THÔNG TIN CHUNG CỦA NHÓM

- Link YouTube video của báo cáo (tối đa 5 phút): *https://youtu.be/nFJngeoIEEk*
- Link slides (dạng .pdf đặt trên Github của nhóm): https://github.com/anah19uit/CS2205.FEB2025/blob/main/%C3%82n%20%C3%82u%20H%E1%BB%93ng%20-%20CS2205.FEB2025.DeCuong.FinalReport.Template.Slide.pdf
- *Mỗi thành viên của nhóm điền thông tin vào một dòng theo mẫu bên dưới*
- *Sau đó điền vào Đề cương nghiên cứu (tối đa 5 trang), rồi chọn Turn in*
- *Lớp Cao học, mỗi nhóm một thành viên*

- Họ và Tên: Âu Hồng Ân
- MSSV: 240101033

- Lớp: CS2205.FEB2025
- Tự đánh giá (điểm tổng kết môn): 7/10
- Số buổi vắng: 0
- Số câu hỏi QT cá nhân: 4
- Số câu hỏi QT của cả nhóm: 0
- Link Github: https://github.com/anah19uit/CS2205.FEB2025

ĐỀ CƯƠNG NGHIÊN CỨU

TÊN ĐỀ TÀI (IN HOA)

SINH VĂN BẢN TĂNG CƯỜNG TRUY XUẤT CÓ NHẬN THỨC ĐỒ THỊ: KHAI THÁC ĐỒ THỊ TRI THỨC VÀ ĐÁNH GIÁ DỰA TRÊN MÔ HÌNH NGÔN NGỮ LỚN

TÊN ĐỀ TÀI TIẾNG ANH (IN HOA)

GRAPH-AWARE RETRIEVAL-AUGMENTED GENERATION: LEVERAGING KNOWLEDGE GRAPHS AND LLM-BASED EVALUATION

TÓM TẮT *(Tối đa 400 từ)*

RAG (Retrieval-Augmented Generation) là một phương pháp kết hợp giữa truy xuất thông tin (retrieval) và mô hình sinh ngôn ngữ (generation) để cải thiện khả năng trả lời câu hỏi hoặc sinh văn bản dựa trên kiến thức ngoài dữ liệu huấn luyện của mô hình. Tuy nhiên, các phương pháp như RAG cơ bản (naive RAG)[1] và các mô hình ngôn ngữ lớn (LLMs) sẽ gặp các thách thức về truy xuất thông tin, khó khăn tổng quát hóa, khó khăn tăng cường các phản hồi dễ lặp lại. Để khắc phục các hạn chế, chúng tôi tập trung vào cải tiến chính: RAG kết hợp đồ thị tri thức [2]. Việc tích hợp đồ thị tri thức vào RAG giúp cải thiện đáng kể khả năng suy luận. Khác với naive RAG có thể bị kẹt trong tối ưu cục bộ, cách tiếp cận của chúng tôi sử dụng [2] để hướng đến tối ưu toàn cục và đồng thời mô hình hóa rõ ràng các mối quan hệ giữa các thực thể. Phương pháp này cũng giảm hiện tượng ảo giác nhờ vào việc liên kết chặt chẽ câu trả lời với đồ thị tri thức.

Những tiến bộ này cho thấy tầm quan trọng của việc tích hợp tri thức có cấu trúc trong quá trình hỏi đáp. Hướng phát triển tiếp theo có thể tập trung vào cập nhật và mở rộng đồ thị tri thức, cũng như tối ưu tốc độ suy luận để phục vụ các ứng dụng thời gian thực. Bằng cách kết hợp các kỹ thuật trên, giải pháp hỏi đáp có thể đáp ứng tốt hơn các yêu cầu đến từ nhiều lĩnh vực khác nhau.

GIỚI THIỆU *(Tối đa 1 trang A4)*

Trong bối cảnh phát triển mạnh mẽ của các LLMs và phương pháp RAG, việc nâng cao khả năng trả lời câu hỏi và sinh văn bản dựa trên tri thức bên ngoài dữ liệu huấn luyện đang là một thách thức quan trọng. Mặc dù naive RAG [1] đã cải thiện đáng kể khả năng truy xuất thông tin, các hệ thống này vẫn gặp phải nhiều hạn chế như truy xuất không chính xác, khó khăn trong tổng quát hóa, và hiện tượng lặp lại phản hồi. Đặc biệt, naive RAG thường bị tối ưu cục bộ và thiếu khả năng suy luận dựa trên mối quan hệ giữa các thực thể, dẫn đến câu trả lời kém chính xác hoặc mắc lỗi "ảo giác" (hallucination).

Để giải quyết việc này chúng tôi sử dụng GraphRAG[2] là phương pháp mới nổi bật nhờ khả năng tổng hợp ý nghĩa toàn cục (global sensemaking) trên toàn bộ dữ liệu và sử dụng phương pháp xây dựng đồ thị tri thức sử dụng GPT4. Nghiên cứu sử dụng kỹ thuật LLM-as-a-judge [3], trong đó nhóm nghiên cứu này tạo ra một LLM để đưa ra tập hợp đa dạng các câu hỏi để có thể tạo ra một global sensemaking dựa trên các trường hợp của ngữ liệu, trước khi dùng LLM thứ hai để đánh giá giữa hai hệ thống RAG. Kết quả thử nghiệm trên của nhóm nghiên cứu cho thấy phương pháp này của vượt trội hơn hẳn vector RAG khi dùng GPT4, đặc biệt với các câu hỏi rộng, không có đáp án cố định.

Input: Một yêu cầu từ người dùng bằng text, một đồ thị tri thức.

Output: Một đoạn text đáp ứng được yêu cầu.

MỤC TIÊU *(Viết trong vòng 3 mục tiêu)*

1. Phản hồi từ LLMs đáp ứng được yêu cầu từ người dùng.
2. Trong tương lai xây dựng đồ thị tri thức không cần sử dụng GPT4 như trong [2]. Sử dụng phương pháp xây dựng đồ thị tri thức của [4] và có khả năng mở rộng đồ thị từ phản hồi người dùng.
3. Trong một số văn bản có kèm ảnh, thực tế thì có thể tách ảnh từ văn bản và nó liên quan đến nội dung liền kề với các đoạn văn xung quanh bức ảnh đó, nhưng

trong một số trường hợp thì **không**. Do đó sẽ có một bài toán là “xác định được mức độ liên quan giữa văn bản và ảnh”.

NỘI DUNG VÀ PHƯƠNG PHÁP

Để tiết kiệm chi phí chúng tôi xây dựng đồ thị tri thức và phân cụm đồ thị bằng kỹ thuật tính toán trước (precomputing technique) sử dụng GPT4 tương ứng với tài liệu. Để không cần phải thực hiện lại việc xây dựng đồ thị mỗi lần nhận được yêu cầu từ người dùng. Thực hiện 6 bước sau:

1. **Chia các tài liệu thành đoạn văn nhỏ**
    - Chia tài liệu thành các đoạn nhỏ hơn
    - Kích thước đoạn ảnh hưởng đến chi phí (ít lần gọi LLM hơn với đoạn lớn) nhưng có thể giảm khả năng nhớ lại (recall) thông tin ở phần đầu đoạn.
2. **Trích xuất thực thể & mối quan hệ**
    - LLM trích xuất các thực thể (entities, ví dụ: công ty, người), mối quan hệ (relationships) và các khẳng định (claims - tuyên bố thực tế) từ mỗi đoạn.
    - Mô tả được tạo cho từng thực thể/mối quan hệ (ví dụ: "NeoChip là công ty chuyên về chip bán dẫn công suất thấp").
    - Các prompt (lệnh) giúp cải thiện độ chính xác khi trích xuất.
3. **Xây dựng đồ thị tri thức**
    - Các thực thể trở thành nút (nodes), mối quan hệ trở thành cạnh (edges), và claims.
    - Các thực thể trùng lặp được hợp nhất
    - Trọng số cạnh (edge weights) phản ánh tần suất xuất hiện của mối quan hệ.
4. **Phân cụm đồ thị từ đồ thị tri thức**
    - Các cộng đồng (communities - cụm các nút liên quan) được phát hiện theo cấu trúc phân cấp sử dụng thuật toán Leiden [5].

- Cho phép tóm tắt theo phương pháp chia để trị (divide-and-conquer) ở nhiều mức độ chi tiết.

5. **Tóm tắt các communities từ phân cụm đồ thị**
    - Mỗi cộng đồng được tóm tắt bằng cách ưu tiên các nút/cạnh có độ quan trọng cao (high-degree) và các claims.
    - Leaf-level communities (Cộng đồng lá): Tóm tắt từng thành phần riêng lẻ.
    - Higher-level communities (Cộng đồng cấp cao): Kết hợp các bản tóm tắt từ cộng đồng con để phù hợp giới hạn token.
6. **Dựa vào các tóm của communities để trả lời câu hỏi từ người dùng.**
    - Xử lý truy vấn cho các bản tóm tắt communities và chia thành các đoạn, sau đó dùng một LLM khác trả lời các câu hỏi trung gian và được đánh giá điểm số từ 0-100.
    - Tổng hợp các câu trả lời có điểm cao nhất được kết hợp thành câu trả lời cuối cùng.

KẾT QUẢ MONG ĐỢI

Phương pháp này được đánh giá vượt trội hơn so với RAG Naive dựa trên hai tiêu chí chính: độ bao quát thông tin và tính đa dạng của câu trả lời. Độ bao quát được đo lường bằng mức độ đầy đủ và toàn diện của nội dung được phản hồi so với yêu cầu từ người dùng, trong khi tính đa dạng phản ánh sự khác biệt về thông tin và cấu trúc giữa các câu trả lời được tạo ra.

Ngoài ra, hiệu quả sử dụng tài nguyên cũng là một tiêu chí quan trọng, được đánh giá thông qua số lượng token tiêu thụ trong quá trình tạo câu trả lời. Phương pháp này cho thấy khả năng tối ưu hóa đáng kể ở khía cạnh này khi áp dụng các cấp độ tóm tắt cộng đồng, đặc biệt là cấp độ tóm tắt khái quát.

Phương pháp này còn được đánh giá về khả năng trích xuất thông tin, dựa trên số lượng và mức độ liên quan của các thông tin được truy xuất từ nguồn dữ liệu đầu vào. Sự phù hợp giữa cấu trúc tóm tắt và loại dữ liệu cũng là yếu tố được phân tích,

phản ánh qua khả năng điều chỉnh chiến lược tóm tắt theo đặc thù nội dung.
Cuối cùng, mức độ hỗ trợ người dùng được đánh giá thông qua khả năng cung cấp ví dụ, trích dẫn chính xác, và trải nghiệm sử dụng tổng thể – những yếu tố có thể bị ảnh hưởng bởi chất lượng của các prompt trích xuất thông tin.

TÀI LIỆU THAM KHẢO *(Định dạng DBLP)*

[1] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, Haofen Wang:
Retrieval-Augmented Generation for Large Language Models: A Survey. CoRR abs/2312.10997 (2023)

[2] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Jonathan Larson:
From Local to Global: A Graph RAG Approach to Query-Focused Summarization. CoRR abs/2404.16130 (2024)

[3] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Zhuang Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, et al.:
Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. NeurIPS 36 (2024)

[4] Marvin Hofer, Daniel Obraczka, Alieh Saeedi, Hanna Köpcke, Erhard Rahm:
Construction of Knowledge Graphs: State and Challenges. CoRR abs/2206.09508 (2022)

[5] Vincent A. Traag, Ludo Waltman, Nees Jan van Eck:
From Louvain to Leiden: Guaranteeing Well-Connected Communities. CoRR abs/1810.08473 (2018)