Title: Building a Retrieval Augmented Generation System for Pittsburgh and CMU Question Answering

Chu Huỳnh Đức_22022612

Abstract

Báo cáo này trình bày chi tiết về việc phát triển hệ thống Retrieval Augmented Generation - RAG được thiết kế để trả lời các câu hỏi thực tế về Pittsburgh và Đại học Carnegie Mellon (CMU). Hệ thống RAG tăng cường LLM với thông tin liên quan được truy xuất từ một cơ sở kiến thức được biên soạn tùy chỉnh. Dự án này bao gồm việc thu thập dữ liệu từ các tài nguyên web được chỉ định, chú thích các tập dữ liệu thử nghiệm và huấn luyện, và phát triển lặp đi lặp lại một quy trình RAG. Một số biến thể của hệ thống RAG đã được triển khai và đánh giá, tập trung vào tác động của các mô hình nhúng khác nhau và prompt engineering. Hệ thống cuối cùng, sử dụng mô hình nhúng thenlper/gte-small và một prompt với LLM Gemma 3 4B, đã cho thấy những cải tiến so với RAG cơ sở và phương pháp LLM closed-book, đạt điểm F1 là 78,06% và Exact Match là 58,00% trên một tập thử nghiệm. Phân tích hiệu suất trên một tập test_questions cũng nhấn mạnh sự phụ thuộc của hệ thống vào phạm vi knowledge của nó.

1. Introduction

Trả lời câu hỏi thực tế QA là một nhiệm vụ quan trọng trong Xử lý Ngôn ngữ Tự nhiên. Mặc dù các Mô hình Ngôn ngữ Lớn (LLM) đã cho thấy những khả năng ấn tượng, kiến thức của chúng thường bị giới hạn trong dữ liệu huấn luyện, khiến chúng gặp khó khăn với thông tin chuyên biệt theo lĩnh vực hoặc rất gần đây. Retrieval Augmented Generation - RAG giải quyết vấn đề này bằng cách cung cấp cho LLM các tài liệu liên quan từ một cơ sở kiến thức tại thời điểm suy luận, cho phép nó tạo ra các câu trả lời có thông tin và chính xác hơn.

2. Data Creation

2.1. Knowledge Resource Compilation

Knowledge được tạo dựa trên danh sách các trang web được đề xuất trong github. Chúng bao gồm các trang Wikipedia về Pittsburgh và lịch sử của nó, trang web của Thành phố Pittsburgh, Encyclopedia Britannica, Visit Pittsburgh, các trang giiới thiệu" của CMU và các liên kết cụ thể liên quan đến quy định thuế của thành phố, ngân sách, lịch sự kiện địa phương, các tổ chức văn hóa (dàn nhạc giao hưởng, opera, bảo tàng), lễ hội ẩm thực và các đội thể thao.

Quy trình thu thập dữ liệu chủ yếu nhắm vào các URL được liệt kê rõ ràng. Đối với Quy định Thuế của Thành phố Pittsburgh, các liên kết PDF trực tiếp đã được xác định thủ công từ trang pittsburghpa.gov/finance/tax-forms trong cột regulations và được thêm vào danh sách thu thập. Tương tự, liên kết trực tiếp đến tệp PDF đã được bao gồm. Đối với các trang web

sự kiện đề xuất điều hướng đến các trang theo tháng cụ thể, chỉ trang đích chính được thu thập ban đầu, với sự hiểu biết rằng mức độ liên quan của sự kiện sau ngày 27 tháng 10 sẽ là một tiêu chí lọc trong quá trình tạo câu hỏi hoặc xử lý RAG.

2.2. Data Extraction

Raw text data được extracted từ compiled resources

- Đối với các trang HTML, thư viện requests được sử dụng để tìm nạp nội dung và BeautifulSoup4 để phân tích HTML. Việc làm sạch cơ bản đã được thực hiện để loại bỏ các yếu tố mẫu phổ biến như thanh điều hướng, chân trang, tập lệnh và thẻ kiểu.
- For PDF documents (e.g., tax regulations, city budget), sử dụng pypdf

Văn bản raw được trích xuất từ mỗi URL đã được lưu vào các tệp .txt riêng lẻ, tạo thành folder knowledge_base_raw.

2.3. Data Annotation

Question-answer pairs được tạo thủ công để (training) và evaluation (testing). nguồn chính là knowledge_base_raw directory.

- **Test Data:** 100 question-answer pairs được tạo test set (data/test/). câu hỏi đa dạng về nhiều khía cạnh như lịch sử văn hóa, âm nhạc, thể thao,...
- **Training Data:** 20 question-answer pairs được tạo để training set (data/train/). chủ yếu để few-shot examples nếu cần prompting, hoặc insant check.
- **Annotation Process:** mỗi câu hỏi sẽ có một câu trả lời tương ứng, ngắn ngọn, đầy đủ, có thể có chú thích nếu cần.

3. Model Details

The RAG system bao gồm document embedder, document retriever (vector store), và document reader (LLM). Langchain là primary framework.

3.1. RAG Pipeline Components

- **Document & Query Embedder:** Converts text into dense vector representations.
- **Document Retriever:** sử dụng vector store (FAISS) để lưu trữ document chunk embeddings và retrieve top-k chunks tương tự nhất về ngữ nghĩa với question embedding. Documents từ knowledge_base_raw were được load dùng DirectoryLoader and TextLoader, rồi chia nhỏ sử dụng RecursiveCharacterTextSplitter (chunk size 750, overlap 100).
- **Document Reader (LLM):** The Gemma 3 4B model, accessed via Ollama, được sử dụng là model chính. nó nhận original question và content của retrieved document chunks để sinh câu trả lời. The RetrievalQA chain from Langchain (with chain_type="stuff") được sử dụng.

3.2. System Variations

có 4 biến thể được triển khai:

1. **Variation 1 (Baseline RAG - V1_MiniLM_OriginalPrompt):**
    - **Embedding Model:** sentence-transformers/all-MiniLM-L6-v2
    - **LLM:** Gemma 3 4B (via Ollama)
    - **Retriever:** FAISS, k=3 (top 3 documents)

Prompt (Original):

Use the following pieces of context to answer the question at the end.
If you don't know the answer from the context, just say that you don't know, don't try to make up an answer.
Keep the answer concise and directly related to the question.

Context:
{context}

Question: {question}

Helpful Answer:

2. **Variation 2 (Prompt Engineering - V2_MiniLM_StrictPrompt):**
    - **Embedding Model:** sentence-transformers/all-MiniLM-L6-v2
    - **LLM:** Gemma 3 4B (via Ollama)
    - **Retriever:** FAISS, k=3

Prompt (Stricter V2): câu trả lời ngắn gọn, thẳng thắn hơn V1.

Based *only* on the following context, answer the question.
If the answer is not found in the context, respond with "I don't know". Do not make up information.
Provide only the most direct and concise answer to the question. For example, if the question asks for a name, provide only the name. If it asks for a year, provide only the year. Do not include extra phrases, repeat the question, or offer additional details unless absolutely necessary to answer the question directly.

Context:
{context}

Question: {question}

Concise Answer:
IGNORE_WHEN_COPYING_START
content_copy download
IGNORE_WHEN_COPYING_END

Justification: giải quyết tính dài dòng quan sát được ở V1 và khuyến khích các câu trả lời trực tiếp hơn, tập trung vào thực thể.

3. **Variation 3 (Improved Embedding - V3_gte-small_StrictPrompt):**

- **Embedding Model:** thenlper/gte-small (a higher-performing lightweight model).
- **LLM:** Gemma 3 4B (via Ollama)
- **Retriever:** FAISS, k=3. (The FAISS index was rebuilt using gte-small embeddings).
- **Prompt:** Stricter V2 "Concise Answer:" prompt.
- *Justification:* thay đổi model nhúng vì V2 có 5 câu hỏi i dont know vì đáp án không có trong k=3 file (trong khi câu trả lời có trong txt file khác, có thể k=3 hơi ít nhưng phần lớn là do model embedding chưa hiệu quả).

4. **Closed-Book System (LLM Only - ClosedBook_Gemma):**
   - **LLM:** Gemma 3 4B (via Ollama)
   - **No retrieval component.** Questions were passed directly to the LLM with a simple prompt: Answer the following question concisely.\nQuestion: {question_text}\nAnswer:
   - *Justification:* so sánh hiệu suất với RAG.

4. Results

Systems đánh giá trên self-created test set of 100 question-answer pairs dùng Exact Match (EM) và F1 Score. EM là khớp tuyệt đối giữa prediction và reference . F1 score measures độ chính xác tổng thể.

Table 1: Performance of System Variations on Self-Created Test Set (100 Questions)

System Variation	Avg. EM (%)	Avg. F1 (%)	Avg. Time/Q (s)
V1: MiniLM + Original Prompt + Gemma 3 4B	46.00	69.93	~3.00
V2: MiniLM + Stricter Prompt V2 + Gemma 3 4B	58.00	75.47	~2.78
V3: gte-small + Stricter Prompt V2 + Gemma 3 4B	**58.00**	**78.06**	**~2.80**
Closed-Book: Gemma 3 4B alone	6.00	20.15	~1.68

do giới hạn về con người chỉ có một mình em và giới hạn về thiết bị phần cứng nên chỉ thực hiện những kết quả trên đã là tốt nhất có thể, từ V1 tới V3 có sự tiến triển rõ rệt.

5. Analysis

5.1. Impact of Prompt Engineering (V1 vs. V2)

so sánh giữa original và stricter prompt có sự khác biệt rõ rệt:

- EM tăng từ 46% lên 58%.
- F1 tăng từ 69.93% lên 75.47%.

Sự cải thiện này chủ yếu là do tính dài dòng của LLM. ví dụ, Q13 ("In what neighborhood does Little Italy Days take place?"), V1 predicted "Little Italy Days takes place in Bloomfield" (F1=0.25), while V2 correctly and concisely predicted "Bloomfield" (EM=1, F1=1.0). tương tự với nhiều câu khác, model giải thích dài dòng không cần thiết.

tuy nhiên cũng có câu rút gọn quá mức như Q67 ("What general expelled the French from Fort Duquesne in 1758?"), V1 correctly gave "General John Forbes," trong khi V2 outputted only "Forbes," giảm F1.

5.2. Impact of Embedding Model (V2 vs. V3)

thay đổi embedding model cũng giúp F1 score tăng (từ 75.47% lên 78.06%), EM giữ nguyên 58.00%.

gte-small thể hiện tốt hơn miniLm ở nhiều câu hỏi như:

- **Q1 ("What two rivers meet to form the Ohio in Pittsburgh?"):** MiniLM (V2) failed (F1=0.0), predicting "Ohio River." gte-small (V3) correctly identified "Allegheny River and Monongahela River" (F1=0.75).
- **Q8 ("Who served as mayor of Pittsburgh from 1946 to 1959?"):** MiniLM (V2) produced "I don't know." gte-small (V3) correctly answered "David Lawrence" (F1=0.8).
- **Q65 ("What is the business district of Pittsburgh called...?"):** MiniLM (V2) failed. gte-small (V3) correctly answered "Golden Triangle" (EM=1, F1=1.0).
- **Q68 ("Who laid out Pittsburgh in 1764?"):** MiniLM (V2) produced "I don't know." gte-small (V3) correctly answered "John Campbell" (EM=1, F1=1.0).

cho thấy rằng gte-small: model mới hơn có hiệu suất tốt hơn model vốn đã ra mắt từ lâu như MiniLm ở nhiều trường hợp.

5.3. Effectiveness of RAG vs. Closed-Book LLM

so sánh giữa RAG system (V3: EM 58.00%, F1 78.06%) và Closed-Book Gemma 3 4B (EM 6.00%, F1 20.15%) làm nổi bật rõ rệt sự cần thiết và hiệu quả của phương pháp RAG cho nhiệm vụ chuyên biệt theo lĩnh vực này.

- The Closed-Book LLM gặp khó với hầu hết câu hỏi yêu cầu kiến thức chuyên sâu về lĩnh vực cụ thể (e.g., names of mayors, specific event details, regulatory information), often hallucinating incorrect answers (ví dụ, Q13: Little Italy Days in "San Francisco"; Q8: Mayor "Carl Hays")
- mặc dù Closed-Book LLM nhanh hơn (1.68s vs. 2.80s for V3 RAG), nhưng chất lượng câu trả lời thấp hơn đáng kể.

5.4. Analysis of "I don't know" Responses on Official Test Set

trong test_quetions.csv bao gồm 575 questions. The best RAG system (V3) trả lời "I don't know" tới 106 questions (18.4%). nguyên nhân chủ yếu là do knowledge_base quá ít thông tin về những sự kiện đó, nhưng thay vì made up lên câu trả lời, em cho model thú nhận luôn là nó không biết để tránh sai lệch thông tin. nhiều trang web không truy cập được, và chủ yếu là do em làm 1 mình khó so được với nhóm 4 người, đồng thời phần cứng có hạn nên em chỉ crawl lượng web tương đối ít, nếu knowledge đầy đủ hơn thì chắc chắn model sẽ dễ dàng trả lời những câu hỏi này. tóm lại nguyên nhân không nằm ở model mà ở data không đủ.

6. Conclusion

Dự án này đã phát triển và đánh giá thành công một hệ thống Retrieval Augmented Generation để trả lời các câu hỏi về Pittsburgh và CMU. Những cải tiến lặp đi lặp lại, bao gồm prompt engineering và thay embedding model (thenlper/gte-small), dẫn đến final system (V3) đạt F1 score of 78.06% và EM of 58.00% trên self-annotated test set. Phân tích đã chứng minh vai trò quan trọng của cả prompt for concise generation và a high-quality embedding model.

Quan trọng nhất, phương pháp RAG cho thấy hiệu suất vượt trội hơn nhiều so với closed-book LLM, nhấn mạnh giá trị của nó đối với việc trả lời câu hỏi thực tế chuyên biệt theo lĩnh vực. Xu hướng của hệ thống trả lời "i dont know" cho một phần đáng kể các câu hỏi trong tập thử nghiệm chính thức làm nổi bật sự phụ thuộc của các hệ thống RAG vào phạm vi cơ sở kiến thức của chúng và được hiểu là hành vi đúng đắn trong việc tránh tạo ra thông tin sai lệch cho các truy vấn ngoài phạm vi.

Future work có thể là more advanced retrieval strategies (e.g., re-ranking, query expansion), alternative chunking methods, với chủ đề rộng hơn và llm thông minh hơn như deepseek r1 hay qwen 3,..

7. References

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv preprint arXiv:2005.11401.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). *Llama 2: Open foundation and fine-tuned chat models*. arXiv preprint arXiv:2307.09288.

Vu, T., Lai, V., Ha, N. T. T., Ngo, D. L., Nguyen, B. T., & Nguyen, M. L. (2023). *FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation*. arXiv preprint arXiv:2310.03214.