NGHIÊN CỨU VÀ PHÁT TRIỂN KHUNG LÀM VIỆC LẶP THÍCH ỨNG CHO BÀI TOÁN HỎI ĐÁP TRÊN TÀI LIỆU DÀI ĐA PHƯƠNG THỨC

Lê Nguyễn Anh Khoa - 23520742

Cáp Kim Hải Anh - 23520036

Tóm tắt

- Lớp: CS519.Q11.KHTN
- Link Github của nhóm: https://github.com/LeNguyenAnhKhoa/CS519.Q11
- Link YouTube video: https://youtu.be/E7iFQ5xhF7g
- Họ tên các thành viên:

Lê Nguyễn Anh Khoa - 23520742

Cáp Kim Hải Anh - 23520036

Giới thiệu

- Bối cảnh thực tế và lý do chọn đề tài:
  - Sự bùng nổ dữ liệu phi cấu trúc (pdf, slide) với các tài liệu doanh nghiệp/khoa học chứa văn bản lẫn nhiều hình ảnh hoặc biểu đồ phức tạp.
  - Người dùng cần một số câu hỏi để lấy thông tin từ các tài liệu
  - *Ví dụ:* So sánh doanh thu Bắc Mỹ trong biểu đồ (Trang 5) với dự báo (Trang 20)
  - → ***Thách thức:*** Thông tin bị đứt gãy ngữ cảnh. Để trả lời, máy tính cần nhìn trang 5 và thấy cần tìm tiếp thông tin ở trang 20
- Các giải pháp hiện tại và hạn chế:
  - Các mô hình RAG (Retrieval-Augmented Generation): tìm kiếm 1 lần dựa trên từ khoá/ vector tương đồng → không có khả năng lập kế hoạch, thất bại khi thông tin nằm rải rác ở nhiều trang hoặc cần suy luận chéo giữa ảnh và text
  - Multimodal LLMs: đưa cả nhiều trang tài liệu vào context window → chi phí quá cao và vấn đề mất thông tin ở giữa chuỗi context dài (Lost in the middle).
- → Cần một phương pháp thông minh hơn thay vì chỉ tìm kiếm mù quáng.

Giới thiệu

- Khoảng trống nghiên cứu:
  - Hiện chưa có một cơ chế định lượng "Khi nào thì đủ thông tin?". Các mô hình hiện tại thường hallucination thay vì biết mình thiếu dữ liệu để đi tìm tiếp.
  - Thiếu sự kết hợp giữa Vision Encoder và Tư duy tìm kiếm (Iterative Reasoning).
- Giải pháp đề xuất:
  - Chuyển từ Truy xuất tĩnh sang Lặp thích ứng (Adaptive Iterative) để tự động đánh giá và tìm kiếm lại những gì còn thiếu sót, thay vì trả lời ngay lập tức dựa trên dữ liệu không đủ
  - Điểm mới: Coi việc tìm kiếm là bài toán Tối ưu hóa thông tin (Information Optimization).
    - Sử dụng hàm InfoNCE để đo lường độ lợi thông tin (Information Gain).
    - Hệ thống tự động sinh câu hỏi phụ (Sub-query) để lấp đầy các khoảng trống kiến thức
- Bài toán: Xây dựng hệ thống hỏi đáp tự động trên tài liệu dài đa phương thức.
  - Input: Tập dữ liệu D gồm N trang (chứa text và ảnh hoặc biểu đồ) + Câu truy vấn ngôn ngữ tự nhiên Q từ người dùng
  - Output: Câu trả lời A chính xác dạng văn bản + Trích dẫn nguồn (số trang, vùng ảnh) dùng để trả lời

Mục tiêu

- Để giải quyết vấn đề "đứt gãy ngữ cảnh" và các hạn chế nêu trên, đề tài tập trung vào 03 mục tiêu cụ thể:
  1. **Mục tiêu 1:** Mô hình hóa toán học cho "Độ lợi thông tin" (Information Gain)
     - **Vấn đề giải quyết:** Các mô hình hiện tại tìm kiếm dựa trên cảm tính hoặc từ khóa, không biết *khi nào thì đủ thông tin*.
  2. **Mục tiêu 2:** Xây dựng pipeline hoàn chỉnh: Hiện thực hóa quy trình hỏi đáp lặp, tự động sinh câu hỏi phụ để tìm tin thiếu.
     - **Vấn đề giải quyết:** RAG đơn lượt thất bại với câu hỏi suy luận đa bước (cần tìm A rồi mới tìm B).
     - **Nhiệm vụ:** Phát triển thuật toán **Dynamic Retrieval Loop** với 2 thành phần:
       - *Generator:* Tự động sinh truy vấn phụ (Sub-queries) để lấp đầy khoảng trống.
       - *Judge (LLM):* Đánh giá và điều hướng luồng tìm kiếm.
  3. **Mục tiêu 3:** Kiểm chứng khả năng suy luận đa bước: Đánh giá tính khả thi và độ chính xác trên tập dữ liệu chuẩn (MP-DocVQA/MMVQA) so với RAG cơ bản.

Nội dung và Phương pháp

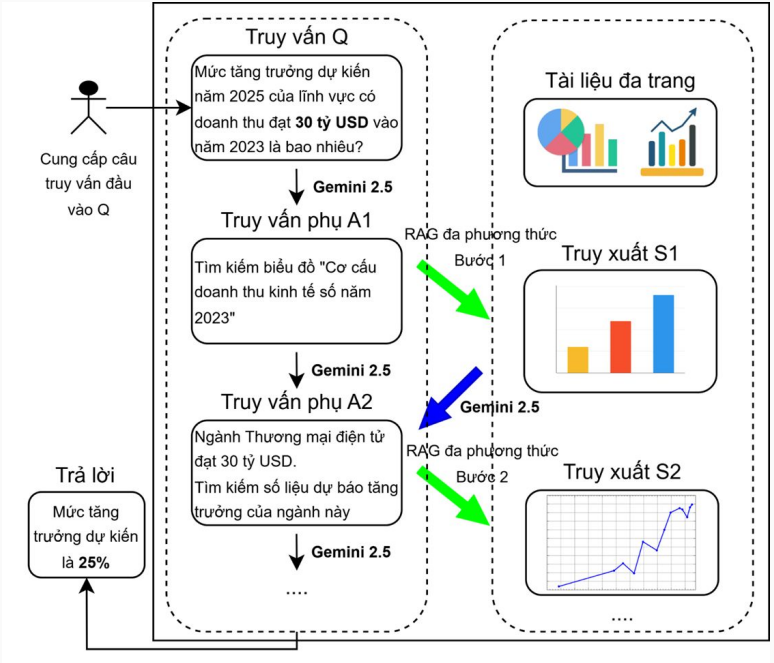
- **Nội dung 1: Cơ sở lý thuyết - Tối ưu hoá thông tin (Information Gain)**
  - **Giải pháp:** Sử dụng hàm mất mát **InfoNCE** để tối ưu hóa **Độ lợi thông tin**
  - *Ý tưởng:* Coi trang tài liệu chứa câu trả lời là "mẫu dương", các trang khác là "mẫu âm". Hệ thống học cách tối đa hóa khoảng cách giữa mẫu dương và mẫu âm trong không gian vector.
  - → *Lý do: InfoNCE* xếp hạng các trang tài liệu không chỉ dựa trên từ khóa mà dựa trên độ đóng góp thông tin (Mutual Information). Trong khi *Cosine Similarity* chỉ đo độ giống nhau về mặt từ ngữ, một trang tài liệu có thể chứa từ khóa giống câu hỏi nhưng nội dung lại sai lệch.
- **Nội dung 2: Xây dựng hệ thống:** Chia thành 3 module chính độc lập để dễ tối ưu hoá:
  - **Module 1: Generator (LLM Agent):** Phân tích câu hỏi phức tạp thành các truy vấn đơn giản.
    - *Công cụ:* **OpenAI GPT-4o** (hoặc Qwen-3 cho open-source).
    - → *Lý do chọn:* Cần khả năng *Instruction Following* mạnh mẽ để tuân thủ quy trình suy luận.
  - **Module 2: Retriever:** Tìm kiếm trang tài liệu dựa trên ngữ nghĩa bằng model embedding
    - *Công cụ:* **SigLIP** (Sigmoid Loss for Language Image Pre-Training).
    - → *Lý do: Sử dụng SigLIP* vì đây là mô hình state-of-the-art của multimodel embeddings
  - **Module 3: Judge:** Đánh giá mức độ đủ của thông tin.

Nội dung và Phương pháp

- **Ví dụ:** *Câu hỏi Q:* "Doanh thu mảng X tại Bắc Mỹ (Trang 5) chiếm bao nhiêu % so với tổng toàn cầu (Trang 10)?"
- **Algorithm Design:** Quy trình xử lý lặp
  - Bước 1: Nhận câu hỏi Q, sinh ra sub-query (VD: *Tìm biểu đồ doanh thu mảng X tại Bắc Mỹ")*
  - Bước 2: Sử dụng SigLiP để tìm ra thông tin ban đầu liên quan đến sub-query từ tập **D,** thêm vào tập **S**
  - Bước 3: LLM đánh giá phần thông tin này, đưa ra **information gain** nếu nó vượt qua ngưỡng có thể giữ lại, ngược lại bỏ qua thông tin này để tránh việc bị nhét thông tin không cần thiết. *(VD: Thấy Trang 5 có số liệu Bắc Mỹ, nhưng thiếu số liệu Toàn cầu* → thông tin chưa đủ*)*
  - Bước 4: Sinh câu hỏi phụ (sub-query) tiếp theo dựa trên phần thiếu *(VD: Tìm bảng số liệu tổng doanh thu toàn cầu)*
  - Bước 5: Dùng sub-query tiếp theo này để tìm kiếm lại chính xác hơn, cập nhật tập **S** mới và LLM tiếp tục đánh giá.
  - Bước 6: Cập nhật thông tin và lặp lại B3 cho đến khi đủ tin hoặc đạt giới hạn lặp tối đa.

Nội dung và Phương pháp

- **Nội dung 3: Thực nghiệm kiểm chứng**
  - **Dataset:** MP-DocVQA (Tài liệu nhiều trang) và MMVQA.
  - **Kịch bản A/B Testing:** So sánh *Proposed Framework* với *Standard RAG*.
- **Thách thức, khó khăn & Giải pháp:**
  - **Vòng lặp vô hạn:** LLM liên tục cảm thấy "chưa đủ tin" và tìm kiếm mãi mãi → *Giải pháp:* Thiết lập ngưỡng bão hòa Entropy. Nếu sau 3 bước mà lượng thông tin mới (Information Gain) gần bằng 0, hệ thống buộc phải dừng và trả lời tốt nhất có thể.
  - **Chi phí & Độ trễ:** Gọi LLM nhiều lần làm chậm hệ thống → *Giải pháp:* Cắt tỉa không gian tìm kiếm (Pruning). Ở vòng 1, loại bỏ ngay lập tức 80% các trang không liên quan bằng vector search nhẹ, chỉ dùng LLM để đọc sâu 20% trang còn lại.



Kết quả dự kiến

- Báo cáo kỹ thuật: So sánh độ chính xác (Accuracy/F1) giữa phương pháp đề xuất và Baseline (RAG truyền thống).
- Phân tích lỗi: Đánh giá các trường hợp thất bại để hiểu rõ giới hạn của phương pháp xấp xỉ thông tin.
- Mức độ khả thi: Dựa trên framework có sẵn, tập trung vào tinh chỉnh logic.
- Đóng góp của đề tài:
  - Đề xuất được **công thức toán học** áp dụng InfoNCE vào bài toán truy xuất tài liệu (chuyển từ Semantic Search sang Information Optimization).
  - Chứng minh được hiệu quả của cơ chế lặp so với cơ chế một lượt trên dữ liệu phức tạp.

Tài liệu tham khảo

- [1]. Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao, "ReAct: Synergizing reasoning and acting in language models," ICLR, 2023.
- [2] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny, "Hierarchical multimodal transformers for Multi-Page DocVQA," *Pattern Recognition*, vol. 144, p. 109834, 2023.
- [3] Yihao Ding, Kaixuan Ren, Jiabin Huang, Siwen Luo, and Soyeon Caren Han, "MMVQA: A Comprehensive Dataset for Investigating Multipage Multimodal Information Retrieval in PDF-based Visual Question Answering," IJCAI, 2024
- [4] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer, "Sigmoid Loss for Language Image Pre-Training," ICCV, 2023
- [5] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” NeurIPS, 2020