TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ CƯƠNG KHOÁ LUẬN TỐT NGHIỆP

TÍCH HỢP KĨ THUẬT TRUY XUẤT THÔNG TIN VÀ MÔ HÌNH NGÔN NGỮ ĐỂ TẠO CHATBOT TRA CỨU VĂN BẢN TỰ ĐỘNG NỘI BỘ

Applying retrieval-augmented generation techniques in the development of a chatbot supports internal knowledge management.

1 THÔNG TIN CHUNG

Người hướng dẫn:

– TS. Phạm Nguyễn Cương (Khoa Công nghệ Thông tin)

Nhóm sinh viên thực hiện:

1. Mạch Vĩ Kiệt (MSSV: 21127634)
2. Nguyễn Duy Đăng Khoa (MSSV: 21127078)

Loại đề tài: Ứng dụng

Thời gian thực hiện: Từ *9/2024* đến *3/2025*

2 NỘI DUNG THỰC HIỆN

2.1 Giới thiệu về đề tài

Từ khi mới ra mắt, những mô hình LLM (Large Language Model) như ChatGPT của OpenAI đã tạo ra cơn sốt trên toàn cầu. Những mô hình ngôn ngữ lớn này có khả năng hiểu rõ văn bản và phản hồi lại người dùng bằng ngôn ngữ tự nhiên. Tuy nhiên, thời điểm đó mô hình này chỉ được cập nhật dữ liệu đến năm 2021. Hiện nay, với sự cạnh tranh khóc liệt khi có sự tham gia của các nhà cung cấp khác như Google, Meta với 2 mô hình ngôn ngữ lớn là Gemini và Llama, OpenAI đã cho ra mắt bản cập nhật phiên bản Plus với dữ liệu được huấn luyện nâng cấp. Tuy nhiên, tất cả mô hình hiện nay đều cho thấy khuyết điểm rất lớn đó là:

- Dữ liệu dễ dàng bị lỗi thời nếu không cập nhật thường xuyên.
- Do được huấn luyện trên tập dữ liệu tổng quát do đó rất khó để sử dụng cho doanh nghiệp với các ngành nghề đặc thù và nghiệp vụ thông tin được thay đổi liên tục.
- Chi phí re-train (tái huấn luyện) một mô hình hiện nay tốn rất nhiều chi phí và đòi hỏi trình độ chuyên môn cao.

Để giải quyết vấn đề này, một kiến trúc triển khai mới đã được xây dựng để khai thác khả năng của LLM đó chính là kiến trúc RAG (Retrieval-Augmented-Generation). RAG cố gắng giải quyết vấn đề dữ liệu của LLM bằng cách kết nối với kho dữ liệu của người dùng để cung cấp ngữ cảnh cho LLM để trả lời các câu hỏi từ người dùng một cách chính xác. Qua các năm, có vô số cải tiến được đề ra cho kiến trúc này, đa số đều tập trung vào 3 bước cơ bản trong một quy trình RAG: xây dựng chỉ mục (Indexing), tìm kiếm (Retrieval) và tạo sinh câu trả lời (Generation). [1]

Trong một hệ thống RAG cơ bản nhất, 3 bước trên sẽ được thực hiện:

- **Indexing**: Các tập tin có trong cơ sở dữ liệu của người dùng sẽ được làm sạch và chuyển thể thành văn bản thuần. Các văn bản này thường sẽ thông qua một bước phân cắt thành các đoạn nhỏ hơn, sau đó tất cả các đoạn sẽ được nhúng thành các vector biểu diễn (sử dụng một mô hình embedding). Dữ liệu này sẽ được đưa vào một kho dữ liệu vector và được tạo chỉ mục, sẵn sàng để thực hiện tìm kiếm.
- **Retrieval**: Để thực hiện tìm kiếm trong kho dữ liệu, câu hỏi (query) của người dùng sẽ được nhúng bằng mô hình embedding (Mô hình này phải là mô hình đã được sử dụng trong bước Indexing). Vector biểu diễn của query sẽ được dùng để tìm kiếm các văn bản tương đồng nhất trong kho dữ liệu. Những văn bản này sẽ được dùng làm ngữ cảnh để cung cấp cho mô hình LLM.
- **Generation**: Với ngữ cảnh được cung cấp, ta sẽ kết hợp với một số hướng dẫn để yêu cầu mô hình LLM tạo sinh ra một câu trả lời cho câu hỏi của người dùng.

Ở giai đoạn Indexing, qua thời gian các hướng tiếp cận đã thay đổi nhiều như từ token đơn giản [2] đến trích xuất entity [3] hoặc chia văn bản thành khối (chunks) [4]. Việc lựa chọn phương thức nào phù hợp là một sự cân bằng giữa hiệu quả, chi phí và thời gian.

Giai đoạn Retrieval có thể được thực hiện 1 hoặc qua nhiều lần lặp (iterations) để cải thiện kết quả tìm kiếm, hoặc thực hiện tìm kiếm tổng hợp nhiều bước.

Để đạt được hiệu quả cao ở giai đoạn Generation, việc tạo ra một prompt hướng dẫn hiệu quả là cực kì cần thiết và cần trải qua một quá trình Prompt Engineering để có thể đảm bảo mô hình LLM giảm thiểu ảo giác (hallucination), trả lời đúng format và yêu cầu riêng của người dùng/domain dữ liệu.

Đa số các cải tiến được đưa ra đều tập trung vào việc tăng hiệu quả/độ chính xác của quá trình, trông cậy vào các mô hình LLM và embedding để hiểu và biểu diễn chính xác thông tin được cung cấp. Tuy nhiên, theo [5], đối với các domain

doanh nghiệp cụ thể trong thực tế (tư vấn tài chính,...), các dữ liệu này là riêng tư và không thể được dùng làm dữ liệu huấn luyện cho LLM. Do vậy, các mô hình LLM không dễ dàng nắm bắt được ý nghĩa của từng ngữ cảnh chúng ta cung cấp nếu không trải qua quá trình finetuning cực kì tốn kém. Vì thế, ta cần thiết kế 1 hệ thống RAG có thể tận dụng tốt được kiến thức domain, hay nói cách khác là semantic layer của doanh nghiệp. Hệ thống này cần phải được xây dựng với mục đích có thể tích hợp trơn tru với nhiều domain, sử dụng kiến thức của các chuyên gia trong doanh nghiệp.

Chúng tôi đề xuất một thiết kế hệ thống RAG với trọng tâm đặt vào việc tích hợp với domain dữ liệu một cách hiệu quả và nhanh nhẹn. Hệ thống được xây dựng với mục đích cho phép các chuyên gia đóng góp vào hiệu quả của hệ thống thông qua các kiến thức chuyên sâu của họ và lớp metadata trong kho dữ liệu. Hệ thống sẽ đảm bảo khả năng mở rộng, triển khai dễ dàng (scalable) và khả năng cải thiện thông qua feedback thực tế.

Bằng phương pháp này, chúng tôi mong tạo được hướng giải quyết mới cho vấn đề domain trong các hệ thống RAG. Hướng giải quyết này giúp doanh nghiệp tránh khỏi việc finetuning tốn kém, và có được một hệ thống tiết kiệm chi phí và hiệu quả hơn.

2.2 Mục tiêu đề tài

Việc tra cứu thông tin nội bộ trong các tổ chức, doanh nghiệp hay trường học thường mất nhiều thời gian vì người dùng phải tìm kiếm thủ công qua hệ thống lưu trữ hoặc nắm rõ cấu trúc tài liệu. Điều này gây lãng phí tài nguyên và giảm hiệu quả làm việc. Tích hợp kỹ thuật truy xuất thông tin và mô hình ngôn ngữ tiên tiến sẽ tự động hóa quy trình, giúp người dùng nhận được câu trả lời chính xác từ tài liệu mà không cần tìm kiếm thủ công.

Đề tài mang đến tính ứng dụng rất cao khi chúng sẽ mang lại hệ thống hỏi đáp tự động có khả năng tự động tra cứu chính xác và dễ dàng, Tiết kiệm thời gian

và tăng hiệu quả làm việc, Dễ dàng tích hợp với nhiều nguồn kho dữ liệu nội bộ cần sự bảo mật.

Kết quả: Đề tài đem đến giải pháp tối ưu hóa quy trình tra cứu thông tin nội bộ giúp nâng cao hiệu quả làm việc và giảm thiểu thời gian tìm kiếm. Hệ thống này cũng đóng góp vào nghiên cứu AI và NLP, mở ra tiềm năng ứng dụng rộng rãi trong các lĩnh vực khác nhau như quản lý thông tin, chăm sóc khách hàng, và giáo dục. Việc đánh giá của hệ thống cũng sẽ góp phần xem xét hiệu quả của phương pháp đối với các cải tiến khác trong thực tế.

2.3 Phạm vi của đề tài

Trong phạm vi nghiên cứu chính của luận văn, chúng tôi sẽ xây dựng một hệ thống chatbot thông minh hỗ trợ sinh viên hỏi đáp các vấn đề hành chính nội bộ của trường đại học Khoa Học Tự Nhiên - ĐHQG TPHCM.

Đối tượng nghiên cứu chính: Sinh viên, Giáo vụ, các câu hỏi thường gặp liên quan đến vấn đề như lịch học, nội quy, thông tin sự kiện trường học, ...

Tập dữ liệu bao gồm: Các trang tin tức của trường, khoa công nghệ thông tin, các tài liệu được cung cấp bởi giáo vụ, sổ tay sinh viên, ... Những nguồn dữ liệu này nằm ở nhiều nơi và dưới nhiều dạng khác nhau như PDF, DOCX, HTML,...

Ràng buộc đề tài: Đề tài sẽ tập trung chính vào các câu hỏi mang tính chất tra cứu văn bản trong khuôn khổ nghiệp vụ trường học, nội quy, công văn và các chủ đề được xác định cụ thể trước đó. Qua đó đánh giá hiệu quả của hệ thống trong nghiệp vụ này.

2.4 Cách tiếp cận dự kiến

Đã có rất nhiều nhóm nghiên cứu tiếp cận việc xây dựng các hệ thống RAG với nhiều thay đổi khác nhau, một số cũng đã tập trung vào việc làm giàu lớp ngữ nghĩa metadata của dữ liệu [6][7]. Cách chúng tôi tiếp cận đề tài cũng sẽ tương tự như cách phổ biến và nhìn chung sẽ qua các bước sau:

- **Chuẩn bị dữ liệu:** Đối với những đề tài này, việc phân tích và làm sạch dữ liệu trong domain được chọn là cực kì cần thiết. Thông qua đó, ta sẽ định nghĩa được kho dữ liệu, các trường cần thiết và cách tổ chức dữ liệu sao cho hiệu quả và hợp lý nhất đối với domain đã cho. Đối với đề tài này, bước phân tích domain là quan trọng để khai thác và chứng minh được giá trị của kiến thức chuyên gia đối với một hệ thống RAG.
- **Đánh giá và lựa chọn kiến trúc RAG:** Có rất nhiều cải tiến và cài đặt ta có thể thêm vào kiến trúc RAG. Những thay đổi này có thể ở cả ba bước *Indexing, Retrieval và Generation*[1]. Do hạn chế về mặt mô hình, đa số cải tiến chúng tôi thực hiện khả năng cao sẽ nằm ở các bước Indexing và Retrieval.
- **Prompt Engineering:** Để đạt được chất lượng tốt nhất ở câu trả lời của mô hình, quá trình Prompt Engineering là cực kì cần thiết. Lựa chọn prompt đúng đắn cần có quá trình thử nghiệm và đánh giá cẩn thận.[8]
- **Xây dựng hệ thống:** Bắt đầu xây dựng hệ thống cũng như quá trình đánh giá. Đối với đề tài này, chúng tôi sẽ xây dựng hệ thống theo một cách không ràng buộc vào domain dữ liệu và tạo cơ hội để các chuyên gia sẽ đóng góp được vào hiệu năng của hệ thống.
- **Tổng hợp và báo cáo kết quả**

Kết quả cuối cùng là một hệ thống hoàn chỉnh hoạt động theo đầy đủ các bước của kiến trúc RAG đề ra và có thể tương tác với người dùng. Ngoài ra, nhiều nghiên cứu còn xây dựng hệ thống đánh giá riêng để nhận xét hiệu quả và độ chính xác của các bước trong kiến trúc RAG đã xây dựng. [5][9]

2.5 Kết quả dự kiến của đề tài

2.5.1 Số liệu định lượng

Để đánh giá một hệ thống RAG, chúng tôi sẽ đánh giá về cả 2 mặt: triển khai và hiệu quả của RAG.

- Hiệu quả RAG: sử dụng những thang đo như Accuracy, Recall để đánh giá độ chính xác chung của câu trả lời.
- Triển khai: Đánh giá về khả năng thực tế của hệ thống như: chi phí, tốc độ, khả năng mở rộng (scalability).

2.5.2 Sản phẩm đầu ra

Sản phẩm cuối cùng của đề tài là một hệ thống được xây dựng đầy đủ, với UI người dùng là website và backend liên lạc với nhau qua các giao thức API. Hệ thống sẽ tự vận hành, cập nhật liên tục và có bảo mật cơ bản.

2.6 Kế hoạch thực hiện

Thời gian	**Nội dung**	**Phân công & Tiến độ công việc**
Tuần 1-2 *01/09/24 - 14/09/24*	1. Xác định domain sẽ thực hiện & các nguồn dữ liệu có thể có. 2. Tìm hiểu thêm về domain đã chọn. 3. Thực hiện thu thập dữ liệu.	**Đã hoàn thành** *21127078*: Phần 1, 2, 3 *21127634*: Phần 1, 2
Tuần 3-4-5 *15/09/24 - 05/10/24*	1. Phân tích & làm sạch dữ liệu. 2. Tìm hiểu và đánh giá các công nghệ sẽ sử dụng (Frontend/Vector Database/LLM/...). 3. Định nghĩa kho dữ liệu và cách tổ chức dữ liệu.	**Đã hoàn thành** *21127078*: Phần 1, 3 *21127634*: Phần 2, 3

Tuần 6-7 *06/10/24 - 19/10/24*	1. Phân tích & làm sạch dữ liệu. 2. Tìm hiểu và đánh giá các công nghệ sẽ sử dụng (Frontend/Vector Database/LLM/...). 3. Định nghĩa kho dữ liệu và cách tổ chức dữ liệu.	**Đã hoàn thành** *21127078*: Phần 1, 3 *21127634*: Phần 2, 3
Tuần 8-9 *20/10/24 - 02/11/24*	1. Xác định rõ ràng kiến trúc của hệ thống RAG (sử dụng thủ thuật nào, cải tiến ở đâu...) 2. Thực hiện quá trình Prompt Engineering. 3. Lựa chọn kho dữ liệu vector và tiến hành ingest dữ liệu. 4. Nghiên cứu các bài báo khoa học liên quan. 5. Xác định user và các flow chính của ứng dụng.	**Đã hoàn thành** *21127078*: Phần 1, 3 *21127634*: Phần 2, 3, 4

Tuần 10-11-12-13 *03/11/24 - 30/11/24*	1. Xây dựng hoàn chỉnh version đầu tiên của hệ thống. 2. Hoàn thành và nộp đề cương khóa luận (21/11/24). 3. Nghiên cứu các phương án triển khai cho hệ thống. (deployment).	**Đang thực hiện** *21127078*: Phần 1, 2 *21127634*: Phần 1, 2, 3
Tuần 13-14 *01/12/24 - 14/12/24*	1. Xác định và lựa chọn quy trình đánh giá hệ thống. 2. Chuẩn bị hệ thống sẵn sàng cho việc triển khai. 3. Nghiên cứu những người dùng thử và khả năng triển khai cho Giáo vụ Trường.	**Chưa thực hiện** *21127078*: Phần 1, 2, 3 *21127634*: Phần 1, 2, 3
Tuần 15-16-17 *15/12/24 - 04/01/25*	1. Thực hiện những cải tiến tồn đọng (backlog) (nếu có). 2. Triển khai hệ thống và thu thập feedback.	**Chưa thực hiện** *21127078*: Phần 1, 2, 3 *21127634*: Phần 1, 2, 3 **Ghi chú:** Do quá trình triển khai có thể gặp khó khăn về chi phí và thủ tục, thời gian thực tế triển khai có thể khác với dự kiến.

Tuần 18-19 *05/01/25 - 18/01/25*	1. Đánh giá hệ thống RAG về mặt hiệu quả. 2. Đánh giá hệ thống RAG về mặt triển khai. 3. Tổng hợp các nghiên cứu, references để chuẩn bị viết luận văn tốt nghiệp.	**Chưa thực hiện** *21127078*: Phần 1, 3 *21127634*: Phần 2, 3
Tuần 20-21 *19/01/25 - 01/02/25*	1. Tổng hợp và báo cáo kết quả. 2. Bắt đầu viết luận văn.	**Chưa thực hiện** *21127078*: Phần 1, 2 *21127634*: Phần 1, 2
Tuần 22-23-24 *02/02/25 - 22/02/25*	1. Tiếp tục hoàn thành luận văn. 2. Sửa lỗi/cải tiến cập nhật hệ thống (nếu cần).	**Chưa thực hiện** *21127078*: Phần 1, 2 *21127634*: Phần 1, 2
Tuần 25-26 *23/02/25 - 08/03/25*	1. Tiếp tục hoàn thành luận văn. 2. Sửa lỗi/cải tiến cập nhật hệ thống (nếu cần).	**Chưa thực hiện** *21127078*: Phần 1, 2 *21127634*: Phần 1, 2
Tuần 27-28 *09/03/25 - 22/03/25*	1. Hoàn thành luận văn. 2. Thực hiện thủ tục cho ngày bảo vệ đề tài.	**Chưa thực hiện** *21127078*: Phần 1, 2 *21127634*: Phần 1, 2
Tuần 29-END *23/03/25 - 04/25*	**Nghiệm thu, thực hiện phản biện và bảo vệ đề tài.**	**Chưa thực hiện**

Tài liệu

[1] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2024.

[2] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," 2020.

[3] S. Nishikawa, R. Ri, I. Yamada, Y. Tsuruoka, and I. Echizen, "Ease: Entity-aware contrastive learning of sentence embedding," 2022.

[4] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, "In-context retrieval-augmented language models," 2023.

[5] S. Wang, J. Liu, S. Song, J. Cheng, Y. Fu, P. Guo, K. Fang, Y. Zhu, and Z. Dou, "Domainrag: A chinese benchmark for evaluating domain-specific retrieval-augmented generation," 2024.

[6] M. Poliakov and N. Shvai, "Multi-meta-rag: Improving rag for multi-hop queries using database filtering with llm-extracted metadata," 2024.

[7] L. Mombaerts, T. Ding, A. Banerjee, F. Felice, J. Taws, and T. Borogovac, "Meta knowledge for retrieval augmented large language models," 2024.

[8] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," 2024.

[9] A. Hikov and L. Murphy, "Information retrieval from textual data: Harnessing large language models, retrieval augmented generation and prompt engineering," *Journal of AI, Robotics & Workplace Automation*, vol. 3, no. 2, pp. 142–150, 2024.

[10] A. Einstein, "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]," *Annalen der Physik*, vol. 322, no. 10, pp. 891–921, 1905.

[11] M. Goossens, F. Mittelbach, and A. Samarin, *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

[12] D. Knuth, "Knuth: Computers and typesetting." `https://www-cs-faculty.stanford.edu/~knuth/abcde.html`.

[13] Overleaf, "Learn LaTeX in 30 minutes." `https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`.

[14] S. Simon, A. Mailach, J. Dorn, and N. Siegmund, "A methodology for evaluating rag systems: A case study on configuration dependency validation," 2024.

XÁC NHẬN
CỦA NGƯỜI HƯỚNG DẪN
(Ký và ghi rõ họ tên)

TP. Hồ Chí Minh, ngày... tháng... năm...
NHÓM SINH VIÊN THỰC HIỆN
(Ký và ghi rõ họ tên)