Developing a Legal Chatbot System

Nguyễn Anh Duy 21522000

Đỗ Quốc Duy 21520768

Tô Duy Nguyên Hoàng 21522100

GVHD: Nguyễn Vinh Tiệp

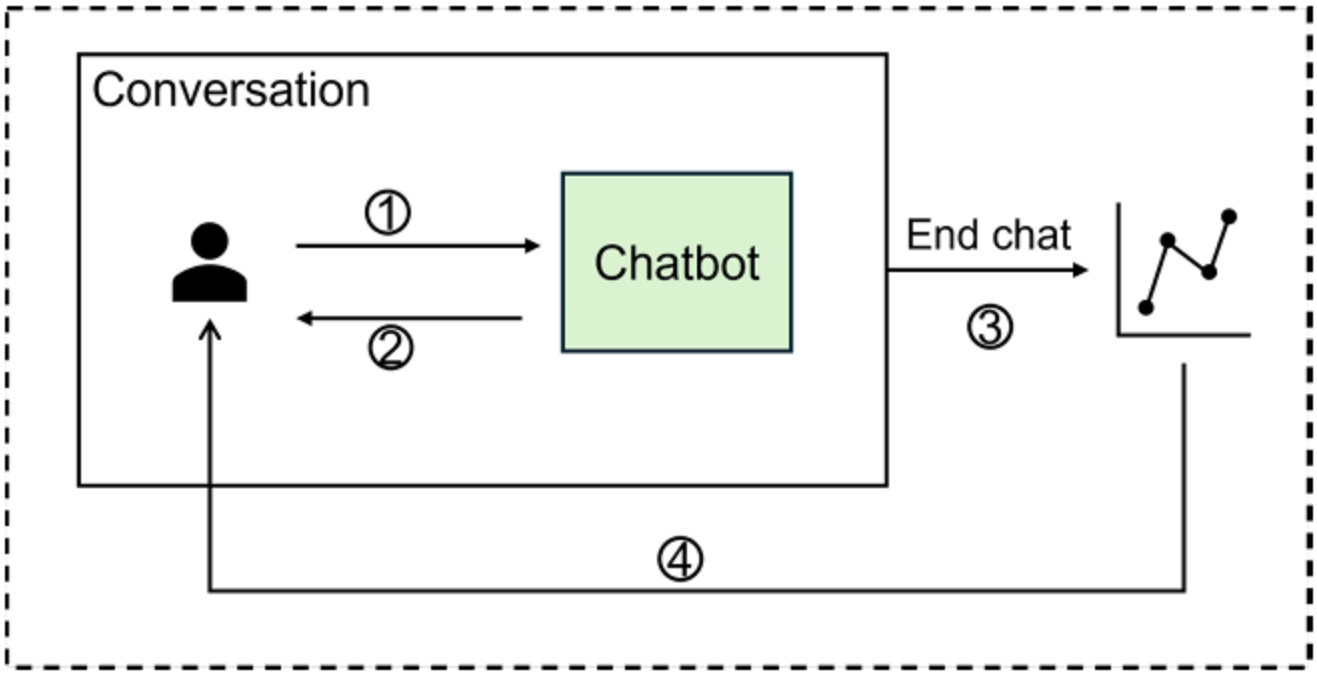
Abstract

Hệ thống tư vấn pháp luật tự động là một giải pháp tiềm năng nhằm giúp người dùng tiếp cận thông tin pháp lý một cách nhanh chóng, chính xác và hiệu quả. Trong nghiên cứu này, chúng tôi giới thiệu Hệ thống Chatbot Pháp Lý Tích Hợp AI, một nền tảng thông minh kết hợp các mô hình ngôn ngữ lớn (LLM) như GPT và PhoBERT với các kỹ thuật mã hóa tiên tiến (Bi-Encoder, Cross-Encoder) và hệ thống tìm kiếm tương đồng FAISS. Hệ thống được thiết kế để hỗ trợ người dùng giải đáp các câu hỏi pháp luật phổ biến, xử lý linh hoạt các tình huống pháp lý khác nhau mà không cần tái huấn luyện. Kết quả thực nghiệm cho thấy hệ thống mang lại độ chính xác cao, phản hồi nhanh chóng, và khả năng mở rộng tốt, mở ra tiềm năng ứng dụng rộng rãi trong các lĩnh vực như giáo dục, dịch vụ công, và tư vấn pháp luật chuyên nghiệp.

1 Introduction

Trong bối cảnh hiện đại, nhu cầu tiếp cận thông tin pháp luật ngày càng trở nên cấp thiết. Với hệ thống pháp luật phức tạp và sự thay đổi liên tục trong các quy định, việc hiểu và áp dụng đúng pháp luật không chỉ là thách thức đối với công chúng mà còn đối với cả các chuyên gia pháp lý. Các hệ thống chatbot pháp luật hiện nay, dù đã có những tiến bộ nhất định, vẫn gặp hạn chế khi chỉ dựa vào dữ liệu được huấn luyện trước. Điều này làm giảm khả năng xử lý các tình huống mới hoặc phức tạp, đòi hỏi sự phát triển của các giải pháp linh hoạt và hiệu quả hơn.

Để hiện thực hóa mục tiêu, dự án sử dụng các công nghệ tiên tiến như mô hình ngôn ngữ lớn (LLM) và phương pháp Retrieval-Augmented Generation (RAG). Hệ thống tích hợp công cụ FAISS, cùng với các kỹ thuật mã hóa Bi-Encoder và Cross-Encoder, giúp tối ưu hóa quá trình truy xuất thông tin và đảm bảo câu trả lời phù hợp với ngữ cảnh. Bằng cách kết hợp các công nghệ hiện đại, hệ thống này không chỉ hỗ trợ hiệu quả cho người dùng phổ thông mà còn đóng góp tích cực vào công việc của các chuyên gia pháp lý, đồng thời giảm thiểu thời gian và chi phí tìm kiếm thông tin.



Hình 1: Hệ thống thực hiện trò chuyện với người dùng (1)(2), phân tích và trả lời (3), theo dõi và lưu dữ liệu. (4)

Bắt đầu khi người dùng gửi câu hỏi, hệ thống sẽ tiếp nhận truy vấn đầu vào và xử lý thông qua mô hình ngôn ngữ lớn (LLM). Chatbot sau đó phản hồi bằng cách sử dụng dữ liệu đã truy xuất để tạo ra các câu trả lời chính xác và phù hợp với ngữ cảnh. Sau khi kết thúc phiên trò chuyện, toàn bộ dữ liệu liên quan được lưu trữ nhằm phục vụ cho việc phân tích và cải thiện hiệu suất của hệ thống trong tương lai. Nhờ đó, hệ thống có thể học hỏi từ dữ liệu đã lưu và liên tục cải thiện độ chính xác cũng như tính linh hoạt, đảm bảo chất lượng cao hơn trong các tương tác tiếp theo.

2 Related Work

.Nhiều nghiên cứu trên thế giới đã chỉ ra tiềm năng lớn của các mô hình ngôn ngữ lớn (LLM) và các phương pháp xử lý ngôn ngữ tự nhiên (NLP) trong việc phát triển chatbot pháp luật. Các mô hình như GPT-3, BERT, và các phiên bản cải tiến của chúng đã được ứng dụng rộng rãi trong nhiều lĩnh vực, từ trả lời câu hỏi đến hỗ trợ ra quyết định. Những nghiên cứu này đã chứng minh khả năng xử lý văn bản mạnh mẽ của các LLM, đặc biệt trong việc tạo phản hồi ngữ cảnh tự nhiên và chính xác.

Tại Việt Nam, PhoBERT, một mô hình Bi-Encoder tối ưu hóa cho tiếng Việt, đã trở thành công cụ quan trọng trong các ứng dụng NLP. PhoBERT hỗ trợ mã hóa dữ liệu tiếng Việt hiệu quả, cho phép xây dựng các hệ thống thông minh với độ chính xác cao. Cụ thể, PhoBERT đã được sử dụng trong các hệ thống truy vấn thông tin và phân loại văn bản, tạo nền tảng cho các ứng dụng đòi hỏi xử lý dữ liệu tiếng Việt phức tạp.

Một hướng tiếp cận nổi bật khác là phương pháp Retrieval-Augmented Generation (RAG), kết hợp giữa khả năng xử lý ngôn ngữ tự nhiên của LLM và truy xuất thông tin từ cơ sở dữ liệu bên ngoài. Phương pháp này không chỉ giảm thiểu hiện tượng "ảo giác thông tin" (hallucination) – khi hệ thống đưa ra các câu trả lời không chính xác – mà còn giúp cung cấp phản hồi dựa trên dữ liệu thực tế và đáng tin cậy.

Ngoài ra, các công cụ lưu trữ và truy xuất thông tin, chẳng hạn như FAISS (Facebook AI Similarity Search), đã được áp dụng để tăng cường hiệu quả xử lý và tìm kiếm dữ liệu trong các hệ thống lớn. FAISS cho phép lưu trữ embedding và tìm kiếm dựa trên khoảng cách vector, giúp cải thiện tốc độ và hiệu suất của hệ thống khi xử lý các truy vấn phức tạp.

Dựa trên các thành tựu này, dự án của chúng tôi không chỉ kế thừa các công nghệ hiện đại mà còn tối ưu hóa chúng để phù hợp với đặc điểm ngôn ngữ và nhu cầu pháp luật tại Việt Nam. Hệ thống chatbot được kỳ vọng sẽ trở thành công cụ hỗ trợ hiệu quả, vừa phục vụ cộng đồng người dùng phổ thông, vừa hỗ trợ đắc lực cho các chuyên gia pháp lý.

2.1 Dataset Description

Dữ liệu sử dụng trong dự án được cung cấp bởi cuộc thi SoICT Hackathon 2024, do Viện Nghiên cứu BKAI thuộc Đại học Bách Khoa tổ chức. Tập dữ liệu này được thiết kế để hỗ trợ các bài toán xử lý ngôn ngữ tự nhiên trong lĩnh vực pháp luật, bao gồm hai tập chính:

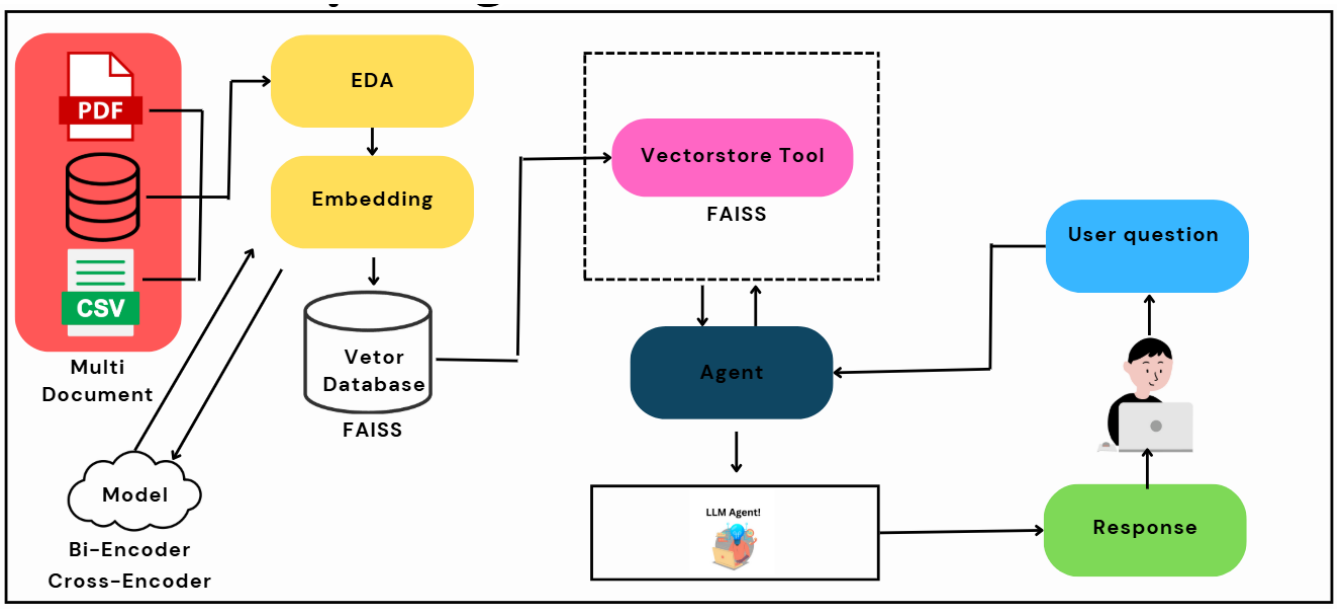
train.csv: Bao gồm hơn 260.000 mẫu dữ liệu, với các cặp câu hỏi và câu trả lời pháp luật. Đây là nguồn dữ liệu chính được sử dụng để huấn luyện mô hình chatbot, đảm bảo hệ thống có khả năng xử lý và trả lời các câu hỏi từ cơ bản đến phức tạp.

corpus.csv: Gồm hơn 110.000 mẫu dữ liệu, chứa tri thức liên quan đến các câu trả lời trong tập train.csv. Tập dữ liệu này đóng vai trò như một cơ sở kiến thức bổ sung, giúp hệ thống nâng cao độ chính xác và cung cấp các câu trả lời phù hợp với ngữ cảnh.

Các tập dữ liệu này không chỉ có kích thước lớn mà còn được cấu trúc chặt chẽ, đảm bảo độ tin cậy và tính toàn diện trong việc huấn luyện và đánh giá mô hình chatbot. Với sự hỗ trợ từ tập dữ liệu này, hệ thống có thể phát triển khả năng trả lời hiệu quả các câu hỏi pháp lý đa dạng, đồng thời mở rộng phạm vi ứng dụng trong thực tế.

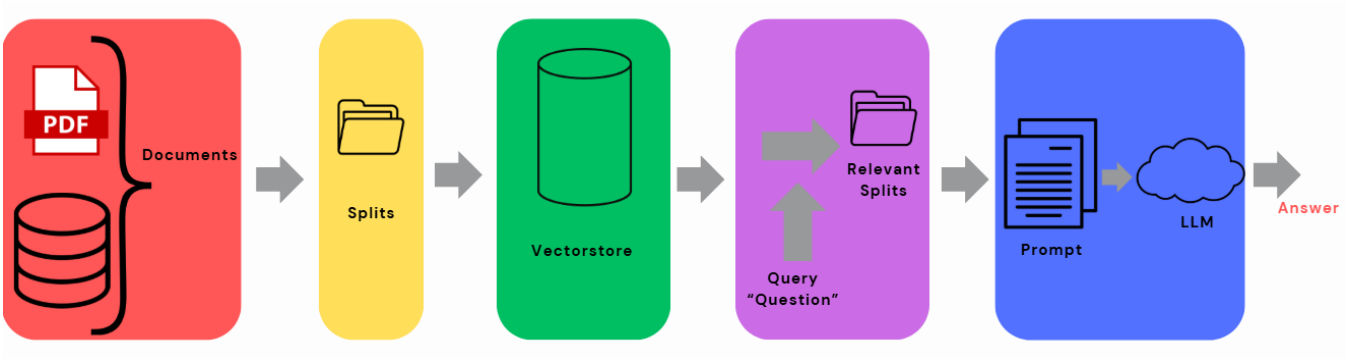
3 The proposed approach

Để xây dựng hệ thống chatbot pháp luật thông minh, chúng tôi đề xuất một kiến trúc tích hợp dựa trên các công nghệ học sâu tiên tiến và các phương pháp truy xuất thông tin hiện đại. Hệ thống được thiết kế để kết hợp sức mạnh của các mô hình ngôn ngữ lớn (LLM) và phương pháp Retrieval-Augmented Generation (RAG), cho phép truy vấn dữ liệu nhanh chóng và chính xác. Bên cạnh đó, chúng tôi sử dụng các công cụ tối ưu hóa như FAISS và các kỹ thuật mã hóa Bi-Encoder và Cross-Encoder để cải thiện hiệu suất và độ chính xác của hệ thống.



Hình 2: Mô hình hệ thông tổng quát của chat bot tư vấn luật tương tác với người dùng.

3.1 Retrieval-Augmented Generation (RAG)



Hình 3: Hệ thống tổng quát của mô hình RAG.

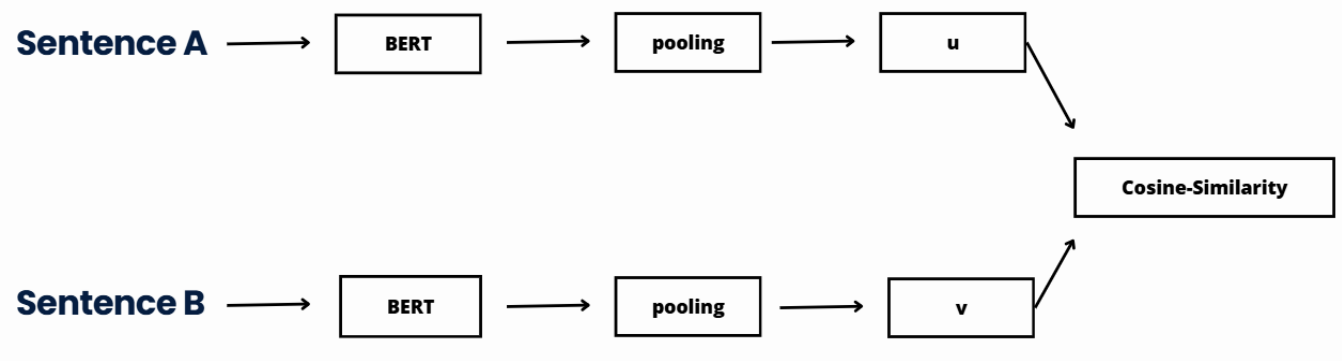
Trong hệ thống chatbot pháp luật, phương pháp Retrieval-Augmented Generation (RAG) được sử dụng nhằm kết hợp giữa sức mạnh của các mô hình ngôn ngữ lớn (LLM) và cơ chế truy xuất thông tin từ cơ sở dữ liệu. Phương pháp này không chỉ giúp hệ thống cung cấp các câu trả lời có ngữ cảnh chính xác mà còn giảm thiểu hiện tượng "ảo giác thông tin" thường gặp ở các mô hình LLM thuần túy.

Quy trình hoạt động của RAG được thể hiện trong Hình 3.1 và bao gồm các bước chính như sau: Đầu tiên, hệ thống xử lý tập dữ liệu gốc (các tài liệu PDF hoặc văn bản) và chia nhỏ chúng thành các đoạn (splits) để dễ dàng quản lý. Các đoạn này sau đó được mã hóa thành các vector bằng kỹ thuật Bi-Encoder, rồi lưu trữ trong cơ sở dữ liệu vector (vectorstore).

Khi người dùng gửi truy vấn, hệ thống sẽ mã hóa câu hỏi thành vector và so sánh nó với các vector đã lưu để tìm ra những đoạn văn bản liên quan nhất. Các đoạn văn bản này sau đó được tổng hợp vào một prompt và gửi đến mô hình LLM để tạo ra câu trả lời cuối cùng.

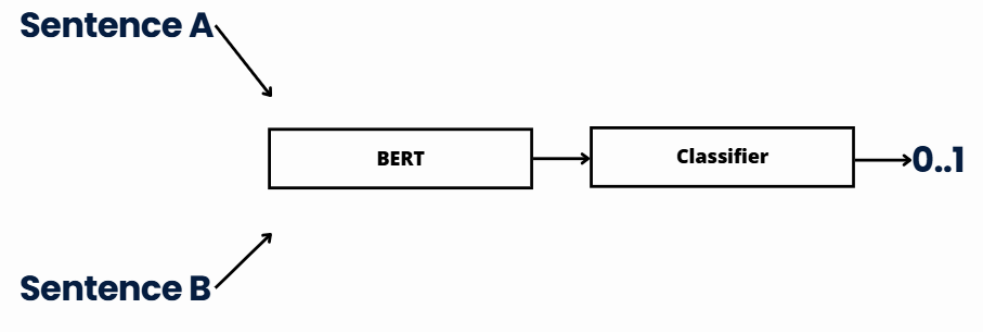
Điểm nổi bật của phương pháp RAG là khả năng tích hợp dữ liệu bên ngoài vào quá trình xử lý của mô hình LLM. Điều này không chỉ giúp tăng độ chính xác của câu trả lời mà còn cải thiện khả năng xử lý các tình huống mới hoặc phức tạp mà hệ thống chưa được huấn luyện trước. Nhờ đó, phương pháp này mang lại hiệu quả cao, đặc biệt trong các ứng dụng pháp luật, nơi mà độ chính xác và ngữ cảnh đóng vai trò quan trọng.

3.2 Bi Encoder/Cross Encoder



Hình 4: Hệ thống tổng quát của mô hình Bi-Encoder.

Trong phương pháp này, hai câu đầu vào, **Sentence A** và **Sentence B**, được mã hóa độc lập bằng mô hình BERT. Quá trình này tạo ra hai vector đặc trưng, gọi là **u** và **v**, đại diện cho ngữ nghĩa của từng câu. Sau khi mã hóa, hai vector này được so sánh với nhau bằng phép đo **Cosine Similarity** để đánh giá mức độ tương đồng. Bi-Encoder đặc biệt phù hợp cho các tác vụ yêu cầu xử lý lượng lớn dữ liệu, nhờ khả năng mã hóa từng câu một cách riêng biệt và giảm đáng kể thời gian tính toán khi tìm kiếm thông tin trong cơ sở dữ liệu.



Hình 5: Hệ thống tổng quát của mô hình Cross-Encoder.

Sentence A và Sentence B, được kết hợp với nhau và đưa vào cùng một mô hình BERT. Kết quả của quá trình này là một đầu ra duy nhất, biểu thị dưới dạng xác suất trong khoảng từ 0 đến 1, cho biết mức độ liên quan giữa hai câu. Cross-Encoder có lợi thế trong việc cung cấp kết quả chính xác hơn so với Bi-

Encoder, nhờ khả năng xử lý và so sánh các câu trong cùng một ngữ cảnh. Tuy nhiên, mô hình này yêu cầu thời gian tính toán lớn hơn, vì mỗi cặp câu phải được xử lý cùng lúc.

Phương pháp kết hợp: Để tận dụng ưu điểm của cả hai mô hình, chúng tôi kết hợp Bi-Encoder và Cross-Encoder trong hệ thống chatbot pháp luật. Cụ thể, Bi-Encoder được sử dụng để nhanh chóng lọc ra các câu trả lời tiềm năng từ cơ sở dữ liệu lớn, nhờ khả năng tìm kiếm hiệu quả với Cosine Similarity. Sau đó, các câu trả lời tiềm năng này được xếp hạng lại bằng Cross-Encoder, đảm bảo độ chính xác cao hơn trong việc đánh giá mức độ liên quan đến truy vấn của người dùng.

3.3 Vector Store

Hệ thống chatbot pháp luật sử dụng Vector Store để lưu trữ và quản lý dữ liệu đã mã hóa dưới dạng vector, hỗ trợ quá trình truy xuất thông tin một cách nhanh chóng và chính xác. Trong dự án này, chúng tôi sử dụng FAISS (Facebook AI Similarity Search) – một công cụ mạnh mẽ cho việc tìm kiếm và so khớp vector.

Quy trình hoạt động của Vector Store bắt đầu từ việc mã hóa các tài liệu gốc (PDF, CSV) thông qua kỹ thuật embedding, biến đổi nội dung văn bản thành vector số học. Các vector này được lưu trữ trong cơ sở dữ liệu vector, nơi FAISS thực hiện tìm kiếm dựa trên các phép đo tương đồng như khoảng cách cosine hoặc Euclidean.

Khi người dùng gửi truy vấn, hệ thống sẽ mã hóa truy vấn thành vector và tìm kiếm các vector liên quan trong cơ sở dữ liệu. Nhờ FAISS, hệ thống có thể truy xuất thông tin từ hàng triệu vector một cách nhanh chóng, đảm bảo tính hiệu quả và độ chính xác cao trong các tác vụ pháp luật phức tạp.

3.3 LLM Agent

LLM Agent đóng vai trò là trung tâm điều phối hoạt động của hệ thống chatbot pháp luật. Đây là thành phần kết hợp giữa khả năng xử lý ngôn ngữ tự nhiên (NLP) của mô hình LLM và các công cụ hỗ trợ truy xuất thông tin như Vector Store.

Quá trình vận hành của LLM Agent diễn ra như sau:

1. Người dùng gửi câu hỏi đến hệ thống.
2. LLM Agent nhận truy vấn và xác định các bước cần thực hiện, bao gồm mã hóa câu hỏi và truy xuất thông tin liên quan từ Vector Store thông qua FAISS.
3. Các đoạn văn bản liên quan được tổng hợp và tích hợp vào **prompt** để gửi đến mô hình ngôn ngữ lớn (LLM).
4. LLM Agent sử dụng khả năng xử lý ngữ cảnh của LLM để tạo ra câu trả lời phù hợp, sau đó gửi phản hồi cho người dùng.

4. Phương pháp

4.1. Mô hình Bi-Encoder

4.1.1. Mô hình cơ sở và kiến trúc

Nhóm chúng tôi sử dụng kiến trúc bi-encoder để mã hóa các câu hỏi pháp lý và các tài liệu tương ứng thành các biểu diễn vector dày đặc (dense vector representations). Mô hình bi-encoder bao gồm hai encoder riêng biệt:

- **Question Encoder:** Mã hóa các truy vấn từ người dùng thành các embedding có kích thước cố định.
- **Document Encoder**: Mã hóa các tài liệu pháp lý thành các embedding có kích thước cố định.

Cả hai encoder được khởi tạo bằng mô hình bi-encoder tiếng Việt (*bkai-foundation-models/vietnamese-bi-encoder*), một mô hình transformer đã được huấn luyện trước (*pre-trained*) và tinh chỉnh (*fine-tuned*) cho văn bản tiếng Việt. Mô hình này dựa trên kiến trúc PhoBERT, vốn được huấn luyện trên một tập dữ liệu tiếng Việt lớn. Việc chọn mô hình này xuất phát từ thực nghiệm khi chúng tôi đã chạy thử và đánh giá, kết quả cho thấy mô hình này có khả năng vượt trội hơn so với base model là XLM-RoBERTa (sẽ được trình bày trong phần V) trong việc hiểu ngữ nghĩa và thực hiện các tác vụ truy xuất thông tin đối với văn bản tiếng Việt .

4.1.2. Phương pháp huấn luyện

Mô hình bi-encoder được huấn luyện với hàm mục tiêu là *contrastive learning*. giúp mô hình tối đa hóa sự tương đồng giữa một câu hỏi và tài liệu đúng tương ứng, đồng thời giảm thiểu sự tương đồng với các tài liệu không đúng. Quá trình huấn luyện bao gồm các bước sau:

- **Chuẩn bị dữ liệu**:
  - Bộ dữ liệu gốc của BKAI bao gồm các cặp câu hỏi pháp lý và tài liệu tương ứng. Mỗi cặp câu hỏi-tài liệu được coi là một ví dụ dương (*positive example*).
  - Để có thể train mô hình bi-encoder một cách hiệu quả. Chúng tôi đã xử lí qua bộ dữ liệu gốc bằng cách thực hiện phân đoạn từ (*word segmentation*). Khác với base model là XLM-RoBERTa, mô hình PhoBERTcần dữ liệu được phân đoạn để hoạt động hiệu quả.
  - Các ví dụ âm khó (*hard negatives*) được khai thác trong quá trình huấn luyện nhằm cải thiện khả năng phân biệt của mô hình. Hard negatives là các tài liệu sai nhưng có ngữ nghĩa tương tự với câu hỏi, khiến chúng trở thành thách thức cho mô hình phân biệt.
- **Hàm loss**:
  - Hàm loss dựa trên cross-entropy được tính với giá trị đầu vào là similarity score giữa các embedding của câu hỏi và tài liệu và nhãn thực. Các điểm tương đồng này được điều chỉnh bởi tham số nhiệt độ (*temperature = 0.05*).
  - Hàm mất mát kết hợp cả các mẫu trong batch và hard negatives (các tài liệu không đúng nhưng có ngữ nghĩa tương tự được khai thác trong quá trình huấn luyện).
- **Chi tiết huấn luyện**:
  - Mô hình được huấn luyện trong 2 epoch với kích thước batch là 16.

    - Tốc độ học (*learning rate*) được thiết lập là 1e−5, sử dụng optimizer AdamW với weight decay là 0.01.
    - Sử dụng scheduler điều chỉnh tốc độ học cosine (*cosine annealing learning rate scheduler*) để điều chỉnh động tốc độ học trong quá trình huấn luyện.
- **Kĩ thuật gradual unfreezing**:
    - Để ổn định quá trình huấn luyện và tránh hiện tượng "quên thảm họa" (*catastrophic forgetting*), chúng tôi áp dụng kĩ thuật gỡ đóng băng tuần tự(*gradual unfreezing*). Ban đầu, tất cả các lớp của encoder bị khóa (trừ lớp pooling). Khi quá trình huấn luyện tiến triển, các lớp được mở khóa dần:
        - Sau 25% số bước huấn luyện: Mở khóa 2 lớp trên cùng.
        - Sau 50% số bước huấn luyện: Mở khóa 4 lớp trên cùng.
        - Sau 75% số bước huấn luyện: Mở khóa 6 lớp trên cùng.
        - Sau 90% số bước huấn luyện: Mở khóa 8 lớp trên cùng.
    - Phương pháp này cho phép mô hình thích nghi với dữ liệu đặc thù của tác vụ, đồng thời bảo toàn kiến thức đã huấn luyện trước.
- **Khai thác hard negatives**:
    - Các hard negatives được khai thác định kỳ trong quá trình huấn luyện bằng cách tính toán điểm tương đồng giữa các embedding của câu hỏi và tài liệu. Với mỗi câu hỏi, top-k tài liệu không đúng nhưng có độ tương đồng cao nhất được chọn làm hard negatives. Các hard negative này sẽ được thêm vào quá trình training nhằm làm tang cường tuần suất xuất hiện của chúng.
    - Quá trình này được thực hiện hai lần mỗi epoch để đảm bảo mô hình tiếp xúc với các ví dụ khó trong suốt quá trình huấn luyện.

4.2. Mô hình Cross-Encoder

4.2.1. Mô hình cơ sở và kiến trúc

Mô hình cross-encoder được thiết kế để xếp hạng lại (*rerank*) các tài liệu ứng viên (*candidate document*) được truy xuất bởi mô hình bi-encoder. Khác với bi-encoder, mô hình cross-encoder xử lý đồng thời cặp câu hỏi-tài liệu để tính điểm mức độ liên quan (*relevance score*). Điều này cho phép mô hình nắm bắt các tương tác chi tiết giữa câu hỏi và tài liệu.

Mô hình cross-encoder được khởi tạo bằng mô hình bi-encoder tiếng Việt giống như trên (*bkai-foundation-models/vietnamese-bi-encoder*). Việc lựa chọn như vậy là để đảm bảo tính nhất quán với bi-encoder và bên cạnh đó mô hình này cũng đã chứng tỏ bằng thực nghiệm là nó tốt hơn trong việc sử lí dữ liệu tiếng Việt so với base model.

4.2.2. Phương pháp huấn luyện

Mô hình cross-encoder được huấn luyện với mục tiêu phân lớp nhị phân (*binary classification*), trong đó mô hình học cách dự đoán một cặp câu hỏi-tài liệu có liên quan (nhãn = 1) hay không (nhãn = 0). Quá trình huấn luyện bao gồm các bước sau:

- **Chuẩn bị dữ liệu**:
    - Bộ dữ liệu huấn luyện bao gồm các cặp câu hỏi-tài liệu, được gán nhãn là "liên quan" hoặc "không liên quan".
    - Bộ dữ liệu này không nằm trong bộ dữ liệu gốc mà được sinh ra bằng các hard negative khi sử dụng mô hình bi-encoder lên tập dữ liệu huấn luyện nhằm cải thiện khả năng phân biệt giữa các tài liệu không chính xác nhưng có độ tương đồng cao. Cụ thể với mỗi câu truy vấn trong tập huấn luyện ta sử dụng bi-encoder để lấy top 10 kết quả trả về. Sau đó ta lấp top 3 tài liệu sai có xếp hạng cao nhất cùng với tài liệu đúng để build tập dữ liệu này.
- **Hàm mất mát**:
    - Mô hình được huấn luyện bằng hàm mất mát binary cross-entropy, đo lường sự khác biệt giữa điểm mức độ liên quan dự đoán và nhãn thực tế.
- **Chi tiết huấn luyện**:
    - Mô hình được huấn luyện trong 1 epoch với kích thước batch là 16.
    - Tỷ lệ *warmup* là 0.02 được áp dụng trong các bước huấn luyện ban đầu để tăng dần learning rate, giúp ổn định quá trình huấn luyện.
    - Optimizer AdamW được sử dụng với các siêu tham số mặc định.
- **Đánh giá trong quá trình huấn luyện**:
    - Hiệu suất của mô hình được đánh giá trên tập validation sau mỗi 2000 bước, sử dụng độ chính xác nhị phân (*binary accuracy*) làm chỉ số chính.
    - Các điểm dừng (*checkpoints*) được lưu lại trong quá trình huấn luyện để cho phép lựa chọn mô hình dựa trên hiệu suất kiểm định.

4.3. Pipeline truy vấn

Pipeline truy xuất kết hợp sức mạnh của mô hình bi-encoder và cross-encoder để xây dựng một hệ thống truy xuất tài liệu vừa hiệu quả vừa chính xác. Qúa trình truy vấn diễn ra qua hai bước chính:

- **Truy xuất bằng bi-encoder**: Nhanh chóng truy xuất một tập hợp các tài liệu ứng viên thông qua tìm kiếm dựa trên độ tương đồng vector. Việc này được thực hiện bằng cách sử dụng nhánh question encoder để mã hoá câu truy vấn thành vector, sau đó so sánh với các vector nằm trong FAISS vector database và trả về top 50 tài liệu liên quan nhất.
- **Xếp hạng lại bằng cross-encoder**: Xếp hạng lại các tài liệu ứng viên bằng cách tính toán điểm mức độ liên quan chi tiết (*fine-grained relevance score*) cho từng cặp truy vấn-tài liệu. Mô hình này sẽ nhận đầu vào là câu truy vấn của người dùng và lần lượt từng tài liệu trong 50 ứng cử viên trả về từ bước bi-encoder. Sau cùng lấy ra 10 tài liệu có điểm tương đồng cao nhất trong 50 tài liệu ứng cử viên. Đây sẽ là các tài liệu được cung cấp cho LLM để trả lời người dùng.

Hướng tiếp cận trên vừa đảm bảo tốc độ nhanh (nhờ bi-encoder) vừa đạt độ chính xác cao (nhờ cross-encoder).

4.4. LLM Agent.

Vì mục tiêu bài toán là xây dựng một hệ thống chatbot để tư vấn luật cho người dùng. Chúng tôi đã sử dụng mô hình GPT-4o mini và build theo hướng agent. Nguyên nhân của sự lựa chọn này là để hệ thống không cần phải thực hiện các truy vấn vô nghĩa khi câu hỏi của người dùng không liên quan đến văn bản

luật, với hướng tiếp cận này bản thân agent sẽ tự đưa ra lựa chọn có nên truy vấn cơ sở dữ liệu không dựa trên câu hỏi của người dùng. Việc này giúp tiết kiệm token đầu vào và để hệ thống có thể dễ dàng mở rộng hơn khi ta có thêm các module mới.

Bản thân pipeline truy vấn đã được trình bày bên trên sẽ được đóng gói thành một công cụ (*tool*) và được cung cấp cho LLM agent để sử dụng.

5. Đánh giá mô hình

5.1. Phương thức đánh giá

Hệ thống được đánh giá trên một tập kiểm tra gồm 11,946 mẫu để đo lường hiệu suất truy xuất. Chỉ số chính được quan tâm là **Recall@10**, vì 10 tài liệu được truy xuất hàng đầu sẽ được cung cấp cho mô hình ngôn ngữ lớn (LLM) để tạo phản hồi cuối cùng. Ngoài ra, các metric sau cũng được tính để để đánh giá toàn diện:

- MAP (Mean Average Precision): Đo lường độ chính xác của các tài liệu được truy xuất trên tất cả các tài liệu liên quan.
- MRR (Mean Reciprocal Rank): Đánh giá vị trí của tài liệu liên quan đầu tiên.
- NDCG (Normalized Discounted Cumulative Gain): Đánh giá chất lượng xếp hạng bằng cách xem xét vị trí của các tài liệu liên quan.

Quá trình đánh giá được thực hiện trong hai kịch bản:

1. **Không xếp hạng lại**: Chỉ sử dụng bi-encoder để truy xuất.
2. **Có xếp hạng lại**: Bi-encoder truy xuất tài liệu ứng viên, sau đó cross-encoder xếp hạng lại chúng.

Hai biến thể của bi-encoder được đánh giá:

- **XLM-RoBERTa**: Một mô hình đa ngôn ngữ được pretrained trên một tập hợp dữ liệu lớn, bao gồm tiếng Việt.
- **PhoBERT**: Một mô hình đơn ngôn ngữ được huấn luyện chuyên biệt trên văn bản tiếng Việt.

5.2. Kết quả

5.2.1. Không xếp hạng lại

Mô hình	MAP	MRR	NDCG	Recall@10
XLM-RoBERTa	0.2724	0.2839	0.2762	0.5018
PhoBERT	0.4707	0.4900	0.4772	0.7223

- **Nhận xét:** PhoBERT vượt trội hơn XLM-RoBERTa trên tất cả các chỉ số, thể hiện lợi thế của việc sử dụng mô hình đơn ngôn ngữ được thiết kế riêng cho tiếng Việt. Recall@10 của PhoBERT đạt 0.7223, chỉ ra rằng tài liệu đúng được truy xuất trong top 10 kết quả cho 72.23% các truy vấn.

5.2.2. Có xếp hạng lại

Mô hình	MAP	MRR	NDCG	Recall@10
XLM-RoBERTa	0.3558	0.3613	0.3499	0.5545
PhoBERT	0.5598	0.5766	0.5655	0.7714

Nhận xét: Việc xếp hạng lại bằng cross-encoder cải thiện hiệu suất của cả hai mô hình. Đối với PhoBERT, Recall@10 tăng từ 0.7223 lên 0.7714, minh chứng cho hiệu quả của việc xếp hạng lại trong việc tinh chỉnh kết quả truy xuất. Các cải thiện về MAP, MRR, và NDCG cũng xác nhận rằng cross-encoder nâng cao chất lượng xếp hạng bằng cách phân biệt tốt hơn giữa các tài liệu có độ tương đồng cao.

6 Conclusion

Chúng tôi đã xây dựng một hệ thống truy vấn hiệu quả và mạnh mẽ dành cho chatbot tư vấn pháp luật, kết hợp ưu điểm của kiến trúc bi-encoder và cross-encoder. Hệ thống được thiết kế để xử lý các phức tạp trong bài toán truy vấn văn bản pháp luật tiếng Việt, kết hợp giữa truy vấn dựa trên embedding và kỹ thuật reranking để đảm bảo độ chính xác và tốc độ truy vấn nhanh. Kết quả thực nghiệm cho thấy bi-encoder dựa trên PhoBERT vượt trội hơn so với các mô hình khác, đạt Recall@10 là 0.7714 khi kết hợp với cross-encoder reranking. Việc tích hợp hệ thống truy vấn này với một LLM agent giúp chatbot có khả năng cung cấp các câu trả lời chính xác và phù hợp với ngữ cảnh, trở thành một công cụ hữu ích cho cả chuyên gia pháp lý và người dùng phổ thông. Về hướng đi trong tương lai một số đề xuất được chúng tôi cân nhắc bao gồm cải thiện chất lượng vector embedding vì mô hình PhoBERT hiện chỉ chấp nhận chiều dài chuỗi đầu vào là 256 token, nên các vector embedding hiện tại không chứa hoàn toàn ngữ nghĩa của các tài liệu dài. Một số hướng đi khác là sử dụng các kĩ thuật xử lí dữ liệu trước khi đưa vào huấn luyện hoặc sử dụng các phương pháp kết hợp nhiều mô hình (*ensemble*) cũng đáng được thử nghiệm

References

[1] [2005.11401] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

[2] GitHub - NguyenDinhTiem/AIO-MENTAL-HEALTH: Ứng dụng chăm sóc sức khỏe tinh thần thông minh

[3] Bi-encoder vs Cross encoder?When to use which one?

[4] Cross-Encoders — Sentence Transformers documentation

[5] HARD NEGATIVE MINING. A SIMPLE EXPLAINATION