỉ thu thập các văn bản pháp luật liên quan đến lĩnh vực giao thông.

[1] https://dantri.ai/giaothong
[2] https://aitracuuluat.vn/
[3] https://vbpl.vn/

b) Công cụ sử dụng

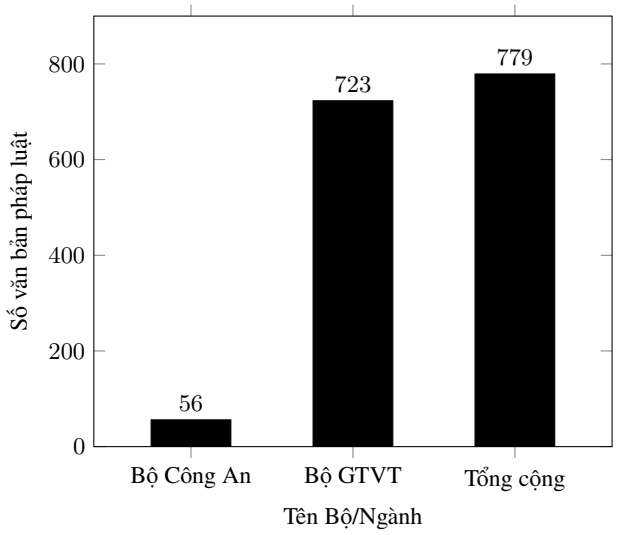
Chúng tôi sử dụng hai công cụ chính là **Selenium** và **Beautifulsoup** để truy cập và trích xuất văn bản từ trang web. Sau đó, chúng tôi sử dụng **Pandas** để lưu dữ liệu dạng bảng *.csv*.

c) Kết quả

Sau quá trình thu thập dữ liệu, chúng tôi kết hợp văn bản pháp luật ở cả hai bộ và thu được một tập dữ liệu bao gồm 779 văn bản pháp luật (số lượng chi tiết được mô tả ở **Hình 1**).

Tập dữ liệu chứa 3 cột bao gồm:

- *doc_id*: số hiệu của từng văn bản pháp luật.
- *context*: nội dung chính của văn bản pháp luật.
- *effective_day*: ngày văn bản pháp luật có hiệu lực.



Hình 1: Biểu đồ thể hiện số lượng văn bản pháp luật thu thập được theo các Bộ/Ngành

2.1.2 Tiền xử lí dữ liệu

Sau khi hoàn tất quá trình thu thập dữ liệu, chúng tôi tiến hành tiền xử lí với mục đích "làm sạch" cơ bản dữ liệu.

Với hai cột *doc_id* và *effective_day*, đây là các thuộc tính không thể thiếu cho mỗi vbpl, vì vậy chúng tôi tiến hành rà soát, kiểm tra để đảm bảo tính đầy đủ của hai thuộc tính này.

Với cột *context*, chúng tôi tiến hành các bước xử lí như sau: xác định các phần tử cấu thành: tiêu đề, mục lục, điều, khoản, điểm, phụ lục, chú giải; loại bỏ các phần tử không cần thiết: tiêu đề, biểu ngữ, thời gian, chữ kí; chuẩn hóa Unicode, chính tả.

Sau quá trình tiền xử lí này, chúng tôi thu được một tập dữ liệu đảm bảo tính đầy đủ và "sạch sẽ" để chuẩn bị cho giai đoạn tiếp theo.

2.2 Giai đoạn 2

Sau khi làm sạch cơ bản dữ liệu ở **Giai đoạn 1**, chúng tôi tiến hành chuỗi các bước chuyển đổi dữ liệu phù hợp cho việc xây dựng hệ thống dựa trên RAG.

Quá trình chuyển đổi này sẽ được chúng tôi trình bày chi tiết ở phần **3.2.1** để phù hợp với việc mô tả luồng xử lí chính của hệ thống.

3 Phương pháp thực hiện

Chúng tôi quyết định sử dụng kĩ thuật **Retrieval-Augmented Generation (RAG)** làm phương pháp chính để thực hiện xây dựng hệ thống hỏi đáp về pháp luật dựa trên bộ dữ liệu đã thu thập được.

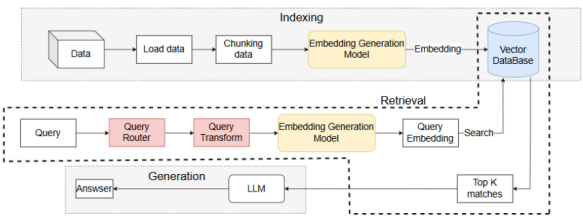
3.1 RAG

Retrieval-Augmented Generation (RAG) là một kỹ thuật tiên tiến kết hợp mô hình sinh ngôn ngữ với một hệ thống truy xuất thông tin từ nguồn tri thức bên ngoài. Cách tiếp cận này bao gồm hai thành phần chính:

- **Retrieval (Truy xuất):** Tìm kiếm và lấy ra những thông tin liên quan nhất đến câu hỏi của người dùng từ một kho dữ liệu lớn.
- **Generation (Sinh nội dung):** Sử dụng một mô hình ngôn ngữ lớn (LLM) để tạo ra câu trả lời tự nhiên, mạch lạc dựa trên thông tin đã được truy xuất.

3.2 Kiến trúc hệ thống

Hệ thống được xây dựng dựa trên kiến trúc RAG, bao gồm 3 giai đoạn chính: Indexing, Retrieval và Generation. So với pipeline RAG cơ bản, chúng tôi giới thiệu một hệ thống Advanced RAG (Hình 2) có thêm các module được tùy chỉnh là Query Router và Query Transformation để tối ưu hóa hiệu suất.

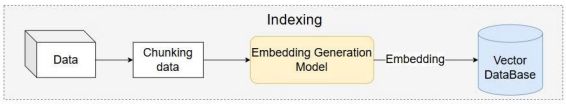


Hình 2: Sơ đồ kiến trúc Advanced RAG.

3.2.1 Indexing

Indexing là giai đoạn quan trọng nhằm chuyển đổi dữ liệu đầu vào thành các vector số có khả năng phản ánh ngữ nghĩa, phục vụ hiệu quả cho bước truy xuất thông tin (Retrieval).

Quá trình này bao gồm ba bước chính: (1) phân chia dữ liệu thu thập thành các đoạn nhỏ (chunks) có ý nghĩa **(Chunking)**; (2) ánh xạ từng đoạn thành vector số thông qua mô hình embedding **(Embedding)**; (3) lưu trữ các vector này vào cơ sở dữ liệu vector để truy vấn sau này **(Storing)**. **Hình 3** minh họa luồng xử lý tổng quát của quá trình Indexing trong hệ thống RAG.



Hình 3: Sơ đồ minh hoạ quá trình Indexing trong hệ thống RAG

a) Chunking

Bước đầu tiên trong quá trình này là chunking, tức là chia documents thành nhiều chunks, hoặc gọi là phân đoạn nhỏ hơn nhưng vẫn có nghĩa.

Chúng tôi khảo sát hai cách chunking chính thường được áp dụng là:

- **Semantic chunking**: được thực hiện dựa trên việc phân tích ngữ nghĩa của văn bản thay vì dựa trên cấu trúc của văn bản. Chúng ta sẽ chia văn bản dựa trên việc tính toán độ tương đồng giữa các câu.
- **Recursive chunk**: được thực hiện dựa vào những kí tự phân cách được thiết lập sẵn như: (".", "?", ";"...). Chúng ta sẽ lần lượt "cắt" theo thứ tự ưu tiên dựa trên những kí tự này sao cho mỗi chunk được lấy ra vừa là dài nhất vừa giữ được nội dung nguyên vẹn.

Chunking là một bước cực kì quan trọng, nó ảnh hưởng trực tiếp đến hiệu suất và chất lượng của hệ thống. Vì vậy, sau khi nghiên cứu, thử nghiệm, chúng tôi quyết định sử dụng **Recursive chunk** làm phương pháp chính bởi vì: phương pháp này có thể giữ lại được cấu trúc văn bản rõ ràng, đặc biệt phù hợp cho loại văn bản về pháp luật, được cấu trúc bởi các Điều/ Khoản cụ thể. Việc chia các văn bản theo cấu trúc này không chỉ giúp tăng độ chính xác về nội dung cho mỗi chunk mà còn thuận tiện cho việc tìm kiếm và truy xuất sau này.

Quá trình chunking được thực hiện trên mỗi văn bản pháp luật lưu trữ trong cột *context*. Trước tiên, toàn bộ văn bản được phân tách theo từng Điều/ Khoản. Tiếp đó, mỗi Điều/Khoản được tách nhỏ hơn nữa thành các câu dựa trên các ký tự phân cách định trước. Các câu sau khi tách sẽ được nhóm thành các đoạn có độ dài tối đa 200 từ; đồng thời, giữa hai đoạn liên tiếp luôn phải đảm bảo độ trùng lặp khoảng hai câu để duy trì tính liên tục về ngữ nghĩa. Trong trường hợp một đoạn có độ dài dưới 50 từ, nó sẽ được ghép với đoạn kế tiếp sao cho vẫn thỏa mãn các quy tắc trên, trừ khi đó là đoạn cuối cùng của một Điều/Khoản.

Mỗi đoạn (chunk) được tách ra từ văn bản pháp luật sẽ được lưu trữ cùng một bộ siêu dữ liệu (metadata) như mô tả trong **Bảng 1**, bao gồm các thông tin giúp đảm bảo tính chi tiết và nhất quán khi tra cứu, tạo điều kiện thuận lợi cho việc cập nhật hoặc thay thế dữ liệu khi văn bản gốc có sự thay đổi. **Hình 4** minh họa cấu trúc một chunk hoàn chỉnh sau khi hoàn tất quá trình này. Kết thúc quá trình Chunking, chúng tôi thu được tổng cộng **30769** chunks, đây sẽ là "tập dữ liệu" mới phục vụ cho quá trình tiếp theo - Embedding.

b) Embedding

Embedding là quá trình biến đổi mỗi đoạn văn bản (chunk) thành một vector số trong không gian nhiều chiều, nhờ đó mô tả được ngữ nghĩa nội dung và hỗ trợ tìm kiếm thông tin dựa trên ngữ nghĩa thay vì từ khóa đơn thuần. Trong nghiên cứu này, chúng tôi sử dụng hai mô hình chính là **halong_embedding**[4] và **Vietnamese PhoBERT base**[5] để thực hiện điều này; đồng thời so sánh kết quả tạo vector từ cả hai mô hình nhằm đánh giá khách quan chất lượng embedding và hiệu quả của bước truy xuất (Retrieval).

c) Storing

Giai đoạn Storing chịu trách nhiệm lưu trữ, đánh chỉ mục và thực hiện truy vấn tương tự (similarity search) trên không gian vector đa chiều. Việc sử dụng một vector database cho phép truy xuất các embedding với độ trễ thấp và thông lượng cao, đồng thời đơn giản hóa quá trình bảo trì: khi có dữ liệu mới hoặc cải tiến mô hình, chỉ cần tái sinh các vector tương ứng mà không phải điều chỉnh lại toàn bộ pipeline. Trong nghiên cứu này, chúng tôi triển

[4] https://huggingface.co/hiieu/halong_embedding
[5] https://huggingface.co/VoVanPhuc/sup-SimCSE-VietNamese-phobert-base

```
{"id": 592883,
 "doc_id": "73/2024/TT-BCA",
 "effective_date": "01/01/2025",
 "chapter": "Chương III",
 "article": "Điều 25",
 "chunk_index": 7,
 "context": "Công an xã, phường, thị trấn khi nhận được thông báo có trách nhiệm chuyển đến chủ phương
tiện, đề nghị chủ phương tiện thực hiện theo thông báo và thông báo lại cho cơ quan Công an đã gửi thông
báo (theo mẫu số 04 ban hành kèm theo Thông tư này)."
}
```


Hình 4: Cấu trúc một chunk hoàn chỉnh

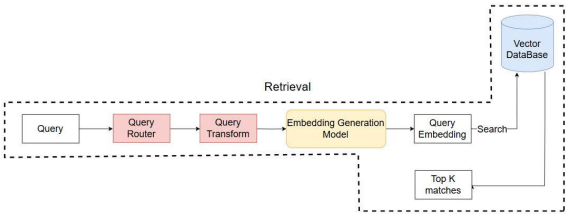
Thuộc tính	Mô tả
id	Mã định danh duy nhất cho mỗi chunk.
doc_id	số hiệu của văn bản pháp luật gốc
effective_date	Ngày có hiệu lực của văn bản pháp luật gốc
chapter	Định danh Chương gốc của mỗi chunk
article	Định danh Điều/ Khoản gốc của mỗi chunk
chunk_index	Thứ tự của chunk trong mỗi điều được tách (bắt đầu từ 0).
context	Nội dung của mỗi chunk.

Bảng 1: Các thuộc tính metadata lưu trữ kèm mỗi chunk

khai **Qdrant** — một cơ sở dữ liệu vector mã nguồn mở — để lưu trữ các embedding. Bên cạnh truy vấn theo khoảng cách cosine hoặc dot-product, **Qdrant** hỗ trợ lọc kết quả dựa trên các điều kiện metadata, phù hợp với cấu trúc của dữ liệu, nhờ đó nâng cao độ chính xác của kết quả truy xuất.

3.2.2 Retrieval

Giai đoạn Retrieval chịu trách nhiệm truy vấn kho embedding để tìm và chọn lọc những đoạn văn bản (chunk) liên quan nhất với truy vấn đầu vào. Sau khi biến đổi truy vấn thành embedding, hệ thống sử dụng các phương pháp tìm kiếm để trả về các chunk phù hợp, làm đầu vào cho bước sinh văn bản (Generation) nhằm đảm bảo câu trả lời chính xác và giàu thông tin.



Hình 5: Advance Retrieval: kết hợp hai mô-đun chính là Query Router và Query Transform

a) Advance Retrieval

So với kiến trúc hệ thống truy xuất truyền thống, chúng tôi đề xuất một cơ chế Retrieval nâng cao, trong đó hai mô-đun bổ sung **Query Router** và **Query Transform** — được tích hợp vào pipeline trước bước tính toán nhằm tối ưu hóa khả năng tìm kiếm và tăng cường độ phù hợp của kết quả truy xuất. Quá trình này được minh họa trong **Hình 5**.

Cụ thể, Query Router có chức năng phân loại truy vấn đầu vào thành hai nhóm chính: in-domain (nội miền, phù hợp với phạm vi dữ liệu hệ thống) và out-of-domain (ngoại miền, không phù hợp hoặc nằm ngoài phạm vi tri thức hệ thống). Dựa trên kết quả phân loại, mô-đun này sẽ định tuyến truy vấn đến không gian tìm kiếm thích hợp, giúp giảm nhiễu trong kết quả truy xuất, tăng tốc độ truy vấn cho hệ thống.

Trong khi đó, Query Transform đóng vai trò cải thiện biểu diễn ngữ nghĩa của truy vấn bằng cách khai thác ngữ cảnh lịch sử từ các lượt truy vấn trước đó. Nếu người dùng đang trong một phiên hỏi đáp liên tục, mô-đun này sẽ tái cấu trúc truy vấn hiện tại bằng cách bổ sung thông tin ngữ cảnh từ câu truy vấn trước, nhằm bảo toàn mạch hội thoại và tăng độ đầy đủ về ngữ nghĩa. Điều này đặc biệt hữu ích trong các tình huống hỏi đáp theo chuỗi, nơi mỗi câu hỏi có thể phụ thuộc vào ngữ cảnh được thiết lập trước đó.

Sự kết hợp giữa hai mô-đun giúp hệ thống không chỉ giúp hệ thống truy xuất chính xác hơn mà còn phản hồi hiệu quả trong các kịch bản truy vấn phức tạp và liên tục.

b) Information Retrieval

Mục tiêu chính của Retrieval là với một truy vấn bất kỳ, xác định và truy xuất top-K đoạn văn bản (chunks) có mức độ phù hợp cao nhất với nội dung truy vấn.

Trong lĩnh vực Information Retrieval, ba phương pháp truy xuất chính thường được sử dụng gồm:

- **Full-text Search**: Truy xuất dựa trên từ khóa, sử dụng các thuật toán như TF-IDF hoặc BM25 để tính độ quan trọng của từ trong tài liệu.
- **Vector Search**: Sử dụng mô hình học sâu để ánh xạ truy vấn và văn bản vào không gian vector, sau đó đánh giá mức độ tương đồng ngữ nghĩa bằng các phép đo như cosine similarity hoặc dot product.
- **Hybrid Search**: Kết hợp giữa hai phương pháp trên với tỉ lệ nhất định để tận dụng ưu điểm của từng loại, cho phép truy xuất hiệu quả cả về từ khóa lẫn ngữ nghĩa.

Trong nghiên cứu này, chúng tôi triển khai và so sánh cả ba phương pháp truy xuất nêu trên, bao gồm full-text search, vector search và hybrid search, nhằm đánh giá hiệu quả truy xuất của từng phương pháp từ đó đưa ra lựa chọn tối ưu về chiến lược truy xuất phù hợp nhất với hệ thống. Kết quả so sánh ba phương pháp được trình bày ở Bảng 2.

3.2.3 Generation

Generation đóng vai trò tạo ra câu trả lời cuối cùng dựa trên ngữ cảnh đã được truy xuất từ các tài liệu pháp luật. Mục tiêu chính của giai đoạn này là sinh văn bản mang tính chính xác về pháp lý, rõ ràng về lập luận, và trung lập về văn phong, sao cho phù hợp với các tiêu chuẩn của một hệ thống trợ lý pháp lý chuyên nghiệp.

a) Đặc điểm thiết kế Generation

Generation trong RAG được thiết kế theo các tiêu chí nghiệm ngặt nhằm đảm bảo tính chính xác, phù hợp và thẩm mỹ ngôn ngữ trong ngữ cảnh ứng dụng cụ thể.

Chúng tôi sử dụng mô hình **gpt-o3** của OpenAI, vốn dựa trên kiến trúc Transformer với khả năng xử lý ngữ nghĩa sâu và sinh văn bản mạch lạc. Mô hình này đảm bảo tuân thủ ngữ cảnh đã truy xuất, đồng thời hỗ trợ đa ngôn ngữ và giữ được phong cách pháp lý nghiêm túc. Nhờ vậy, thành phần Generation có thể sinh ra các câu trả lời chính xác, chuyên nghiệp và phù hợp với phong cách pháp lý, đảm bảo chất lượng và độ tin cậy cao cho hệ thống của chúng tôi.

Để hạn chế tình trạng trả lời một cách "ảo giác" (hallucination), chúng tôi chỉ cho phép mô hình sinh ra thông tin nằm gọn trong các đoạn văn bản đã được truy xuất, qua đó đảm bảo mọi kết quả đầu ra đều dựa trên nguồn dữ liệu đã được xác thực. Phần Generation được cấu trúc nhằm duy trì văn phong pháp lý nghiêm ngặt, với các câu logic rõ ràng và khách quan tương tự như trong các văn bản luật hoặc tài liệu tư vấn chuyên ngành.

Bên cạnh đó, hệ thống luôn phải ưu tiên trả lời đúng trọng tâm bằng cách đi thẳng vào yêu cầu của người dùng, từ đó giảm thiểu các nội dung không cần thiết và tối ưu thời gian đọc hiểu. Cuối cùng, nhằm đáp ứng nhu cầu đa dạng về ngôn ngữ, chúng tôi tích hợp một khả năng cho phép mô hình sinh câu trả lời bằng tiếng Việt hoặc bất kỳ ngôn ngữ đầu vào nào, giúp nâng cao tính linh hoạt và mở rộng phạm vi ứng dụng.

b) Cấu trúc Prompt và Tối ưu hóa

Trong hệ thống RAG, prompt đóng vai trò then chốt trong việc định hướng và tối ưu hóa quá trình Generation, đảm bảo mô hình sinh ra câu trả lời chính xác, nhất quán và phù hợp với ngữ cảnh thu hồi thông tin.

Cấu trúc prompt được chúng tôi thiết kế linh hoạt dưới dạng dynamic prompt nhằm tối ưu hóa chất lượng và tính liên tục của quá trình sinh đáp án. Đầu tiên, phần tóm tắt ngữ cảnh truy xuất cung cấp các trích đoạn đã được truy hồi, tạo nền tảng thông tin vững chắc cho bước sinh ngôn ngữ. Tiếp theo, câu hỏi của người dùng được lồng ghép một cách rõ ràng để đảm bảo mô hình hiểu đúng yêu cầu. Phần lệnh định hướng sau đó đưa ra chỉ dẫn cụ thể về cách thức trả lời, trong khi bộ ràng buộc đảm bảo thông tin không vượt ra ngoài tài liệu gốc, giữ giọng điệu trung lập và tuân thủ văn phong pháp lý.

Quá trình này còn được mở rộng bằng khả năng ghi nhớ lịch sử hội thoại, giúp duy trì mạch truy vấn – đáp giữa người dùng và hệ thống. Đặc biệt, prompt có thể tự điều chỉnh theo độ dài ngữ cảnh hoặc mức độ trang trọng của câu hỏi, đồng thời hỗ trợ cơ chế dịch ngược kết quả đầu ra khi người dùng tương tác bằng tiếng Anh hoặc các ngôn ngữ khác, từ đó nâng cao tính nhất quán và trải nghiệm sử dụng.

IR	Hit Rate@10	MRR@10	MAP@10
BM25	0.7838	0.8021	0.5129
Semantic search	0.7529	0.6592	0.5997
Hybrid search	0.9649	0.8439	0.7904

Bảng 2: Kết so sánh các phương pháp IR được thực hiện trong quá trình Retrieval sử dụng mô hình Ha_long embedding.

Model	Hit Rate@10	MRR@10	MAP@10
Phobert_base	0.9649	0.8439	0.7904
Halong_embedding	0.9321	0.8627	0.7525

Bảng 3: Kết quả so sánh hai mô hình Halong_embedding và Vietnamese PhoBERT base sử dụng phương pháp Hybrid.

4 Đánh giá

4.1 Metric

Để đánh giá hiệu quả của việc truy vấn tài liệu pháp luật, chúng tôi sử dụng ba chỉ số phổ biến trong lĩnh vực Information Retrieval là:

- **Hit Rate@k**: Tỷ lệ các truy vấn mà trong top-k kết quả có ít nhất một tài liệu liên quan.
- **Mean Reciprocal Rank (MRR@k)**: Đánh giá mức độ ưu tiên tài liệu liên quan trong thứ hạng kết quả.
- **Mean Average Precision (MAP@k)**: Trung bình độ chính xác ở từng vị trí có tài liệu đúng trong top-k.

4.2 Kết quả

Về phương pháp IR

Trong thí nghiệm này, ba phương pháp truy xuất - BM25, Semantic Search và Hybrid Search - được đánh giá dựa trên mô hình Ha_long embedding nhằm so sánh khả năng truy xuất xếp hạng tài liệu liên quan.

Theo **Bảng 2**, trước hết, so sánh kết quả giữa hai phương pháp BM25 và Semantic Search, BM25 ghi nhận MRR@10 cao hơn (0,8021 so với 0,6592), cho thấy khả năng ưu tiên đưa tài liệu liên quan nhất lên vị trí hàng đầu; bên cạnh đó, Semantic Search lại vượt trội về MAP@10 (0,5997 so với 0,5129), phản ánh năng lực hiểu sâu ngữ nghĩa khi phân phối các kết quả phù hợp đều trong top-k dù không nhất thiết ở top-1. Đáng chú ý nhất, Hybrid Search tổng hợp được lợi thế của cả hai phương pháp trên, với Hit Rate@10 lên tới 0,9649 và đồng thời đạt 0,8439 với MRR@10 cùng MAP@10 là 0,7904 cao nhất. Điều này khẳng định hiệu quả tổng thể vượt trội của Hybrid Search trong việc vừa nhanh chóng truy xuất tài liệu quan trọng, vừa duy trì khả năng hiểu ngữ nghĩa sâu.

Như vậy, Hybrid Search được xác định là phương pháp thích hợp nhất cho giai đoạn Retrieval trong hệ thống RAG.

Về mô hình embedding

Để đánh giá hiệu suất của hai mô hình embedding, chúng tôi áp dụng phương pháp Hybrid Search đã được chứng minh là tối ưu trong giai đoạn Retrieval cho cả Halong_embedding và PhoBERT_base. Phương pháp này cho phép so sánh trực tiếp khả năng kết hợp giữa truy xuất tần suất và hiểu ngữ nghĩa sâu của mỗi embedding.

Kết quả ở **Bảng 3** cho thấy PhoBERT_base đạt Hit Rate@10 cao hơn (0,9649 so với 0,9321 của Halong), cho thấy khả năng thu hồi đủ tài liệu liên quan trong top-10 vượt trội. Điều này minh chứng rằng PhoBERT_base, được huấn luyện chuyên sâu trên dữ liệu tiếng Việt, có lợi thế trong việc nhận diện rộng các tài liệu phù hợp. Đồng thời, chỉ số MAP@10 của PhoBERT_base (0,7904 so với 0,7525) cũng cao hơn, cho thấy độ chính xác trung bình trên toàn bộ nhóm kết quả của nó ổn định hơn.

Ngược lại, Halong_embedding ghi nhận MRR@10 cao hơn (0,8627 so với 0,8439 của PhoBERT_base), cho thấy mô hình này có khả năng đưa tài liệu liên quan nhất lên vị trí đầu tiên hiệu quả hơn. Điểm mạnh này phản ánh ưu thế của Halong_embedding trong việc tối ưu hóa thứ tự xếp hạng tài liệu quan trọng nhất, mặc dù tổng thể nó thu hồi ít tài liệu hơn và có độ chính xác trung bình kém hơn một chút so với PhoBERT_base.

Tóm lại, với kết quả so sánh thu được,

PhoBERT_base thể hiện ưu thế về khả năng thu hồi và độ chính xác trung bình trong top-k, trong khi Halong_embedding nổi bật ở khả năng xếp hạng tài liệu quan trọng nhất lên vị trí đầu. Những khác biệt này gợi ý rằng lựa chọn embedding có thể dựa vào mục tiêu cụ thể của hệ thống: nếu ưu tiên thu hồi đầy đủ và ổn định, PhoBERT_base là lựa chọn phù hợp; nếu cần tối ưu vị trí top-1, Halong_embedding đem lại lợi thế rõ rệt.

5 Nhận xét và Hướng phát triển

5.1 Nhận xét

Hệ thống RAG đã chứng minh được khả năng sinh các phản hồi chính xác và tuân thủ pháp lý, mang lại kết quả gần với các ứng dụng hiện nay. Bộ dữ liệu nền tảng, được xây dựng từ tập hợp các văn bản luật có giá trị thực tiễn cao, đã cung cấp cơ sở thông tin vững chắc cho quá trình truy xuất và sinh ngôn ngữ; tuy nhiên, tính toàn diện của dữ liệu vẫn còn hạn chế khi chưa bao quát đầy đủ các ngoại lệ và tình huống đặc thù trong thực tế.

Chất lượng đầu ra của hệ thống phụ thuộc chặt chẽ vào giai đoạn Retrieval, đặc biệt đối với những câu hỏi phức tạp hoặc kịch bản dài dòng, nơi việc truy xuất đầy đủ ngữ cảnh và tài liệu liên quan là then chốt. Bên cạnh đó, dù mô hình Generation đã thể hiện tốt về độ chính xác và tính hợp pháp nhưng cần có thêm sự tinh chỉnh để gia tăng khả năng diễn giải logic mạch lạc và trích dẫn điều khoản một cách phù hợp. Những cải tiến này sẽ giúp nâng cao hơn nữa độ tin cậy và tính tham khảo của hệ thống, hướng đến mục tiêu trở thành công cụ hỏi - đáp pháp lý toàn diện và hiệu quả.

5.2 Hướng phát triển

Trong giai đoạn tiếp theo, chúng tôi dự định mở rộng và hoàn thiện bộ dữ liệu pháp luật bằng cách liên tục cập nhật những văn bản mới ban hành, loại bỏ hoặc điều chỉnh những quy định hết hiệu lực và bổ sung các quy định địa phương cùng các tình huống xử phạt thực tế. Việc này không chỉ giúp tăng tính thời sự và toàn diện cho hệ thống mà còn làm cơ sở vững chắc cho quá trình truy xuất thông tin, giảm thiểu việc bỏ sót các ngoại lệ hay các tình huống đặc thù phát sinh trong thực tiễn.

Song song với đó, chúng tôi sẽ tích hợp dữ liệu từ các diễn đàn pháp lý, phản hồi của người dân và hồ sơ xử phạt vi phạm để xây dựng một tập dữ liệu tình huống thực tế đa dạng. Nguồn dữ liệu này sẽ hỗ trợ đáng kể cho khả năng tổng quát hóa (generalization) của mô hình, giúp hệ thống không chỉ đưa ra được các đáp án đúng theo luật lý thuyết, mà còn ứng dụng linh hoạt trong những kịch bản phức tạp, đòi hỏi xử lý ngữ cảnh phong phú.

Để nâng cao trải nghiệm người dùng, giao diện sẽ được thiết kế đa phương thức, hỗ trợ tương tác bằng văn bản, thoại và thậm chí xử lý hình ảnh (chẳng hạn bảng hiệu hay biển số) thông qua tích hợp OCR, từ đó tạo nên một công cụ tư vấn pháp lý toàn diện, linh hoạt và dễ tiếp cận.

6 Tài liệu tham khảo

1. Luật Giao thông Đường bộ Việt Nam, 2008.
2. K. Guu, M. Chang, E. Z. Zhang, and K. Toutanova, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” NeurIPS, 2020.
3. OpenAI API Documentation. https://platform.openai.com/docs
4. Hugging Face Transformers Library. https://huggingface.co/transformers/
5. L. Van Phuc et al., "sup-SimCSE-VietNamese-phobert-base," Hugging Face, 2023.
6. Manning et al., "Introduction to Information Retrieval," Cambridge University Press, 2008.