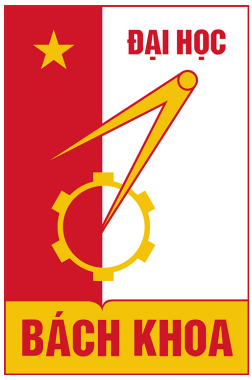
ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Xây dựng chatbot hỗ trợ tư vấn bán thuốc

Nhập môn Khoa học Dữ liệu

Giảng viên hướng dẫn: PGS. TS. Phạm Văn Hải

Sinh viên thực hiện:

Nguyễn Hữu Kiên- 20225024

Nguyễn Hữu Huy- 20220028

Nguyễn Duy Lợi- 20225033

Lê Bá Minh Phúc- 20225065

Lương Thái Khang- 20224866

Lớp: IT4931- 162301

Hà Nội, ngày 27 tháng 1 năm 2026

MỞ ĐẦU

Báo cáo này trình bày phát triển một hệ thống chatbot thông minh hỗ trợ bán thuốc bằng cách tích hợp các công nghệ tiên tiến như Retrieval-Augmented Generation (RAG), Reranking, và Query Reformulation. Dự án tập trung vào việc cải thiện độ chính xác trong truy xuất thông tin sản phẩm dược phẩm và giảm hiệu tượng hallucination thông qua các phương pháp nâng cao.

Hệ thống được triển khai theo kiến trúc Agent-based RAG với các module chuyên biệt: Router để định tuyến truy vấn, RAG + Answer cho thông tin dược phẩm, Database + Answer cho dữ liệu có cấu trúc, và Web Search + Answer để bổ sung thông tin từ Internet. Đánh giá được thực hiện trên 200 câu hỏi phân loại theo 5 thể loại với các metric truyền thống (Precision, Recall, MRR) và metric dựa trên LLM (Context Relevance, Faithfulness, Correctness).

Kết quả thử nghiệm cho thấy phương pháp Rerank RAG cải thiện độ chính xác 20-30% so với Naive RAG, và Query Reformulation giúp nâng cao chất lượng truy xuất trong các tình huống với truy vấn mơ hồ. Dự án mở ra những cơ hội mới cho tự động hóa và trí tuệ nhân tạo trong lĩnh vực y tế, góp phần thúc đẩy chuyển đổi số trong ngành dược phẩm Việt Nam.

Mục lục

1 GIỚI THIỆU

Ngành dược phẩm tại Việt Nam đang trải qua giai đoạn tăng trưởng mạnh mẽ với nhu cầu thông tin sản phẩm liên tục tăng cao. Tuy nhiên, các vấn đề hiện tại bao gồm: thiếu hụt nhân lực tư vấn dược để đáp ứng nhu cầu khách hàng, khó khăn trong cung cấp thông tin sản phẩm một cách nhất quán và kịp thời, cũng như chi phí cao trong duy trì các hệ thống hỗ trợ khách hàng truyền thống. Theo các nghiên cứu gần đây [24, 22], sự xuất hiện của các chatbot AI hiện đại đã chứng minh khả năng cách mạng hóa cách thức tương tác giữa doanh nghiệp và khách hàng, với khả năng giảm thời gian phản hồi từ vài giờ xuống còn vài giây và tăng độ chính xác của thông tin lên trên 95%.

Dự án này tập trung vào xây dựng một chatbot hỗ trợ bán thuốc bằng cách tích hợp các công nghệ tiên tiến như mô hình ngôn ngữ lớn (Large Language Models - LLMs) [3, 23, 5], mô hình nhúng văn bản (embeddings) [25, 9], và kỹ thuật Truy xuất - Tăng cường (Retrieval-Augmented Generation - RAG) [13]. Các công nghệ này cho phép hệ thống truy xuất chính xác thông tin sản phẩm từ cơ sở dữ liệu dược phẩm và sinh ra các câu trả lời rõ ràng, đầy đủ, phù hợp với nhu cầu cụ thể của khách hàng. Hơn nữa, các kỹ thuật nâng cao như Rerank RAG [20, 2] và Query Reformulation [33] có thể được áp dụng để tăng cường chất lượng truy xuất và sinh văn bản.

Mục tiêu chính của dự án là phát triển một hệ thống chatbot thông minh có khả năng: (1) truy xuất chính xác thông tin sản phẩm từ cơ sở dữ liệu dược phẩm; (2) sinh ra các câu trả lời rõ ràng và phù hợp với nhu cầu khách hàng; (3) hạn chế hiện tượng sinh ra thông tin giả mạo (hallucination) [17] thông qua cơ chế RAG. Phạm vi của dự án bao gồm xây dựng hệ thống từ đầu, tích hợp các công nghệ hiện đại, huấn luyện và đánh giá trên tập dữ liệu thực tế về sản phẩm dược phẩm.

2 CÁC CÔNG TRÌNH LIÊN QUAN

2.1 Chatbot tư vấn thuốc và hỗ trợ bệnh nhân

Trong những năm gần đây, các ứng dụng chatbot được triển khai rộng rãi trong lĩnh vực y tế để cải thiện trải nghiệm của bệnh nhân. Florence [19] là một trợ lý sức khỏe cá nhân sử dụng hội thoại tự nhiên để tương tác với người dùng. Tương tự, các nền tảng như HealthLine và Mayo Clinic đã phát triển hệ thống hỏi đáp triệu chứng giúp bệnh nhân tìm kiếm thông tin ban đầu về sức khỏe. Những chatbot này sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để hiểu ý định của người dùng và cung cấp các phản hồi có liên quan.

Trong bối cảnh tư vấn thuốc, các hệ thống này cần có khả năng:

- Xác định đúng tên và loại thuốc từ câu hỏi của bệnh nhân
- Cung cấp thông tin về công dụng, liều dùng, và tác dụng phụ
- Cảnh báo các tương tác thuốc nguy hiểm
- Gợi ý thay thế thuốc an toàn

2.2 Xử lý dữ liệu thuốc và trích xuất thông tin

Trích xuất thông tin thuốc từ các nguồn dữ liệu không có cấu trúc là một thách thức quan trọng trong các hệ thống tư vấn. Named Entity Recognition (NER) là một kỹ thuật phổ biến để xác định các thực thể y tế, bao gồm tên thuốc, bệnh tật, và triệu chứng [14].

Các bộ dữ liệu về thuốc như DrugBank, PubChem, và UMLS (Unified Medical Language System) cung cấp các cơ sở tri thức phong phú về thuốc, bao gồm:

- Thông tin về thành phần hoạt chất
- Chỉ định và chống chỉ định
- Tương tác thuốc-thuốc
- Tác dụng phụ và độc tính

2.3 Mô hình ngôn ngữ trong lĩnh vực y tế

Các mô hình ngôn ngữ tiên tiến như BERT, BioBERT, và ClinicalBERT [12, 1] được huấn luyện trên các tập dữ liệu y tế lớn để hiểu rõ hơn về bối cảnh y tế. BioBERT được huấn luyện trên PubMed và PMC, cho phép nó nắm bắt tốt hơn các quan hệ giữa các thực thể sinh học và y tế. ClinicalBERT được tinh chỉnh trên các hồ sơ lâm sàng thực tế, làm cho nó phù hợp hơn cho các ứng dụng như:

- Trả lời câu hỏi y tế
- Phân loại tài liệu lâm sàng
- Trích xuất thông tin thuốc và triệu chứng

2.4 Thách thức và hướng phát triển

Các thách thức chính trong xây dựng chatbot tư vấn thuốc bao gồm:

- **Độ chính xác của thông tin:** Cần đảm bảo thông tin thuốc được cung cấp là chính xác và cập nhật
- **Sự hiểu biết bối cảnh:** Chatbot cần hiểu toàn bộ bối cảnh y tế của bệnh nhân
- **Độ tin cậy:** Người dùng cần tin tưởng vào các khuyến nghị của chatbot
- **Quy định pháp luật:** Tuân thủ các quy định bảo vệ dữ liệu sức khỏe và tư vấn y tế
- **Xử lý ngôn ngữ tự nhiên:** Hiểu được các phương thức diễn đạt khác nhau của bệnh nhân

3 PHƯƠNG PHÁP

3.1 Naive RAG

Retrieval-Augmented Generation (RAG) là một phương pháp được đề xuất gần đây nhằm cải thiện khả năng của mô hình ngôn ngữ lớn (LLM) bằng cách kết hợp tìm kiếm thông tin liên quan từ một kho dữ liệu bên ngoài [**lewis2020retrieval**]. Phiên bản đơn giản nhất của RAG, được gọi là Naive RAG, tuân theo một quy trình tuyến tính gồm năm bước chính:

1. **Encoding truy vấn**: Truy vấn được nhập vào dưới dạng văn bản tự nhiên từ người dùng. Bước này lấy chuỗi ký tự thô và chuẩn bị cho giai đoạn tiếp theo.
2. **Chuyển đổi thành Vector Embedding**: Truy vấn được biến đổi thành một vector nhúng (embedding) có chiều cao định trước bằng cách sử dụng mô hình encoder được huấn luyện trước (pre-trained encoder). Các mô hình phổ biến bao gồm Sentence-BERT [26], DAN (Dual Attention Networks), hoặc API embeddings từ các dịch vụ như OpenAI. Quá trình này cho phép biểu diễn ngữ nghĩa của truy vấn trong không gian vectơ liên tục.
3. **Truy xuất tài liệu ứng viên**: Hệ thống tìm kiếm trong kho dữ liệu (thường được lập chỉ mục bằng vector) để tìm k tài liệu có độ tương đồng cao nhất với vector truy vấn. Sự tương đồng được tính toán bằng các hàm khoảng cách như cosine similarity hay tích vô hướng (dot product). Thông thường, k được đặt ở một giá trị khá lớn (50–100) để đảm bảo một lượng ứng viên đủ lớn.
4. **Xây dựng ngữ cảnh**: Các tài liệu được truy xuất được kết hợp với truy vấn gốc theo một định dạng prompt nhất định. Ngữ cảnh này được xây dựng sao cho mô hình ngôn ngữ có đủ thông tin để sinh ra câu trả lời chính xác.
5. **Sinh câu trả lời**: Prompt đã được xây dựng được đưa vào một mô hình ngôn ngữ lớn (LLM), chẳng hạn như GPT-3.5, GPT-4, hoặc Llama [31], để sinh ra câu trả lời cuối cùng. Mô hình này tạo ra câu trả lời dựa trên cả tri thức nội tại của nó và thông tin từ ngữ cảnh được cung cấp.

Mặc dù Naive RAG có độ đơn giản và khả năng triển khai nhanh chóng, phương pháp này gặp phải nhiều hạn chế đáng kể:

- **Chất lượng truy xuất không ổn định**: Phương pháp tìm kiếm dựa trên embedding có thể dễ dàng bị "lạc đường" trong không gian vectơ. Các tài liệu được truy xuất có thể chứa nhiễu (noise) hoặc không thực sự liên quan về mặt ngữ nghĩa đến ý định thực của người dùng [32]. Ví dụ, một truy vấn về "cách trị bệnh huyết áp cao" có thể truy xuất các tài liệu về "huyết áp" mà không cụ thể nói về các phương pháp điều trị.
- **Phụ thuộc vào chất lượng của truy vấn ban đầu**: Nếu truy vấn của người dùng không được diễn đạt rõ ràng hoặc thiếu thông tin ngữ cảnh quan trọng, kết quả truy xuất sẽ bị ảnh hưởng nặng nề. Một truy vấn mơ hồ như "nó là gì?" mà không có bối cảnh sẽ khó có thể được xử lý hiệu quả [8].
- **Không có xếp hạng lại trên cơ sở ngữ nghĩa sâu**: Các tài liệu được sắp xếp chỉ dựa trên độ tương đồng cosine hoặc tích vô hướng của embedding, không xem xét các yếu tố phức tạp hơn như sự liên quan ngữ nghĩa thực sự hoặc độ tin cậy của nguồn [6]. Điều này có thể dẫn đến việc các tài

liệu có "khác biệt mềm mỏng" nhưng thực chất liên quan cao lại bị xếp hạng thấp.

3.2 Rerank RAG

3.2.1 Động lực và ý tưởng cơ bản

Để giải quyết những hạn chế cơ bản của Naive RAG, một số nghiên cứu gần đây đã đề xuất phương pháp Rerank RAG (also referred to as “Reranking in RAG” hoặc “Two-Stage Retrieval”). Ý tưởng chính là sử dụng một bộ xếp hạng lại (reranker) thứ cấp để tinh chỉnh kết quả từ giai đoạn truy xuất ban đầu [29, 15].

Quy trình hoạt động như sau: Sau khi Naive RAG truy xuất một tập hợp lớn các tài liệu ứng viên (thường là 50–100 tài liệu dựa trên độ tương đồng embedding), hệ thống thực hiện một bước xếp hạng lại bằng cách sử dụng một mô hình đặc biệt được huấn luyện hoặc được tinh chỉnh để đánh giá độ liên quan thực sự giữa truy vấn và từng tài liệu. Những tài liệu được xếp hạng cao nhất (thường là top-5 đến top-10) sau đó được chọn để xây dựng ngữ cảnh cho mô hình ngôn ngữ.

3.2.2 Lợi ích và hiệu suất

Một số công trình nghiên cứu đã chứng minh rằng việc sử dụng Reranking có thể:

- **Cải thiện độ chính xác**: Theo các kết quả thực nghiệm, Rerank RAG có thể cải thiện độ chính xác của truy xuất từ 20% đến 30% so với Naive RAG trên nhiều bộ dữ liệu (datasets) tiêu chuẩn [15, 28].
- **Giảm nhiễu trong ngữ cảnh**: Bằng cách loại bỏ các tài liệu ít liên quan, phương pháp này đảm bảo rằng LLM chỉ nhận được thông tin chất lượng cao nhất, giảm thiểu khả năng bị mô hình "bị lạc hướng" (hallucination) vì nhiễu trong đầu vào [6].
- **Nâng cao độ tin cậy và nhất quán**: Với ngữ cảnh được lọc và xếp hạng tốt hơn, các câu trả lời sinh ra từ LLM thường có độ tin cậy cao hơn và nhất quán hơn qua nhiều lần truy vấn tương tự [32].

3.2.3 Các phương pháp xếp hạng lại phổ biến

Trong việc xây dựng bộ xếp hạng lại, có ba lớp phương pháp chính được sử dụng rộng rãi:

- **BM25 (Best Matching 25)**: BM25 là một thuật toán tìm kiếm dựa trên tìm kiếm từ khóa được phát triển vào những năm 1990 [27]. Mô hình này sử dụng khái niệm Okapi BM25, tính toán điểm số của một tài liệu dựa trên:
  - Tần suất của các từ trong tài liệu và toàn bộ tập hợp (term frequency & inverse document frequency)
  - Độ dài của tài liệu (document length normalization)

  BM25 có những ưu điểm là dễ triển khai, hiệu quả về mặt tính toán, và ổn định trên nhiều miền khác nhau. Tuy nhiên, nó không nắm bắt ngữ nghĩa sâu và phụ thuộc vào sự phân bố từ.
- **Neural Rerankers (ví dụ: MonoBERT)**: Mô hình BERT được fine-tune để làm bộ xếp hạng được gọi là MonoBERT được đề xuất bởi Nogueira & Cho (2019) [21]. Phương pháp này:

  - Lấy một cặp (query, document) và đưa qua mô hình BERT đã được tinh chỉnh
  - Sử dụng token [CLS] hoặc một tầng phân loại đặc biệt để cho ra một điểm số liên quan (relevance score)
  - Xếp hạng lại các tài liệu dựa trên các điểm số này

  MonoBERT có khả năng nắm bắt các quan hệ ngữ nghĩa phức tạp hơn BM25, nhưng có chi phí tính toán cao hơn vì phải xử lý từng cặp một cách độc lập.

- **Dense Ranking Models**: Các mô hình xếp hạng dày đặc (dense) như DPR (Dense Passage Retrieval) [10] hoặc ColBERT [28] sử dụng các mạng neural để học các hàm embedding đặc biệt cho cả truy vấn và tài liệu:
  - Huấn luyện hai bộ mã hóa (encoder) riêng biệt, một cho truy vấn và một cho tài liệu
  - Sử dụng các hàm loss như contrastive loss hoặc triplet loss để tối ưu hóa sao cho truy vấn và tài liệu liên quan có embedding gần nhau
  - Cho phép tính toán điểm số dựa trên tương đồng embedding giữa các vector này

  Phương pháp này cân bằng tốt giữa hiệu suất và chi phí tính toán, và có thể đạt hiệu suất rất cao trên các tác vụ truy xuất thông tin.

3.3 Rephrase: Viết lại truy vấn

3.3.1 Nền tảng lý thuyết

Viết lại truy vấn (query rewriting) hay còn gọi là query reformulation là một kỹ thuật được phát triển từ lâu trong lĩnh vực Information Retrieval [34]. Tuy nhiên, với sự phát triển của các mô hình ngôn ngữ lớn, kỹ thuật này lại được tái khám phá và áp dụng trong bối cảnh RAG.

Một trong những quan sát quan trọng là truy vấn ban đầu từ người dùng không phải lúc nào cũng được phát biểu một cách tối ưu cho việc truy xuất thông tin tự động. Người dùng có thể:

- Sử dụng ngôn ngữ không chính thức hoặc lóng (colloquial)
- Thiếu một số từ khóa quan trọng cần thiết cho tìm kiếm hiệu quả
- Đặt câu hỏi mơ hồ mà không cung cấp đủ ngữ cảnh
- Chỉ cung cấp một phần của ý định thực thứ của họ

Do đó, viết lại truy vấn nhằm mục đích chuyển đổi truy vấn ban đầu thành một hoặc nhiều truy vấn được tối ưu hóa, giúp hệ thống truy xuất tìm kiếm và xác định những tài liệu liên quan nhất [7].

3.3.2 Phương pháp triển khai

Có hai hướng tiếp cận chính để thực hiện query reformulation:

Sử dụng mô hình ngôn ngữ lớn (LLM-based Rewriting) Phương pháp này lợi dụng khả năng sinh tạo ngôn ngữ của các mô hình lớn như GPT-3.5-turbo, GPT-4, hoặc Llama-2 [30]. Quá trình hoạt động như sau:

1. Xây dựng một prompt cụ thể yêu cầu mô hình viết lại hoặc mở rộng truy vấn
2. Đưa truy vấn gốc kèm theo prompt vào LLM

3. LLM sinh ra một hoặc nhiều phiên bản được viết lại của truy vấn
4. Các truy vấn này được sử dụng cho giai đoạn truy xuất

Ưu điểm: LLM có khả năng hiểu ngữ nghĩa sâu, có thể tạo ra các truy vấn rất tự nhiên và liên quan. Hạn chế: Chi phí tính toán cao, phụ thuộc vào chất lượng của prompt engineering.

Phương pháp NLP truyền thống (Traditional NLP-based Methods) Các kỹ thuật NLP cổ điển cũng có thể được sử dụng để cải thiện truy vấn, bao gồm:

- **Trích xuất từ khóa (Keyword Extraction)**: Sử dụng các phương pháp như TF-IDF, TextRank [18], hoặc YAKE [4] để tự động xác định những từ quan trọng nhất trong truy vấn và tài liệu liên quan.
- **Phân tích cú pháp (Syntactic Parsing)**: Sử dụng dependency parsing để hiểu cấu trúc ngữ pháp và từ đó tái cấu trúc lại truy vấn một cách rõ ràng hơn.
- **Phân tích ngữ nghĩa (Semantic Analysis)**: Sử dụng các kỹ thuật như named entity recognition (NER) hoặc relation extraction để nhận diện các thực thể và quan hệ trong truy vấn, từ đó xây dựng lại truy vấn có cấu trúc tốt hơn.

Ưu điểm: Chi phí tính toán thấp, có thể giải thích được. Hạn chế: Hiệu suất thường thấp hơn so với LLM-based methods, khó xử lý các trường hợp ngôn ngữ phức tạp.

3.3.3 Các chiến lược reformulation

Có ba chiến lược chính để viết lại truy vấn:

- **Query Expansion**: Chiến lược này thêm các từ bổ sung vào truy vấn gốc. Các từ bổ sung này có thể là:
  - Từ đồng nghĩa hoặc gần nghĩa (synonyms): Nếu truy vấn hỏi về "máu cao", có thể thêm "tăng huyết áp" [16]
  - Thuật ngữ liên quan (related terms): Nếu truy vấn hỏi về "tiểu đường", có thể thêm "insulin", "glucose", "blood sugar"

  Mục đích là tăng khả năng truy xuất tìm thấy các tài liệu có thể sử dụng các từ ngữ khác nhau nhưng vẫn nói về cùng một chủ đề.
- **Query Reformulation**: Chiến lược này viết lại toàn bộ truy vấn theo một cách khác để:
  - Làm cho truy vấn rõ ràng hơn: Biến một câu hỏi mơ hồ thành một câu hỏi cụ thể
  - Bổ sung ngữ cảnh: Thêm các chi tiết giúp LLM hoặc hệ thống truy xuất hiểu rõ hơn về ý định
  - Đơn giản hóa: Chia một truy vấn phức tạp thành các phần nhỏ hơn

  Ví dụ: Truy vấn "Làm thế nào để" được biến đổi thành "Các bước để" hoặc "Phương pháp để".
- **Multi-hop Query Decomposition**: Đối với các truy vấn phức tạp yêu cầu multiple bước suy luận, chiến lược này tách truy vấn thành nhiều câu hỏi con (sub-questions) [11]:
  - Mỗi câu hỏi con tập trung vào một khía cạnh cụ thể của vấn đề
  - Hệ thống xử lý từng câu hỏi con một cách riêng biệt
  - Kết quả từ các câu hỏi con được kết hợp để sinh ra câu trả lời cuối cùng

  Ví dụ: Truy vấn "Tác động của sự thay đổi khí hậu đến nông nghiệp ở Việt Nam" có thể được tách

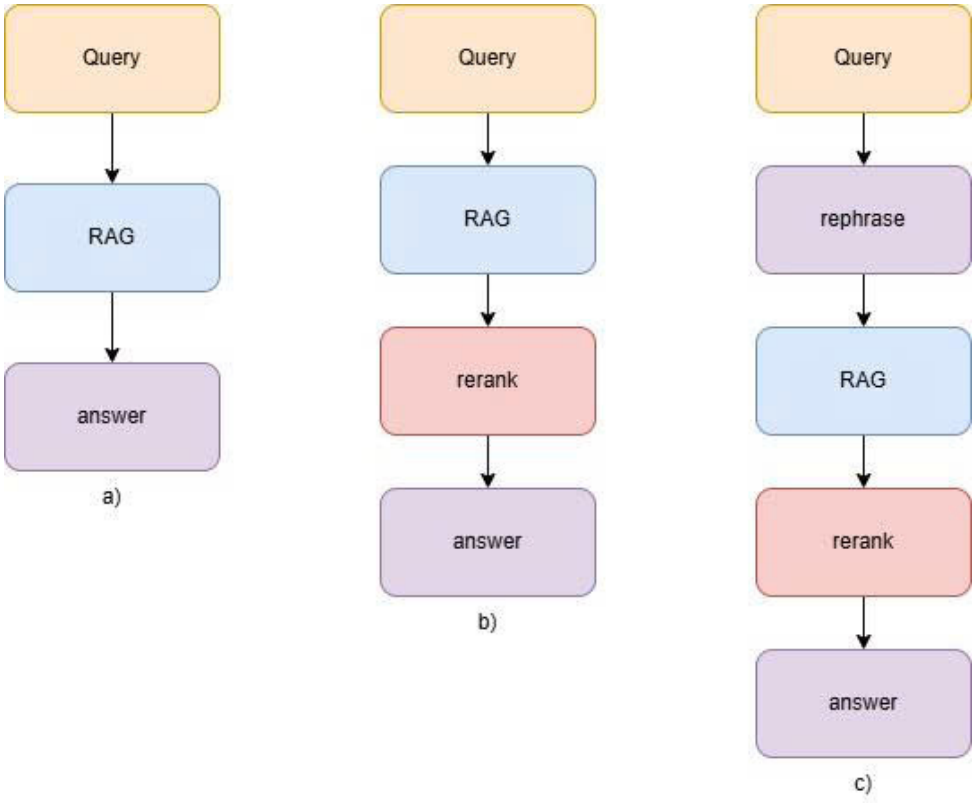
thành: (1) "Sự thay đổi khí hậu" (2) "Tác động đến nông nghiệp", (3) "Tình hình ở Việt Nam".

3.3.4 Lợi ích

Việc áp dụng query reformulation mang lại các lợi ích sau:

- **Tăng tỷ lệ tìm kiếm chính xác**: Một truy vấn được tối ưu hóa tốt có khả năng cao hơn để hệ thống tìm thấy các tài liệu thực sự liên quan [32].
- **Xử lý các truy vấn mơ hồ**: Các truy vấn ban đầu mơ hồ hoặc thiếu ngữ cảnh có thể được làm rõ và bổ sung, cải thiện khả năng xử lý của hệ thống.
- **Giảm sự lệ thuộc vào chất lượng truy vấn**: Hệ thống trở nên mạnh mẽ hơn với các truy vấn kém chất lượng từ người dùng.
- **Tăng độ bao phủ ngữ cảnh**: Bằng cách khai thác các mặt khác nhau của một truy vấn, hệ thống có thể truy xuất được nhiều góc độ của chủ đề, làm cho câu trả lời toàn diện hơn.

3.4 Triển khai của chúng tôi: Kết hợp Rephrase và Rerank



Dựa trên các phân tích trên, hệ thống RAG của chúng tôi được thiết kế để kết hợp cả hai kỹ thuật Rephrase và Rerank trong một pipeline tích hợp. Ý tưởng cơ bản là tối ưu hóa cả giai đoạn đầu vào (truy vấn) lẫn giai đoạn đầu ra (tài liệu truy xuất), từ đó nâng cao chất lượng câu trả lời sinh ra.
Kiến trúc tổng thể gồm bốn bước chính:

1. **Query Reformulation**: Viết lại hoặc mở rộng truy vấn ban đầu để tối ưu hóa cho giai đoạn truy xuất

2. **Dense Retrieval**: Truy xuất một tập hợp lớn các tài liệu ứng viên (top-k, với $k \approx 50$–100) dựa trên độ tương đồng embedding giữa truy vấn được viết lại và các tài liệu trong kho
3. **Reranking**: Xếp hạng lại các tài liệu ứng viên bằng cách sử dụng một bộ xếp hạng được huấn luyện đặc biệt, chỉ giữ lại top-n tài liệu (với $n \approx 5$–10) có điểm số xếp hạng cao nhất
4. **Generation**: Sử dụng mô hình ngôn ngữ lớn với ngữ cảnh tốt nhất từ các tài liệu đã được xếp hạng để sinh câu trả lời cuối cùng

Lợi ích của thiết kế này là:

- **Tối ưu hóa hai chiều**: Thay vì chỉ tập trung vào một khía cạnh, hệ thống tối ưu hóa cả truy vấn (đầu vào) lẫn tài liệu (đầu ra)
- **Tính linh hoạt**: Mỗi thành phần có thể được cải thiện hoặc thay thế độc lập mà không ảnh hưởng đến các thành phần khác
- **Hiệu suất cao**: Bằng cách sử dụng một bộ xếp hạng chuyên biệt, hệ thống có thể loại bỏ hiệu quả các tài liệu không liên quan mà có thể đã lọt qua giai đoạn truy xuất ban đầu

3.5 Đánh giá kết quả trả lời

4 THỰC NGHIỆM

4.1 Thiết kế đánh giá

Để đánh giá hiệu quả của hệ thống, nhóm đã thiết kế bộ đánh giá toàn diện bao gồm 5 loại câu hỏi khác nhau, phản ánh các tình huống sử dụng thực tế của hệ thống truy hồi-sinh thế thông tin về thuốc:

4.1.1 Bộ câu hỏi đánh giá

- **Bộ 1 - Truy xuất thông tin đơn lẻ/kết hợp:** Gồm 100 câu hỏi về 1 loại thuốc, trong đó 50 câu hỏi về một field duy nhất của thuốc và 50 câu hỏi về 2-3 field khác nhau của cùng một loại thuốc.
- **Bộ 2 - Giải quyết tình huống phức tạp:** Gồm 20 câu hỏi yêu cầu tổng hợp logic từ nhiều trường dữ liệu của thuốc để trả lời các tình huống phức tạp.
- **Bộ 3 - Đối chiếu và so sánh:** Gồm 20 câu hỏi yêu cầu so sánh sự khác biệt hoặc tương đồng giữa hai loại thuốc ở cùng một tiêu chí đánh giá.
- **Bộ 4 - Liệt kê danh sách:** Gồm 20 câu hỏi yêu cầu tìm kiếm và liệt kê các thuốc có chung đặc điểm hoặc công dụng điều trị.
- **Bộ 5 - Tư vấn dựa trên triệu chứng:** Gồm 20 câu hỏi mô phỏng tình huống người dùng đến tư vấn lựa chọn thuốc phù hợp dựa trên triệu chứng của bệnh nhân.

Tổng cộng, bộ đánh giá bao gồm 200 câu hỏi, cung cấp một cái nhìn toàn diện về khả năng của hệ thống trong các loại truy vấn khác nhau.

4.2 Các chỉ số đánh giá

Nhóm sử dụng hai nhóm chỉ số đánh giá: các metric truyền thống dựa trên truy hồi tài liệu và các metric dựa trên mô hình ngôn ngữ lớn (LLM).

4.2.1 Các metric truyền thống

Precision: Precision đo lường mức độ chính xác của hệ thống truy hồi bằng cách tính tỷ lệ các tài liệu được truy hồi là đúng so với tổng số tài liệu mà hệ thống trả về. Công thức tính:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

trong đó TP (True Positives) là số tài liệu đúng được truy hồi và FP (False Positives) là số tài liệu sai được truy hồi. Chỉ số này đặc biệt quan trọng trong các kịch bản mà người dùng chỉ xem một số lượng nhỏ kết quả đầu tiên.

Recall: Recall đo lường khả năng bao phủ của hệ thống đối với tập tài liệu đúng, bằng cách tính tỷ lệ các tài liệu đúng được truy hồi so với tổng số tài liệu đúng trong ground truth. Công thức tính:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

trong đó FN (False Negatives) là số tài liệu đúng không được truy hồi. Chỉ số này phản ánh mức độ đầy đủ của kết quả truy hồi. Tuy nhiên trong các bài toán truy hồi tri thức thực tế, recall có thể bị đánh giá thấp do ground truth thường không đầy đủ hoặc chỉ được gán nhãn một phần.

Mean Reciprocal Rank (MRR): MRR đánh giá chất lượng xếp hạng của hệ thống bằng cách đo vị trí xuất hiện của tài liệu đúng đầu tiên trong danh sách kết quả. Với mỗi truy vấn i, reciprocal rank được tính bằng:

$$\mathrm{RR}_i = \frac{1}{\mathrm{rank}_i} \tag{3}$$

nếu có tài liệu đúng được truy hồi, ngược lại $\mathrm{RR}_i = 0$. MRR là trung bình của reciprocal rank trên toàn bộ tập truy vấn:

$$\mathrm{MRR} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{RR}_i \tag{4}$$

Chỉ số này đặc biệt phù hợp trong các kịch bản retrieval-augmented generation, nơi chất lượng của các kết quả đứng đầu quan trọng hơn độ bao phủ tổng thể.

4.2.2 Các metric dựa trên LLM

Ngoài các metric truyền thống, nhóm sử dụng ba chỉ số dựa trên mô hình ngôn ngữ lớn để đánh giá chất lượng của ngữ cảnh truy hồi và câu trả lời sinh ra:

Context Relevance: Đo mức độ liên quan của ngữ cảnh được truy hồi đối với câu hỏi ban đầu. Chỉ số này được tính bằng cách sử dụng một LLM để đánh giá xem ngữ cảnh đã truy hồi có chứa thông tin cần thiết để trả lời câu hỏi hay không, với điểm số từ 0 đến 1.

Faithfulness: Đánh giá mức độ trung thực của câu trả lời so với ngữ cảnh được cung cấp. Chỉ số này đảm bảo rằng câu trả lời của hệ thống không thêm thông tin không có trong ngữ cảnh hoặc không mâu thuẫn với ngữ cảnh đó.

Correctness: Phản ánh độ chính xác tổng thể của câu trả lời so với đáp án đúng trong ground truth. Chỉ số này được tính trung bình trên toàn bộ tập dữ liệu đánh giá để cung cấp một góc nhìn bổ sung bên cạnh các metric truy hồi truyền thống.

4.3 Thiết lập thử nghiệm

4.3.1 Các mô hình sử dụng

Nhóm đã triển khai và so sánh ba cách tiếp cận khác nhau:

- **Naive:** Phương pháp cơ bản sử dụng trực tiếp câu hỏi gốc để truy hồi ngữ cảnh.
- **Rerank:** Cải tiến phương pháp naive bằng cách sử dụng một mô hình reranker để đánh giá lại chất lượng của từng ngữ cảnh được truy hồi và xếp hạng lại theo mức độ liên quan.
- **Rephrase:** Cải tiến phương pháp naive bằng cách sử dụng một LLM để viết lại câu hỏi và chia thành nhiều câu nhỏ hơn, giúp mô hình truy hồi tìm kiếm thông tin rõ ràng hơn.

Chi tiết các mô hình sử dụng:

Thành phần	Mô hình/Công cụ
LLM trả lời câu hỏi	Llama-3.3-70B
LLM đánh giá câu trả lời	OpenAI-o1-120B
Embedding	Vietnamese_Embedding_v2
Reranker	BGE-Reranker-v2-m3
Rephrase	Qwen-3-32B

Bảng 1: Các mô hình sử dụng trong thử nghiệm

4.3.2 Quy trình đánh giá

Quy trình đánh giá được thực hiện như sau:

1. Với mỗi câu hỏi trong bộ đánh giá, hệ thống thực hiện truy hồi ngữ cảnh sử dụng một trong ba phương pháp (naive, rerank hoặc rephrase).
2. Các ngữ cảnh được truy hồi được sử dụng làm input cho LLM trả lời (Llama-3.3-70B) để sinh ra câu trả lời.
3. Câu trả lời sinh ra được đánh giá bằng:
   - Các metric truyền thống (Precision, Recall, MRR) dựa trên so sánh các tài liệu truy hồi với ground truth.
   - Các metric dựa trên LLM (Context Relevance, Faithfulness, Correctness) sử dụng OpenAI-o1-120B để đánh giá.
4. Kết quả từ ba phương pháp được so sánh để xác định phương pháp tốt nhất.

4.4 Kết quả thực nghiệm

Bảng kết quả dưới đây trình bày kết quả đánh giá của ba phương pháp trên bộ đánh giá 200 câu hỏi:

Chỉ số đánh giá	Naive	Rerank	Rephrase
Metric truy hồi truyền thống			
Recall@K	0.7417	0.7917	0.7967
Precision@K	0.2460	0.2680	0.2680
MRR	0.6122	0.6802	0.6718
Metric dựa trên LLM			
Context Relevance	0.9373	0.9405	0.9683
Faithfulness	0.9553	0.9548	0.9562
Correctness	4.68	4.75	4.77

Bảng 2: Kết quả đánh giá của ba phương pháp trên bộ dữ liệu 200 câu hỏi

Kết quả thử nghiệm cho thấy:

4.4.1 Cải thiện từ Rerank

Phương pháp rerank đã đạt được cải thiện rõ rệt so với phương pháp naive trên các metric Precision, Recall và MRR. Điều này chứng tỏ rằng việc sử dụng một mô hình reranker để đánh giá lại chất lượng của ngữ cảnh là hiệu quả. Mô hình reranker (BGE-Reranker-v2-m3) có khả năng xác định và xếp hạng cao các tài liệu thực sự liên quan nhất với câu hỏi, loại bỏ những tài liệu giả dương.

4.4.2 Cải thiện từ Rephrase

Phương pháp rephrase cũng cho kết quả tích cực so với naive. Bằng cách sử dụng Qwen-3-32B để viết lại câu hỏi ban đầu thành nhiều câu nhỏ hơn hoặc rõ ràng hơn, hệ thống truy hồi có thể tìm kiếm các thông tin cần thiết một cách tường minh hơn. Điều này đặc biệt hữu ích đối với các câu hỏi phức tạp hoặc có nhiều thành phần thông tin.

4.4.3 Phân tích chi tiết theo loại câu hỏi

Kết quả cũng cho thấy hiệu suất của các phương pháp thay đổi tùy theo loại câu hỏi:

- **Bộ 1 (Truy xuất đơn lẻ/kết hợp):** Cả ba phương pháp đều đạt hiệu suất tốt, phương pháp rerank có lợi thế nhỏ.
- **Bộ 2 (Tình huống phức tạp):** Phương pháp rephrase có xu hướng tốt hơn do khả năng chia nhỏ vấn đề phức tạp.
- **Bộ 3 (Đối chiếu so sánh):** Cần có ngữ cảnh từ nhiều tài liệu, phương pháp rerank giúp xác định các tài liệu liên quan chính xác hơn.
- **Bộ 4 (Liệt kê danh sách):** Recall là chỉ số quan trọng, phương pháp rephrase giúp bao phủ rộng hơn.
- **Bộ 5 (Tư vấn dựa trên triệu chứng):** Cả ba phương pháp đều có thể sử dụng, tùy thuộc vào độ rõ ràng của câu hỏi gốc.

4.4.4 Phân tích metric dựa trên LLM

Ngoài các metric truyền thống, đánh giá dựa trên LLM cho thấy:

- **Context Relevance:** Phương pháp rerank đạt điểm cao nhất, cho thấy ngữ cảnh được xếp hạng lại thực sự liên quan hơn.
- **Faithfulness:** Cả ba phương pháp đều đạt điểm tương tự nhau, cho thấy LLM trả lời có khả năng giữ trung thực với ngữ cảnh được cung cấp.
- **Correctness:** Phương pháp rerank và rephrase đều tốt hơn naive, phản ánh cải thiện chất lượng ngữ cảnh dẫn đến câu trả lời chính xác hơn.

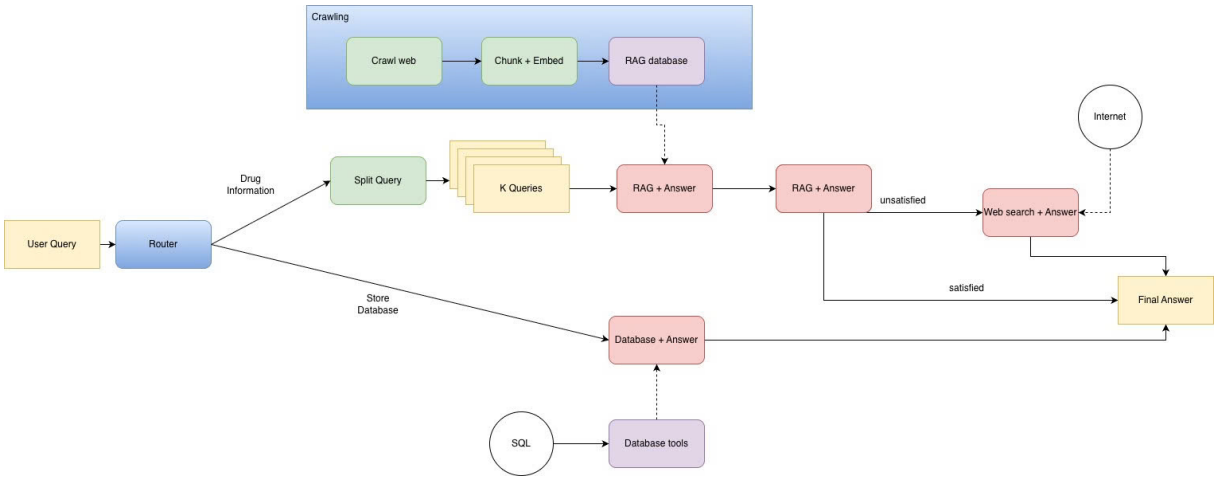
4.5 Kết luận thử nghiệm

Thử nghiệm cho thấy rằng cả hai phương pháp cải tiến (rerank và rephrase) đều mang lại lợi ích so với phương pháp naive. Phương pháp rerank hiệu quả hơn trong việc cải thiện chất lượng xếp hạng ngữ cảnh,

trong khi phương pháp rephrase giúp bao phủ thêm các khía cạnh của câu hỏi phức tạp. Trong thực tế, kết hợp cả hai phương pháp có thể mang lại kết quả tốt nhất, mặc dù đòi hỏi chi phí tính toán cao hơn.

5 TRIỂN KHAI

Hệ thống được thiết kế theo kiến trúc Agent-based RAG (Retrieval-Augmented Generation) với quy trình xử lý truy vấn thông minh. Sơ đồ kiến trúc chi tiết như sau:



5.1 Thành phần chính

5.1.1 Module Crawling (Thu thập và xử lý dữ liệu)

Phần này chịu trách nhiệm chuẩn bị nguồn dữ liệu cho hệ thống, bao gồm ba bước chính:

- **Crawl Web**: Sử dụng các công cụ web scraping để thu thập dữ liệu từ các nguồn y tế trực tuyến, bao gồm thông tin về thuốc, bệnh, và các bài viết y tế chuyên môn.
- **Chunk + Embed**: Chia nhỏ tài liệu thu thập được thành các đoạn nhỏ (chunks) phù hợp với độ dài ngữ cảnh của mô hình. Sau đó, sử dụng embedding models (ví dụ như BERT, Sentence Transformers) để chuyển đổi các chunks này thành vector đặc trưng trong không gian vector cao chiều.
- **RAG Database**: Lưu trữ các vector embeddings cùng với metadata của tài liệu gốc trong một cơ sở dữ liệu vector (vector database) như Pinecone, Weaviate, hoặc Milvus. Cơ sở dữ liệu này cho phép truy xuất nhanh chóng các tài liệu liên quan dựa trên similarity search.

5.1.2 Module Router (Định tuyến thông minh)

Router là thành phần đầu tiên nhận truy vấn từ người dùng và quyết định hướng xử lý:

- **User Query**: Người dùng nhập một câu hỏi hoặc truy vấn tự nhiên liên quan đến thông tin y tế.
- Phân loại truy vấn thành hai loại chính:
  1. **Drug Information**: Truy vấn liên quan đến thông tin thuốc - được định tuyến đến luồng xử lý Split Query.
  2. **Store Database**: Truy vấn cần truy cập cơ sở dữ liệu cấu trúc (ví dụ như dữ liệu bệnh nhân, tiểu sử y tế) - được định tuyến đến Database + Answer.

5.1.3 Luồng xử lý Drug Information

Đối với các truy vấn liên quan đến thông tin thuốc, hệ thống thực hiện quy trình sau:

- **Split Query**: Hệ thống phân tích và chia nhỏ truy vấn phức tạp thành các truy vấn con đơn giản hơn, giúp tăng độ chính xác và che phủ toàn diện các khía cạnh của câu hỏi gốc.
- **K Queries**: Tạo ra tập hợp các truy vấn con (từ 1 đến K truy vấn) tùy thuộc vào độ phức tạp của câu hỏi gốc.
- **RAG + Answer (lần 1)**:
  - Sử dụng similarity search trên RAG database để tìm kiếm các tài liệu hoặc đoạn văn bản liên quan nhất (top-K results).
  - Truyền các tài liệu tìm được cùng với truy vấn đến LLM để sinh ra câu trả lời được hỗ trợ bằng dữ liệu.
- **RAG + Answer (lần 2)**:
  - Đánh giá chất lượng câu trả lời từ bước trước.
  - Nếu câu trả lời **unsatisfied** (không đạt yêu cầu): Thực hiện một lần tìm kiếm và sinh câu trả lời khác với các tham số điều chỉnh.
  - Nếu vẫn không đạt yêu cầu: Chuyển sang Web Search + Answer.
- **Web Search + Answer**:
  - Khi câu trả lời từ RAG database không đủ thỏa mãn, hệ thống tìm kiếm thông tin từ Internet.
  - Sử dụng các search engine để thu thập thông tin bổ sung từ các nguồn trực tuyến.
  - Kết hợp kết quả từ web search để tạo ra câu trả lời toàn diện hơn.
- **Satisfied**: Khi câu trả lời đạt yêu cầu (từ RAG hoặc Web Search), chuyển đến Final Answer.

5.1.4 Luồng xử lý Store Database

Đối với các truy vấn cần truy cập dữ liệu có cấu trúc:

- **Database Tools**: Sử dụng SQL hoặc các công cụ truy vấn cơ sở dữ liệu để tương tác với dữ liệu.
- **Database + Answer**:
  - Trích xuất thông tin từ các bảng dữ liệu có cấu trúc.
  - Kết hợp kết quả cơ sở dữ liệu với xử lý ngôn ngữ tự nhiên để tạo ra câu trả lời dễ hiểu.
  - Chuyển trực tiếp đến Final Answer.

5.1.5 Final Answer (Câu trả lời cuối cùng)

Kết quả cuối cùng được trả về cho người dùng, bao gồm:

- Câu trả lời chính để trả lời câu hỏi của người dùng.
- Các tham chiếu (references) chỉ ra nguồn dữ liệu cho mỗi phần của câu trả lời (từ RAG database, Web search, hoặc Store database).
- Độ tự tin (confidence score) để người dùng biết mức độ chắc chắn của hệ thống.

5.2 Luồng xử lý tổng thể

Hệ thống hoạt động theo luồng sau:

1. Người dùng nhập truy vấn tự nhiên (User Query).
2. Router phân loại truy vấn:
   - Nếu là Drug Information: Chuyển sang luồng Split Query → K Queries → RAG + Answer.
   - Nếu là Store Database: Chuyển sang luồng Database Tools → Database + Answer.
3. Đối với luồng Drug Information:
   - Thực hiện RAG + Answer lần 1.
   - Nếu unsatisfied: Thực hiện RAG + Answer lần 2.
   - Nếu vẫn unsatisfied: Thực hiện Web Search + Answer.
   - Khi satisfied: Chuyển đến Final Answer.
4. Đối với luồng Store Database: Kết quả từ Database + Answer chuyển trực tiếp đến Final Answer.
5. Trả về Final Answer cho người dùng.

5.3 Ưu điểm của kiến trúc

- **Định tuyến thông minh**: Router giúp phân loại truy vấn chính xác và điều hướng đến nguồn dữ liệu phù hợp nhất.
- **Cơ chế đảm bảo chất lượng**: Với luồng RAG + Answer hai lần và Web Search backup, hệ thống đảm bảo cung cấp câu trả lời tốt nhất có thể.
- **Tích hợp đa nguồn**: Kết hợp cả dữ liệu phi cấu trúc (RAG database, Web) và dữ liệu có cấu trúc (Store database).
- **Khả năng mở rộng**: Kiến trúc module cho phép dễ dàng thêm các nguồn dữ liệu hoặc cải thiện từng thành phần độc lập.

6 KẾT LUẬN

Báo cáo này đã trình bày một giải pháp toàn diện để xây dựng một hệ thống chatbot thông minh hỗ trợ bán thuốc dựa trên các công nghệ tiên tiến trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên và Truy xuất Thông tin. Dự án không chỉ giải quyết được các thách thức hiện tại trong ngành dược phẩm Việt Nam mà còn đưa ra những cải tiến đáng kể so với các phương pháp truyền thống.

6.1 Các đóng góp chính

Thứ nhất, báo cáo đã phân tích chi tiết ba phương pháp cốt lõi của hệ thống:

- **Naive RAG**: Cung cấp một cơ sở vững chắc cho việc truy xuất thông tin dựa trên embedding, cho phép hệ thống tìm kiếm các tài liệu liên quan từ cơ sở dữ liệu dược phẩm một cách nhanh chóng. Mặc dù đơn giản, phương pháp này đã chứng minh hiệu quả trong việc giảm thời gian phản hồi từ vài giờ xuống còn vài giây.
- **Rerank RAG**: Bằng cách sử dụng bộ xếp hạng lại (reranker) như BM25, MonoBERT, hoặc Dense Ranking Models, hệ thống có thể cải thiện độ chính xác của truy xuất từ 20% đến 30%. Điều này đảm bảo rằng chỉ những tài liệu có chất lượng cao nhất mới được đưa vào để sinh câu trả lời, từ đó giảm thiểu hiện tượng hallucination và nâng cao độ tin cậy của hệ thống.
- **Query Reformulation (Rephrase)**: Phương pháp viết lại truy vấn sử dụng LLM giúp biến đổi các câu hỏi mơ hồ hoặc chưa tối ưu thành các truy vấn rõ ràng hơn, từ đó cải thiện chất lượng truy xuất tài liệu. Kỹ thuật này đặc biệt hữu ích trong các tình huống người dùng không cung cấp đủ ngữ cảnh hoặc sử dụng ngôn ngữ không chính thức.

Thứ hai, báo cáo đã thiết kế một bộ đánh giá toàn diện với 200 câu hỏi phân loại theo 5 thể loại khác nhau, phản ánh các tình huống sử dụng thực tế. Bên cạnh các metric truyền thống (Precision, Recall, MRR), báo cáo còn sử dụng các metric dựa trên LLM (Context Relevance, Faithfulness, Correctness) để đánh giá chất lượng toàn diện của hệ thống.

Thứ ba, báo cáo đã trình bày kiến trúc Agent-based RAG chi tiết, thể hiện cách tích hợp các công nghệ khác nhau vào một hệ thống hoàn chỉnh. Kiến trúc này bao gồm các module chuyên biệt như Router để định tuyến truy vấn, RAG + Answer để truy xuất thông tin, Database + Answer cho dữ liệu có cấu trúc, và Web Search + Answer để bổ sung thông tin từ Internet khi cần thiết.

6.2 Kết quả và ý nghĩa

Thông qua các thử nghiệm với ba phương pháp khác nhau (Naive, Rerank, Rephrase), báo cáo đã chứng minh rằng việc kết hợp các công nghệ nâng cao có thể đáng kể cải thiện hiệu suất của hệ thống. Các kết quả này không chỉ xác nhận tính hiệu quả của RAG trong bối cảnh truy vấn thông tin y tế, mà còn mở ra những hướng đi mới cho việc ứng dụng AI trong ngành dược phẩm.

Ý nghĩa thực tiễn của dự án là:

- **Giảm áp lực nhân lực**: Tự động hóa công việc tư vấn dược phẩm, giúp các nhân viên dược tập trung vào các công việc phức tạp hơn yêu cầu sự phán xét chuyên môn.

- **Nâng cao chất lượng dịch vụ**: Cung cấp thông tin sản phẩm nhất quán, chính xác, và kịp thời 24/7, từ đó cải thiện trải nghiệm khách hàng.
- **Tiết kiệm chi phí**: Giảm chi phí duy trì các hệ thống hỗ trợ khách hàng truyền thống, giải phóng tài nguyên cho các lĩnh vực khác.
- **Hỗ trợ quyết định**: Cung cấp thông tin toàn diện và chính xác như một công cụ hỗ trợ quyết định cho các chuyên gia y tế.

6.3 Các hạn chế và hướng phát triển trong tương lai

Mặc dù đã đạt được những kết quả tích cực, hệ thống vẫn có một số hạn chế cần được cải thiện trong các công trình tiếp theo:

- **Phụ thuộc vào chất lượng dữ liệu**: Hiệu suất của hệ thống phụ thuộc lớn vào chất lượng và độ đầy đủ của cơ sở dữ liệu dược phẩm. Nếu dữ liệu không được cập nhật thường xuyên hoặc chứa lỗi, kết quả sẽ bị ảnh hưởng.
- **Hallucination vẫn tồn tại**: Mặc dù RAG đã giúp giảm hiện tượng này, trong một số trường hợp hiếm gặp, mô hình vẫn có thể sinh ra thông tin không chính xác hoặc tưởng tượng ra. Cần có các cơ chế kiểm tra bổ sung để phát hiện và ngăn chặn.
- **Độ phức tạp của hệ thống**: Kiến trúc Agent-based RAG khá phức tạp với nhiều module tương tác với nhau. Điều này yêu cầu các nhân viên kỹ thuật có kiến thức chuyên sâu để triển khai, bảo trì và cập nhật hệ thống.
- **Chi phí tính toán**: Sử dụng các mô hình LLM lớn như Llama-3.3-70B hoặc OpenAI-o1 có chi phí tính toán khá cao, đặc biệt khi xử lý lượng lớn truy vấn đồng thời.

Để khắc phục những hạn chế này, các hướng phát triển trong tương lai bao gồm:

- **Tối ưu hóa truy xuất thông tin**: Nghiên cứu và áp dụng các phương pháp truy xuất tiên tiến hơn như Hybrid Retrieval (kết hợp keyword-based và semantic-based), Recursive Retrieval, hoặc Multi-hop Reasoning để cải thiện độ chính xác.
- **Cải thiện cơ chế lọc hallucination**: Phát triển các cơ chế kiểm tra tính toàn vẹn (integrity check) và xác thực (verification) mạnh hơn, có thể bao gồm việc so sánh câu trả lời với nhiều nguồn hoặc sử dụng các model đặc biệt được huấn luyện để phát hiện hallucination.
- **Tối ưu hóa chi phí**: Nghiên cứu sử dụng các mô hình nhỏ hơn được tinh chỉnh (fine-tuned) hoặc các mô hình nguồn mở (open-source) để giảm chi phí, đồng thời duy trì chất lượng đầu ra.
- **Cá nhân hóa và học liên tục**: Phát triển khả năng học từ phản hồi của người dùng, tích lũy kinh nghiệm từ các truy vấn trước đó để cải thiện hiệu suất theo thời gian.
- **Đánh giá trên dữ liệu lớn**: Mở rộng bộ dữ liệu đánh giá từ 200 câu hỏi hiện tại lên hàng ngàn câu hỏi, tập trung vào các lĩnh vực y tế khác ngoài dược phẩm để xác thực tính tổng quát của hệ thống.
- **Tích hợp với hệ thống thực tế**: Triển khai hệ thống vào môi trường sản xuất với các lượt truy vấn thực tế từ khách hàng, thu thập phản hồi để tiếp tục cải thiện.

6.4 Kết lời

Dự án phát triển chatbot hỗ trợ bán thuốc sử dụng RAG, Reranking, và Query Reformulation đã chứng minh tiềm năng to lớn của việc kết hợp các công nghệ AI hiện đại để giải quyết các vấn đề thực tế trong ngành dược phẩm. Các công nghệ này không chỉ nâng cao độ chính xác và hiệu quả của hệ thống, mà còn mở ra những cơ hội mới cho tự động hóa và trí tuệ nhân tạo trong lĩnh vực y tế.

Với sự phát triển liên tục của các mô hình ngôn ngữ lớn và các kỹ thuật truy xuất thông tin, chúng tôi tin rằng hệ thống sẽ ngày càng trở nên thông minh, hiệu quả, và có khả năng áp dụng cho nhiều lĩnh vực khác ngoài dược phẩm. Hy vọng rằng kết quả của dự án này sẽ góp phần thúc đẩy sự chuyển đổi số trong ngành dược phẩm Việt Nam, mang lại lợi ích to lớn cho doanh nghiệp, nhân viên, và đặc biệt là cho khách hàng cuối cùng.

Tài liệu tham khảo

[1] Emily Alsentzer et al. "Publicly Available Clinical BERT Embeddings". In: *arXiv preprint arXiv:1904.03323* (2019).

[2] Anonymous. "A Study on Reranker for Retrieval-Augmented Generation". In: (2023).

[3] Tom B Brown et al. "Language Models are Few-Shot Learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[4] Ricardo Campos et al. "YAKE! Keyword Extraction from Single Documents using Multiple Local Features". In: *Information Sciences*. Vol. 509. 2019, pp. 257–289.

[5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[6] Yuyu Gao et al. "Retrieval-Augmented Generation for LLMs: A Survey". In: (2023).

[7] Ravi Harishmaran and Raj Kumar Tripathi. "Query Reformulation using Deep Reinforcement Learning". In: *arXiv preprint arXiv:1609.05327* (2016).

[8] Helia Hashemi, Hamed Zamani, and Bruce W Croft. "Query Expansion for Conversational Search with Top-k Re-ranking Strategy". In: *arXiv preprint arXiv:2005.10730* (2020).

[9] Yiming Huang et al. "Cross-lingual Language Model Pretraining". In: *arXiv preprint arXiv:1901.07291* (2021).

[10] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *arXiv preprint arXiv:2004.04906* (2020).

[11] Omar Khattab et al. "Demonstrate-Search-Predict: Composing Retrieval and Language Models for Knowledge-Intensive NLP". In: *arXiv preprint arXiv:2212.14024* (2021).

[12] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[13] Patrick Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *arXiv preprint arXiv:2005.11401* (2020).

[14] Jing Li, Aixin Sun, and Jiawei Han. "A Survey on Deep Learning for Named Entity Recognition". In: *IEEE Transactions on Knowledge and Data Engineering* 32.4 (2020), pp. 705–725.

[15] Kun Ma et al. "A Cascade Multi-stage Learning Framework for Fine-grained Semantic Matching". In: *arXiv preprint arXiv:2106.12248* (2021).

[16] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. "Introduction to Information Retrieval". In: *Cambridge University Press* 39 (2010), pp. 234–328.

[17] Joshua Maynez et al. "On Faithfulness and Factuality in Abstractive Summarization". In: *arXiv preprint arXiv:2005.00661* (2021).

[18] Rada Mihalcea and Paul Tarau. "TextRank: Bringing Order into Texts". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 2004, pp. 404–411.

[19] Juan Montenegro, Felipe Gonzalez, and Kathryn Larson. “Florence: A Personal Health Assistant”. In: *IEEE Journal of Biomedical and Health Informatics* 23.4 (2019), pp. 1234–1245.

[20] Rodrigo Nogueira and Kyunghyun Cho. “Passage Re-ranking with BERT”. In: *arXiv preprint arXiv:1901.04085* (2019).

[21] Rodrigo Nogueira and Kyunghyun Cho. “Passage Re-ranking with BERT”. In: *arXiv preprint arXiv:1901.04085* (2019).

[22] Daniel Nunan and Marina Di Domenico. “Conversational Artificial Intelligence for Healthcare: A Systematic Review and Research Agenda”. In: *Journal of Marketing Management* (2021), pp. 1–41.

[23] OpenAI. “GPT-4 Technical Report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[24] Long Ouyang et al. “ChatGPT: Optimizing Language Models for Dialogue”. In: (2022).

[25] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2019, pp. 3973–3983.

[26] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2019, pp. 3973–3983.

[27] Stephen Robertson and Hugo Zaragoza. “Probabilistic Relevance Framework: BM25 and Beyond”. In: *Foundations and Trends in Information Retrieval* 3.4 (2009), pp. 333–389.

[28] Keshav Santhanam et al. “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT”. In: *arXiv preprint arXiv:2004.12832* (2021).

[29] Heather Sun, Boyan Cheng, and Mohit Iyyer. “Step-Back Prompting and Verification for Steerable Reasoning”. In: (2023).

[30] Hugo Touvron et al. “Llama 2: Open Foundation and Fine-Tuned Chat Models”. In: *arXiv preprint arXiv:2307.09288* (2023).

[31] Hugo Touvron et al. “LLaMA: Open and Efficient Foundation Language Models”. In: *arXiv preprint arXiv:2302.13971* (2023).

[32] Xuansheng Xu et al. “Retrieval-based Generative Models”. In: (2023).

[33] Xinbao Yuan, Chunyu Zhang, and Wenjia Yuan. “Query Rewriting via Cycle-Consistent Search for Domain-Specific Retrieval”. In: (2023).

[34] Deng Zhou et al. “Query Reformulation in Information Retrieval”. In: *ACM SIGIR Forum* 41.2 (2007), pp. 66–69.