BÁO CÁO ĐỒ ÁN MÔN KỸ THUẬT LẬP TRÌNH TRÍ TUỆ NHÂN TẠO - PHẦN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Le Van Hoang - 22520465
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
{22520465}@gm.uit.edu.vn

Tóm tắt

Đây là báo cáo đồ án môn học **Kĩ Thuật Lập Trình Trí Tuệ Nhân Tạo - CS311** về ứng dụng chatbot trong hỏi đáp các vấn đề về Machine Learning. Đồ án này xoay quanh việc xử dữ liệu đúng cách, kết hợp semantic Route để truy vấn một cách hiệu quả hơn. Ngoài ra, thiết kế prompt cũng là một trong những phần quan trọng giúp chatbot tránh được hiện tượng 'hallucination'. Chatbot này hoạt động dựa trên cơ chế RAG (Retrieval-augmented generation)

1 Giới thiệu

Đồ án này được thực hiện với mục tiêu chính là nghiên cứu và phát triển một hệ thống chatbot hỗ trợ hỏi đáp các vấn đề liên quan đến lĩnh vực Machine Learning, nằm trong khuôn khổ môn học Kĩ Thuật Lập Trình Trí Tuệ Nhân Tạo - CS311. Với sự phát triển nhanh chóng của trí tuệ nhân tạo và nhu cầu ngày càng cao về các công cụ hỗ trợ học tập, nghiên cứu, việc xây dựng một chatbot thông minh có khả năng trả lời chính xác các câu hỏi chuyên môn là một bài toán có ý nghĩa thực tiễn lớn.

Trọng tâm của đồ án xoay quanh việc xử lý dữ liệu một cách hiệu quả và chính xác, đảm bảo rằng thông tin được chatbot cung cấp là đáng tin cậy và phù hợp với ngữ cảnh. Để đạt được điều này, đồ án áp dụng kỹ thuật semantic Route, giúp tối ưu hóa quá trình truy vấn thông tin bằng cách hiểu sâu hơn về ý nghĩa và ngữ nghĩa của câu hỏi. Điều này không chỉ giúp chatbot trả lời nhanh chóng mà còn đảm bảo tính chính xác và phù hợp với yêu cầu của người dùng.

Một trong những thách thức lớn khi xây dựng chatbot là hiện tượng 'hallucination', tức là chatbot đưa ra thông tin sai lệch hoặc không có cơ sở. Để giải quyết vấn đề này, đồ án tập trung vào việc thiết kế prompt (gợi ý đầu vào) một cách cẩn thận, giúp chatbot hiểu rõ hơn về phạm vi và yêu cầu của câu hỏi, từ đó tránh được việc đưa ra các câu trả lời không chính xác hoặc không liên quan.

Chatbot trong đồ án này hoạt động dựa trên cơ chế RAG (Retrieval-augmented generation), một phương pháp kết hợp giữa truy xuất thông tin và tạo lập câu trả lời. Cơ chế này cho phép chatbot không chỉ dựa vào kiến thức có sẵn trong mô hình ngôn ngữ mà còn tận dụng các nguồn dữ liệu bên ngoài để tìm kiếm và trích xuất thông tin phù hợp. Nhờ đó, chatbot có khả năng đưa ra câu trả lời chính xác, chi tiết và đáng tin cậy hơn, đồng thời giảm thiểu nguy cơ mắc lỗi hoặc đưa ra thông tin không chính xác.

2 Kiến trúc hệ thống

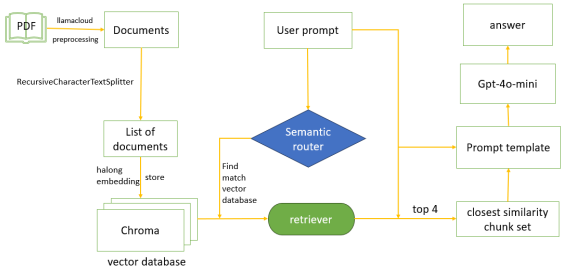

Figure 1: Kiến trúc hệ thống

Giới thiệu

Dữ liệu mà chúng ta đang xử lý là một cuốn sách được lưu trữ dưới định dạng PDF. Tuy nhiên, pipeline xử lý dữ liệu này có thể dễ dàng mở rộng để áp dụng cho nhiều file PDF khác nhau. Quy trình xử lý dữ liệu có thể được tóm tắt thành các bước chính như sau: **Parse** (Phân tích), **Chunking** (Chia nhỏ), **Embedding** (Biểu diễn vector), **Router** (Định tuyến), và **Prompt** (Tạo prompt). Các phần tiếp theo sẽ đi sâu vào từng thành phần này, đồng thời thảo luận về những vấn đề phát sinh và cách thức xử lý chúng.

Parse (Phân tích dữ liệu)

Một trong những điểm cần lưu ý khi đọc các file PDF dài, đặc biệt là những tài liệu có nhiều trang,

là phải đảm bảo tính liền mạch của nội dung giữa các trang. Thông thường, các hàm đọc file PDF mặc định sẽ xử lý mỗi trang như một đối tượng `Document` riêng biệt, điều này vô tình thực hiện việc chia nhỏ dữ liệu (*chunking*) ngay từ bước đầu tiên. Điều này có thể dẫn đến việc mất đi sự liên kết giữa các phần nội dung quan trọng.

Đối với các cuốn sách, thường xuất hiện những nội dung thừa ở đầu và cuối mỗi trang. Ví dụ, ở đầu trang có thể chứa thông tin về số chương hoặc tiêu đề phụ, trong khi cuối trang có thể chứa các ghi chú hoặc số trang. Những nội dung này không cần thiết và cần được loại bỏ để đảm bảo tính chính xác và sự liền mạch của dữ liệu.

Ngoài ra, cuốn sách mà chúng ta đang xử lý là một tài liệu cơ bản về Machine Learning, do đó, nó chứa rất nhiều công thức toán học và các ký tự đặc biệt. Việc phân tích (*parse*) các công thức toán học một cách hiệu quả là một thách thức không nhỏ. Để giải quyết vấn đề này, chúng ta cần lựa chọn các công cụ phù hợp để đảm bảo rằng các công thức và ký tự toán học được xử lý chính xác.

Giải pháp: Trong trường hợp này, chúng ta sử dụng `llamaparse` của `llamacloud` để chuyển đổi file PDF sang định dạng markdown. Lý do chọn định dạng markdown là vì nó cung cấp các tín hiệu rõ ràng cho từng phần nội dung, giúp việc xử lý và phân tích trở nên dễ dàng hơn. Sau đó, chúng ta sử dụng thư viện `re` (regular expression) để loại bỏ các nội dung thừa dựa trên các mẫu (*pattern*) được xác định trước.

Chunking (Chia nhỏ dữ liệu)

Sau khi đã phân tích và làm sạch dữ liệu, bước tiếp theo là chia nhỏ dữ liệu thành các phần nhỏ hơn để phục vụ cho các bước xử lý tiếp theo. Có nhiều phương pháp chia nhỏ dữ liệu (*chunking*) khác nhau, chẳng hạn như:

- `CharacterTextSplitter`: Phương pháp này chia nhỏ văn bản dựa trên số lượng ký tự. Tuy đơn giản nhưng nó thường không hiệu quả trong hầu hết các bài toán thực tế, đặc biệt là khi văn bản có cấu trúc phức tạp.
- `MarkdownHeaderTextSplitter`: Phương pháp này phù hợp với định dạng markdown, nó chia nhỏ văn bản dựa trên các phần (*section*) của mỗi chương. Mặc dù hiệu quả hơn so với `CharacterTextSplitter`, nhưng nó vẫn có nhược điểm là các section có thể quá dài, dẫn đến việc xử lý không được tối ưu.

Giải pháp: Để khắc phục những hạn chế trên, chúng ta sử dụng `RecursiveCharacterTextSplitter`. Phương pháp này cho phép chia nhỏ văn bản một cách linh hoạt hơn bằng cách điều chỉnh các dấu phân cách (*separators*). Cụ thể, chúng ta sử dụng các dấu phân cách như `['#', '\n\n', '\n']` để đảm bảo rằng văn bản được chia nhỏ một cách hợp lý, đồng thời giữ được cấu trúc và ý nghĩa của từng phần.

Embedding (Biểu diễn vector)

Trong phần này, chúng ta cần lựa chọn một mô hình embedding phù hợp để áp dụng vào bài toán cụ thể. Mô hình embedding sẽ chuyển đổi văn bản thành các vector số học, giúp máy tính có thể hiểu và xử lý ngôn ngữ tự nhiên một cách hiệu quả. Đối với bài toán này, các yêu cầu chính của mô hình embedding bao gồm:

- Hỗ trợ đa ngôn ngữ (multilingual), đặc biệt là tiếng Việt, vì đây là ngôn ngữ chính được sử dụng trong bài toán.
- Phục vụ tốt cho các bài toán Hỏi đáp (Question Answering - QA), đảm bảo khả năng trích xuất thông tin chính xác từ ngữ cảnh.
- Nhẹ và tối ưu hóa về mặt hiệu suất, đặc biệt khi triển khai trên các hệ thống có tài nguyên hạn chế.

Một lựa chọn phù hợp cho bài toán này là mô hình `SentenceTransformer` với phiên bản `hiieu/halong_embedding`. Mô hình này được thiết kế để hỗ trợ tiếng Việt và có khả năng xử lý các tác vụ liên quan đến biểu diễn văn bản một cách hiệu quả.

Lưu ý quan trọng: Nếu hệ thống không có GPU hoặc tài nguyên tính toán hạn chế, có thể sử dụng API của các dịch vụ embedding có sẵn. Tuy nhiên, việc sử dụng API có một số nhược điểm như khả năng hỗ trợ tiếng Việt không tốt bằng các mô hình cục bộ và không thể tinh chỉnh (fine-tune) để phù hợp với dữ liệu riêng của bài toán.

Router

Semantic Router là một kỹ thuật quan trọng trong việc xây dựng chatbot thông minh. Nó sử dụng trí tuệ nhân tạo (AI) để định hướng và phân loại các yêu cầu của người dùng dựa trên ngữ nghĩa (semantics) thay vì chỉ dựa vào từ khóa hoặc cú pháp cụ thể. Kỹ thuật này giúp chatbot hiểu được ý định (intent) của người dùng một cách chính xác

hơn, từ đó chuyển hướng cuộc trò chuyện đến các chức năng hoặc dịch vụ phù hợp.

Trong đồ án này, Semantic Router được sử dụng với mục đích chính là tìm kiếm và chọn lựa cơ sở dữ liệu vector phù hợp để xử lý các truy vấn của người dùng. Điều này đặc biệt hữu ích khi làm việc với các nguồn dữ liệu lớn và phức tạp, chẳng hạn như một cuốn sách.

Ví dụ, trong một cuốn sách, nội dung của các chương thường độc lập và có tính lặp lại. Chẳng hạn, khái niệm về hàm mất mát (loss function) có thể được định nghĩa trong cả chương về Linear Regression và chương về SVM. Tuy nhiên, nếu thực hiện truy vấn trên toàn bộ cuốn sách, kết quả có thể không chính xác do sự trùng lặp thông tin. Thay vào đó, nếu truy vấn chỉ được thực hiện trong phạm vi một chương cụ thể (ví dụ: chương về Linear Regression), kết quả sẽ phù hợp và chính xác hơn.

→ Ý tưởng chính ở đây là chia cuốn sách thành nhiều chương riêng biệt, thực hiện embedding cho từng chương và lưu trữ các vector tương ứng. Sau đó, sử dụng Semantic Router để xác định và tìm kiếm cơ sở dữ liệu vector phù hợp nhất cho từng câu truy vấn của người dùng.

Prompt

Việc định nghĩa prompt (câu lệnh đầu vào) là một bước quan trọng trong việc xây dựng hệ thống chatbot. Mặc dù việc tạo prompt có vẻ đơn giản và ít tốn công sức, nhưng việc lựa chọn một prompt phù hợp lại là thách thức lớn. Một prompt tốt cần đảm bảo rằng mô hình ngôn ngữ lớn (LLM) sẽ tập trung vào thông tin trong ngữ cảnh được cung cấp, thay vì sử dụng kiến thức có sẵn của nó.

Một vấn đề khác cần lưu ý là dữ liệu của chúng ta có thể không chứa thông tin liên quan đến một số câu hỏi của người dùng. Do đó, cần định nghĩa prompt sao cho chatbot có thể thông báo cho người dùng biết rằng câu hỏi của họ nằm ngoài phạm vi kiến thức của hệ thống.

→ Dưới đây là prompt mẫu được định nghĩa cho bài toán này:

```
"""Hãy hiểu context bên dưới và hãy
hiểu câu hỏi.
Xem thông tin chỉ có trong context
có thể dùng để trả lời một cách đầy
đủ cho câu hỏi hay không.
Nếu không, thì trả lời: "Xin lỗi,
tôi không có thông tin về câu
hỏi này".
Nếu có, thì hãy dùng nó để trả
lời câu hỏi.
Context: {context}
Câu hỏi: {question}"""
```

Prompt này đảm bảo rằng chatbot sẽ chỉ sử dụng thông tin từ ngữ cảnh được cung cấp để trả lời câu hỏi, đồng thời thông báo rõ ràng khi không có thông tin phù hợp.

3 Đánh giá hệ thống

Ở đây mình sẽ tạo 30 câu hỏi và tự đánh giá kết quả (dựa vào dữ liệu)

3.1 Kết quả đánh giá

Table 1: Tiêu đề bảng

Câu hỏi	Satisfy?
Làm thế nào để nhận biết overfitting và underfitting trong một mô hình machine learning?	No
Ma trận đơn vị là gì?	Yes
Null space là gì?	Yes
Đạo hàm của một vector là một vector hả?	Yes
Cách đạo hàm một vector?	Yes
Cách đạo hàm một ma trận?	No
Mục đích của cuốn sách là gì?	Yes
Những ai đóng góp để viết cuốn sách?	Yes
Xác xuất biên là gì?	Yes
Thế nào là quy tắc bayes?	Yes
Trong trường hợp nào thì dùng phân phối categorical?	Yes
Maximum likelihood estimation là gì?	Yes
Maximum likelihood estimation để làm gì?	Yes
Bài toán machine translation được hiểu như thế nào?	Yes
Thế nào là bài toán clustering?	Yes
Mô hình chung cho các bài toán machine learning?	No
Hàm mất mát của linear regression là gì?	No
Hàm mất mát trong linear regression là gì?	Yes
Trong trường hợp nào thì dùng linear regression?	Yes
Các kĩ thuật để tránh overfitting?	Yes
Tại sao phải validation?	Yes
Thuật toán K-nearest neighbor có cần training hay không?	Yes
K-nearest neighbor hoạt động như thế nào?	Yes
So sánh thuật toán SVM và KNN	No
SVM hoạt động như thế nào?	Yes
Bài toán tối ưu lồi là gì?	Yes
Thế nào là local optimum?	Yes
Trường hợp nào thì SVM hoạt động tệ?	No
Trường hợp nào thì SVM hoạt động tốt?	Yes

3.2 Nhận xét

Kết quả đạt 80% tỉ lệ trả lời thỏa mãn

4 Kết luận

Đồ án môn học Kỹ thuật Lập trình Trí tuệ Nhân tạo - CS311 này đã tập trung vào việc xây dựng một chatbot có khả năng hỗ trợ người dùng trong việc tìm hiểu và trả lời các câu hỏi liên quan đến lĩnh vực Machine Learning. Bằng cách áp dụng quy trình xử lý dữ liệu hiệu quả, kết hợp kỹ thuật Semantic Router để truy vấn thông tin chính xác, và thiết kế prompt cẩn thận để giảm thiểu hiện tượng 'hallucination', chatbot đã đạt được những kết quả khả quan.

Cụ thể, đồ án đã giải quyết thành công các vấn đề sau:

- Xử lý dữ liệu từ định dạng PDF phức tạp, bao gồm cả việc phân tích cú pháp, loại bỏ nội dung không liên quan, và xử lý các công thức toán học.
- Chia nhỏ dữ liệu thành các phần nhỏ hơn (chunking) một cách hợp lý, đảm bảo tính liên kết và ngữ nghĩa của thông tin.
- Lựa chọn và sử dụng mô hình embedding phù hợp để biểu diễn văn bản, hỗ trợ tốt cho việc truy vấn và tìm kiếm thông tin.

- Áp dụng kỹ thuật Semantic Router để định tuyến truy vấn, giúp chatbot tìm kiếm và truy xuất thông tin từ cơ sở dữ liệu vector một cách hiệu quả.
- Thiết kế prompt tối ưu để hướng dẫn mô hình ngôn ngữ lớn (LLM) tập trung vào thông tin được cung cấp, giảm thiểu nguy cơ đưa ra thông tin sai lệch.

Kết quả đánh giá cho thấy chatbot đạt tỷ lệ trả lời thỏa mãn là 80%. Đây là một kết quả đáng khích lệ, cho thấy tiềm năng ứng dụng của chatbot trong việc hỗ trợ học tập và nghiên cứu về Machine Learning.

Tuy nhiên, đồ án cũng còn một số hạn chế cần được cải thiện trong tương lai:

- Cần mở rộng tập dữ liệu để chatbot có thể trả lời được nhiều loại câu hỏi hơn, bao gồm cả những câu hỏi phức tạp và chuyên sâu.
- Cần cải thiện khả năng xử lý và hiểu ngôn ngữ tự nhiên của chatbot, đặc biệt là đối với các câu hỏi có nhiều nghĩa hoặc sử dụng thuật ngữ chuyên môn.
- Cần tối ưu hóa hiệu suất của hệ thống để chatbot có thể trả lời nhanh chóng và hiệu quả hơn.
- Cần thử nghiệm và đánh giá chatbot trên một tập người dùng lớn hơn để có được cái nhìn khách quan và toàn diện hơn về khả năng của hệ thống.

Nhìn chung, đồ án đã đạt được các mục tiêu đề ra và đóng góp vào việc nghiên cứu và phát triển các ứng dụng chatbot trong lĩnh vực giáo dục và đào tạo. Hy vọng rằng, những kết quả và kinh nghiệm thu được từ đồ án này sẽ là nền tảng cho các nghiên cứu và phát triển tiếp theo trong tương lai.