Báo cáo: Quy trình Xây dựng và Đánh giá Pipeline RAG với Mô hình Ngôn ngữ Lớn (LLMs)

Phạm Thành Long-22022604, Nguyễn Đức Minh-22022533, and Trần Tiến Nam-22022594

Trường Đại học Công nghệ, K67A-AI2

Ứng dụng AI cho ngôn ngữ, AIT3009

Tóm tắt nội dung

Báo cáo này trình bày quy trình xây dựng pipeline Retrieval-Augmented Generation (RAG) cơ bản sử dụng mô hình ngôn ngữ lớn (LLMs) trên miền dữ liệu về VNU và các trường thành viên. Xây dựng quy trình từ thu thập dữ liệu, xử lý dữ liệu, thiết kế pipeline, đến đánh giá hiệu quả bằng các chỉ số như BLEU, ROUGE.

1 Giới thiệu

Retrieval-Augmented Generation (RAG) là kỹ thuật kết hợp truy xuất thông tin và sinh văn bản để nâng cao khả năng trả lời của mô hình ngôn ngữ lớn (LLMs). RAG hữu ích khi xử lý các truy vấn yêu cầu thông tin chuyên sâu hoặc dữ liệu cập nhật, vốn là hạn chế của LLMs thông thường.

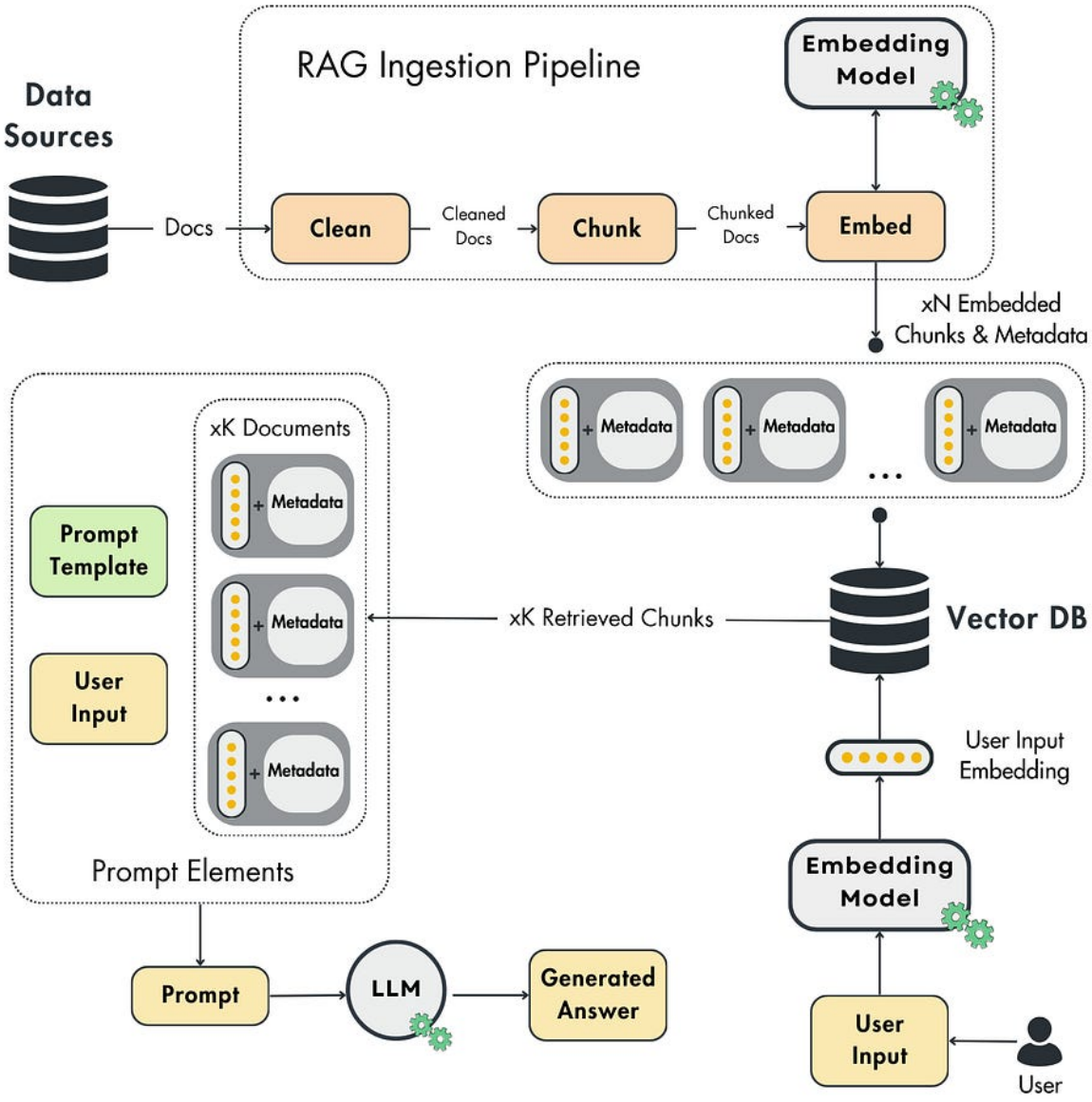
Báo cáo này mô tả quy trình triển khai pipeline RAG cơ bản, bao gồm:

(1) Thu thập dữ liệu từ các nguồn trực tuyến.

(2) Xử lý và chuyển đổi dữ liệu thành định dạng phù hợp.

(3) Thiết kế và triển khai pipeline RAG.

(4) Đánh giá hiệu quả bằng các chỉ số BLEU, ROUGE.



Hình 1: End2End RAG Pipeline

2 Phương pháp

Quy trình xây dựng pipeline RAG gồm bốn giai đoạn: (1) Thu thập dữ liệu, (2) Xử lý dữ liệu,(3) Xây dựng bộ Q&A,(4) Thiết kế pipeline RAG, và (5) Đánh giá hiệu quả.

2.1 Thu thập Dữ liệu

Nguồn dữ liệu: Dữ liệu được thu thập từ các trang web công khai VNU và các trường thành viên như UET, UEB, ULIS. Công cụ thu thập được sử dụng là `BeautifulSoup` và `Scrapy`.

Quy mô: Thu thập 500 tài liệu, mỗi tài liệu khoảng 500 từ.

Xử lý ban đầu: Loại bỏ nội dung không liên quan (quảng cáo, liên kết hỏng), trích xuất văn bản thô và lưu dưới dạng TXT.

2.2 Xử lý Dữ liệu

Làm sạch dữ liệu: Trong quá trình thu thập dữ liệu từ các trang web, chúng tôi nhận thấy nhiều văn bản chứa các ký tự HTML dư thừa, nội dung bị trùng lặp, các đoạn văn thiếu ngữ nghĩa rõ ràng hoặc câu văn lộn xộn. Thay vì xử lý dữ liệu hoàn toàn bằng phương pháp thủ công, nhóm đã lựa chọn áp dụng các mô hình AI như Gemini nhằm tự động làm sạch và chuẩn hóa nội dung.

Đối với mỗi tệp văn bản TXT, hệ thống sẽ gửi yêu cầu đến API kèm theo đoạn prompt sau: *"Dưới đây là dữ liệu đã được thu thập nhưng ở dạng thô, hãy làm sạch (chỉnh sửa lại, chứ không phải gợi ý code) để tôi có thể lấy câu trả lời của bạn và lưu vào txt và tiến hành embedding cho RAG với LLMs. Nếu dữ liệu không có giá trị, hãy trả về NULL (lưu ý, ký tự | thể hiện cho bảng): {text}"*

Nhúng văn bản: Trong bài tập này, nhóm đã thử nghiệm với nhiều mô hình embedding khác nhau nhằm tìm ra phương án tối ưu cho việc mã hóa văn bản. Sau quá trình đánh giá và so sánh, nhóm đã lựa chọn mô hình `intfloat/multilingual-e5-large` từ nền tảng Hugging Face, do mô hình này cho kết quả nhất quán và hiệu quả cao đối với dữ liệu tiếng Việt.

Lưu trữ: Vector được lưu trong `FAISS` (index 384 chiều), hỗ trợ tìm kiếm dựa trên độ tương đồng cosine.

2.3 Xây dựng bộ Q&A

Chúng tôi sử dụng mô hình Gemini 2.0 Flash (triển khai qua API của Google AI Studio) để sinh câu hỏi tự động. Mô hình được yêu cầu tóm tắt nội dung từ các tệp văn bản đã được xử lý trước, sau đó tạo ra các cặp câu hỏi – câu trả lời với tiêu chí ngắn gọn, rõ ràng và đúng trọng tâm.

Từ tập dữ liệu gồm các tệp TXT đã được làm sạch, mô hình đã sinh ra khoảng 1000 cặp hỏi – đáp. Chúng tôi tiếp tục lọc lựa các câu hỏi có ý nghĩa, đồng thời phân loại theo mức độ khó. Kết quả cuối cùng là một bộ Q&A gồm hơn 250 cặp câu hỏi – câu trả lời chất lượng.

2.4 Thiết kế Pipeline RAG

Pipeline RAG gồm hai mô-đun chính, dựa theo pipeline RAG cơ bản được sử dụng rộng rãi:

(1) Module Truy xuất: Chuyển đổi truy vấn thành vector bằng `intfloat/multilingual-e5-large`. Sử dụng `FAISS` để tìm top-k tài liệu liên quan. Ghép các đoạn văn thành ngữ cảnh (tối đa 512 token).

(2) Module Sinh văn bản: Sử dụng `vinallama-7b-chat` để sinh câu trả lời. Đầu vào gồm truy vấn và ngữ cảnh từ mô-đun truy xuất. Đầu ra là văn bản ngắn gọn, chính xác. Pipeline được triển khai bằng Python, sử dụng `Ctransformers` và `faiss-cpu`.

2.5 Đánh giá Hiệu quả

Hiệu quả được đánh giá trên 200 truy vấn (UET, ULIS, UEB) với các chỉ số chuyên dụng cho đánh giá hiệu quả sinh văn bản của LLMs.

Exact Match (EM): Độ đo này kiểm tra liệu câu trả lời dự đoán có khớp hoàn toàn với câu trả lời tham chiếu sau khi chuẩn hóa hay không. Giá trị là 1 nếu trùng khớp tuyệt đối, ngược lại là 0. EM đo lường độ chính xác tuyệt đối.

F1 Score: Là trung bình điều hòa giữa Precision và Recall ở cấp độ từ. Độ đo này phản ánh sự giao nhau giữa các từ trong câu trả lời dự đoán và tham chiếu, phù hợp với các bài toán có nhiều câu trả lời đúng tiềm năng.

BLEU Score: BLEU (Bilingual Evaluation Understudy) đánh giá dựa trên độ trùng lặp n-gram giữa câu trả lời dự đoán và tham chiếu. Độ đo này phổ biến trong dịch máy, có thể sử dụng hàm làm mượt để cải thiện ổn định khi câu trả lời ngắn.

ROUGE-L Score: ROUGE-L dựa trên độ dài chuỗi con chung dài nhất (Longest Common Subsequence – LCS) giữa hai câu. Độ đo này nhấn mạnh cấu trúc ngữ nghĩa và thứ tự từ trong câu, phù hợp để đánh giá tóm tắt hoặc sinh văn bản.

3 Kết quả

Do hạn chế về tài nguyên phần cứng, nên nhóm có sử dụng mô hình nhỏ đã quantized và sử dụng trên `Ctransformers` để giảm thiểu sức nặng tính toán lên phần cứng. Do đó, câu trả lời được tạo ra bởi LLMs không hoàn toàn chính xác, chỉ đáp ứng điều kiện là có chứa nội dung câu trả lời trong đó.

Exact Match	F1 Score	BLEU	ROUGE-L
0.0952	0.5124	0.3314	0.4725

Bảng 1: Kết quả đánh giá mô hình theo các metrics.


```
{
  "question": "Học bổng Annex HKII năm học 2024-2025 được thông báo vào ngày nào?",
  "answer": "Học bổng Annex HKII năm học 2024-2025 được thông báo vào ngày 03/03/2025.",
  "llm_answer": "Ngày 03 tháng 03 năm 2025.\n"
},
{
  "question": "Thông tin về học bổng K-T năm học 2024-2025 được thông báo vào ngày nào?",
  "answer": "Thông tin về học bổng K-T năm học 2024-2025 được thông báo vào ngày 28/02/2025.",
  "llm_answer": "Ngày 28/02/2025.\n"
},
{
  "question": "Lễ trao học bổng Mitsubishi năm học 2024-2025 diễn ra khi nào?",
  "answer": "Lễ trao học bổng Mitsubishi năm học 2024-2025 diễn ra vào ngày 21/02/2025 (Thứ Sáu).",
  "llm_answer": "Lễ trao học bổng Mitsubishi năm học 2024-2025 diễn ra vào ngày 21 tháng 2 năm 2025.\n"
},
{
  "question": "Trường Đại học Công nghệ tặng bao nhiêu suất học bổng Vietcombank năm học 2024-2025?",
  "answer": "Trường Đại học Công nghệ tặng 20 suất học bổng Vietcombank.",
  "llm_answer": "Trường Đại học Công nghệ tặng 20 suất học bổng Vietcombank năm học 2024-2025.\n"
},
```


Hình 2: Một số kết quả tham khảo

4 Thảo luận

Pipeline RAG cải thiện chất lượng câu trả lời so với LLMs độc lập, đặc biệt với truy vấn chuyên sâu vào các domain công việc cụ thể. Tuy nhiên, hạn chế bao gồm:

(1) Tài nguyên phần cứng: Để có thể nhận được câu trả lời chính xác, phần cứng mạnh và đáp ứng được mô hình là điều tất yếu.

(2) Chất lượng dữ liệu: Dữ liệu crawl chứa nhiễu (quảng cáo, nội dung không liên quan).

(3) Thời gian xử lý: Sinh câu trả lời chiếm phần lớn thời gian.

(4) Giới hạn ngữ cảnh: Cửa sổ ngữ cảnh của `VinaLLaMA-7B` giới hạn lượng thông tin.

Các cải tiến tiềm năng được đề xuất bao gồm:

(1) Sử dụng công cụ crawl tiên tiến hơn.

(2) Tối ưu `FAISS` bằng GPU.

(3) Tích hợp LLMs mạnh hơn.

(4) Có thể cân nhắc đến việc dùng API LLMs từ bên thứ 3 nếu có thể gác lại vấn đề bảo mật dữ liệu.

5 Kết luận

Pipeline RAG cải thiện hiệu quả trả lời của LLMs. Tối ưu hóa crawl dữ liệu, truy xuất và mô hình LLMs sẽ giúp nâng cao hiệu suất, mở ra ứng dụng trong giáo dục, nghiên cứu và hỗ trợ khách hàng.

Tài liệu

[1] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*.

[2] Hugging Face. (2023). Transformers: State-of-the-art Natural Language Processing. `https://huggingface.co/transformers`.

[3] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

[4] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Jegou, H. (2023). LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.