ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Đức Anh

HỆ THỐNG CHATBOT ĐỌC HIỂU VÀ PHÂN TÍCH BÁO CÁO TÀI CHÍNH

KHÓA LUẬN TỐT NGHIỆP ĐÀO TẠO HỆ CHÍNH QUY
Ngành: Trí tuệ nhân tạo

HÀ NỘI - 2025

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Đức Anh

HỆ THỐNG CHATBOT ĐỌC HIỂU VÀ PHÂN TÍCH BÁO CÁO TÀI CHÍNH

KHÓA LUẬN TỐT NGHIỆP ĐÀO TẠO HỆ CHÍNH QUY
Ngành: Trí tuệ nhân tạo

Cán bộ hướng dẫn: TS. Trần Hồng Việt

HÀ NỘI - 2025

LỜI CẢM ƠN

Trước tiên em xin được gửi lời cảm ơn chân thành tới TS. Trần Hồng Việt, người đã trực tiếp hướng dẫn em trong suốt quá trình thực hiện khóa luận. Thầy đã có những hướng dẫn chỉ bảo tận tình, đưa ra những lời khuyên giải pháp vô cùng hữu ích, giúp em không chỉ hoàn thành cải thiện khóa luận tốt hơn mà còn học được nhiều về kỹ năng làm việc, giải quyết vấn đề, và các kỹ thuật chuyên môn.

Bên cạnh đó, em cũng xin gửi lời cảm ơn chân thành tới các thầy cô trong Viện Trí tuệ nhân tạo, Trường đại học Công Nghệ ĐHQG Hà Nội, đã tạo điều kiện cho em có không gian cơ hội được học tập làm việc trong môi trường chuyên nghiệp và có những lời khuyên cho em để cải thiện tốt hơn. Em xin gửi lời cảm ơn đến các thầy, cô của trường, viện đã đồng hành trang bị cho em những kiến thức kỹ năng quý báu giúp em phát triển cả về kỹ năng chuyên môn lẫn kỹ năng làm việc.

Cuối cùng em xin gửi lời tri ân sâu sắc đến các anh chị, bạn bè đã luôn đồng hành ủng hộ giúp đỡ em giúp em đạt được kết quả tốt nhất trong suốt quá trình thực hiện khóa luận.

Hà Nội, ngày … tháng 12 năm 2025

Sinh viên

Nguyễn Đức Anh

LỜI CAM ĐOAN

Em xin cam đoan toàn bộ kết quả đạt được trong khóa luận này là sản phẩm của riêng em, là kết quả của quá trình nghiên cứu học tập tại Phòng thí nghiệm Xử lý ngôn ngữ tự nhiên dưới sự dẫn dắt của TS. Trần Hồng Việt.

Những nội dung được trình bày trong khóa luận chưa từng được nộp như một báo cáo khóa luận tại Trường Đại học Công Nghệ - Đại học Quốc Gia Hà Nội hay bất kỳ trường đại học nào khác.

Em xin cam đoan hệ thống Chatbot và quy trình xử lý dữ liệu báo cáo tài chính dựa trên đồ thị là sản phẩm của riêng em không sao chép mã nguồn từ bất kỳ nguồn nào.

Em xin cam đoan những điều trên là đúng sự thật. Nếu sai em xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định của nhà trường.

Hà Nội, ngày … tháng 12 năm 2025

Sinh viên

Nguyễn Đức Anh

TÓM TẮT

Tóm tắt: Tài chính là một lĩnh vực rộng lớn và thu hút nhiều mối quan tâm, đầu tư đặc biệt là trong thời đại sự phát triển của công nghệ thông tin rộng lớn, vấn đề áp dụng các công nghệ hiện đại như blockchain, AI, vào để tăng hiệu quả làm việc trở thành cơ hội cũng như thách thức đối với các nhà đầu tư. Trong những năm gần đây với sự bùng nổ của các mô hình ngôn ngữ lớn như Chat GPT, Gemini có khả năng đọc hiểu ngôn ngữ tự nhiên đáng kinh ngạc, sự xuất hiện của chúng tạo nên một xu hướng mới trong việc tiếp cận công nghệ, việc áp dụng mô hình ngôn ngữ vào nhiều tác vụ trong tài chính như chatbot, dự đoán thị trường, phân tích cảm xúc trở thành nhu cầu tất yếu, mở ra nhiều cơ hội cho các startup công nghệ. Với sự hỗ trợ đáp lực từ các mô hình AI tiên tiến trong xử lý ngôn ngữ tự nhiên có thể giúp ích cho các nhà đầu tự, các chuyên viên phân tích một góc nhìn mời hỗ trợ việc ra quyết định nhanh chóng và chính xác.

Trong khóa luận lần này tập trung việc xử lý tác vụ đọc hiểu và phân tích báo cáo tài chính của một doanh nghiệp bằng việc áp dụng mô hình ngôn ngữ lớn kết hợp với quy trình xử lý dữ liệu được thiết kế riêng biệt với đặc thù dữ liệu báo cáo tài chính dựa trên kiến trúc tăng cường dữ liệu bằng việc truy xuất để giúp AI có thể hiểu được ngữ cảnh ý nghĩa, cấu trúc của báo cáo tài chính từ đó xây dựng hệ thống chatbot có khả năng phân tích câu hỏi của người dùng xử lý báo cáo tài chính với các định dạng khác nhau và đưa ra câu trả lời cho các câu hỏi người dùng xoay quanh nội dung của báo cáo tài chính một cách chính xác truy xuất thông tin, tính toán các chỉ số, đưa ra phân tích theo yêu cầu, nhằm giúp người dùng có cái nhìn khách quan và hỗ trợ đưa ra các quyết định.

Mục tiêu khai thác được nguồn dữ liệu khổng lồ từ báo cáo tài chính của một doanh nghiệp và từ đó xây dựng một giao website với giao diện đơn giản hiện đại thuận tiện dễ tiếp cận dễ sử dụng với bất kỳ đối tượng người dùng nào với nhu cầu đọc hiểu phân tích thông tin nội dung xoay quanh báo cáo tài chính của người dùng.

Từ khóa: báo cáo tài chính, hệ thống chatbot, mô hình ngôn ngữ lớn, tài chính doanh nghiệp, tăng cường dữ liệu bằng việc truy xuất.

MỤC LỤC

LỜI CẢM ƠN I

LỜI CAM ĐOAN II

TÓM TẮT III

DANH MỤC HÌNH ẢNH VI

DANH MỤC BẢNG VII

THUẬT NGỮ VÀ TỪ VIẾT TẮT VIII

CHƯƠNG 1 GIỚI THIỆU CHUNG 1

1.1. Đặt vấn đề 1

1.2. Bài toán 3

1.3. Đóng góp và cấu trúc của khóa luận 4

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 6

2.1. Kiến thức về báo cáo tài chính 6

2.2 Nền tảng công nghệ 6

2.2.1. Mô hình ngôn ngữ lớn 6

2.2.2. Kiến trúc RAG 8

2.2.3. Nhận dạng ký tự quang học 13

2.2.4. Đồ thị tri thức và cơ sở dữ liệu Neo4j 13

2.2.5. NoSQL và MongoDB 14

2.2.6. Các thư viện và công cụ hỗ trợ 15

2.3 Một số nghiên cứu liên quan 17

2.4 Kết chương 19

CHƯƠNG 3 ỨNG DỤNG ĐỒ THỊ TRI THỨC TRONG CHATBOT BÁO CÁO TÀI CHÍNH 20

3.1. Các phương pháp khai thác truy xuất dữ liệu từ báo cáo tài chính 20

3.1.1 Basic RAG 20

3.1.2. Hybrid search + Re-rank RAG 22

3.1.3. Graph-RAG trong báo cáo tài chính (Financial Report Graph-RAG) 26
Biểu diễn báo cáo tài chính dưới dạng đồ thị tri thức (Financial Report Graph) 27
1. Đối với doanh nghiệp sản xuất trên tất cả lĩnh vực 30
2. Đối với doanh nghiệp ngân hàng và chứng khoán 37
3.2. Tối ưu hóa prompt cho việc truy vấn dữ liệu và phân tích báo cáo tài chính ..41
3.2.1. Trong truy vấn dữ liệu 41
3.3.2. Trong phân tích báo cáo tài chính 47
3.3. Hệ thống chatbot đọc hiểu và phân tích báo cáo tài chính 47
3.4 Kết chương 48
CHƯƠNG 4 ĐÁNH GIÁ KẾT QUẢ THỰC NGHIỆM 50
4.1. Mục tiêu thực nghiệm 50
4.2. Đánh giá chất lượng câu trả lời của chatbot qua các phương pháp đánh giá khác nhau 50
4.2.1. Bộ dữ liệu đánh giá 50
4.2.2. Các phương pháp đánh giá 51
4.2.3. Kết quả đánh giá 53
4.3. Kết chương 57
KẾT LUẬN 58
TÀI LIỆU THAM KHẢO 60

DANH MỤC HÌNH ẢNH

Hình 1.1 Sự phát triển của LLM qua các giai đoạn (Nguồn: labellerr). 1
Hình 2.1 LLM sinh ra từ tiếp theo dựa trên các từ trước đó (*nguồn*) 7
Hình 2.2 Kiến trúc RAG 9
Hình 2.3 Minh họa luồng hoạt động cơ bản của hệ thống Naive RAG. 10
Hình 2.4 Minh họa luồng hoạt động cơ bản của hệ thống HyDE RAG 11
Hình 2.5 Minh họa luồng hoạt động cơ bản của hệ thống Corrective RAG 11
Hình 2.6 Minh họa luồng hoạt động cơ bản của hệ thống Graph RAG 11
Hình 2.7 Minh họa luồng hoạt động cơ bản của hệ thống Hybrid RAG 12
Hình 2.8 Minh họa luồng hoạtđộng cơ bản của hệ thống Agentic RAG. 12
Hình 2.9 Ví dụ về một đồ thị tri thức (nguồn). 14
Hình 3.1 Luồng hoạt động của hệ thống 21
Hình 3.2 Luồng hoạt động được bổ sung thêm bước requery 22
Hình 3.3 Luồng hoạt động đầy đủ của hệ thống. 24
Hình 3.4 Luồng hoạt động của Graph-RAG 27
Hình 3.5 Cấu trúc đồ thị tri thức cho báo cáo tài chính. 30
Hình 3.6 Báo cáo thành phần Bảng cân đối kế toán trong báo cáo tài chính 32
Hình 3.7 Bảng cân đối kế toán được trình bày trong báo cáo tài chính 33
Hình 3.8 Hình minh họa các mục con của Báo cáo kết quả hoạt động kinh doanh 36
Hình 3.9 Đồ thị của một báo tài chính trong cơ sở dữ liệu đồ thị 37
Hình 3.10 Hình minh họa đồ thị của báo cáo tài chính ngân hàng trong cơ sở dữ liệu đồ thị 39
Hình 3.11 Minh họa đồ thị của báo cáo tài chính công ty chứng khoán 41
Hình 3.12 Giao diện tổng quan của hệ thống 48
Hình 4.1 Câu hỏi, trả lời của chatbot và câu trả lời mẫu 53
Hình 4.2 Câu trả lời của chatbot phân tích 56
Hình 4.3 Câu trả lời được phân tích bởi con người 56

DANH MỤC BẢNG

Bảng 1.1 Bảng so sánh hai kiến trúc mạng học sâu truyền thông và Transformer 8

Bảng 3.1 Tên gọi khác có thể của một số mục lớn 33

Bảng 3.2 Thông tin về mục lớn và các mục con tương ứng 34

Bảng 3.3 Thông tin về các mục con có trong báo cáo tài chính của ngân hàng 38

Bảng 3.4 Thông tin về các mục con (subsection) của công ty chứng khoán. 40

Bảng 4.1 Kết quả đánh giá LLM và SLM dựa trên các chỉ số 52

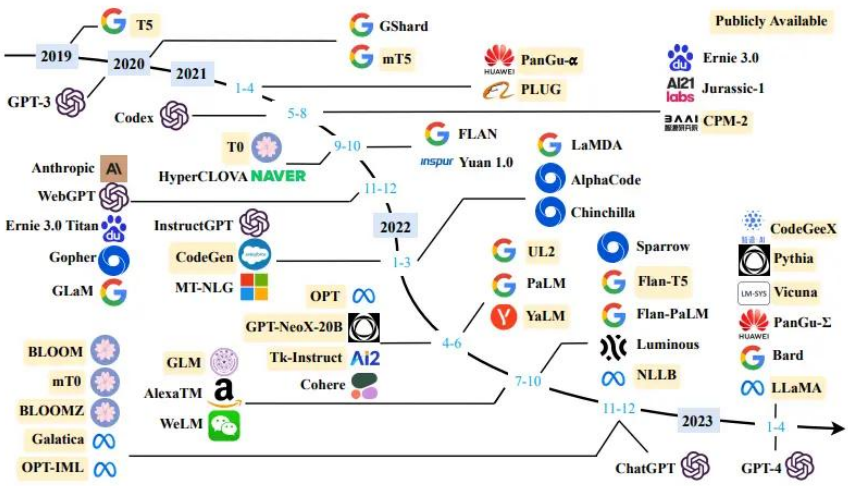
THUẬT NGỮ VÀ TỪ VIẾT TẮT

Từ viết tắt	Từ gốc	Ý nghĩa
API	Application Programming Interface	Giao diện lập trình ứng dụng
Chunk	Chunk	Đoạn văn bản
CoT	Chain of Thought	Một kỹ thuật đặt prompt cho LLM
KD	Knowledge Graph	Đồ thị tri thức
Keyword	Keyword	Từ khóa
LLM	Large Language Model	Mô hình ngôn ngữ lớn
LSTM	Long short-term memory	Mạng học sâu cải tiến dựa trên RNN
No-SQL	Non-relational SQL	Cơ sở dữ liệu phi quan hệ
OCR	Optical Character Recognition	Nhận dạng ký tự quang học
RAG	Retrieval Augmented Generation	Truy xuất thông tin tăng cường
RNN	Recurrent Neural Network	Mạng nơ ron học sâu chuyên cho dữ liệu tuần tự
ROA	Return on assets	Đo lường lợi nhuận tạo ra từ tổng tài sản của doanh nghiệp
ROE	Return on Equity	Đo lường lợi nhuận tạo ra từ vốn chủ sở hữu của doanh nghiệp
SDK	Software Development Kit	Bộ công cụ phát triển
Section	Section	Mục lớn
Subsection	Subsection	Mục con
UI	User interface	Giao diện người dùng

CHƯƠNG 1 GIỚI THIỆU CHUNG

1.1. Đặt vấn đề

Trong bối cảnh sự phát triển vượt bậc của công nghệ đặc biệt là trong lĩnh vực Trí tuệ nhân tạo. Với xử lý ngôn ngữ tự nhiên sự ra đời của kiến trúc Transformer [22] đã tạo nên một bước đột phá trong lĩnh vực ứng dụng AI trong các tác vụ liên quan đến xử lý ngôn ngữ tự nhiên. Đó cũng là nền tảng cho sự ra đời của các mô hình ngôn ngữ lớn (LLM). Với Transformer, cho phép các LLM có khả năng hiểu ngôn ngữ một cách đáng kinh ngạc, nó giúp LLM có thể hiểu sâu sắc ngữ cảnh, ngữ nghĩa của câu hỏi do người dùng đặt ra đồng thời trả lời một cách chính xác trên tập dữ liệu khổng lồ đã được huấn luyện trước đó trên toàn bộ lĩnh vực của đời sống. Sự ra đời của LLM đã làm ảnh hưởng lớn tới thói quen sử dụng tìm kiếm thông tin của người dùng: giờ đây hầu hết các câu hỏi thắc mắc trên tất cả các lĩnh vực của đời sống chúng ta đều có thể đặt câu hỏi cho LLM và nhận được câu trả lời một cách nhanh chóng, đúng trọng tâm câu hỏi hơn là việc tìm kiếm trên các công cụ tìm kiếm bình thường.



Hình 1.1 Sự phát triển của LLM qua các giai đoạn (Nguồn: labellerr).

Trải qua quá trình hình thành và phát triển. Các mô hình LLM gần như đã đạt tới khả năng hiểu biết ngôn ngữ tự nhiên, hiểu biết kiến thức về các ngành nghề, lĩnh vực cụ thể trong đời sống vượt xa một người bình thường. Các nghiên cứu gần đây cho thấy các LLM gần như đã được huấn luyện trên các bộ dữ liệu khổng lồ với kiến thức được bao phủ trên toàn bộ ngành nghề lĩnh vực do đó khả năng hiểu biết của chúng vượt xa một người bình thường và có thể cung cấp thông tin kiến thức cho con người trên bất kỳ chủ đề gì. Tuy nhiên bên cạnh những khả năng đáng kinh ngạc mà LLM, khi nhìn nhận thực tế, chúng vẫn còn một số hạn chế như khả năng suy luận

(seasoning) từ các tác vụ phức tạp liên quan đến tính toán số liệu, hay khả năng giải thích (interpretability), tính chính xác của thông tin (Safety & bias), bịa đặt thông tin (hallucination), hay các vấn đề liên quan đến bảo mật thông tin riêng tư. Với kiến thức được trang bị đầy đủ như vậy việc áp dụng LLM vào các trong một lĩnh vực cụ thể trở thành xu hướng mới trong những năm gần đây. Việc tích hợp LLM và trong các hệ thống thay vì chỉ đơn giản hỏi và đáp làm cải thiện các hạn chế bên trên của LLM đồng thời giúp xây dựng các hệ thống tích hợp AI phù hợp với từng ngành nghề lĩnh vực riêng biệt.

Tài chính là một lĩnh vực vừa là cơ hội vừa là thách thức đối với LLM. Ngày nay trên lĩnh vực tài chính LLM được ứng dụng vô cùng rộng rãi như: trợ lý khách hàng, chatbot tài chính, phân tích xử lý tài liệu, tư vấn cá nhân hóa danh mục đầu tư, phát hiện gian lận, phân tích tin tức tài chính, phân tích cảm xúc thị trường,…Do đó việc xây dựng các hệ thống kết hợp với LLM trở thành chủ đề hot đối với các nhà đầu tư startup công nghệ cũng như các doanh nghiệp tài chính ngân hàng. Việc sử dụng LLM giúp cho các chuyên gia trong lĩnh vực tài chính của các doanh nghiệp có thể có các góc nhìn mới về thị trường, tổng hợp thông tin từ nhiều nguồn khác nhau, hỗ trợ việc tìm kiếm thông tin phân tích, đưa ra quyết định giúp việc phân tích nhận định tình hình tài chính của doanh nghiệp một cách hiệu quả hơn, tiết kiệm chi phí vận hành nhân lực. Hay đối với nhà đầu tư việc tận dụng LLM kết hợp với kiến thức tài chính của bản thân có thể tạo nền tảng vững vàng hơn trước khi đưa ra quyết định đầu tư. Hoặc đối với khách hàng, người mới làm quen với tài chính, các hệ thống này cũng giúp họ có thể học hỏi quan sát, tìm kiếm thông tin, kiến thức tài chính. Tuy nhiên đây cũng là một lĩnh vực khó do tính chất thay đổi liên tục của thị trường, các con số thống kê có thể là một thách thức lớn đối với cả LLM và con người, việc đưa ra các phân tích dự đoán yêu cầu khả năng tổng hợp thông tin từ rất nhiều nguồn số liệu qua nhiều khoảng thời gian, có thể tốn nhiều thời gian, công sức, chi chí. Việc thế có thể tận dụng LLM để xử lý lượng lớn thông tin, rút ra các nhận định, phân tích sơ bộ cho người dùng trở thành việc có ý nghĩa đối với con người giúp tiết kiệm công sức và tăng năng suất độ hiệu quả công việc.

Báo cáo tài chính là một tài liệu vô cùng quan trọng của bất kỳ doanh nghiệp thuộc bất kỳ lĩnh vực nào. Việc phân tích báo cáo tài chính một cách chính xác đòi hỏi người phân tích cần có kiến thức chuyên môn rất sâu về lĩnh vực tài chính kế toán. Theo thông tin từ Misa meInvoice [6] "***Phân tích báo cáo tài chính****: là quá trình xem xét, kiểm tra, đối chiếu, so sánh số liệu và đưa ra đánh giá về tình hình tài chính*

doanh nghiệp trong kỳ hiện tại với các kỳ kinh doanh đã qua. Từ đó, giúp doanh nghiệp, ngân hàng, nhà đầu tư và các bên liên quan ra quyết định kinh tế phù hợp nhất".

Do là một dạng dữ liệu ngôn ngữ tự nhiên, do đó việc LLM có thể đọc hiểu là hoàn toàn khả thi và có nhiều tiềm năng để phát triển. Việc sử dụng LLM để phân tích đọc hiểu báo cáo tài chính trở thành một trong những tác vụ quan trọng và có tiềm năng trong tương lai. Với LLM các nhà phân tích không chỉ dựa vào khả năng của mình mà còn có thêm trợ lý đồng hành giúp việc tìm kiếm thông tin, đọc hiểu, tính toán các chỉ số tài chính, phát hiện bất thường, nhân định về tình hình tài chính,… từ đó có thể đưa ra quyết định một cách có hiệu quả hơn dựa vào sự hỗ trợ của LLM và khả năng chuyên môn của mình. Bên cách đó LLM cũng giúp cho những người dùng, khách hàng những người không chuyên về tài chính vẫn có thể đọc hiểu và nhìn ra tình hình tài chính của công ty thông qua báo cáo tài chính.

1.2. Bài toán

Trong bối cảnh trên, khóa luận này tập chung vào việc phát triển một hệ thống ứng dụng LLM trong tác vụ việc đọc hiểu và phân tích báo cáo tài chính từ đó có khả năng trả lời các câu hỏi của người dùng yêu cầu xoay quanh một báo cáo tài chính. Mục tiêu là xây dựng một hệ thống, luồng xử lý dữ liệu báo cáo tài chính một cách chính xác nhất đảm bảo LLM có thể có khả năng truy xuất chính xác thông tin, cùng với cung cấp đủ thông tin để thực hiện các bước phân tích đánh giá sâu hơn. Báo cáo tài chính của một doanh nghiệp là một loại tài liệu dài và chứa rất nhiều thông tin, trung bình một báo cáo có thể dài tới 40 - 60 trang hoặc thậm chí là dài hơn đối với các doanh nghiệp lớn việc khai thác và tận dụng được nguồn dữ liệu lớn này vừa là cơ hội vừa là thách thức. Để LLM có thể thực hiện truy xuất thông tin trả lời câu hỏi một cách chính xác nhất, đòi hỏi hệ thống cần có các khung kỹ thuật xử lý tăng cường dữ liệu như RAG giúp LLM có đủ thông tin để thực hiện tác vụ do người dùng yêu cầu và phân tích đánh giá tình hình tài chính của công ty một cách chính xác nhất và các kỹ thuật Chain-Of-Thought (CoT), Instruction prompt phát huy tối đa khả năng hiểu biết của LLM.

Với mục tiêu trên câu hỏi nghiên cứu được đặt ra là:

- Có những cách tiếp cận nào để khai thác lượng thông tin từ một báo cáo tài chính dài như vậy?

- Liệu LLM có thể phân tích đánh giá và suy luận trên báo cáo tài chính của một doanh nghiệp tốt dựa trên các thông tin được cung cấp và hướng dẫn ?

Phạm vi nghiên cứu của khóa luận này tập chung vào việc xử lý đơn lẻ một báo cáo tài chính của một doanh nghiệp trong một giai đoạn nhất định ở Việt Nam. Các báo cáo tài chính được trình bày theo chuẩn báo cáo tài chính theo quy định của Bộ tài chính Việt Nam gồm đầy đủ bốn mục chính Bảng cân đối kế toán, Báo cáo kết quả hoạt động, Báo cáo lưu chuyển tiền tệ và Thuyết minh báo cáo tài chính. Ngày nay phần lớn các doanh nghiệp đều công bố báo cáo tài chính của mình trên các diễn đàn về tài chính trên internet vì thế việc truy cập đến các tài liệu này trở nên vô cùng phổ biến và dễ dàng đối với bất kỳ người dùng internet nào chỉ cần có kết nối online, bất kỳ ai cũng có thể có được báo cáo tài chính của bất kỳ doanh nghiệp nào.

Từ đó phát triển một giao diện website Chatbot tương tác giữa người dùng và máy giúp người dùng có thể dễ dàng tương tác thông qua giao diện website. Người dùng có thể dễ dàng tải lên các file báo cáo tài chính cùng với câu hỏi của mình một cách thuận tiện nhất . Hướng đến đối tượng người dùng là các nhà phân tích tài chính, việc sử dụng LLM như một trợ lý bên cạnh khả năng chuyên môn giúp các chuyên gia có thể tiết kiệm thời gian tính toán các chỉ số, dò tìm dữ liệu thủ công, hỗ trợ việc phân tích nhận định tình hình. Bên cạnh đó hệ thống cũng giúp cho những người không chuyên cũng có thể học tập, đọc hiểu được báo cáo tài chính bằng việc đặt các câu hỏi đáp xung quanh đó, nhận định tình hình tài chính của bất kỳ doanh nghiệp nào đóng vai trò như một ứng dụng giáo dục về đọc hiểu báo cáo tài chính.

1.3. Đóng góp và cấu trúc của khóa luận

Đóng góp chính của khóa luận là đề xuất một luồng xử lý dữ liệu chuyên biệt cho báo cáo tài chính, nhằm khai thác tối đa thông tin và tối ưu hóa khả năng truy xuất tri thức. Luồng xử lý này được xây dựng dựa trên kiến trúc RAG (Retrieval-Augmented Generation) kết hợp với đồ thị khi thức Knowledge Graph: biểu diễn cấu trúc của các báo cáo tài chính hỗ trợ việc truy xuất thông tin một cách nhanh chóng và chính xác, giúp mô hình LLM có thể tiếp cận đầy đủ thông tin cần thiết để trả lời câu hỏi và phân tích tình hình tài chính một cách chính xác. Bên cạnh đó, khóa luận cũng áp dụng các kỹ thuật prompting cải tiến nhằm giúp LLM hiểu rõ nhiệm vụ, nắm bắt chính xác ngữ cảnh câu hỏi, và tuân theo các hướng dẫn chuyên môn trong lĩnh vực phân tích báo cáo tài chính. Qua đó, hệ thống chatbot có thể đưa ra câu trả lời đúng trọng tâm, đúng định dạng, đồng thời hạn chế tối đa các hạn chế thường gặp của LLM như sai lệch trong suy

luận, thiếu khả năng giải thích, tính thiên vị, hay việc "bịa thông tin" khi không có dữ liệu phù hợp.

Khóa luận được trình bày theo cấu trúc:

- **Chương 2:** Tổng quan về báo cáo tài chính, các chuẩn mực về lập báo cáo tài chính cho các doanh nghiệp ở Việt Nam và các nền tảng công nghệ được sử dụng để nghiên cứu áp dụng xử lý dữ liệu báo cáo tài chính và xây dựng hệ thống chatbot.
- **Chương 3:** Chi tiết về các phương pháp RAG đã triển khai trong hệ thống và phương pháp đề xuất sử dụng đồ thị tri thức để biểu diễn báo cáo tài chính. Từ đó xây dựng giao diện website chatbot giúp người dùng dễ dàng tương tác với hệ thống.
- **Chương 4:** Một số phân tích đánh giá thử nghiệm hệ thống qua các chỉ số đánh giá dành cho các hệ thống RAG.
- **Kết luận:** Kết luận nhận định về các kết quả đạt được và các hạn chế còn tồn tại của phương pháp từ đó đề xuất hướng phát triển trong tương lai.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1. Kiến thức về báo cáo tài chính

Theo thông tư 99/2025/TT-BTC [2] Báo cáo tài chính được hiểu là một bản tổng hợp các số liệu kinh tế, được trình bày bằng các bảng biểu, nhằm phản ánh toàn diện thực trạng tài chính, kết quả hoạt động kinh doanh và dòng tiền của doanh nghiệp trong một kỳ nhất định. Tại điều 17 trong thông tư quy định rõ hệ thống báo cáo tài chính của doanh nghiệp gồm có:

- Báo cáo tình hình tài chính
- Báo cáo kết quả hoạt động kinh doanh
- Báo cáo lưu chuyển tiền tệ
- Bản thuyết minh Báo cáo tài chính

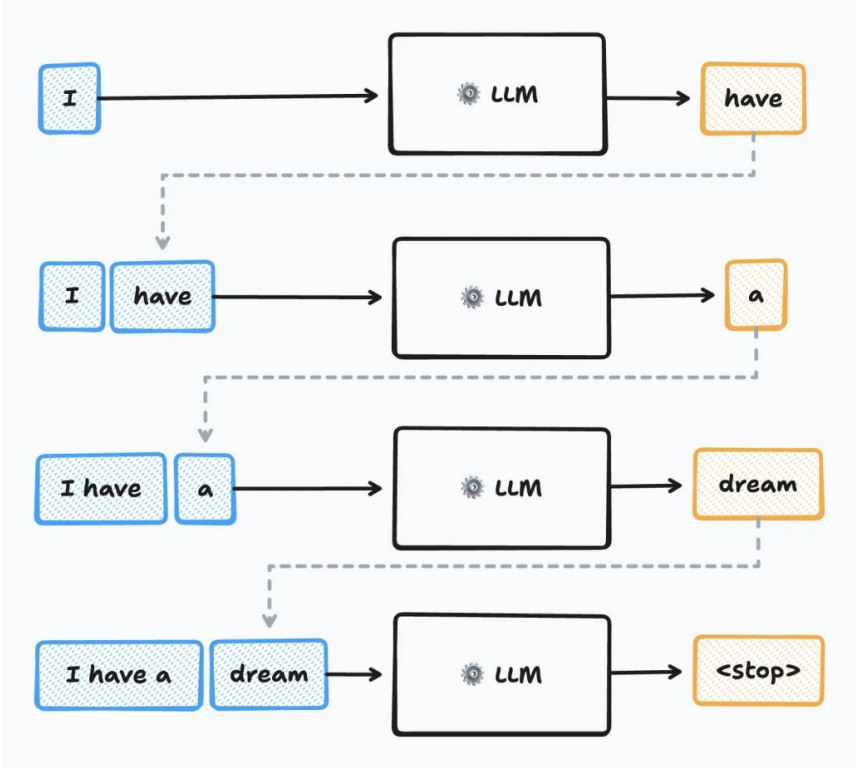
Đối với mỗi loại báo cáo thành phần đều có mẫu đi kèm tương ứng với từng loại báo cáo tài chính theo năm, giữa niên độ, theo từng loại doanh nghiệp, dạng đầy đủ hoặc tóm lược. Tại điều 20 trong thông tư quy định rõ việc lập và trình bày báo cáo tài chính phải tuân thủ theo Chuẩn mực kế toán Việt Nam số 21 (VAS 21) [3].

Bên cách đó việc đọc hiểu và phân tích báo cáo tài chính liên quan đến các phần được trình bày trong đó bao gồm: nguồn vốn, tài sản, doanh thu và lợi nhuận, chi phí, lưu chuyển tiền tệ, các chỉ số tài chính quan trọng, tình hình tài chính tổng thể, các sự kiện và chính sách đặc biệt, dự báo và phân tích tương lai, phân tích rủi ro.

2.2 Nền tảng công nghệ

2.2.1. Mô hình ngôn ngữ lớn

Mô hình ngôn ngữ lớn LLM (Large Language Model) [11] là một loại mô hình trí tuệ nhân tạo dựa trên các kiến trúc mạng nơ-ron nhân tạo học sâu (Deep Learning) rất lớn được huấn luyện với lượng dữ liệu văn bản khổng lồ để có khả năng hiểu biết, nhận thức, kiến thức về ngôn ngữ giống như con người. Về bản chất LLM huấn luyện để sinh ra, dự đoán một từ tiếp theo dựa trên ngữ cảnh đầu vào mà người dùng cung cấp dựa trên một mạng nơ-ron học sâu phức tạp.



Hình 2.1 LLM sinh ra từ tiếp theo dựa trên các từ trước đó ([19])

Trước khi có sự ra đời của kiến trúc Transformer: một kiến trúc mạng nơ-ron hiện đại được sử dụng trong hầu hết LLM hiện nay, thì phần lớn sử dụng kiến trúc mạng nơ-ron hồi quy cho dữ liệu dạng chuỗi như RNN, LSTM,... Các kiến trúc này được thiết kế để xử lý dữ liệu dạng chuỗi liên tục như văn bản, tín hiệu, với ưu điểm là có thể xử lý và ghi được các dạng thông tin tuần tự, thông tin sau phụ thuộc vào thông tin trước đó, giúp cho mô hình có thể hiểu được mối liên hệ giữa các từ câu, khung thời gian, …Do đó giúp mô hình có khả năng ghi nhớ ngắn hạn các thông tin giúp cho việc hiểu ngữ cảnh của một câu văn, một đoạn văn bản trở lên tốt hơn, được ứng dụng trong các bài toán về dịch máy, nhận dạng giọng nói. Tuy nhiên khi lượng dữ liệu trở lên lớn hơn, đầu vào của các mô hình không chỉ là một câu văn ngắn, mà có thể là cả một đoạn văn bản dài tới hàng trăm hàng nghìn token, việc tính toán và xử lý tuần tự trở thành một vấn đề lớn khi các kiến trúc này mất quá nhiều thời gian để tính toán việc phải tính toán lại từ đầu do kết quả phụ thuộc vào phép tính trước đó làm cho thời gian tính toán tăng tuyến tính. Điều này không chỉ gân nên vấn đề về thời gian mà còn cả độ chính xác ngữ cảnh, hiện tượng vanishing gradient xảy ra khi lượng đầu vào quá lớn dẫn đến mất mát đi các thông tin về ngữ cảnh khiến cho việc đưa ra dự đoán sinh văn bản trở nên không còn chính xác phù hợp với ngữ cảnh câu hỏi đầu vào ban đầu.

Transformer ra đời để giải quyết đi vấn đề đó, với cơ chế chú ý (Attention Mechanism) [22] của mình kiến trúc đó đã giải quyết đi hoàn toàn các nhược điểm các của mạng nơ-ron hồi quy. Với cơ chế này cho phép mô hình xử lý được song song các chuỗi đầu vào thay vì xử lý tuần tự. Việc so sánh và "chú ý" tới các từ quan trọng có ý nghĩa trong ngữ cảnh của đầu vào với toàn bộ đầu vào một cách song song giúp mô hình tránh được hiện tượng mất đi thông tin ngữ cảnh, đảm bảo việc hiểu đúng ý nghĩa mục đích của câu hỏi.

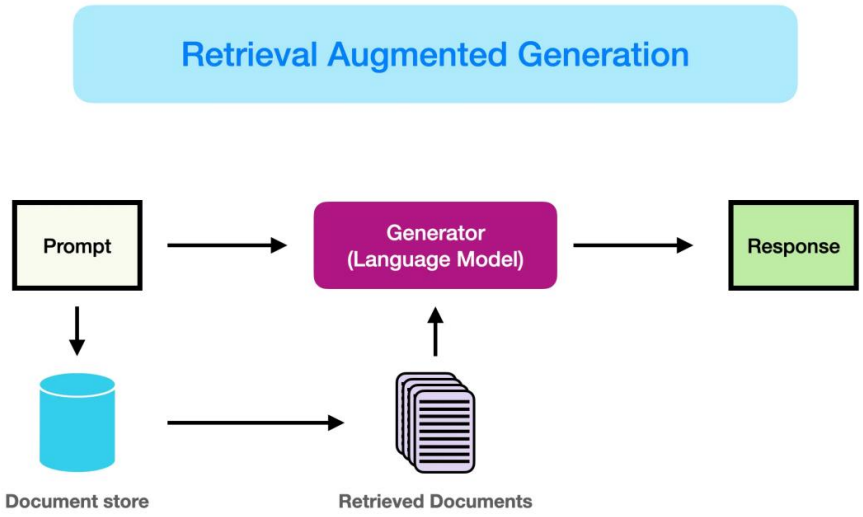
Bảng 1.1 Bảng so sánh hai kiến trúc mạng học sâu truyền thông và Transformer

	Mạng nơ-ron hồi quy (RNN, LSTM, …)	**Transformer**
Cơ chế tính toán	Tuần tự	Song song
Khả năng ghi nhớ ngữ cảnh	Ngắn hạn	Dài hạn
Tốc độ huấn luyện	Chậm do phải tính toán tuần tự từ đầu cho mỗi từ mới.	Rất nhanh khi xử lý đồng thời toàn bộ các từ.
Khả năng diễn giải	Hạn chế do khả năng ghi nhớ kém	Cao do cơ chế self-attention giữa các từ.
Bộ nhớ khi huấn luyện	Yêu cầu ít bộ nhớ do việc lưu trữ các trạng thái ẩn	Yêu cầu bộ nhớ lớn do lưu giữ các ma trận Attention.

Nhờ đó ngày này Transformer trở thành trái tim của hầu hết các mô hình ngôn ngữ lớn giúp cho chúng có khả năng học hỏi đáng kinh ngạc vượt xa khả năng của một người bình thường. LLM được huấn luyện trên các bộ dữ liệu văn bản khổng lồ trên toàn bộ lĩnh vực của đời sống ngày này, với kích thước lớn từ hàng tỷ đến trăm tỷ tham số. Nhớ đó LLM được ứng dụng vào trong rất nhiều các bài toán và cho ra kết các kết quả vượt bậc so với trước đây như dịch máy, việc sử dụng LLM giúp việc dịch các câu văn bản trở nên chính xác hơn phù hợp với ngữ cảnh ý nghĩa mà người dùng mong muốn. Các tác vụ hỏi đáp chatbot cũng được cải thiện đáng kể nhờ khả năng hiểu biết của LLM giúp cho chúng giờ đây trở thành trợ lý của bất kỳ ai sử dụng internet với bất kỳ ngành nghề lĩnh vực nào.

2.2.2. Kiến trúc RAG

Mặc dù được đánh giá rất cao trong khả năng ngôn ngữ, hiểu biết tuy nhiên LLM vẫn có những điểm hạn chế nhất định do chỉ trả lời được trong phạm vi kiến thức được huấn luyện. Mặc dù được huấn luyện trên một bộ dữ liệu khổng lồ nhưng điều này chỉ giúp LLM có được kiến thức tổng quát chung về lĩnh vực cụ thể đó chưa thể tiếp xúc với các kiến thức nhỏ chi tiết trên các lĩnh vực, hay các thông tin mới nhất hiện nay. Ví dụ LLM có thể có kiến thức về tài chính, về báo cáo tài chính tuy nhiên lại không thể biết được chính xác tình hình tài chính cụ thể của một doanh nghiệp do không được huấn luyện cụ thể trên bộ dữ liệu của doanh nghiệp đó và không có thông tin về báo cáo tài chính của doanh nghiệp đó. Vì thế RAG (Retrieval Augmented Generation) ra đời kết hợp với LLM để tăng cường và tận dụng tối đa các kiến thức nền tảng của LLM.



Hình 2.2 Kiến trúc RAG (nguồn [20])

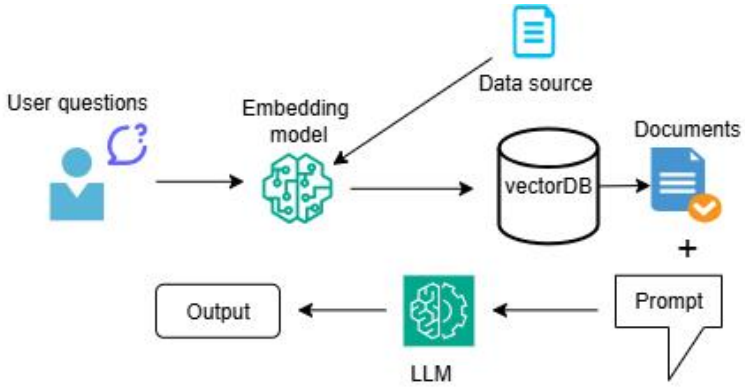
RAG [20] (Tạo sinh tăng cường truy xuất) là một kỹ thuật nhằm mục đích cung cấp thông tin dữ liệu từ bên ngoài cho LLM giúp cho LLM có kiến thức để trả lời câu hỏi do người dùng đặt ra:

- RA - Retrieval-Augmented: Truy xuất, từ câu hỏi của người dùng hệ thống sẽ tìm kiếm các thông tin có liên quan đến câu hỏi từ đó cung cấp ngữ cảnh và thông tin cho LLM giúp LLM sinh ra một cách chính xác đúng trọng tâm tránh tình trạng bịa đặt thông tin hay thông tin không có căn cứ.
- G- Generation: Sau khi đã có đủ thông tin LLM sẽ tiến hành sinh câu trả lời dựa trên thông tin đó.

Ngày nay RAG hoạt động phổ biến nhất dựa trên cơ sở dữ liệu vector: một hệ thống lưu trữ các vector đặc trưng ngữ nghĩa của tài liệu thay vì dữ liệu văn bản. Đây là một dạng cơ sở dữ liệu đặc biệt phục vụ cho việc truy xuất thông tin theo ngữ nghĩa. Trong xử lý ngôn ngữ tự nhiên, vector ngữ nghĩa là một loại dữ liệu đặc biệt quan trọng nó giúp máy tính hiểu được sự tương đồng ngữ nghĩa giữa các từ trong ngôn ngữ giống con người thông qua các con số. VectorDB đóng vai trò quan trọng trong các kiến trúc RAG là cầu nối giữa LLM và hệ thống tài liệu cung cấp. Ví dụ từ một câu hỏi của người dùng, hệ thống sẽ tiến hành đưa về dạng vector ngữ nghĩa sau đó thực hiện so sánh với dữ liệu có trong cơ sở dữ liệu Vector từ đó chọn ra đoạn có thông tin có ngữ cảnh phù hợp nhất với câu hỏi và đưa vào LLM. Hiện nay có rất nhiều kiến trúc RAG từ cơ bản đến nâng cao nhưng tất cả đều xoay quanh việc truy xuất thông tin trên cơ sở dữ liệu vector.

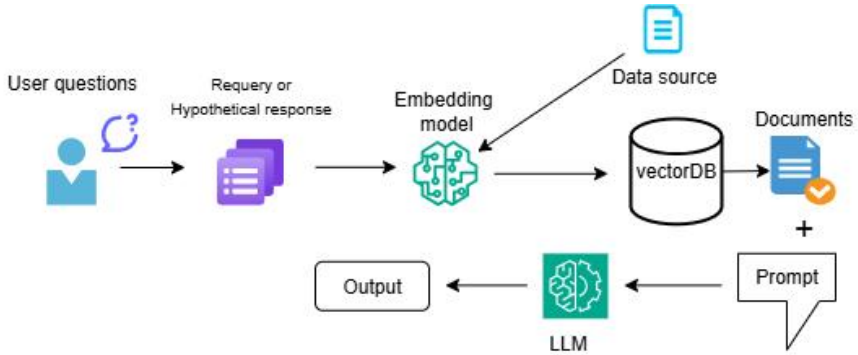
Một số kiến trúc RAG hiện nay [7]:

- **Naive RAG**: Đây là kiến trúc đơn giản nhất dễ dàng triển khai nhất chỉ bằng việc truy vấn trực tiếp vào cơ sở dữ liệu vector rồi lấy ra đoạn văn bản phù hợp nhất đưa vào prompt của LLM.



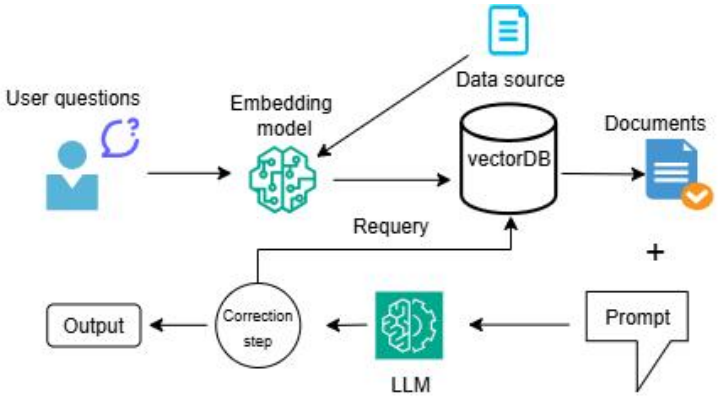
Hình 2.3 Minh họa luồng hoạt động cơ bản của hệ thống Naive RAG.

- **HyDE (Hypothetical Document Embeddings)**: Trước khi đưa vào tìm kiếm trong cơ sở dữ liệu vector. LLM cần hình dung ra mình cần những thông tin gì để trả lời câu hỏi từ đó tìm kiếm thông tin dựa trên những thông tin đó thay vì sử dụng trực tiếp câu hỏi của người dùng. Kiến trúc này thường dùng cho các bài toán có yêu cầu tính toán số liệu không có sẵn trực tiếp hay những giả định mô tả cần thông tin để tổng hợp.



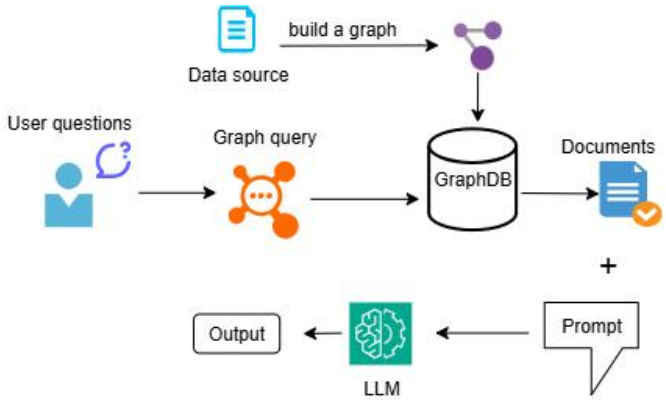
Hình 2.4 Minh họa luồng hoạt động cơ bản của hệ thống HyDE RAG

- **Corrective RAG:** Sau khi truy vấn xong hệ thống tự kiểm tra đối chiếu với các nguồn đáng tin cậy khác như search web để kiểm tra lại thông tin vừa có được hoặc câu trả lời vừa sinh ra đã đúng và phù hợp chưa.



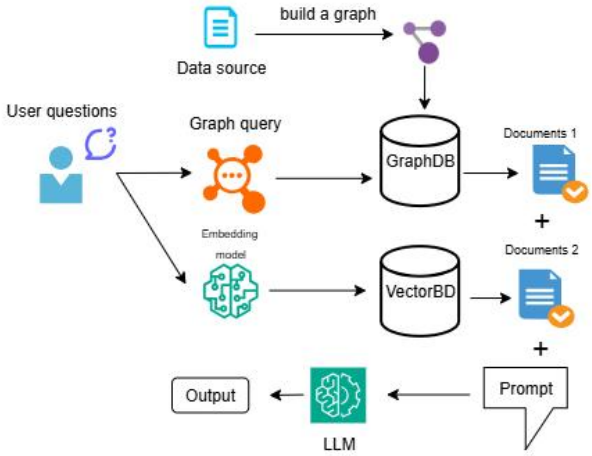
Hình 2.5 Minh họa luồng hoạt động cơ bản của hệ thống Corrective RAG

- **Graph RAG:** Truy vấn dựa trên đồ thị tri thức, đây là cách tiếp cận sử dụng cơ sở dữ liệu đồ thị biểu diễn các quan mối quan hệ trong tài liệu được cung cấp giúp cho việc truy xuất thông tin trở nên nhanh hơn chính xác đảm thông tin chính xác.



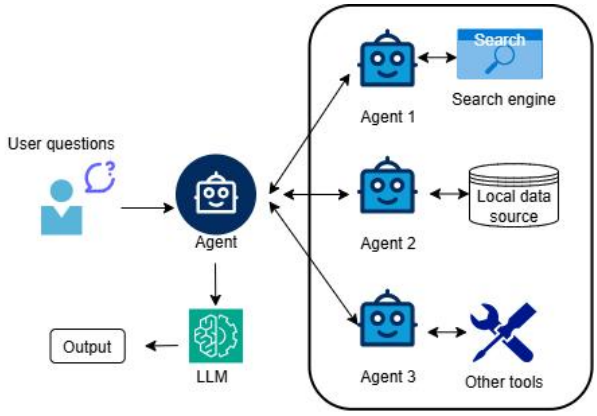
Hình 2.6 Minh họa luồng hoạt động cơ bản của hệ thống Graph RAG

- **Hybrid RAG:** Sự kết hợp giữa cơ sở dữ liệu vector ngữ nghĩa và cơ sở dữ liệu đồ thị giúp truy xuất thông tin theo cấu trúc và ngữ nghĩa.



Hình 2.7 Minh họa luồng hoạt động cơ bản của hệ thống Hybrid RAG

- **Agentic RAG:** Đây là kiến trúc xu hướng hiện này bằng việc kết hợp tận dụng sức mạnh của AI Agent, mỗi Agent sẽ đảm nhiệm một vai trò riêng cho phép hệ thống tự quyết định thực hiện tìm kiếm thông tin bằng nhiều phương pháp khác nhau (search web, duyệt đồ thị, vector ngữ nghĩa), tìm kiếm thông tin riêng rồi tổng hợp lại đưa ra câu trả lời cuối cùng.



Hình 2.8 Minh họa luồng hoạt động cơ bản của hệ thống Agentic RAG.

So với fine-tuning RAG cho phép khả năng linh hoạt hơn dễ sử dụng và tích hợp trong nhiều trường hợp bài toán không quá phức tạp có thể thay đổi dễ dàng phụ thuộc vào bài toán. Tránh được các vấn đề khi fine-tune như hao tốn tài nguyên, khối lượng dữ liệu. Trong bối cảnh LLM đã được huấn luyện rất lớn thì việc fine-tune đối với những bài toán tác vụ nhỏ trở nên không cần thiết, thay vào đó RAG sẽ là lựa chọn tối

ưu cân bằng vừa nhanh chóng dễ dàng vừa tận dụng được lượng tri thức khổng lồ có sẵn của LLM. Tuy nhiên trong một số bài toán với những yêu cầu đặc thù fine-tune vẫn là sự lựa chọn đáng cân nhắc [14].

2.2.3. Nhận dạng ký tự quang học

Optical Character Recognition - OCR [12]: Nhận dạng ký tự quang học là công cụ giúp máy tính có thể hiểu được nội dung của một hình ảnh, các tài liệu viết tay, scan hay các loại tài liệu mà không thể trực tiếp sao chép, tìm kiếm. Mục tiêu của công nghệ này là biến đổi một hình ảnh hay một dạng tài liệu đặc biệt nào đó sang một loại mà máy tính có thể xử lý được như PDF, Word Markdown.

Quy trình OCR thường gồm 3 bước chính:

1. **Tiền xử lý ảnh (Preprocessing):** Làm sạch ảnh (lọc nhiễu, tăng độ tương phản, làm phẳng trang giấy…) để văn bản rõ nét hơn.
2. **Nhận dạng ký tự (Character Recognition):** Mô hình OCR (thường là CNN, RNN hoặc Transformer) sẽ **phân tích từng ký tự hoặc chuỗi ký tự**, nhận dạng chữ, số, ký hiệu.
3. **Hậu xử lý (Postprocessing):** Hiệu chỉnh lỗi chính tả, ghép dòng, nhận diện ngữ cảnh để tăng độ chính xác.

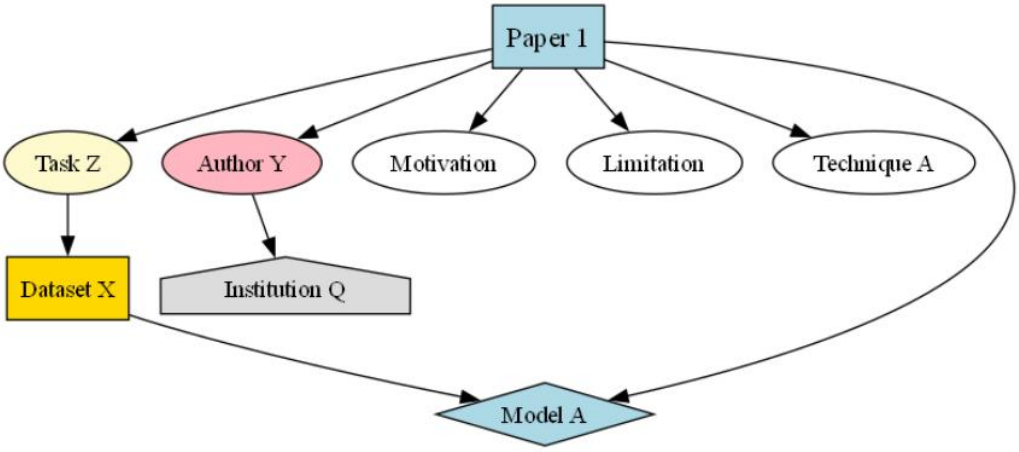
Ngày nay LLM cũng được sử dụng trong OCR để nâng cao độ chính xác qua việc kiểm tra phát hiện lỗi ngôn ngữ, kiểm tra cấu trúc, bố cục các văn bản đảm bảo tính hợp lý.

Với báo cáo tài chính thì đây là bước không thể thiếu. Hầu hết khi được công bố các báo cáo tài chính của công ty thường ở dạng PDF scan không thể sao chép hay tìm kiếm số liệu một cách trực tiếp vì thế việc phải xử lý OCR để có thể lấy được thông tin từ đó là việc bắt buộc.

2.2.4. Đồ thị tri thức và cơ sở dữ liệu Neo4j

Đồ thị tri thức (Knowledge Graph) [17] là mô hình biểu diễn tri thức dưới dạng đồ thị bao gồm các:

- Nút: đại diện cho các thực thể chính trong đó các thực thể cũng có thể có các thuộc tính riêng đi kèm.
- Cạnh: biểu diễn mối quan hệ giữa các thực thể đó.



Hình 2.9 Ví dụ về một đồ thị tri thức (nguồn [18])

Đồ thị tri thức giúp chúng ta có thể quan sát được trực quan mối quan hệ giữa các thực thể với nhau hỗ trợ việc suy diễn giúp việc khai thác thông tin trở nên dễ dàng hơn trực quan hơn.

Mở rộng hơn không chỉ quan hệ giữa các thực thể trong một đồ thị mà nhiều đồ thị với nhau lúc này một cơ sở dữ liệu đồ thị tri thức được ra đời, nó giống như một bản đồ tri thức giúp xác định và truy xuất đến bất kỳ nguồn thông tin nào có trong đồ thị dựa trên các biểu diễn quan hệ giữa các thực thể đã được xác định trước đó.

Neo4j là một hệ cơ sở dữ liệu đồ thị được phát triển bởi Neo4j Inc chuyên dụng giúp cho việc biểu diễn và truy vấn tới đồ thị thông qua các nút, các cạnh và thuộc tính một các dễ dàng nhanh chóng. Cypher là ngôn ngữ truy vấn riêng được sử dụng giúp cho việc nhập xuất dữ liệu mạnh mẽ nhanh chóng phù hợp.

Với tính chất của các báo cáo tài chính việc có thể phân tách cấu trúc của báo cáo tài chính và biểu diễn dưới dạng đồ thị giúp việc tìm kiếm thông tin trở nên dễ dàng, nhanh chóng hơn tận dụng được mối quan hệ tự nhiên giữa các báo cáo tài chính.

2.2.5. NoSQL và MongoDB

NoSQL thường được hiểu là Not Only SQL hoặc non-relational là một loại hệ quản trị cơ sở dữ liệu nhưng không tuân theo mô hình quan hệ truyền thông bao gồm các hàng cột. Nhớ đó cho phép việc lưu trữ dữ liệu nhiều định dạng khác nhau như document, key-value, wide-column, … linh hoạt và không theo một schema cố định từ đó giúp cho việc mở rộng và khả năng xử lý dữ liệu lớn không đồng nhất cấu trúc một cách dễ dàng tiện lợi.

Một số loại NoSQL tiêu biểu:

- **Key-Value Stores**: mỗi mục dữ liệu được lưu trữ dưới dạng cặp khóa-giá trị (ví dụ Redis, DynamoDB).
- **Document Stores**: lưu trữ dữ liệu như tài liệu JSON/BSON (ví dụ MongoDB)
- **Wide-Column Stores**: lưu trữ theo dạng bảng nhưng có cột linh hoạt và phù hợp cho dữ liệu lớn (ví dụ Apache Cassandra).
- **Graph Databases**: lưu trữ và truy vấn dữ liệu dưới dạng đồ thị (nodes & edges) như Neo4j.

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL được phát triển bởi MongoDB Inc lưu trữ dữ liệu dưới dạng tài liệu (document) có cấu trúc như JSON. Điều này đặc biệt phù hợp với những loại dữ liệu có cấu trúc thay đổi nhanh, không rõ cấu trúc hay không chú trọng đến các biểu diễn quan hệ. Do đó các ứng dụng web thường phù hợp với MongoDB nhờ khả năng mở rộng, linh hoạt dễ triển khai và truy vấn linh hoạt.

Trong hệ thống website chatbot, việc lưu trữ các đoạn hội thoại của người dùng và thông tin về các tài liệu mà họ tải lên vốn không chứa dữ liệu nhạy cảm nhưng có mối quan hệ phức tạp đòi hỏi một cơ sở dữ liệu linh hoạt và dễ mở rộng. MongoDB là lựa chọn tối ưu cho mục đích này, nhờ khả năng lưu trữ dữ liệu phi cấu trúc, truy xuất nhanh chóng và hỗ trợ mở rộng quy mô hiệu quả.

2.2.6. Các thư viện và công cụ hỗ trợ

Django.

Django là một framework phát triển ứng dụng web cao cấp của Python được thiết kế kế để xây dựng một website nhanh chóng, bảo mật dễ bảo trì và mở rộng. Django cũng dựa trên kiến trúc MVT (Model - View - Template) hay tương đương với MVC (Model - View -Control). Django cũng cung cấp sẵn các các chức năng cơ bản thường gặp khi phát triển web như xác thực người dùng, kết nối cơ sở dữ liệu, các template,… Do đó giúp lập trình viên có thể xây dựng các ứng dụng web đầy đủ cả frontend và backend trong thời gian ngắn phục vụ cho các mục đích ngắn như demo sản phẩm hay phát triển hệ thống.

Các thành phần chính:

- **Model**: định nghĩa cấu trúc dữ liệu, liên kết tới cơ sở dữ liệu.
- **View**: xử lý logic ứng dụng khi có yêu cầu từ người dùng.
- **Template**: giao diện hiển thị HTML (hoặc format khác) cho người dùng.

- **URL dispatcher**: định tuyến các đường dẫn URL tới View tương ứng.
- **Admin**: giao diện quản trị tự động tạo dựa trên Models để quản lý dữ liệu.

Đối với các ứng dụng web tích hợp trí tuệ nhân tạo (AI), Django là lựa chọn hoàn hảo nhờ khả năng triển khai linh hoạt và dễ dàng tích hợp các mô hình AI vào hệ thống. Việc Django được xây dựng hoàn toàn bằng ngôn ngữ Python giúp quá trình phát triển trở nên thống nhất, giảm thiểu chi phí tích hợp và tiết kiệm thời gian trong việc xây dựng giao diện người dùng cũng như kết nối giữa frontend và backend.

FAISS

FAISS (Facebook AI Similarity Search) là một thư viện mã nguồn mở của Facebook phát triển. Giúp việc lưu trữ và tìm kiếm thông tin dựa trên vector ngữ nghĩa. Có thể hiểu FAISS giống như một vectorDB hỗ trợ lưu trữ tạm thời dữ liệu dưới dạng vector embedding từ đó giúp việc tìm kiếm thông tin tài liệu dựa trên ngữ cảnh dễ dàng hơn.

Embedding model

Embedding model là một mô hình giúp biến đổi các câu văn từ ngữ từ ngôn ngữ tự nhiên sang biểu diễn vector. Điều này vô cùng quan trọng đối với các bài toán xử lý ngôn ngữ tự nhiên bởi đây có thể coi là bước dịch từ ngôn ngữ tự nhiên sang ngôn ngữ dành cho máy. Việc các câu từ được biểu diễn thành các vector trong không gian giúp truyền tải được mối quan hệ ngữ nghĩa giữa các từ với nhau thông qua khoảng cách giữa chúng trong không gian vector. Embedding model càng tốt thì LLM càng có khả năng hiểu biết về ngôn ngữ sâu.

LlamaCloud services

LlamaCloud [16] là dịch vụ điện toán đám mây do LlamaIndex phát triển, cung cấp các ứng dụng giải pháp để xử lý các tài liệu tạo cơ sở tri thức để kết hợp với LLM và ứng dụng trong các bài toán RAG giúp xử lý các loại tài liệu để có thể đọc được tìm kiếm được. LlamaCloud giúp các nhà phát triển có thể dễ dàng thử nghiệm tích hợp vào hệ thống thông qua API, SDK (Python/TypeScript) và giao diện web.

LlamaCloud gồm 3 thành phần chính:

- **Parse (Phân tích)**: chuyển đổi tài liệu phức tạp thành dữ liệu có thể đọc hiểu tìm kiếm phục vụ cho việc trích xuất thông tin.

- **Extract (Trích xuất)**: từ nội dung vừa được phân tích, trích xuất thông tin theo schema (ví dụ: ngày, tên, số, bảng dữ liệu) thành JSON có cấu trúc.
- **Index (Lập chỉ mục / Tạo cơ sở tri thức)**: đưa nội dung đã phân tích và trích xuất vào vector database hoặc môi trường truy vấn, cho phép ứng dụng RAG hoặc tìm kiếm thông tin một cách nhanh và chính xác.

Google AI studio

Google AI studio là một nền tảng do Google phát triển giúp xây dựng các ứng dụng liên quan đến generative AI nhanh chóng đơn giản. Giúp cho người dùng có thể dễ dàng tương tác thử nghiệm với các mô hình ngôn ngữ lớn, mô hình sinh ảnh, video do Google phát triển thông qua giao diện web, bên cạnh đó Google AI studio cũng cung cấp các API và SDK python.

2.3 Một số nghiên cứu liên quan

Phân tích báo cáo tài chính với LLM

Trong bài báo "Financial Statement Analysis with Large Language Models" [10] của tác giả Alex G. Kim và cộng sự đã nghiên cứu đánh giá khả năng phân tích báo cáo tài tài giữa người và máy bao gồm LLM và một số mạng học sâu được huấn luyện dành riêng cho tài chính. Trong đó mục tiêu chính là dự đoán mức thu thập của doanh nghiệp sẽ tăng hay giảm dựa trong tương lai vào Bảng cân đối kế toán và Báo cáo kết quả hoạt động kinh doanh của doanh nghiệp.

Dữ liệu được sử dụng trong quá trình huấn luyện là dữ liệu tài chính hàng năm của Compustat từ 1968 - 2021. Và lấy dữ liệu của năm 2022 làm tập kiểm tra để dự đoán năm 2023. Toàn bộ tập dữ liệu gồm 150.678 quan sát từ 15.401 công ty riêng biệt. Từ đó áp dụng các kỹ thuật prompt từ cơ bản đến nâng cao mô phỏng quá trình phân tích của con người để giúp LLM có khả năng phân tích đúng hướng và đưa ra dự đoán từ thông tin đã được cung cấp.

Dựa vào kết quả thực nghiệm nghiên cứu đã chỉ ra rằng GPT có thể làm tốt hơn con người trong việc phân tích báo cáo tài chính và cho kết quả ngang ngửa với các mô hình học sâu chuyên biệt. Tuy nhiên, con người thường có thể tận dụng được nhiều thông tin bên ngoài, do đó có lợi thế hơn so với LLM chỉ được cung cấp thông tin ngữ cảnh trong đầu vào. Nhìn chung, con người thường dựa vào thông tin mềm mà máy móc khó có thể tiếp cận được. Việc sử dụng LLM vào trong phân tích báo cáo tài chính giúp các phân tích có giá trị thông tin đáng kể, sự xuất hiện của LLM là sự bổ sung cho con người và AI sẽ đóng góp đáng kể việc ra quyết định của con người.

Phân tích tổng hợp về quy mô và sự phát triển của GenAI trong lĩnh vực xử lý ngôn ngữ tự nhiên trên miền tài chính

Trong bài báo "Prompting the Market? A Large-Scale Meta-Analysis of GenAI in Finance NLP (2022–2025)"[18] của tác giả Paolo Pedinotti và cộng sự khảo sát về sự phát triển của LLM trong lĩnh vực NLP tài chính. Sử dụng phương pháp xây dựng đồ thị tri thức MetaGraph được xây dựng dựa trên LLM giúp việc Trích xuất thông tin định lượng lớn, biểu diễn thông tin có cấu trúc, phát hiện tần suất, các mối quan hệ, một cách hoàn toàn tự động cho lượng dữ liệu lớn. Nghiên cứu được thực hiện trong 3 giai đoạn chính:

- Giai đoạn đầu: áp dụng LLM và đổi mới về nhiệm vụ/tập dữ liệu.
- Giai đoạn phản tư: nhận diện và phân tích các hạn chế của LLM.
- Giai đoạn tích hợp: kết hợp thêm các kỹ thuật phụ trợ vào các hệ thống mô-đun.

Dữ liệu từ 681 bài báo NLP cho tài chính (2022- 2025)

Mục đích nghiên cứu: Đánh giá định lượng về ảnh hưởng và sự thay đổi của GenAI trong NLP tài chính.

Theo kết quả nghiên cứu chỉ ra rằng. Trong giai đoạn đầu khi LLM chưa xuất hiện mạnh mẽ phần lớn các nghiên cứu xoay quanh các tác vụ phân tích cảm xúc, trích xuất thông tin, dự đoán giá cổ phiếu. Sau đó khi LLM ra đời các tác vụ phức tạp hơn như Financial Question Answering (QA) bắt đầu trở lên phổ biến. Trong những giai đoạn đầu khi LLM ra mắt phần lớn tập chung vào việc ứng dụng trực tiếp LLM do khả năng vượt trội của LLM giúp xóa bỏ đi các rào cản về dữ liệu trước đó. Tuy nhiên dần nhận ra các điểm yếu của LLM như:

Reasoning yếu (khó suy luận số học, tài chính phức tạp), Safety & bias (có thể đưa ra thông tin sai, lệch hoặc không an toàn), Interpretability (khó giải thích tại sao mô hình trả lời như vậy), Scalability & latency (tốn kém, chậm khi xử lý quy mô lớn). Do đó các nghiên cứu chuyển dần theo hướng tích hợp LLM vào hệ thống như RAG, Agentic và cải thiện prompting như: chain-of-thought, retrieval-based prompts, self-criticism → giúp LLM "tự suy nghĩ có hướng dẫn" thay vì chỉ dựa vào khả năng few-shot vốn có giúp cải thiện các nhược điểm của LLM.

Mô hình ngôn ngữ lớn riêng biệt cho miền tài chính

FinGPT [15] là một dự án mã nguồn mở được giới thiệu qua bài báo "FinGPT: Open-Source Financial Large Language Models " nhằm xây dựng các mô hình ngôn ngữ lớn chuyên biệt trong miền tài chính được phát triển bởi AI4Finance Foundation. FinGPT cung cấp framework từ việc thu thập dữ liệu đến huấn luyện mô hình thông qua nhiều lớp xử lý.

- Data Source Layer: Thu thập dữ liệu từ nhiều nguồn đáng tin cậy như tin tức, mạng xã hội, báo cáo công ty, xu hướng thị trường.
- Data Engineering Layer: Tiền xử lý dữ liệu như làm sạch, chuẩn hóa, tokenization,…
- LLMs Layer: Sử dụng các mô hình ngôn ngữ lớn đã có sẵn và các phương pháp tinh chỉnh để tạo ra mô hình chuyên biệt cho tài chính.
- Application Layer: Triển khai vào các bài toán tác vụ cụ thể như robo-advisor, phân tích cảm xúc tài chính, giao dịch định lượng, phát triển low-code.

2.4 Kết chương

Chương 2 đã trình bày những cơ sở lý thuyết kiến thức nền tảng để xây dựng hệ thống chatbot đọc hiểu và phân tích báo cáo tài chính. Từ những kiến thức liên quan đến tài chính và cụ thể là báo cáo tài chính của một doanh nghiệp, làm cơ sở lý thuyết giúp việc xử lý dữ liệu sau được đúng đắn và chính xác cho đến những kiến thức chuyên môn về LLM và RAG là cốt lõi của hệ thống và cuối cùng là các công nghệ được sử dụng để phát triển hệ thống và một số nghiên cứu liên quan nổi bật trong việc áp dụng LLM trong việc đọc hiểu phân tích báo cáo tài chính nói riêng và lĩnh vực tài chính nói chung. Những nội dung này đóng vai trò quan trọng, tạo tiền đề cho việc triển khai xây dựng hệ thống trong các chương tiếp theo.

.

CHƯƠNG 3 ỨNG DỤNG ĐỒ THỊ TRI THỨC TRONG CHATBOT BÁO CÁO TÀI CHÍNH

Trong chương 3 của khóa luận sẽ đi tìm hiểu sâu về các phương pháp RAG truyền thống từ cơ bản đến nâng cao khi được áp dụng đối với dữ liệu báo cáo tài chính, các ưu điểm và hạn chế của từng phương pháp. Từ đó đề xuất phương pháp biểu diễn báo cáo tài chính dưới dạng đồ thị tri thức. Chi tiết về cách lập đồ thị từ báo cáo tài chính từ đó xây dựng hệ thống chatbot hỏi đáp giúp người dùng đẩy báo cáo tài chính và tương tác với hệ thống, giúp trả lời các câu hỏi của người dùng xoay quanh báo cáo tài chính.

3.1. Các phương pháp khai thác truy xuất dữ liệu từ báo cáo tài chính

3.1.1 Basic RAG

Phương pháp này tập chung vào việc truy xuất thông tin dựa trên độ tương đồng giữa các vector của câu hỏi từ người dùng và vector báo cáo tài chính được lưu trữ trong cơ sở dữ liệu vector.

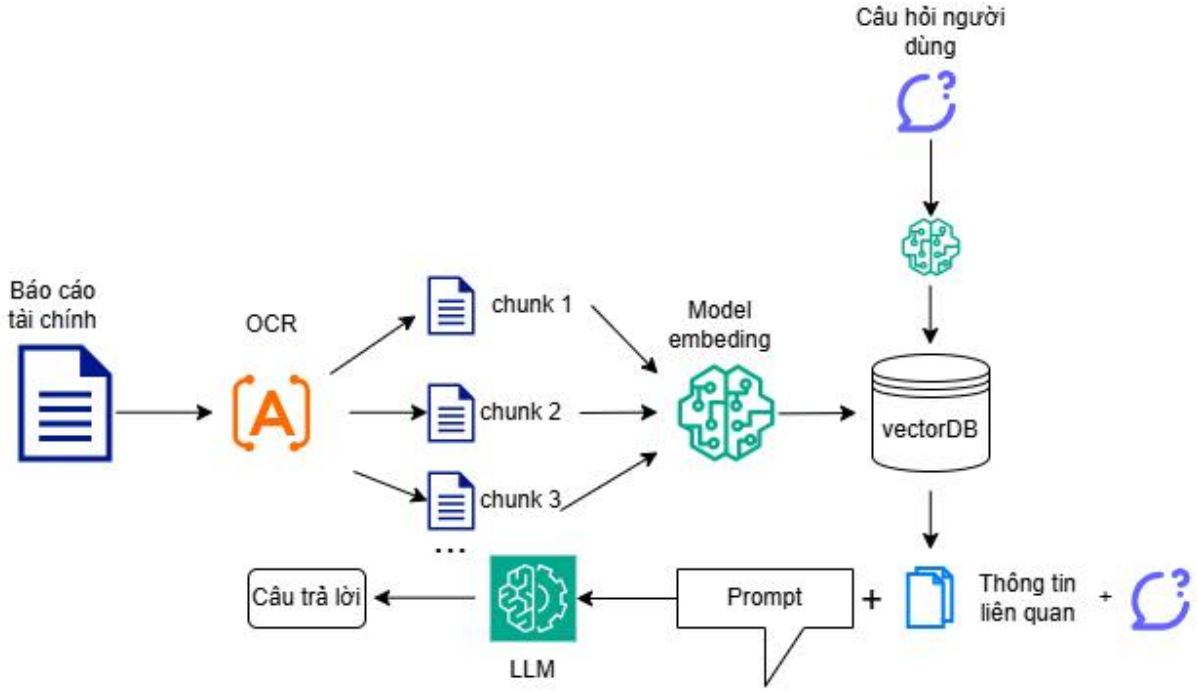
Toàn bộ quy trình xử lý dữ liệu được thực hiện chi tiết qua các bước:

Bước 1 OCR: Sau khi nhận được bản báo cáo từ người dùng, nếu báo cáo ở dạng PDF scan, hệ thống sử dụng dịch vụ parse của LlamaCloud để OCR đưa báo cáo về file markdown: định dạng này phù hợp với LLM bởi hầu hết các LLM đều được huấn luyện trên dữ liệu dạng này và có khả năng hiểu các ký hiệu ngôn ngữ markdown điều này giúp việc thể hiện thông tin về bảng biểu trong báo cáo tài chính dễ dàng đảm bảo thông tin được biểu diễn chính xác và LLM dễ dàng đọc hiểu.

Bước 2 Chunking: Khi hệ thống OCR xong báo cáo tài chính của người dùng các thông tin trong báo cáo lúc này sẽ được chia thành các đoạn nhỏ (chunks) nhỏ hơn phù hợp để lưu trữ và truy xuất các chunks này thường sẽ có độ dài khoảng 500 token và cho phép 150 token chồng chéo nhằm mục đích giúp đảm bảo mỗi chunks sẽ có đầy đủ ngữ nghĩa thông tin nhất định.

Bước 3 Tạo cơ sở dữ liệu vector: Các chunks này sau đó sẽ được đi qua mô hình embedding để tạo ra các vector ngữ nghĩa cho từng chunk từ đó xây dựng một cơ sở dữ liệu vector riêng bao gồm metadata chứa thông tin nội dung văn bản gốc và embeddings chứa vector ngữ nghĩa tương ứng của báo cáo tài chính phục vụ cho việc truy xuất thông tin.

Bước 4 Truy xuất thông tin và trả lời câu hỏi: Sau khi đã có cơ sở dữ liệu vector chứa toàn bộ thông tin của báo cáo tài chính lúc này hệ thống sẵn sàng cho việc tìm kiếm thông tin trả lời cho câu hỏi người dùng. Bằng cách tính toán độ tương đồng giữa các vector ngữ nghĩa của câu hỏi người dùng và trong cơ sở dữ liệu và chọn ra k chunks mang thông tin gần nhất với câu hỏi. Từ đó đưa vào prompt của LLM kết hợp với câu hỏi ban đầu của người dùng đảm bảo kết quả do LLM sinh ra chính xác và đúng trọng tâm câu hỏi nhất.



Hình 3.1 Luồng hoạt động của hệ thống RAG cơ bản

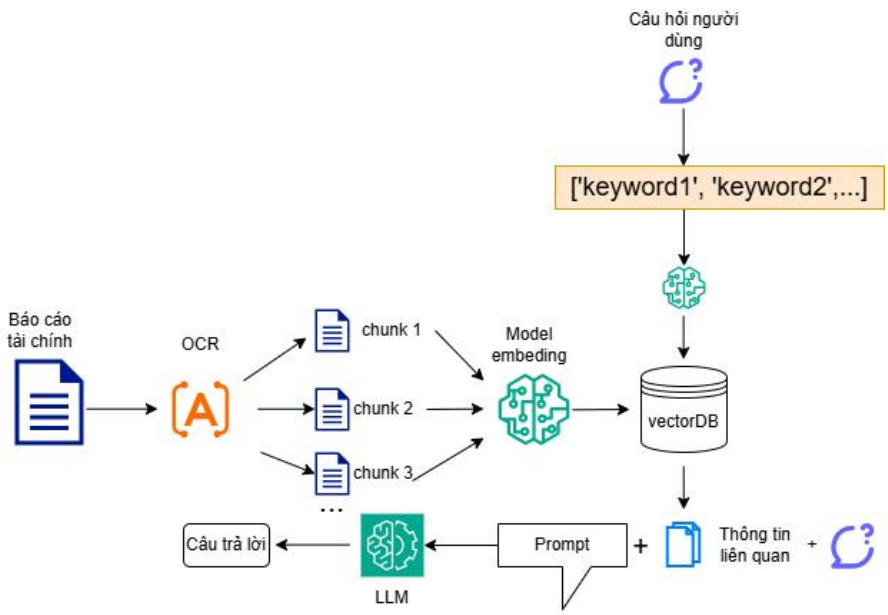
Tuy nhiên việc tìm ra được các chunk chứa đúng thông tin cần thiết để trả lời câu hỏi trong số rất nhiều chunk kia trở thành một thử thách ở bước này. Khác với các loại văn bản khác báo cáo tài chính chưa rất nhiều thông tin và các thông tin này có thể được tìm thấy ở nhiều chunks khác nhau. Ví dụ khi người dùng đưa ra câu hỏi về *"nợ phải trả của doanh nghiệp*", khi thực hiện tìm kiếm bằng độ tương đồng giữa các vector, kết quả trả ra có thể không như mong muốn do thông tin về nợ phải trả có thể khớp với một chunks miêu tả giải thích về nợ phải trả trong phần **thuyết minh báo cáo** chứ không phải là thông tin về nợ phải trả ở **bảng cân đối kế toán**. Điều này nghĩa là cứ chunks nào có nhiều từ khóa "*nợ phải trả*", "*doanh nghiệp*", thì các chunks đó mặc định được hiểu là thông tin cần thiết.

Các câu hỏi yêu cầu tính toán cũng trở thành một thách thức lớn ví dụ khi câu hỏi người là "*Tính chỉ số ROA*", Hệ thống sẽ đi tìm từ khóa "ROA" trong cở sở dữ liệu

vector tuy nhiên trong báo cáo tài chính thường không trình bày chi tiết các chỉ số này thay vào đó chỉ có số liệu để tính toán, khi đó hệ thống sẽ không thể tìm được chính xác chunks nào chứa thông tin ROA dẫn đến việc LLM không có thông tin để tính toán ra chi số này mặc dù các số liệu cần thiết để tính toán đều có sẵn trong báo cáo.

Để xử lý vấn đề này trước khi tìm kiếm trong cơ sở dữ liệu LLM sẽ tiến hành phân tích lại câu hỏi của người dùng (requery) để biết được mô hình cần được cung cấp những thông tin gì để trả lời câu hỏi đó thay vì sử dụng trực tiếp câu hỏi của người dùng. Từ đó LLM tự chọn ra các từ khóa (keyword) chính dùng để tìm thông tin trong cơ sở dữ liệu vector. Các keywords này đảm bảo cho việc tìm kiếm được những chunks mang đầy đủ thông tin cần thiết nhất để trả lời câu hỏi.

Ví dụ câu hỏi về chỉ số ROA như trình bày ở trên thay vì tìm trực tiếp ROA, mô hình sẽ đưa ra các keyword như: 'Lợi nhuận sau thuế', 'Tổng tài sản' từ đó mô hình sẽ tìm được các đoạn thông tin về lợi nhuận và tổng tài sản để phục vụ cho việc tính toán ROA: ROA = Lợi nhuận sau thuế / Tổng tài sản bình quân. Điều này giúp cải thiện khả năng tính toán các chỉ số của LLM tận dụng được kiến thức về ROA mà LLM đã được huấn luyện để tính toán cho một báo cáo tài chính của một doanh nghiệp cụ thể.



Hình 3.2 Luồng hoạt động được bổ sung thêm bước requery

3.1.2. Hybrid search + Re-rank RAG

Mặc dù độ chính xác đã được cải thiện sau khi câu hỏi của người dùng được requery thành các từ khóa (keywords), nhưng đối với báo cáo tài chính, các thuật ngữ chuyên ngành thường có nhiều cách diễn đạt khác nhau. Do đó, việc sử dụng LLM để

tạo ra các từ khóa cần đảm bảo số lượng vừa đủ và bao gồm các tên gọi khác nhau của cùng một thuật ngữ.

Ví dụ, khi câu hỏi của người dùng là "Vốn chủ sở hữu là bao nhiêu?", hệ thống sẽ tự động sinh ra một loạt từ khóa tương đương để tìm kiếm, bao gồm:

"Vốn chủ sở hữu"

"Nguồn vốn chủ sở hữu"

"Tổng vốn chủ sở hữu"

"Tổng nguồn vốn chủ sở hữu"

"Vốn điều lệ"

"Thặng dư vốn cổ phần"

Việc này giúp đảm bảo rằng dù báo cáo tài chính trình bày thuật ngữ theo cách nào, hệ thống vẫn có thể tìm kiếm và thu thập thông tin chính xác.

Tuy nhiên, việc truy xuất bằng vector ngữ nghĩa cho các thuật ngữ tài chính cũng tiềm ẩn một số hạn chế. Bởi đây là các thuật ngữ chuyên ngành, các mô hình embedding chỉ tạo ra biểu diễn ngữ nghĩa chung, đôi khi dẫn đến kết quả độ tương đồng thấp mặc dù thông tin trùng khớp. Ví dụ, độ tương đồng giữa hai câu "Nợ phải trả" và "Tổng nợ phải trả của doanh nghiệp trong quý 1 năm 2025 là: 1.000.000.000" chỉ đạt 0.42, do có nhiều từ bổ sung xung quanh thuật ngữ chính. Trong các chunks dài 500 token việc có nhiều từ nhiều xung quanh từ khóa càng làm giảm độ tương đồng với nhau.

Để khắc phục hạn chế này, bên cạnh truy xuất bằng vector ngữ nghĩa, hệ thống còn kết hợp tìm kiếm bằng từ khóa. Các chunk chứa keyword cũng được lựa chọn để cung cấp cho LLM, củng cố khả năng tìm đúng và đầy đủ thông tin, từ đó đảm bảo LLM có cơ sở dữ liệu đầy đủ để trả lời chính xác câu hỏi người dùng.

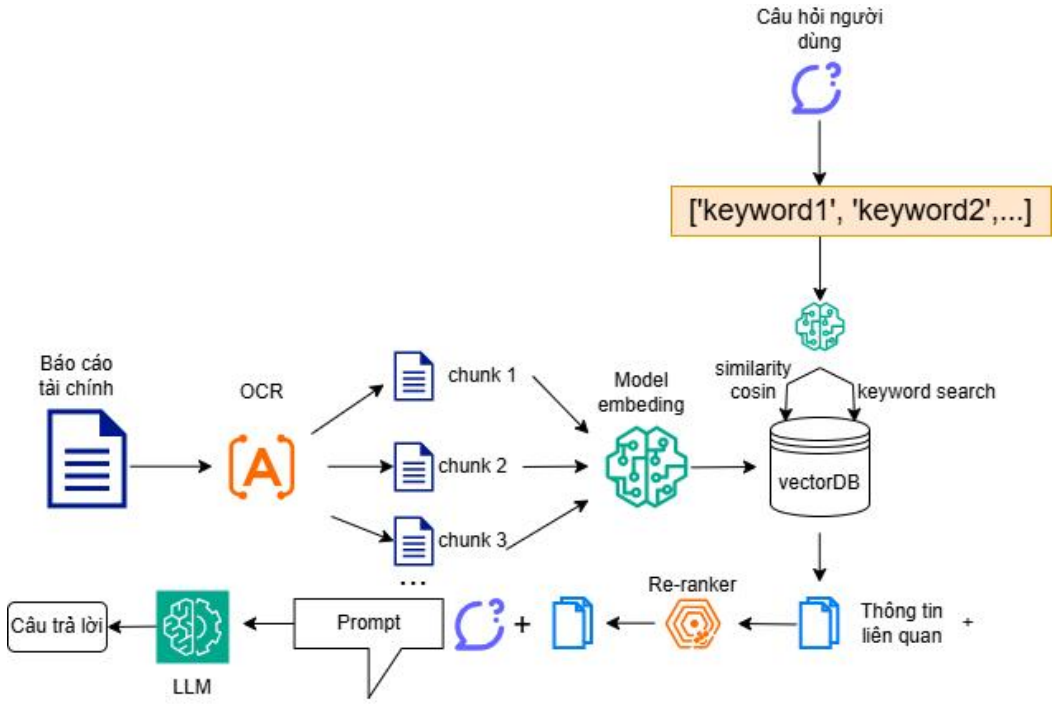
Tuy nhiên khi kết hợp Hybrid search giữa vector ngữ nghĩa và keyword search sẽ đảm bảo được lượng thông tin cần thiết, tuy nhiên với mỗi keyword sẽ có k chunks được chọn điều này dẫn đến có quá nhiều chunks được chọn dẫn tới khi đưa vào prompt làm cho tăng độ dài token ảnh hưởng đến việc trả lời do lượng thông tin bị loãng ngữ cảnh phức tạp các chunks có thể không liên quan đến nhau dẫn tới có thể bị mất thông tin quan trọng và gây ra hiện tượng hallucination bịa đặt thông tin. Do đó yêu cầu một bước quan trọng nữa trong quy trình xử lý dữ liệu trước khi đi vào prompt tránh bị dư thừa các thông tin không cần thiết.

Ở bước này, hệ thống sẽ sử dụng thêm một công cụ re-ranker dựa trên mô hình Cross-Encoder để chọn ra những *k* đoạn (chunks) thật sự liên quan và mang nhiều thông tin nhất đối với câu hỏi. Khác với cách tính embedding thông thường - nơi câu hỏi và đoạn văn được mã hóa độc lập—Cross-Encoder hoạt động bằng cách xử lý trực tiếp cặp (query, chunk) trong cùng một mạng. Điều này cho phép mô hình hiểu sâu ngữ cảnh của câu hỏi, phân tích nội dung từng đoạn, và so sánh mức độ phù hợp giữa chúng một cách chính xác hơn.

Nhờ cơ chế đánh giá trực tiếp này, Cross-Encoder có thể phát hiện:

- Đoạn nào trả lời đúng trọng tâm,
- Đoạn nào chứa đủ bằng chứng hoặc dữ liệu cần thiết,
- Đoạn nào chỉ liên quan mơ hồ hoặc không hữu ích.

Kết quả là hệ thống sẽ tìm được những đoạn *thật sự mang tính giải đáp*, giảm nhiễu, tăng độ chính xác, và đảm bảo LLM nhận đúng nhận đủ phần nội dung có giá trị để tạo ra câu trả lời cuối cùng.



Hình 3.3 Luồng hoạt động đầy đủ của hệ thống

Mặc dù đã khả năng tìm kiếm thông tin với requery + hybrid search + re-rank đã rất tốt và có thể trả lời được hầu hết các câu hỏi của người dùng liên quan đến việc việc trích xuất thông tin số liệu, hay tính toán các chỉ số tài chính rồi đưa ra phân tích

nhận định một cách chính xác đúng trọng tâm câu hỏi. Tuy nhiên vẫn tồn tại một số vấn đề như:

Tốc độ tính toán: Với mỗi keyword hệ thống sẽ phải thực hiện tìm M chunks liên quan bằng vector ngữ nghĩa và N chunks bằng keywords sau đó sẽ tiền hành re-rank đánh giá lần lượt cho từng chunk. Việc này tương đối mất thời gian nếu chạy trên CPU bởi việc embedding rồi tính toán cộng với Cross-Encoder để re-rank khá lâu. Thời gian ước tính cho 1 keyword bằng:

$$t_{keyword} = Mt_{similar_search} + Nt_{keyword_search} + (N+M)t_{re-rank} \quad (1)$$

Trong đó:

- $t_{keyword}$: là thời gian để tìm ra K chunks thực sự chứa thông tin của keyword đó.
- $t_{similar_search}$: là thời gian để tìm được 1 chunk bằng tìm kiếm qua độ tương đồng vector ngữ nghĩa.
- $t_{keyword_search}$: là thời gian để tìm được 1 chunk bằng so khớp keyword.
- $t_{re-rank}$: là thời gian để đánh giá mức độ liên quan của 1 chunk với câu hỏi.
- M: số chunks tìm được bằng *similar_search* (vector ngữ nghĩa).
- N: số chunks tìm được bằng *keyword_search.*

Các câu hỏi phân tích suy luận: Do các thông tin trong báo cáo tài được chia thành chunks vì thế khiến cho trong một số trường hợp ngữ cảnh không liền mạch. Ví dụ thông tin về bảng cân đối kế toán có thể được chia làm những chunks khác nhau việc này dẫn đến không cung cấp một bảng cân đối kế toán hoàn chỉnh cho LLM. Khác với câu hỏi trích xuất thông tin và tính toán thường chỉ tập chung vào số liệu của một mục cụ thể. Nhưng với những câu hỏi phân tích suy luận những câu hỏi này yêu cầu lượng thông tin ngữ cảnh một cách bao quát cụ thể đánh giá trên nhiều tiêu chí chí khác nhau ví dụ như toàn bộ báo cáo kết quả kinh doanh để cho ra nhận xét về kết quả hoạt động của doanh nghiệp. Do đó việc chia thành các chunks và tìm kiếm re-rank đôi khi làm xáo trộn các chunk với nhau dẫn đến thứ tự giữa các chunk bị điều chỉnh ảnh hưởng dẫn đến mất tính liền mạch của thông tin (Context Fragmentation), mất tính mạch lạc tuần tự (Loss of Sequential Coherence) khiến ngữ cảnh trở nên rời rạc gây khó khăn cho LLM. Đặc biệt đối với dữ liệu chứa nhiều thông tin về bảng biểu như báo cáo tài chính.

3.1.3. Graph-RAG trong báo cáo tài chính (Financial Report Graph-RAG)

Với hai hạn chế đã phân tích, câu hỏi đặt ra là: liệu có phương pháp nào truy xuất dữ liệu vừa nhanh, vừa chính xác, đồng thời cung cấp ngữ cảnh liền mạch và đảm bảo các bảng biểu được đưa vào LLM một cách đầy đủ cho từng mục hay không?

Báo cáo tài chính là một dạng dữ liệu đặc thù, được lập theo chuẩn mực và quy định thống nhất của Bộ Tài chính Việt Nam. Vì vậy, phân tách báo cáo theo đúng cấu trúc vốn có của nó trở thành một hướng tiếp cận tối ưu. Khi thực hiện điều này, việc truy xuất sẽ chính xác hơn, logic hơn và ngữ cảnh giữa các phần được bảo toàn thay vì bị chia cắt ngẫu nhiên như trong phương pháp chunk truyền thống.

Sau khi báo cáo được tổ chức lại theo cấu trúc, để lưu trữ và khai thác mối quan hệ giữa các bảng biểu, chỉ tiêu, dòng mục, và các giá trị liên quan, đồ thị tri thức (Knowledge Graph) trở thành lựa chọn tối ưu. Trong đồ thị tri thức, mỗi thông tin từ báo cáo tài chính được ánh xạ thành một nút hoặc một cạnh với vị trí rõ ràng trong không gian dữ liệu. Nhờ đó, thay vì để hệ thống phải "dò từng chunk" và so khớp tuần tự – vốn tốn thời gian và dễ gây nhiễu – thì:

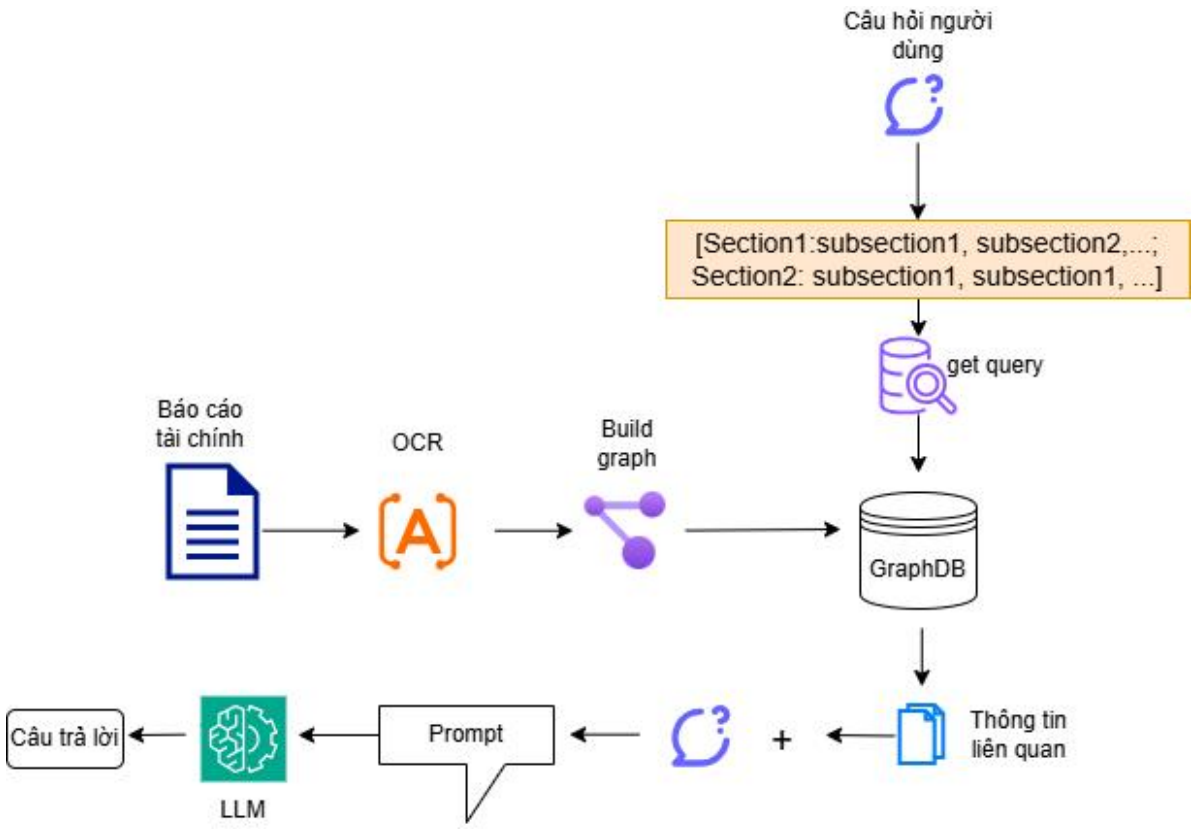
- Mọi thông tin đều có vị trí xác định trong đồ thị
- Các quan hệ giữa các mục được mã hóa rõ ràng
- Việc truy vấn trở nên nhanh chóng, chính xác và có định hướng

Có thể hình dung việc xây dựng đồ thị tri thức cho báo cáo tài chính giống như vẽ một bản đồ tri thức: mỗi bảng biểu, mỗi chỉ tiêu đều được đặt đúng vị trí của nó. Nhờ có "bản đồ" này, hệ thống không còn phải "tìm mò" qua từng đoạn văn rời rạc, mà có thể truy cập đúng thông tin, đúng ngữ cảnh, đúng mối quan hệ ngay lập tức.

Cụ thể khi người dùng hỏi thông tin liên quan đến nợ phải trả, thay vì phải dò tìm theo các keyword như trước thì lúc này hệ thống sẽ thực hiện truy vấn trên đồ thị từ các nút "báo cáo" → "bảng cân đối kế toán" → "nợ phải trả". Rồi lấy thông tin có trong nút đó để làm đầu vào cho LLM. Việc này đảm bảo rằng thông tin về nợ phải trả được trình bày chi tiết cụ thể liền mạnh chính xác tuyệt đối trong báo cáo tài chính.

Việc xây dựng được đồ thị tri thức cho toàn bộ báo cáo tài chính cũng mở ra một định hướng phát triển mạnh mẽ khi có thể tận dụng được các quan hệ giữa những báo cáo trong đồ thị áp dụng cho việc phân tích so sánh trên nhiều báo cáo khác nhau giữa các công ty có cùng lĩnh vực ngành nghề hay giữa những báo cáo tài chính của một doanh nghiệp qua các giai đoạn. Với đồ thị tri thức việc truy vấn đến những thông tin

trong các báo cáo có chung các quan hệ với nhau như vậy trở nên dễ dàng và thuận tiện, nhanh chóng hơn so với việc lưu trữ trong cơ sở dữ liệu truyền thống [23].



Hình 3.4 Luồng hoạt động của Graph-RAG

Biểu diễn báo cáo tài chính dưới dạng đồ thị tri thức (Financial Report Graph)

Vậy thì làm cách nào để biểu diễn một báo cáo tài chính dưới dạng đồ thị đảm bảo chia đúng theo cấu trúc thể hiện được thông tin rõ ràng mạch lạc logic đảm bảo giữ được đúng cấu trúc của các báo cáo tài chính.

Khóa luận này đề xuất phương pháp xây dựng đồ thị tri thức từ báo cáo tài chính bằng cách phân tách báo cáo theo cấu trúc chuẩn được quy định trong Chế độ kế toán doanh nghiệp Việt Nam (VAS), ban hành theo Thông tư 200/2014/TT-BTC [1] ngày 22/12/2014 của Bộ Tài chính. Theo quy định này, hệ thống báo cáo tài chính gồm bốn báo cáo thành phần, bao gồm: (1) Bảng cân đối kế toán (Báo cáo tình hình tài chính), (2) Báo cáo kết quả hoạt động kinh doanh, (3) Báo cáo lưu chuyển tiền tệ, và (4) Thuyết minh báo cáo tài chính.

Trong các lĩnh vực hoạt động khác nhau của doanh nghiệp cách trình bày báo cáo tài chính cũng có sử khác nhau nhẹ những cơ bản tất cả đều vẫn được cấu thành từ bốn

báo cáo thành phần như trên. Ở Việt Nam báo cáo tài chính ở 3 lĩnh vực hoạt động doanh nghiệp có các lập báo cáo tài chính khác nhau là: Ngân hàng, Doanh nghiệp bình thường, Chứng khoán. Nhiệm vụ của hệ thống là phải xác định được báo cáo tài chính của người dùng thuộc lĩnh vực nào rồi sử dụng phương pháp xử lý phân tách cấu trúc phù hợp.

Từ đó trong khóa luận này đề xuất ontology tự định nghĩa dựa trên cấu trúc chuẩn của các báo cáo tài chính: Khung cấu trúc của đồ thị tri thức: hay nói cách khác, Ontology - mô tả các loại thực thể, thuộc tính và quan hệ giữa chúng trong báo cáo tài chính. Ontology đóng vai trò như bộ khung xương giúp tổ chức thông tin một cách logic và nhất quán. Từ đó tạo ra một bộ cấu trúc đồ thị chuẩn hóa với cho các báo cáo tài chính trong cùng một lĩnh vực hay giữa các lĩnh vực khác nhau như trên. Cụ thể như sau:

Trong đồ thị sẽ gồm 6 nút chính bao gồm:

- **Company**: Đây là nút gốc của mỗi báo cáo trong đó có chưa thuộc tính *name*: lưu trữ tên của doanh nghiệp

- **Report**: Đây là nút chứa thông tin của các báo cáo trong đó chứa các thuộc tính như:

report_id: để xác định giữa các báo cáo, *type:* chứa thông tin về loại báo cáo.

- **Sector**: Đây là nút chứa thông tin về lĩnh vực hoạt động của doanh nghiệp bao gồm thuộc tính *name*: tên của lĩnh vực hoạt động.

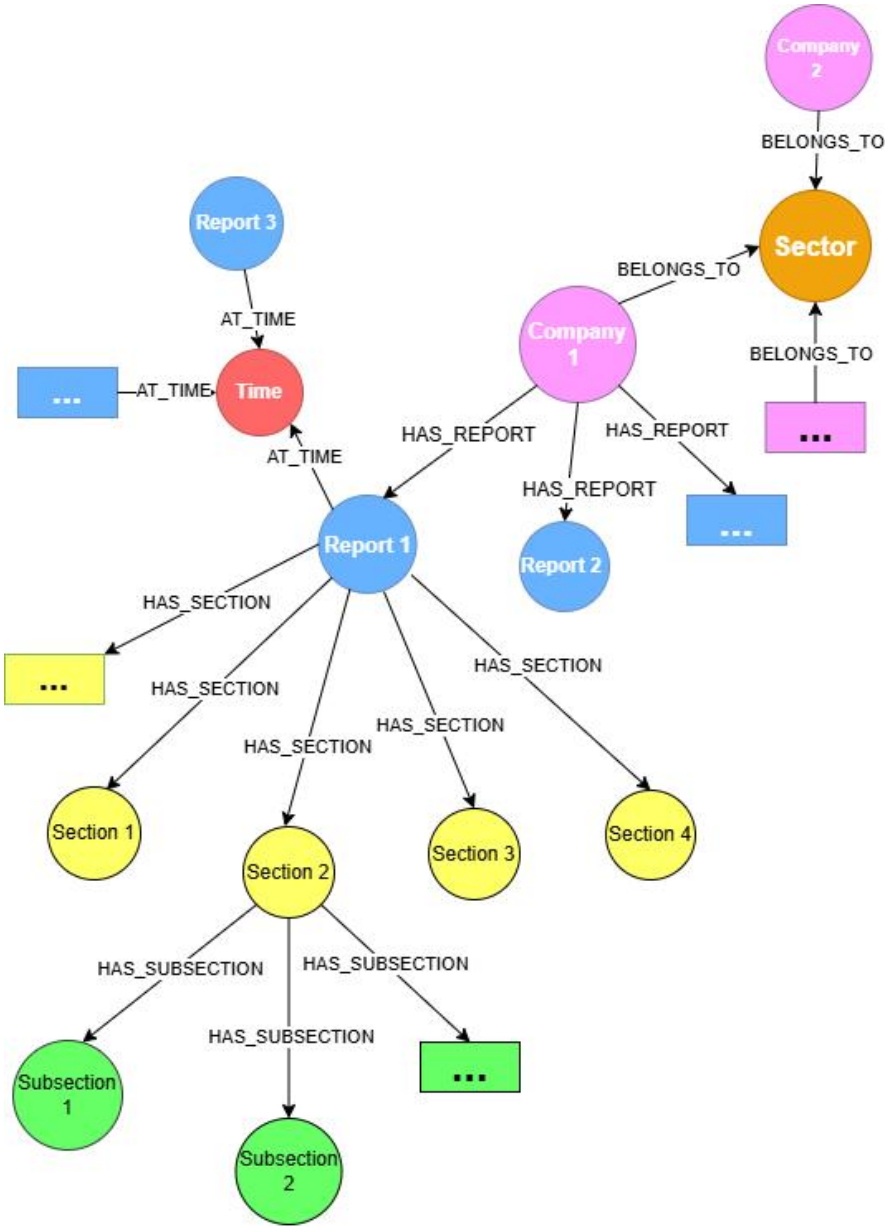
- **Section**: Đây là nút chứa thông tin về các mục lớn (section) của báo cáo tài chính, mỗi báo cáo được chia làm các section. Mỗi section chứa thuộc tính như:

  - *pages:* Số trang của section đó trong báo cáo thực.
  - *section_id:* Dùng để xác định section này thuộc về báo cáo nào.
  - *title*: Tên của mục đó.
  - *raw_text*: thuộc tính này có thể có hoặc không tuỳ vào loại section. Nếu section không chứa các mục con (subsection) thì sẽ chứa *raw_text*: là thông tin văn bản trực tiếp trong báo cáo tài chính.
  - *table_structure*: thuộc chính này cũng có thể có hoặc không. Nếu section này có mục con và chứa bảng thì sẽ có thuộc tính *table_structure*: thể hiện cấu trúc bảng biểu của từng section trong báo cáo tài chính.

- **Subsection**: Đây là nút chưa thông tin về các mục con có thể có trong các mục lớn. Subsection cũng có một số thuộc tính như:
    - *subsection_id*: dùng để xác định subsection này thuộc về section nào và báo cáo nào.
    - *subtitle*: Tên của mục con đó.
    - *raw_text*: Chưa nội dung văn bản bảng biểu, thông tin trực tiếp trong báo cáo tài chính.
- **Time**: Đây là nút chưa thông tin về giai đoạn của báo cáo cáo tài chính đó có một thuộc tính *value:* thể hiện thông tin về thời gian.

Bên cạnh các nút và thuộc tính tương ứng là các quan hệ giữa các nút đó:

- **AT_TIME**: Đây là quan hệ giữa nút Report và Time, mối quan hệ này thể hiện một báo cáo tài chính thể hiện tại một khoảng thời gian nhất định.
- **BELONGS_TO**: Đây là quan hệ giữa Company và Sector, mối quan hệ này thể hiện một doanh nghiệp thuộc về một lĩnh vực cụ thể.
- **HAS_REPORT**: Đây là quan hệ giữa Company và Report, mỗi công ty sẽ có các báo cáo tài chính tương ứng.
- **HAS_SECTION**: Đây là quan hệ giữa Report và Section, mỗi báo cáo sẽ được chia thành các mục lớn riêng biệt.
- **HAS_SUBSECTION**: Đây là quan hệ giữa Section và SubSection: mỗi section được chia thành các subsection tương ứng.



Hình 3.5 Cấu trúc đồ thị tri thức cho báo cáo tài chính

Chi tiết cách xác định giá trị cho từng thuộc tính của các nút cho một báo cáo tài chính như nhau:

1. Đối với doanh nghiệp sản xuất trên tất cả lĩnh vực

Sau khi xử lý OCR xong file kết quả sẽ được lưu dưới dạng markdown. Tiếp theo đó là một loạt các bước xử lý nhằm tạo đồ thị cho báo cáo tài chính và lưu trữ trong cơ sở dữ liệu đồ thị.

Xác định thông tin cơ bản của báo cáo: Trước tiên hệ thống sẽ sử dụng thông tin ở một số trang đầu báo cáo để xác định các thông tin cơ bản của báo cáo như: tên doanh nghiệp, loại báo cáo, lĩnh vực hoạt động và giai đoạn của báo cáo. Việc xác

định này sử dụng LLM và dựa trên thông tin từ các trang đầu tiên. Để bảo bảo mỗi kết quả LLM trả ra đều nhất quán qua các lần gọi hệ thống đặt tham số *temperature = 0* cho LLM.

Lưu ý đối với lĩnh vực hoạt động của doanh nghiệp sẽ giới hạn trong 11 lĩnh vực (sectors) trong chuẩn phân ngành quốc tế GICS (Global Industry Classification Standard) [9] tuy nhiên có sự điều chỉnh nhẹ cách tài chính thành 2 lĩnh vực là chứng khoán và ngân hàng bao gồm:

1. Năng lượng
2. Nguyên vật liệu
3. Công nghiệp
4. Hàng tiêu dùng thiết yếu
5. Hàng tiêu dùng không thiết yếu
6. Y tế
7. Chứng khoán
8. Công nghệ thông tin
9. Viễn thông và Truyền thông
10. Hạ tầng tiện ích
11. Bất động sản
12. Ngân hàng

Và các giai đoạn báo cáo gồm có:

- Quý 1 [năm] (Cho kỳ 3 tháng (01/01 - 31/03))
- Quý 2 [năm] (Cho kỳ 3 tháng (01/04 - 30/06))
- Bán niên [năm] (Cho kỳ 6 tháng (01/01 - 30/06))
- Quý 3 [năm] (Cho kỳ 3 tháng (01/07 - 30/09))
- Quý 4 [năm] (Cho kỳ 3 tháng (01/10 - 31/12))
- Năm [năm] (Cho kỳ 12 tháng (01/01 - 31/12))

Gán nhãn cho từng trang: Hệ thống sẽ tiến hành đi tìm thông tin trên mỗi trang của báo cáo và gán nhãn cho từng trang theo các nhãn sau:

BÁO CÁO CỦA BAN GIÁM ĐỐC,

BÁO CÁO SOÁT XÉT,

BÁO CÁO KIỂM TOÁN,

BẢNG CÂN ĐỐI KẾ TOÁN,

BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH,

BÁO CÁO LƯU CHUYỂN TIỀN TỆ,

THUYẾT MINH BÁO CÁO TÀI CHÍNH.

Để gán được nhãn có từng trang dựa vào cấu trúc của báo cáo tài chính hệ thống sẽ lấy 10 dòng đầu tiên của một trang với báo cáo tài chính khi bắt đầu một đầu mục sẽ luôn có một dòng thể hiện tiêu đề chung đánh dấu điểm bắt đầu của mục đó do đó. Điều này đảm bảo 10 dòng đầu sẽ luôn chứa dòng tiêu đề đủ thông tin để phân loại.

Với mỗi dòng hệ thống sẽ tính toán dựa trên độ tương đồng về mặt ngữ nghĩa giữa vector embedding của từng dòng và các nhãn lần lượt kết hợp với so sánh keywords. Từ đó sẽ xác định nhãn tương ứng cho từng dòng là nhãn có số điểm tương đồng cao nhất để tìm ra đâu là dòng chứa tiêu đề. Dòng có số điểm tương đồng cao nhất với một trong cách nhãn chứng tỏ là tiêu đề của một mục nào đó trong báo cáo. Với những dòng có chứa tên của các nhãn trong đó sẽ được cộng thêm điểm vào độ tương đồng đảm bảo việc xác định chính xác.

Nhãn của toàn page sẽ được xác định là nhãn của dòng có độ tương đồng cao nhất.

Trong trường hợp số điểm cao nhất này thấp hơn một ngưỡng nhất định thì sẽ gán nhãn “None” cho page đó. Điều này chứng tỏ rằng không có dòng tiêu đề ở page này. Các page này thường là sự nối tiếp của một mục nào đó trong báo cáo tài chính.

TỔNG CÔNG TY KHÍ VIỆT NAM - CÔNG TY CỔ PHẦN
Tòa nhà PV GAS Tower, số 673 đường Nguyễn Hữu Thọ
xã Nhà Bè, Thành phố Hồ Chí Minh, Việt Nam

MẪU SỐ B 01a-DN/HN
Ban hành theo Thông tư số 202/2014/TT-BTC
ngày 22 tháng 12 năm 2014 của Bộ Tài chính

BẢNG CÂN ĐỐI KẾ TOÁN HỢP NHẤT GIỮA NIÊN ĐỘ
Tại ngày 30 tháng 6 năm 2025

Đơn vị: VND

	TÀI SẢN	Mã số	Thuyết minh	Số cuối kỳ	Số đầu kỳ
A.	TÀI SẢN NGẮN HẠN	100		64.857.286.428.940	57.290.495.430.038
I.	Tiền và các khoản tương đương tiền	110	4	10.377.145.910.363	5.567.983.431.468
1.	Tiền	111		2.211.370.545.474	1.248.083.431.468
2.	Các khoản tương đương tiền	112		8.165.775.364.889	4.319.900.000.000
II.	Đầu tư tài chính ngắn hạn	120	5	30.730.802.707.604	27.494.031.282.702
1.	Đầu tư nắm giữ đến ngày đáo hạn	123		30.730.802.707.604	27.494.031.282.702
III.	Các khoản phải thu ngắn hạn	130		19.777.698.700.821	19.098.417.078.475

Hình 3.6 Báo cáo thành phần Bảng cân đối kế toán trong báo cáo tài chính [8]

TỔNG CÔNG TY KHÍ VIỆT NAM - CÔNG TY CỔ PHẦN
Tòa nhà PV GAS Tower, số 673 đường Nguyễn Hữu Thọ
xã Nhà Bè, Thành phố Hồ Chí Minh, Việt Nam

MẪU SỐ B 01a-DN/HN
Ban hành theo Thông tư số 202/2014/TT-BTC
ngày 22 tháng 12 năm 2014 của Bộ Tài chính

BẢNG CÂN ĐỐI KẾ TOÁN HỢP NHẤT GIỮA NIÊN ĐỘ (Tiếp theo)
Tại ngày 30 tháng 6 năm 2025

Đơn vị: VND

	NGUỒN VỐN	Mã số	Thuyết minh	Số cuối kỳ	Số đầu kỳ
C.	NỢ PHẢI TRẢ	300		19.860.792.028.126	20.283.886.142.528
I.	Nợ ngắn hạn	310		14.439.813.602.726	14.575.489.059.608
1.	Phải trả người bán ngắn hạn	311	15	7.451.864.506.244	6.037.010.202.174
2.	Người mua trả tiền trước ngắn hạn	312		176.527.470.385	270.548.145.715
3.	Thuế và các khoản phải nộp Nhà nước	313	16	1.212.075.265.324	682.837.407.395
4.	Phải trả người lao động	314		452.447.029.567	409.699.768.279
5.	Chi phí phải trả ngắn hạn	315	17	2.982.297.997.174	2.995.123.654.976
6.	Doanh thu chưa thực hiện ngắn hạn	318	18	6.293.177.547	2.241.036.607.643
7.	Phải trả ngắn hạn khác	319	19	344.119.687.572	373.838.581.752

Hình 3.7 Bảng cân đối kế toán được trình bày trong báo cáo tài chính [8]

Ví dụ như trong hình 3.6 và 3.7 dòng **"BẢNG CÂN ĐỐI KẾ TOÁN HỢP NHẤT GIỮA NIÊN ĐỘ"** sẽ có độ tương đồng rất cao với nhãn "BẢNG CÂN ĐỐI KẾ TOÁN" và chắc chắn sẽ cao hơn các dòng còn lại từ đó page đó sẽ được gán nhãn là BẢNG CÂN ĐỐI KẾ TOÁN.

Tuy nhiên đối với mỗi nhãn trên có thể có các tên gọi khác nhau ví dụ "BẢNG CÂN ĐỐI KẾ TOÁN" hãy còn có tên gọi khác là "BÁO CÁO TÌNH HÌNH TÀI CHÍNH". Vì thế khi tính gán nhãn cho các page hệ thống sẽ tính toán độ tương đồng trên cả những tên gọi khác nhau phổ biến nhất của các nhãn rồi quy về một tên gọi duy nhất đảm bảo tính nhất quán khi đưa vào đồ thị. Ví dụ các tên gọi có thể có của các nhãn như sau:

Bảng 3.1 Tên gọi khác có thể của một số mục lớn

Tên gốc	**Tên gọi khác có thể có**
BẢNG CÂN ĐỐI KẾ TOÁN	BÁO CÁO TÌNH HÌNH TÀI CHÍNH
BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH	BÁO CÁO KẾT QUẢ HOẠT ĐỘNG, BÁO CÁO KẾT QUẢ KINH DOANH, KẾT QUẢ HOẠT ĐỘNG KINH DOANH

Đối với các nhãn BÁO CỦA BAN GIÁM ĐỐC, BÁO CÁO SOÁT XÉT, BÁO CÁO KIỂM TOÁN, Các mục này thường nằm ở đầu các báo cáo tài chính đã được kiểm toán thường đối với các báo cáo năm hoặc bán niên do đó không phải báo cáo

nào cũng có. Vì thế các trang mang những nhãn này sẽ được gán nhãn mới là "Giới thiệu" khi đưa vào đồ thị tri thức.

Gộp trang theo nhãn: Tiếp theo sau khi đã có nhãn tương ứng cho từng trang hệ thống sẽ gộp các trang theo các nhãn đã đã gán, các trang có cùng nhãn sẽ được gộp lại với nhau từ đó giúp việc chia báo cáo tài chính thành các báo cáo thành phần hoàn chỉnh. Một lưu ý trong một số trường hợp của báo cáo tài chính quá trình gộp: với các trang mang nhãn "none" điều này chứng tỏ đây là trang mang nhãn của trang có nhãn gần nhất, do đó các trang có nhãn "None" sẽ được gộp vào trang có nhãn trước đó cho tới khi gặp trang có nhãn mới ví dụ: Trang số 3 mang nhãn "BẢNG CÂN ĐỐI KẾ TOÁN ", Trang 4, 5 mang nhãn "None" và Trang số 6 mang nhãn "BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH". Điều này chứng tỏ trang 4, 5 chính là phần tiếp theo của Bảng cân đối kế toán trong báo cáo tài chính do đó sẽ được gộp vào với trang 3 đảm bảo các trang 3, 4, 5 mang đầy đủ thông tin của Bảng cân đối kế toán.

Chia mục con: Sau khi đã chia được báo cáo tài chính thành 4 báo cáo thành phần tương ứng với bốn section trong đồ thị hệ thống sẽ tiến hành chia tiếp các báo cáo thành phần này thành các mục con tương ứng với các subsection trong đồ thị như sau:

Bảng 3.2 Thông tin về mục lớn và các mục con tương ứng đối với doanh nghiệp sản xuất dịch vụ thông thường

Mục lớn (section)	**Mục con(subsection)**	**Các từ khóa để tìm kiếm mục con**
Bảng cân đối kế toán	Tài sản ngắn hạn	Tài sản ngắn hạn
	Tài sản dài hạn	Tài sản dài hạn
	Nợ phải trả	Nợ phải trả
	Vốn chủ sở hữu	Vốn chủ sở hữu
Báo cáo kết quả hoạt động kinh doanh	Doanh thu bán hàng và cung cấp dịch vụ	- Doanh thu bán hàng - Doanh thu cung cấp dịch vụ - Tổng doanh thu
	Doanh thu hoạt động tài chính	- Doanh thu hoạt động tài chính
	Thu thập khác	- Thu nhập khác

Báo cáo lưu chuyển tiền tệ	Lưu chuyển tiền từ hoạt động kinh doanh	- Lưu chuyển tiền từ hoạt động kinh doanh - Lưu chuyển tiền thuần từ hoạt động kinh doanh - Lưu chuyển tiền từ hoạt động sản xuất - Lưu chuyển tiền tệ từ hoạt động sxkd
	Lưu chuyển tiền từ hoạt động đầu tư	- Lưu chuyển tiền từ hoạt động đầu tư - Lưu chuyển tiền thuần từ hoạt động đầu tư
	Lưu chuyển tiền từ hoạt động tài chính	- Lưu chuyển tiền từ hoạt động tài chính - Lưu chuyển tiền thuần từ hoạt động tài chính

Các mục con (subsection) được chia dựa trên lượng thông tin có trong báo cáo thành phần. Các thông tin này được tìm kiếm bằng cách sử dụng keyword tương ứng với từng mục con. Việc sử dụng keyword thay vì trực tiếp nhằm tìm kiếm so khớp một cách chính xác hơn vì trong một số doanh nghiệp với lĩnh vực khác nhau có thể có các mục con trong các báo cáo thành phần khác nhau nhưng mang ý nghĩa tương đương ví dụ với những doanh nghiệp cung cấp dịch vụ không bán hàng thì mục con "DOANH THU BÁN HÀNG VÀ CUNG CẤP DỊCH VỤ" sẽ tương ứng với "DOANH THU CUNG CẤP DỊCH VỤ"tương ứng.

Đối với THUYẾT MINH BÁO CÁO TÀI CHÍNH do không có cấu trúc rõ ràng về các mục trình bày, do đó với báo cáo thành phần này sẽ lưu thông tin toàn bộ và không có mục con.

Với mỗi báo cáo thành phần hệ thống sẽ tìm các mục con theo thứ tự như bằng cách kiểm tra xem trong mỗi dòng có chưa keyword trên không, nếu có thì đó là dòng bắt đầu của mục con đó và lưu lại. Thông tin trong mục con đó gồm toàn bộ các dòng từ dòng đầu phát hiện keyword cho đến khi gặp dòng chứa keyword của mục tiếp theo.

Để lưu trữ được thông tin về cấu trúc bảng cho mỗi báo cáo thành phần, Hệ thống sẽ lưu 5 dòng phía trước dòng phát hiện mục con đầu tiên đảm bảo chứa được đầy đủ thông tin về cấu trúc bảng.

CÔNG TY CỔ PHẦN TẬP ĐOÀN HÒA PHÁT
Khu Công nghiệp Phố Nối A, Xã Nguyễn Văn Linh,
Tỉnh Hưng Yên, Việt Nam

MẪU SỐ B 02a-DN
Ban hành theo Thông tư số 200/2014/TT-BTC
ngày 22 tháng 12 năm 2014 của Bộ Tài chính

BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH RIÊNG GIỮA NIÊN ĐỘ

Cho kỳ hoạt động 6 tháng kết thúc ngày 30 tháng 6 năm 2025

Đơn vị: VND

	CHỈ TIÊU	Mã số	Thuyết minh	Kỳ này	Kỳ trước
1.	**Doanh thu cung cấp dịch vụ**	01	22	**177.662.880.141**	**62.561.626.570**
2.	Các khoản giảm trừ doanh thu	02	22	1.460.500	-
3.	**Doanh thu thuần về cung cấp dịch vụ (10=01-02)**	10	22	**177.661.419.641**	**62.561.626.570**
4.	Giá vốn dịch vụ cung cấp	11	23	126.525.372.761	48.605.318.118
5.	**Lợi nhuận gộp về cung cấp dịch vụ (20=10-11)**	20		**51.136.046.880**	**13.956.308.452**
6.	Doanh thu hoạt động tài chính	21	25	5.499.712.962.181	5.107.061.479.938
7.	Chi phí tài chính	22	26	62.018.961.390	-
	- Trong đó: Chi phí lãi vay	*23*		*62.018.158.906*	-
8.	Chi phí bán hàng	25		10.709.626	-
9.	Chi phí quản lý doanh nghiệp	26	27	108.802.204.269	44.227.832.288
10.	**Lợi nhuận thuần từ hoạt động kinh doanh (30=20+(21-22)-(25+26))**	30		**5.380.017.133.776**	**5.076.789.956.102**
11.	Thu nhập khác	31		30.082.136	231.819.104
12.	Chi phí khác	32		51.027.882	477.435.720
13.	**Lỗ khác (40=31-32)**	40		**(20.945.746)**	**(245.616.616)**
14.	**Tổng lợi nhuận kế toán trước thuế (50=30+40)**	50		**5.379.996.188.030**	**5.076.544.339.486**
15.	Chi phí thuế thu nhập doanh nghiệp hiện hành	51	28	-	-
16.	**Lợi nhuận sau thuế thu nhập doanh nghiệp (60=50-51)**	60		**5.379.996.188.030**	**5.076.544.339.486**

Hình 3.8 Hình minh họa các mục con của Báo cáo kết quả hoạt động kinh doanh [8]

Sau khi đã trích xuất được đầy đủ thông tin cần thiết để tạo đồ thị các thông tin này được lưu trữ dưới dạng file json dùng để tạo đồ thị. Như vậy thuộc tính của các nút trong đồ thị của một báo cáo cho doanh nghiệp bình thường sẽ gồm có:

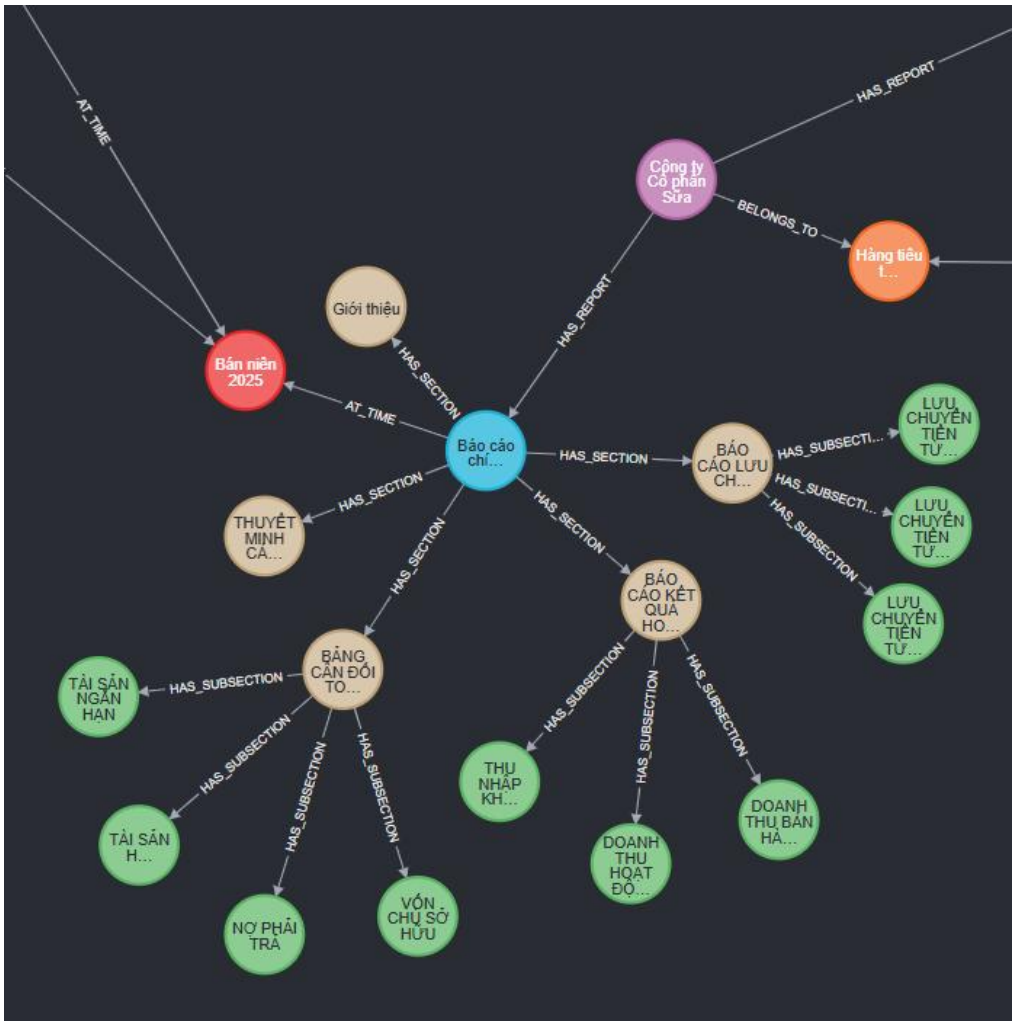
- Với nút Report*: report_id* = tên công ty|loại báo cáo|thời gian báo cáo; *type* = loại báo cáo.
- Với nút Company: *name* = tên công ty.
- Với nút Sector: *name* = lĩnh vực hoạt động.
- Với nút Section: *section_id* = *report_id*|*title*, *title* = tên của section, *table_structure*: cấu trúc bảng của section có thể có hoặc không, *page* = [các trang chứa section trong báo cáo thực], *raw_text*: thông tin của section trong báo cáo có thể có hoặc không.

Các Section của một báo cáo tài chính cho doanh nghiệp bình thường gồm có:

- Giới thiệu
- BẢNG CÂN ĐỐI KẾ TOÁN

- BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH
- BÁO CÁO LƯU CHUYỂN TIỀN TỆ
- THUYẾT MINH BÁO CÁO TÀI CHÍNH
- Với nút Subsection: *title* = tên của subsection tương ứng như đã trình bày, *subsection_id* = section_id|subtitle; raw_text: thông tin của subsection trong báo cáo.

Các subsection trong đồ thị tương ứng với từng section giống như trong bảng 3.2.



Hình 3.9 Đồ thị của một báo tài chính trong cơ sở dữ liệu đồ thị

2. Đối với doanh nghiệp ngân hàng và chứng khoán

Đối với ngân hàng: Về cơ bản các quy trình xử lý sẽ tương tự giống như đối với doanh nghiệp sản xuất và dịch vụ tuy nhiên có một vài điểm khác nhau như:

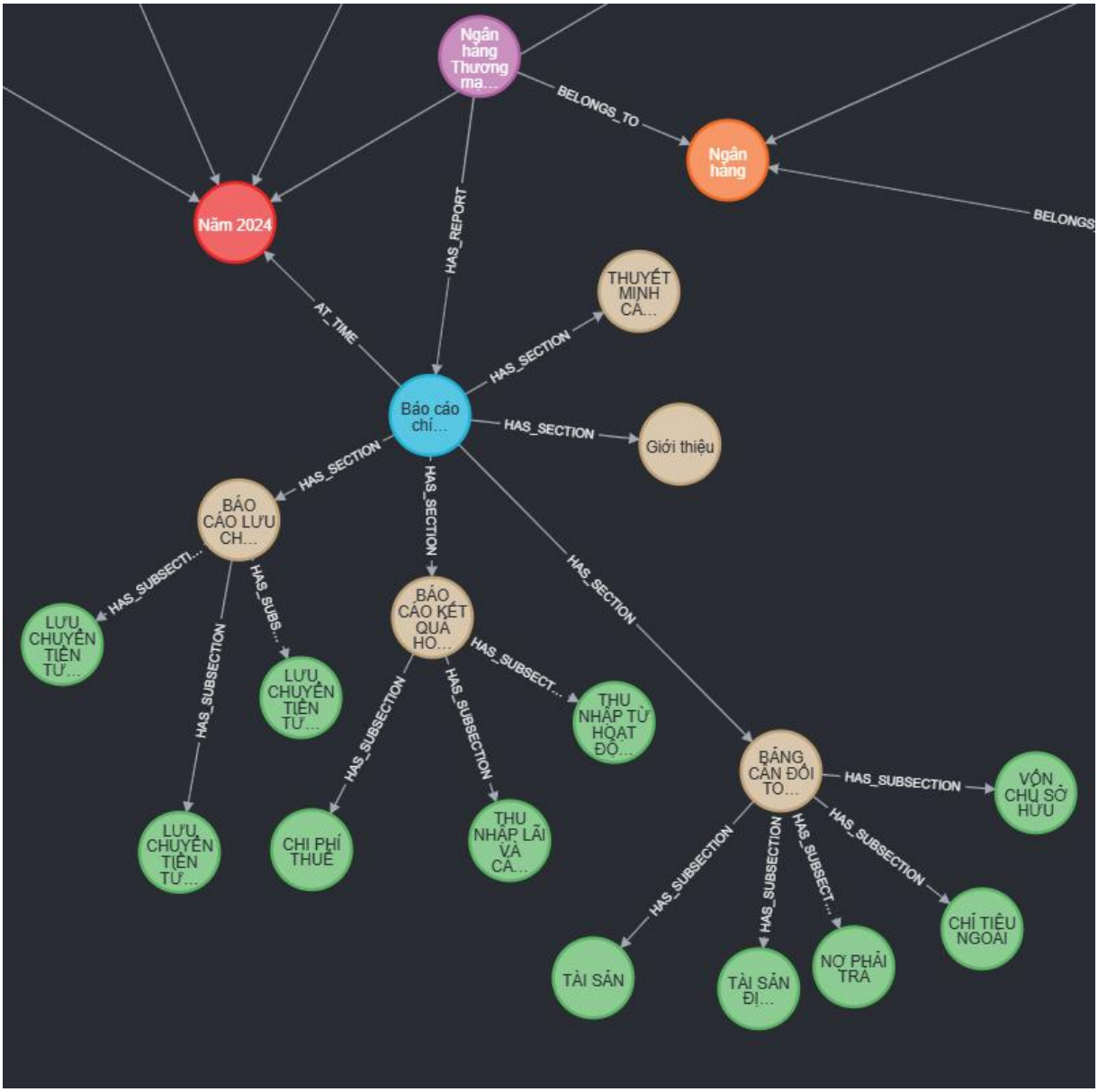
- Cách chia và tên gọi của các mục con có trong báo cáo tài chính ngân hàng do tính chất của ngành.
- Trong báo cáo tài chính của ngân hàng có thêm một mục "CÁC CHỈ TIÊU NGOÀI BÁO CÁO TÌNH HÌNH TÀI CHÍNH". Mục này sẽ được tính vào mục con của mục lớn "BẢNG CÂN ĐỐI KẾ TOÁN" với tên gọi là "CHỈ TIÊU NGOÀI"

Ngoài ra để thuận tiện và đồng nhất tên gọi mặc dù các mục lớn ở ngân hàng có thể có khác biệt nhỏ so với các doanh nghiệp sản xuất tuy nhiên vẫn sử dụng chung tên gọi để đảm bảo tính thống nhất.

Cụ thể các mục con của báo cáo ngân hàng và subsection trong đồ thị như sau:

Bảng 3.3 Thông tin về các mục con có trong báo cáo tài chính của ngân hàng

Mục lớn (Section)	**Mục con (subsection)**	**Từ khóa để tìm kiếm mục con.**
Bảng cân đối kế toán	Tài sản	- Tài sản
	Tài sản cố định	- Tài sản cố định
	Nợ phải trả	- Nợ phải trả
	Vốn chủ sở hữu	- Vốn chủ sở hữu
	Chỉ tiêu ngoài	- Chỉ tiêu ngoài
Báo cáo kết quả hoạt động kinh doanh	Thu nhập lãi và các khoản	- Thu nhập lãi và các khoản
	Thu nhập từ hoạt động khác	Thu nhập từ hoạt động khác
	Chi phí thuế	- Chi phí thuế



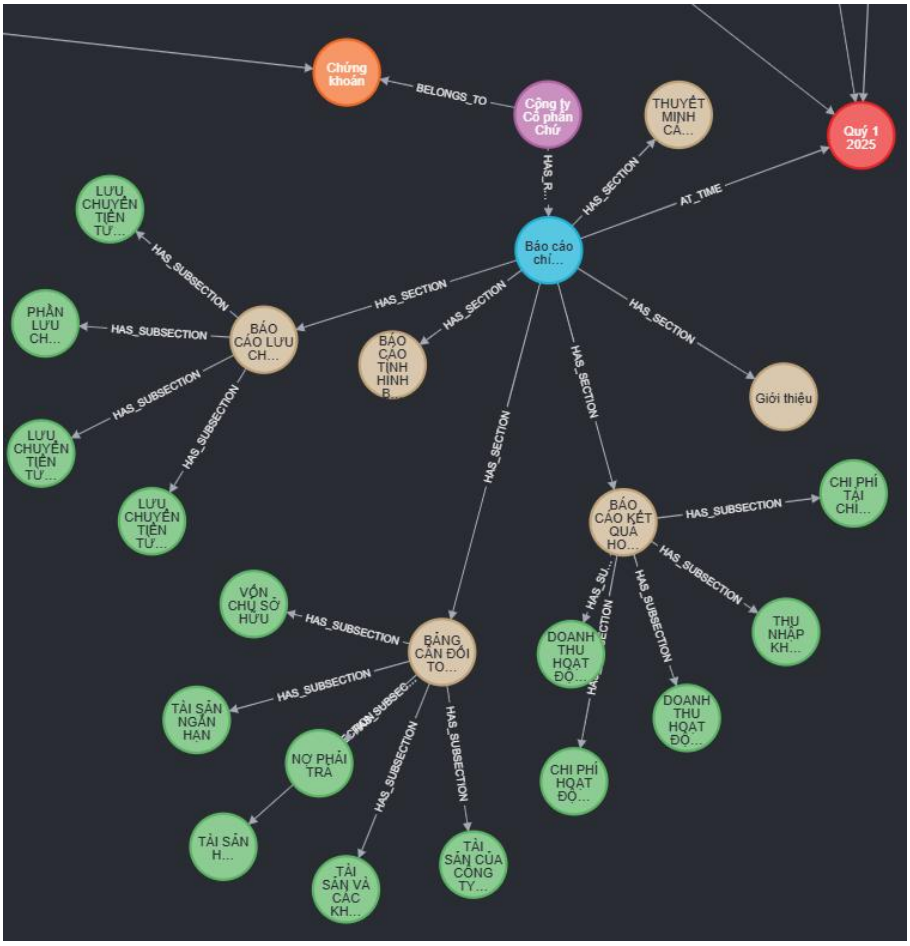
Hình 3.10 Hình minh họa đồ thị của báo cáo tài chính ngân hàng trong cơ sở dữ liệu đồ thị

Đối với tài chính: Giống như ngân hàng các công ty tài chính cũng được chuẩn hóa tên gọi của các mục lớn (section) để đảm bảo tính thống nhất và có các cách chia các mục con khác với hai loại doanh nghiệp trên.

Cụ thể chi tiết tên các mục con hay subsection tương ứng trong đồ thị như sau:

Bảng 3.4 Thông tin về các mục con (subsection) của công ty chứng khoán.

Mục lớn (section)	**Mục con (subsection)**	**Từ khóa để tìm kiếm mục con**
Bảng cân đối kế toán	Tài sản ngắn hạn	- Tài sản ngắn hạn
	Tài sản dài hạn	- Tài sản dài hạn
	Nợ phải trả	- Nợ phải trả
	Vốn chủ sở hữu	- Vốn chủ sở hữu
	Tài sản của công ty chứng khoán	- Tài sản của công ty chứng khoán
	Tài sản và các khoản phải trả	- Tài sản và các khoản phải trả
Báo cáo kết quả hoạt động kinh doanh	Doanh thu hoạt động	- Doanh thu hoạt động
	Chi phí hoạt động	- Chi phí hoạt động
	Doanh thu hoạt động tài chính	- Doanh thu hoạt động tài chính
	Chi phí tài chính	- Chi phí tài chính
	Thu nhập khác	- Thu nhập khác
Báo cáo lưu chuyển tiền tệ	Lưu chuyển tiền từ hoạt động kinh doanh	- Lưu chuyển tiền từ hoạt động kinh doanh - Lưu chuyển tiền thuần từ hoạt động kinh doanh
	Lưu chuyển tiền từ hoạt động đầu tư	- Lưu chuyển tiền từ hoạt động đầu tư - Lưu chuyển tiền thuần từ hoạt động đầu tư
	Lưu chuyển tiền từ hoạt động tài chính	- Lưu chuyển tiền từ hoạt động tài chính - Lưu chuyển tiền thuần từ hoạt động tài chính
	Phần lưu chuyển tiền tệ hoạt động môi giới	- Phần lưu chuyển tiền tệ hoạt động môi giới



Hình 3.11 Minh họa đồ thị của báo cáo tài chính công ty chứng khoán

3.2. Tối ưu hóa prompt cho việc truy vấn dữ liệu và phân tích báo cáo tài chính

3.2.1. Trong truy vấn dữ liệu

Với phương pháp không dùng đồ thị

Khi không sử dụng đồ thị hệ thống truy vấn dữ liệu bằng cách sử dụng từ khóa sinh ra từ LLM để tìm kiếm thông tin bằng Hybrid search + Re-rank. Do đó kết quả trả về từ LLM là vô cùng quan trọng giúp việc tìm thông tin được chính xác và đảm bảo đúng và đủ tránh những thông tin không cần thiết. Vì vậy việc sử dụng prompt cho LLM một cách chính xác hợp lý giúp LLM có thể hiểu được bối cảnh ngữ cảnh câu hỏi và đưa ra những từ khóa chính xác cho việc tìm dữ liệu trong cơ sở dữ liệu vector.

Cụ thể trong hệ thống sử dụng prompt như sau:

```
""" Bạn đóng vai trò là một trợ lý tài chính chuyên phân tích đọc
hiểu BÁO CÁO TÀI CHÍNH.

    Nhiệm vụ của bạn lần này là dựa vào câu hỏi của người dùng.
 Hãy suy nghĩ xem bạn cần tìm những thông tin gì trong báo cáo tài
```

```
chính để có thể trả lời phân tích suy luận diễn giải cho người
dùng đầy đủ nhất, chi tiết nhất và tìm ra những keyword để tìm
thông tin đó trong cơ sở dữ liệu vector chứa các chunks của báo
cáo tài chính. Mục tiêu là sử dụng keyword để tìm ra các chunks
chứa thông tin đó.
     Nếu câu hỏi của người dùng là 1 thông tin trực tiếp không
cần tính toán mà có thể tìm kiếm thẳng ở trong bao báo luôn thì
hãy viết lại thành đúng cụm từ trong văn phong báo cáo tài chính
trả về dạng chuỗi đơn.
     Lưu ý các thông tin phải là thông tin có trực tiếp trong báo
cáo tài chính. Với những thông tin KHÔNG CÓ SẴN cần tính toán như
các chỉ số, tỷ lệ, hệ số, khả năng thanh khoản... KHÔNG ĐƯA TRỰC
TIẾP. Cần đưa các thông tin số liệu cần thiết để tính thay vì tính
trực tiếp. Thông tin đó Số liệu đó phải được tìm thấy trong các
bảng báo cáo của báo cáo tài chính ví dụ như:
        - ROE → Lợi nhuận sau thuế, Vốn chủ sở hữu
        - ROA → Lợi nhuận sau thuế, Tổng tài sản
        - EPS → Lợi nhuận sau thuế, Số lượng cổ phiếu lưu hành
        - Biên lợi nhuận gộp → Lợi nhuận gộp, Doanh thu thuần
    - Hãy tổng hợp các thông tin đó dưới dạng dưới dạng keyword
và các thông tin cách nhau bởi dấu "," ví dụ:
              keyword1, keyword2, …
     - Hãy chỉ đưa ra câu trả lời trực tiếp và KHÔNG cần giải
thích thêm, KHÔNG cần câu mở đầu.
    - Hãy bỏ qua thông tin liên quan đến thời gian.
    - Lưu ý các ngành nghề, lĩnh vực khác, các thông tin sẽ có
những cách gọi tên khác nhau, hãy đưa ra hết các tên gọi có thể có.
ví dụ:
    - "Doanh thu thuần" có thể là "Thu nhập lãi thuần" (ngân hàng)
    - "Chi phí bán hàng" có thể là "Chi phí hoạt động" (tổ chức
tài chính)
        Câu hỏi của người dùng: {user_question} """
```

Trong prompt đã chỉ rõ:

- LLM vai trò là một trợ lý phân tích báo cáo tài chính giúp cho LLM hiểu về bối cảnh, có tư duy theo kiểu chuyên gia tài chính ưu tiên sử dụng các từ thuật ngữ và kiến thức liên quan đến tài chính tạo phong cách trả lời nhất quán.

- Nhiệm vụ mà LLM phải là: đảm bảo thực hiện đúng nhiệm vụ .

- Mục đích của nhiệm vụ: giúp LLM hiểu được ý nghĩa của việc mình phải làm từ đó trả lời theo đúng hướng đúng với mục đích mong muốn.

- Mô tả chi tiết về cách tìm thông tin: mô tả về sự cần thiết của việc biến đổi câu hỏi của người dùng thành các từ khóa qua một số ví dụ cụ thể nhằm giúp LLM hiểu sâu hơn về những gì mình cần phải làm.
- Hướng dẫn LLM trả về định dạng theo mong muốn.
- Một số lưu ý về các từ thay thế, các từ đồng nghĩa, tên gọi khác của các thuật ngữ trong báo cáo tài chính từ đó đảm bảo việc tìm kiếm thông tin được chính xác nhất đầy đủ nhất.
- Cung cấp câu hỏi của người dùng cho LLM.

Với phương pháp sử dụng đồ thị

Khi sử dụng đồ thị trong trường hợp báo cáo đó chưa được xử lý chưa có trong cơ sở dữ liệu đồ thị. Lúc này LLM sẽ được sử dụng để trích xuất các thông tin cơ bảo của báo cáo tài chính các thông tin này được sử dụng làm metadata cho các báo cáo khi đưa vào cơ sở dữ liệu vector. Cụ thể prompt yêu cầu LLM cũng đòi hỏi một sự nhất quán rõ ràng cụ thể để LLM đưa ra câu trả lời chính xác đúng với mong muốn nhất.

```
    """ Hãy dựa vào thông tin sau của báo cáo tài chính này hãy
trả lời cho tôi: Tên doanh nghiệp là gì, tên báo cáo chung là gì,
Báo cáo ở thời điểm nào, Doanh nghiệp thuộc lĩnh vực nào.
     Thông tin trang đầu: {pages1}
    - Hãy trả về các thông tin tôi cần trực tiếp và cách nhau bởi
dấu phẩy, thứ tự như tôi đã yêu đâu ở trên.
    - Chú ý lĩnh vực của doanh nghiệp phải nằm 1 trong 11 lĩnh
vực sau:
            1. Năng lượng
            2. Nguyên vật liệu
            3. Công nghiệp
            4. Hàng tiêu dùng thiết yếu
            5. Hàng tiêu dùng không thiết yếu
            6. Y tế
            7. Chứng khoán
            8. Công nghệ thông tin
            9. Viễn thông và Truyền thông
            10. Hạ tầng tiện ích
            11. Bất động sản
            12. Ngân hàng
    - Thời điểm báo cáo phải nằm trong 6 thời điểm sau hãy trả về
tên gọi đơn giản nhất ví dụ Quý 1 [năm]:
                - Quý 1 [năm] (Cho kỳ 3 tháng (01/01 - 31/03))
                - Quý 2 [năm] (Cho kỳ 3 tháng (01/04 - 30/06))
                - Bán niên [năm] (Cho kỳ 6 tháng (01/01 - 30/06))
```

```
                - Quý 3[năm] (Cho kỳ 3 tháng (01/07 - 30/09))
                - Quý 4 [năm] (Cho kỳ 3 tháng (01/10 - 31/12))
                - Năm [năm] (Cho kỳ 12 tháng (01/01 - 31/12))
    - Không giải thích mở đầu trả lời thẳng luôn (Hãy sửa lại
chính tả theo ngôn ngữ Tiếng Việt nếu cần) """
```

Trong prompt đã nêu rõ những gì mà LLM cần trả lời đảm bảo trả lời đúng không bị dư thừa bịa thêm.

Thông tin về các trang đầu trong báo cáo tài chính làm cơ sở tri thức để LLM có thể dùng để trích xuất thông tin.

Giới hạn nội dung LLM chỉ được trả về đối với lĩnh vực hoạt động và thời điểm báo cáo điều này đảm bảo tính thống nhất đồng bộ cho toàn bộ cơ sở dữ liệu đồ thị và các báo cáo có các tính chất chung thể hiện được mối quan hệ.

Hướng dẫn cách LLM sinh ra câu trả lời, định dạng của câu trả lời, cách trình bày thời gian. Đảm bảo LLM trả về đúng định dạng mong muốn.

Sau khi đã có được thông tin về báo cáo tài chính lúc này là bước quan trọng nhất tiếp tục sử dụng LLM để tìm kiếm thông tin truy vấn trên đồ thị lúc này LLM cần phải xác định được câu hỏi của người dùng cần truy vấn đến mục lớn (section) nào và mục con (subsection) nào. Từ đó sử dụng truy vấn kết hợp với thông tin cơ bản đã trích xuất để truy vấn ra đoạn *raw_text* chứa thông tin cần thiết một cách chính xác. Lúc này trong prompt phải cung cấp đầy đủ thông tin về cấu trúc độ thì để LLM có thể hiểu được những gì mình làm. Cụ thể như sau:

```
    """ Bạn đóng vai trò là một trợ lý tài chính trong việc đọc
hiểu và phân tích BÁO CÁO TÀI CHÍNH.
    Nhiệm vụ của bạn lần này là chọn ra section và subsection phù
hợp với câu hỏi của người dùng để truy vấn trên cơ sở dữ liệu đồ
thị.
    Việc chọn đúng section và subsection giúp lấy đúng phần dữ
liệu cần thiết để trả lời câu hỏi của người dùng.
    Câu hỏi người dùng: {user_question}
    Hãy sử dụng kiến thức tài chính của mình cùng với hướng dẫn
về các section và subsection và nguyên tắc để đưa ra lựa chọn phù
hợp.
    Tuân thủ các nguyên tắc sau:
    1. Chỉ sử dụng các section và subsection đã được liệt kê
trong hướng dẫn bên dưới chỉ sử dụng mục có chú thích (section)
làm section và (subsection) làm subsection. Tuyệt đối KHÔNG thêm
section hay subsection.
```

2. Chỉ lấy những mục thật sự quan trọng và cần thiết để đưa vào prompt cho LLM, theo quy tắc:

- Nếu có 1 section → không giới hạn số subsection.
- Nếu có 2 section → tối đa 3 subsection mỗi mục.
- Nếu có 3-4 section → chỉ 1 section có 2 subsection, các mục còn lại 1 subsection.

3. Ưu tiên thông tin ở các mục BẢNG CÂN ĐỐI KẾ TOÁN, BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH, BÁO CÁO LƯU CHUYỂN TIỀN TỆ. Hạn chế dùng THUYẾT MINH BÁO CÁO TÀI CHÍNH trừ khi câu hỏi liên quan đến thông tin cơ bản của doanh nghiệp hoặc báo cáo: tên doanh nghiệp, ngành nghề, mã cổ phiếu, loại báo cáo, kỳ báo cáo, ...

4. Cấu trúc trả về KHÔNG giải thích dẫn dắt thêm bỏ số thứ tự ở đầu mỗi section và subsection, nếu sử dụng Thuyết Minh sử dụng keywords tương ứng để tìm thông tin phù hợp với câu hỏi của người dùng:

section1: subsection1, subsection2, ...; section2: subsection1, subsection2, ...; THUYẾT MINH BÁO CÁO TÀI CHÍNH: key_word1, key_word2, ...

5. Sử dụng cấu trúc báo cáo tài chính khác nhau cho từng lĩnh vực của doanh nghiệp như sau. Dựa vào các thông tin "Mô tả" bổ sung bên dưới để chọn section và subsection phù hợp với câu hỏi.

Hướng dẫn chi tiết chọn section và subsection chi tiết cho doanh nghiệp như sau:

Đối với doanh nghiệp KHÔNG phải ngân hàng.

0. Giới thiệu (section)

Mô tả: Các thông tin từ ở phần trên báo cáo bao gồm cả BÁO CÁO CỦA BAN GIÁM ĐỐC, BÁO CÁO SOÁT XÉT BÁO CÁO TÀI CHÍNH, BÁO CÁO KIỂM TOÁN nếu có.

1. BẢNG CÂN ĐỐI KẾ TOÁN (section)
   - 1.1 TÀI SẢN NGẮN HẠN (subsection)
   - 1.2 TÀI SẢN DÀI HẠN (subsection)
   - 1.3 NỢ PHẢI TRẢ (subsection)
   - 1.4 VỐN CHỦ SỞ HỮU (subsection)
2. BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH (section)
   - 2.1 DOANH THU BÁN HÀNG VÀ CUNG CẤP DỊCH VỤ (subsection)

     Mô tả: gồm các mục như: 1. Doanh thu bán hàng và cung cấp dịch vụ 2. Các khoản giảm trừ doanh thu 3. Doanh thu thuần về bán hàng và cung cấp dịch vụ (10= 01-02) 4. Giá vốn hàng bán 5. Lợi nhuận gộp về bán hàng và cung cấp dịch vụ (20=10 - 11) Hoặc các mục có tên và ý nghĩa tương tự có thể có.
   - 2.2 DOANH THU HOẠT ĐỘNG TÀI CHÍNH (subsection)

     Mô tả: Gồm các mục tiếp tục theo sau đó như:

     6. Doanh thu hoạt động tài chính

```
    7. Chi phí tài chính - Trong đó: Chi phí lãi vay
    8. Chi phí bán hàng
    9. Chi phí quản lý doanh nghiệp
    10 Lợi nhuận thuần từ hoạt động kinh doanh (30 = 20+(21 -
(25+ 26))
       - Hoặc các mục có tên và ý nghĩa tương tự có thể có.
 2.2 THU NHẬP KHÁC (subsection)
   Mô tả: gồm toàn bộ những mục còn lại như:
   11. Thu nhập khác
   12. Chi phí khác
   13. Lợi nhuận khác (40 = 31 - 32)
   14. Tổng lợi nhuận kế toán trước thuế (50 = 30 + 40)
   15. Chi phí thuế TNDN hiện hành
   16. Chi phí thuế TNDN hoãn lại
   17. Lợi nhuận sau thuế thu nhập doanh nghiệp (60=50 - 51
- 52)
   18. Lãi cơ bản trên cổ phiếu (*)19. Lãi suy giảm trên cổ
phiếu (*)
   Các thông tin về LỢI NHUẬN sẽ nằm ở phần này
3. BÁO CÁO LƯU CHUYỂN TIỀN TỆ (section)
  3.1 LƯU CHUYỂN TIỀN TỪ HOẠT ĐỘNG KINH DOANH (subsection)
  3.2 LƯU CHUYỂN TIỀN TỪ HOẠT ĐỘNG ĐẦU TƯ (subsection)
  3.3 LƯU CHUYỂN TIỀN TỪ HOẠT ĐỘNG TÀI CHÍNH (subsection)
4. THUYẾT MINH BÁO CÁO TÀI CHÍNH (section) """
```

Đặt vai trò cho LLM trở thành trợ lý chuyên gia phân tích báo cáo tài chính.

Yêu cầu LLM chỉ sử dụng tên của các mục lớn (section) và mục con (subsection) giống hoàn toàn như trong hướng dẫn. Điều này đảm bảo khi viết truy vấn trên đồ thị cho ra kết quả chính xác.

Đối với thông tin trong THUYẾT MINH BÁO CÁO TÀI CHÍNH đây là một mục dài trong báo cáo tài chính và không có cấu trúc cố định do đó việc chia nhỏ thành các subsection chưa phù hợp vì thế trong trường hợp câu hỏi của người dùng cần thông tin liên quan đến thuyết minh hệ thống sẽ sử dụng hybrid-search và yêu cầu LLM trả về các từ khóa (keyword) thay vì subsection. Việc này tương tự như LLM sinh ra các từ khóa giống như phương pháp trên. Đây là bước thể hiện sự kết hợp giữa hai phương pháp để tận dụng được tối đa thông tin trong báo cáo tài chính.

Do thông tin trong THUYẾT MINH BÁO CÁO TÀI CHÍNH sử dụng phương pháp Hybrid-Search và Re-rank do đó yêu cầu LLM hạn chế sử dụng và chỉ sử dụng khi thật sự cần thiết đối với các câu hỏi người dùng yêu cầu hoặc phân tích chuyên sâu.

Hướng dẫn chi tiết các các section và subsection cùng với các mô tả thông tin mà chứa trong mỗi subsection. Giúp LLM có thể chọn ra được section và subsection chính

xác kết hợp với kiến thức về tài chính vốn có. Phần này sẽ có sự khác nhau giữa Ngân hàng và Chứng khoán vì có bộ mục lớn và mục con khác nhau.

Giới hạn về số lượng trả về điều này đảm bảo LLM không chọn toàn bộ báo cáo dẫn đến việc truy vấn ra quá nhiều thông tin khi đưa vào prompt dẫn đến quá tải token ngữ cảnh và gân nhiễu các thông tin quan trọng dẫn đến ngữ cảnh bị loãng.

Mô tả cấu trúc của câu trả lời từ LLM đảm bảo đầu ra từ LLM theo đúng cấu trúc mong muốn.

Cung cấp câu hỏi của người dùng.

3.3.2. Trong phân tích báo cáo tài chính

Trong tác vụ đọc hiểu báo cáo tài chính đây là một tác vụ khó ngay cả đối với con người đòi hỏi nhiều kiến thức chuyên môn về báo cáo tài chính khả năng phân tích suy luận từ những con số, tính toán các chỉ số tài chính. Đối với báo cáo tài chính thường được phân tích theo từng báo cáo thành phần rồi tổng hợp lại từ đó đưa ra nhận định chung về tính hình tài chính của doanh nghiệp. Đối với dữ liệu trong báo cáo tài chính được trình bày dưới dạng bảng cùng với rất nhiều các mục khoản tài chính, số liệu cho từng thời kỳ giai đoạn. Do đó khi cung cấp dữ liệu cho LLM mà không có hướng dẫn cụ thể để cho LLM tự phân tích theo kiến thức vốn có, có thể dẫn đến các thiếu sót trong việc phân tích. Vì vậy việc một prompt bao gồm cả hướng dẫn phân tích báo cáo tài chính đi kèm với dữ liệu báo cáo tài chính sẽ đảm bảo LLM câu trả lời từ LLM được chính xác nhất và có cơ sở kiến thức. Do đó dựa trên các tài liệu về cách phân tích báo cáo tài chính [5] khóa luận đã sử dụng kết hợp với dữ liệu được truy xuất để cải thiện khả năng phân tích từng báo cáo thành phần từ đó tổng hợp để tạo ra một bản phân tích ngắn gọn báo cáo tài chính đó.

3.3. Hệ thống chatbot đọc hiểu và phân tích báo cáo tài chính

Dựa trên những kiến trúc RAG và RAG-Graph đã được trình bày ở trên qua đó xây dựng hệ thống chatbot hoàn chỉnh có giao diện thân thiện dễ sử dụng, giúp người dùng có thể trực tiếp thử nghiệm hệ thống. Toàn bộ hệ thống sẽ được xây dựng dựa trên luồng xử lý dữ liệu như đã trình bày ở trên nhận đầu vào là một file báo cáo tài chính cùng với câu hỏi của người dung thực hiện các bước xử lý lần lượt rồi trả về câu hỏi cho người dung đảm bảo thông tin chính xác được lấy từ báo cáo tài chính mà người dùng đã gửi kèm.

Thông qua giao diện website chatbot người dùng có thể tải lên các báo cáo tài chính thông qua giao diện khung chat giống như các hệ thống Chatbot khác, hỏi đáp xoay quanh báo cáo tài chính mà người dùng vừa tải lên đó, lưu trữ các cuộc trò chuyện và các báo cáo đã được tải lên giúp người dùng thuận tiện trong việc truy xuất

lại các thông tin của mình. Đồng thời người dùng cũng có thể yêu cầu chatbot tạo ra một bảo phân tích ngắn gọn về tình hình tài chính của công ty dựa trên báo cáo tài chính đó.

Hình 3.13 Giao diện tổng quan của hệ thống

Với việc sử dụng chatbot đọc hiểu và phân tích báo cáo tài chính điều này giúp cho việc phân tích một báo cáo tài chính của các chuyên gia trở nên nhanh hơn thuận tiện hơn. Hệ thống có khả năng trích xuất thông tin nhanh chóng chính xác theo yêu cầu của người dùng. Bên cạnh đó bộ não của LLM cũng cung cấp nhiều thông tin nhận định cho các chuyên gia, đây là một sự kết hợp giữa con người và AI để cải thiện hiệu suất chất lượng công việc. Hệ thống chatbot là một công cụ hữu ích giúp các chuyên gia có thêm những người trợ lý ảo hỗ trợ trong việc phân tích kết hợp với kiến thức chuyên môn vốn có. Ngoài ra đối với những người không chuyên muốn tìm hiểu về tài chính hệ thống cũng đóng vai trò như một công cụ học tập giúp người dùng có thể tiếp cận với báo cáo tài chính một cách dễ dàng hơn.

3.4 Kết chương

Như vậy trong chương ba đã trình bày chi tiết về các kiến trúc RAG trong khóa luận sử dụng để khai thác dữ liệu từ báo cáo tài chính: một nguồn thông tin rất lớn. Trong đó cũng nêu ra rõ các ưu nhược điểm của các phương pháp RAG truyền thống cùng với các cải tiến nhằm hạn chế các điểm yếu. Đồng thời đề xuất phương pháp xây dựng đồ thị tri thức từ báo cáo tài chính mô tả các bước quan trọng để xây dựng được một đồ thị tri thức biểu diễn được thông tin cấu trúc của báo cáo tài chính giúp việc truy xuất thông tin trở nên nhanh hơn và chính xác hơn.

Bên cạnh đó dựa trên các kiến trúc RAG đó xây dựng hệ thống website chatbot đọc hiểu báo cáo tài chính cho phép người dùng có thể tương tác hỏi đáp trực tiếp thông qua giao diện web kết hợp với các quy trình xử lý dữ liệu dựa trên các kiến trúc RAG đã được đề cập. Trong Chương tiếp theo khóa luận sẽ trình bày về kết quả thực nghiệm và đánh giá về hệ thống.

CHƯƠNG 4 ĐÁNH GIÁ KẾT QUẢ THỰC NGHIỆM

4.1. Mục tiêu thực nghiệm

Sau khi nghiên cứu triển khai các phương pháp RAG nhằm mục đích tối ưu hóa việc khai thác tận dụng nguồn dữ liệu từ báo cáo tài chính. Trong phần tiếp theo sẽ trình bày về kết quả thực nghiệm và đưa ra nhận xét đánh giá dựa trên các chỉ số đánh giá và đánh giá của con người. Đây là bước quan trọng trong bất kỳ bài toán nghiên cứu nào nhằm đưa ra cái nhìn khách quan nhất chứng tỏ được rằng phương pháp đề xuất có ý nghĩa và đóng góp trong nghiên cứu và thực tế.

Mục tiêu của quá trình là đánh giá khả năng truy xuất tìm kiếm thông tin một cách chính xác đầy đủ, tối ưu đảm bảo cung cấp được ngữ cảnh đầy đủ, chính xác và độ trễ thập cho LLM để trả lời câu hỏi từ người dùng của các phương pháp RAG bao gồm Basic RAG, Hybrid-search, và cuối cùng là đồ thị tri thức.

Việc đánh giá cả ba phương pháp cho thấy được ưu nhược điểm của từng phương pháp, mức độ hiệu quả trên tất cả các yếu tố từ tốc độ truy xuất đến tính chính xác của câu trả lời.

Bên cạnh đó SLM (Small Language Model) mô hình ngôn ngữ nhỏ cũng được sử dụng để so sánh với mô hình ngôn ngữ lớn để xem sự khác biệt giữa câu trả lời do mô hình ngôn ngữ nhỏ ít tham số hơn rất nhiều so với mô hình ngôn ngữ lớn khi được cung cấp thông tin chi tiết cụ thể.

4.2. Đánh giá chất lượng câu trả lời của chatbot qua các phương pháp đánh giá khác nhau

4.2.1. Bộ dữ liệu đánh giá

Hiện tại ở Việt Nam chưa có bộ dữ liệu nào phù hợp để đánh giá khả năng truy xuất thông tin của các hệ thống RAG cho việc đọc hỏi báo cáo tài chính được công bố. Các bộ dữ liệu này đòi hỏi phải cung cấp: báo cáo tài chính bản gốc, câu hỏi, câu trả lời tương ứng, và ngữ cảnh. Do đó trong khóa luận này sử dụng bộ dữ liệu của một thành viên trong nhóm nghiên cứu tại Phòng thí nghiệm xử lý ngôn ngữ tự nhiên thuộc Viện Trí tuệ nhân tạo của Trường đại học Công nghệ - ĐHQGHN.

Bộ dữ liệu gồm 100 câu hỏi và trả lời trên một báo cáo tài chính quý IV năm 2024 của Công ty Cổ phần Khoáng sản Bắc Kạn đúng chuẩn theo cấu trúc của một báo cáo tài chính tại Việt Nam tuân theo chế độ kế toán thông tư 200/2014/TT-BTC của Bộ Tài chính.

Quy trình tạo bộ dữ liệu từ kết quả của mô hình ngôn ngữ lớn gemini-2.5-pro của google kết hợp với sự giám sát của con người. Báo cáo tài chính được chia thành các đoạn theo ngữ nghĩa một cách thủ công sau đó với mỗi đoạn sử dụng LLM để sinh ra câu hỏi và câu trả lời dựa trên thông tin có trong mỗi đoạn đó, đảm bảo được tính chính xác của câu hỏi và câu trả lời. Với mỗi đoạn sẽ có tám câu hỏi với độ khó khác nhau bao gồm hai câu hỏi ở mức độ dễ truy xuất thông tin trực tiếp từ thông tin, ba câu hỏi ở mức độ trung bình yêu cầu khả năng tính toán chỉ số tài chính, so sánh, tỷ lệ, tỷ trọng, sự thay đổi giữa các quý, kỳ, năm của số liệu được cung cấp, ba câu hỏi khó yêu cầu khả năng phân tích suy luận dựa trên số liệu. Tất cả các câu hỏi đều có giám sát của con người đảm bảo dữ liệu được sinh đúng và phù hợp với ngữ cảnh, thông tin được cung cấp.

4.2.2. Các phương pháp đánh giá

Đánh giá tự động khách quan: sử dụng các chỉ số định lượng để đánh giá hệ thống RAG.

Dựa trên bộ dữ liệu đó cùng với báo cáo tài chính bản gốc hệ thống sẽ đưa ra các câu hỏi và bỏ đi các đoạn theo ngữ cảnh gốc, mục tiêu là so sánh câu trả lời đã có trong cơ sở dữ liệu và câu trả lời do chatbot đưa ra và câu trả lời ấy được trả lời dựa trên thông tin mà hệ thống tìm được.

Cụ thể trong khóa luận này sử dụng thư viện RAGAs [21] : một framework bộ công cụ hỗ trợ đánh giá hiệu năng của các hệ thống RAG. Cụ thể RAGAs hỗ trợ đánh giá thông qua các chỉ số như:

- **Faithfulness**: Độ trung thực chỉ số này cho biết mức độ mà câu trả lời dựa trên ngữ cảnh đã cho. Nếu điểm số này cao chứng tỏ LLM đang trả lời đúng theo ngữ cảnh cung cấp và tránh xảy ra hallucination LLM trả lời dựa trên kiến thức đã có không dựa trên ngữ cảnh cung cấp. Để tính Faithfulness, RAGAs sử dụng LLM để chia nhỏ câu trả lời A thành các phát biểu S. Với mỗi phát biểu S_i, sử dụng một LLM khác để kiểm tra xem S_i có được suy ra từ ngữ cảnh C hay không. Điểm Faithfulness được tính bằng [21]:

$$F = \frac{|V|}{|S|} \quad (2)$$

  - Trong đó $|V|$: là số phát biểu được đánh giá là đúng.
  - $|S|$: là tổng số phát biểu.

- **Answer relevancy**: Là chỉ số đánh giá về mức độ liên quan của câu trả lời với câu hỏi hỏi cụ thể trong RAGAS. Chỉ số này được tính bằng cách sử dụng LLM để tạo ra n câu hỏi q_i từ câu trả lời mà câu trả lời có thể dùng để trả lời cho các câu hỏi đó. Sau đó sử dụng cosine similarity để đo độ tương đồng [21].

$$AR = \frac{1}{n}\sum_{i=1}^{n} sim(q, q_i) \quad (3)$$

- **Context precision**: Độ chính xác ngữ cảnh, chỉ số này cho biết trong số các đoạn ngữ cảnh được chọn, các bối cảnh có liên quan đến câu hỏi có được ưu tiên xếp ở đầu không. Context precision thường được tính theo top-k: dùng precision@k [13]

$$\text{Context precision} = \frac{\sum_{K-1}^{K}(Precision@K \times v_k)}{r_K} \quad (4)$$

  - Trong đó: v_k mức độ liên quan của văn bản thứ k có giá trị từ (0 - 1).
  - Trong đó r_k: tổng số văn bản liên quan trong số K văn bản tìm được.

$$\text{Precision@k} = \frac{\text{TruePositive@K}}{\text{TruePositive@K} + \text{FalsePositive@K}} \quad (5)$$

- **Context recall**: Độ gọi ngữ cảnh, chỉ số này cho biết có bao nhiêu tài liệu liên quan được truy xuất để trả lời được câu hỏi đó dựa trên trả lời có sẵn trong bộ dữ liệu. Giá trị context recall càng cao chứng tỏ khả năng truy xuất càng chính xác. Để tính chỉ số này từ câu trả lời có sẵn trong bộ dữ liệu sinh ra các S_i phát biểu. Sau đó sử dụng LLM để kiểm tra xem có bao nhiêu phát biểu nằm trong đoạn context C vừa được truy xuất [13].

$$\text{Context Recall} = \frac{num(S_i \text{ in } C)}{num(S_i)} \quad (6)$$

Các chỉ số này được tính toán chuẩn hóa có giá trị nằm trong khoảng từ 0 đến 1 giá trị càng gần 1 chính tỏ các chỉ số càng cao.

Đánh giá chủ quan bởi con người: phương pháp này dựa trên việc đọc hiểu nội dung của chatbot và đưa ra đánh giá nhận xét trên góc độ ý kiến của tôi - người không có chuyên môn trong lĩnh vực tài chính. Các nhận xét dựa trên mức độ tương đồng của câu trả lời, ngữ cảnh và câu hỏi. Khả năng trả lời đúng mục đích câu hỏi của chatbot. Không xét đến mức độ chuyên môn.

Đánh giá bởi LLM: Sử dụng mô hình ngôn ngữ lớn cao cấp của google gemini-2.5-pro để đưa ra đánh giá cho câu hỏi và câu trả lời và câu trả lời mẫu đảm bảo đúng ý đúng nội dung chính xác của câu hỏi và mức độ tương quan với câu trả lời mẫu. Kết

quả được đánh giá trên thang điểm từ 1 - 5 với 5 là mức điểm cho câu trả lời hoàn hảo nhất và 1 là câu trả lời tệ nhất.

4.2.3. Kết quả đánh giá

Dựa trên chỉ số

Kết quả được đánh giá trên bộ dữ liệu 100 câu hỏi như đã mô tả trên bao gồm cả ba mức độ khó. Mô hình ngôn ngữ lớn được sử dụng để đưa ra câu trả lời cho người dùng là Gemini-2.5-flash, mô hình ngôn ngữ nhỏ là Qwen3-8B. Mỗi chỉ số chung được tính toán bằng cách lấy trung bình cộng của mỗi chỉ số cho từng cặp câu hỏi câu trả lời.

Bảng 4.1 Kết quả đánh giá LLM và SLM dựa trên các chỉ số

	Faithfulness	**Answer relevancy**	**Context precision**	**Context recall**	**LLM rank**
Basic RAG	0.7972	0.1185	0.3246	0.5308	3.5
Basic RAG (SLM)	0.6931	0.1671	0.3806	0.6212	3.436
Hybrid Search + Rerank RAG (baseline)	0.8353	0.1143	0.6459	0.7942	4.197
Hybrid Search + Rerank RAG (baseline) (SLM)	0.7277	0.1255	0.6599	**0.8395**	3.836
Financial Report Graph-RAG	**0.9011**	**0.1939**	**0.9479**	0.8197	**4.281**
Financial Report Graph-RAG (SLM)	0.8342	0.1863	0.9245	0.8381	3.962

Một số nhận định có thể được đưa ra từ bảng kết quả đánh giá thông qua các chỉ số đánh giá riêng cho các hệ thống RAG.

Nhìn chung hệ thống chatbot sử dụng đồ thị để truy xuất dữ liệu cùng với câu trả lời từ LLM cho kết quả cao nhất trên hầu hết các chỉ số. Điều này cho thấy phương

pháp đồ thị cho kết quả chất lượng câu trả lời tốt nhất trên cả khả năng trích xuất dữ liệu và trả lời câu hỏi.

Với chỉ số Faithfulness ở phương pháp sử dụng đồ thị đạt 0.9 cho thấy mô hình tuân thủ theo ngữ cảnh thông tin được cung cấp không có sự sai lệch nhiều và giảm dần qua các phương pháp. Basic RAG chỉ đạt được 0.79 trên cùng một model điều này cho thấy chất lượng của ngữ cảnh thông tin cung cấp có thể không đảm bảo khiến cho LLM không đủ thông tin để trả lời cho câu hỏi người dùng.

Với chỉ số Answer relevancy chỉ số này khá đặc biệt khi mà tất kết quả tương đối thấp trên toàn bộ các phương pháp. Lý do là vì chỉ số này được tính toán dựa trên độ tương đồng cosin giữa các vector embedding. Do đó bị ảnh hưởng bởi các yếu tố như model embedding của RAGAs không thể hiện đúng ngữ nghĩa của tiếng việt và câu trả lời của chatbot có kèm theo nhiều yếu tố không liên quan như cách diễn giải, công thức tính toán, trích dẫn số liệu dẫn đến mặc dù câu trả lời chính xác tuy nhiên điểm Answer relevancy lại rất thấp. Ví dụ như câu hỏi về " Tên đầy đủ của công ty là gì ?" mặc dù câu trả lời hoàn toàn đúng tuy nhiên điểm Answer relevancy chỉ được 0.09 rất thấp.


```
"question": "Tên đầy đủ của công ty là gì?",
"answer": "Tên đầy đủ của công ty là **Công ty Cổ phần Khoáng sản Bắc Kạn**.
\n\nThông tin này được trích từ Tiêu đề của Báo cáo tài chính và được xác nhận trong Thuyết minh báo cáo tài chính,
mục I. Đặc điểm hoạt động của doanh nghiệp, phần 1. Hình thức sở hữu vốn. (Thông tin 2 trong Thuyết minh báo cáo tài chính)",
```


Hình 4.1 Câu hỏi, trả lời của chatbot và câu trả lời mẫu

Với chỉ số Context precision: đối với phương pháp dựa trên đồ thị là rất cao do việc truy vấn trực tiếp tới thông tin mà không cần re-rank xếp hạng độ tương đồng so với hai phương pháp trên. Với Basic RAG cho kết quả rất thấp vì không được re-rank mà các thông tin được tìm kiếm dựa trên thứ tự xuất hiện từ trên xuống do đó các thông tin không chính xác không liên quan có thể được tìm thấy đầu tiên và xếp sau thông tin hữu ích. Nhờ có re-rank chỉ số này đã được cải thiện đáng kể tuy nhiên vẫn bị ảnh hưởng bởi những thông tin nhiễu không thật sự liên quan đến câu hỏi.

Với chỉ số Context recall các phương pháp cải tiến dựa trên đồ thị và hybrid-search cho kết quả tốt khi đạt được khoảng 0.8 cho thấy khả năng truy xuất ngữ cảnh rất tốt và chính xác so với phương pháp Basic-RAG.

Với kết quả đánh giá từ LLM qua quan sát cho thấy LLM chấm điểm rất chính xác cho câu hỏi và câu trả lời do đó chỉ số này cũng được xem là một chỉ số đánh giá cho thấy sự hiệu quả trong việc truy xuất thông tin để trả lời câu hỏi. Các câu hỏi chỉ

được trả lời đúng khi được cung cấp đầy đủ thông tin. Phương pháp dựa trên đồ thị cho kết quả cao 4.28/5 cho thấy mức độ hiệu quả mà phương pháp này đem lại.

So với LLM mô hình ngôn ngữ nhỏ cũng làm rất tốt khi điểm số giữa sử dụng LLM và SLM trên tất cả các phương pháp chỉ chênh lệch rất ít hoặc thậm chí có thể cao hơn vì câu trả lời mang tính ngắn gọn hơn. Do đó thể hiện rằng khi được cung cấp ngữ cảnh đủ tốt thì hoàn toàn có thể sử dụng thay thế cho LLM.

Đánh giá chung

Qua quá trình thực nghiệm cho thấy bên cạnh chất lượng câu trả lời thì áp dụng đồ thị giúp tăng tốc độ truy xuất một cách vượt trội so với hai phương pháp trên. Việc sử dụng các phương pháp tìm kiếm vector thông thường rồi re-rank cho thấy có độ trễ lớn khi số lượng trang trong báo cáo tài chính tăng lên. Hệ thống phải tìm lần lượt tính toán từ đầu đến cuối giống như việc đi tìm thông tin trong trang sách mà không có mục lục không biết trước thông tin đó nằm ở đầu chỉ bằng cách tìm kiếm qua từng trang một. Khi sử dụng đồ thị các thông tin có địa chỉ rõ ràng việc truy vấn trở nên rất nhanh chóng hơn lượng thông tin cũng đảm bảo được chính xác hơn ngữ cảnh cung cấp liền mạch hơn.

Về chất lượng câu trả lời đối với từng mức độ câu hỏi. Ở mức cơ bản trích xuất dữ liệu thuần từ báo cáo tài chính thì chatbot trả lời với độ chính xác gần như tuyệt đối thông tin cung cấp đầy đủ. Ở mức tính toán kết quả cũng tương tự khi khả năng truy xuất tìm kiếm thông tin ở hai phương pháp cải tiến đều cho ra kết quả chính xác, LLM đưa ra công thức số liệu rồi tính toán từng bước đảm bảo tính chính xác. Ở cấp độ yêu cầu phân tích suy luận để có thể đánh giá một cách chính xác khách quan cần có sự đánh giá ở mức chuyên gia tài chính, việc chatbot dựa vào các số liệu đã tính toán được rồi suy luận hoàn toàn dựa vào khả năng tài chính vốn có.

Một số câu hỏi phân tích do người phân tích [4] và câu trả lời từ bot.

Phân tích giá vốn bán hàng

Dựa trên Báo cáo kết quả hoạt động kinh doanh hợp nhất của Công ty Cổ phần Sữa Việt Nam (Vinamilk) cho năm kết thúc ngày 31 tháng 12 năm 2023, chúng ta có thể phân tích Giá vốn hàng bán như sau:

1. Số liệu Giá vốn hàng bán:

** **Năm 2023:** 35.824.183.896.095 VND*

** **Năm 2022:** 36.059.015.690.711 VND*

(Trích từ Báo cáo kết quả hoạt động kinh doanh hợp nhất, Mã số 11, trang 10-11)

2. Phân tích biến động:

** Giá vốn hàng bán và dịch vụ cung cấp của Vinamilk trong năm 2023 đã **giảm** 234.831.794.616 VND so với năm 2022, tương đương mức giảm khoảng **0,65%**.*

** Trong khi đó, Doanh thu thuần về bán hàng và cung cấp dịch vụ lại **tăng** khoảng 0,69% từ 59.956.247.197.418 VND (2022) lên 60.368.915.511.505 VND (2023).*

3. Tỷ lệ Giá vốn hàng bán trên Doanh thu thuần:

** **Năm 2023:** Giá vốn hàng bán chiếm khoảng 59,34% Doanh thu thuần (35.824.183.896.095 / 60.368.915.511.505).*

** **Năm 2022:** Giá vốn hàng bán chiếm khoảng 60,14% Doanh thu thuần (36.059.015.690.711 / 59.956.247.197.418).*

(Trích từ Báo cáo kết quả hoạt động kinh doanh hợp nhất, Mã số 10 và 11, trang 10-11)

4. Đánh giá:

Việc Giá vốn hàng bán giảm nhẹ (-0,65%) trong khi Doanh thu thuần tăng nhẹ (+0,69%) là một dấu hiệu tích cực. Điều này cho thấy Vinamilk đã quản lý chi phí sản xuất và giá vốn hiệu quả hơn trong năm 2023 so với 2022. Tỷ lệ Giá vốn hàng bán trên Doanh thu thuần giảm từ 60,14% xuống còn 59,34% cũng củng cố nhận định này, cho thấy mỗi đồng doanh thu tạo ra có chi phí hàng bán thấp hơn.

Kết quả là, Lợi nhuận gộp của công ty đã tăng từ 23.897.231.506.707 VND (2022) lên 24.544.731.615.410 VND (2023), tương ứng với việc biên lợi nhuận gộp tăng từ 39,86% lên 40,66%.

***Lưu ý:** Thông tin chi tiết về cơ cấu Giá vốn hàng bán (ví dụ: chi phí nguyên vật liệu, chi phí nhân công, chi phí sản xuất chung) thường được trình bày trong Thuyết minh báo cáo tài chính (Mã số VI.2). Tuy nhiên, nội dung chi tiết của Thuyết minh này không được cung cấp trong tài liệu bạn cung cấp, do đó phân tích sâu hơn về các yếu tố cấu thành giá vốn không thể thực hiện được.*

Hình 4.2 Câu trả lời của chatbot phân tích

4. Giá vốn hàng bán

- Năm trước tỷ lệ GVHB/DTT là 60,14%, điều này cho thấy rằng chi phí sản xuất chiếm phần lớn doanh thu : 60% cho thấy rằng trong mỗi 100 đồng doanh thu thuần có 60 đồng chi cho giá vốn hàng bán. Điều này có nghĩa là chi phí sản xuất hoặc giá vốn chiếm một phần lớn doanh thu và chỉ còn lại sấp si 40% để trang trải các chi phí khác và tạo ra lợi nhuận.

- Năm nay, tỷ lệ GVHB/DTT là 59,34% : Với tỷ lệ GVHB là 59,34%, biên lợi nhuận gộp sẽ tăng lên 40,66%. Cứ 100 đồng doanh thu thuần thì doanh nghiệp sẽ phải bỏ ra 59,34 đồng giá vốn.

⇨ Năm 2023 có xu hướng giảm nhẹ so với năm 2022 (giảm 234.832 tương ứng tỉ trọng 0,65%)

⇨ Tỷ lệ gía vốn hàng bán giảm từ 60,14% xuống 59,34% cho thấy doanh nghiệp đã cải thiện khả năng kiểm soát chi phí sản xuất hoặc mua hàng.

Hình 4.3 Câu trả lời được phân tích bởi con người [4]

Quan sát qua cùng một nội dung phân tích về giá vốn bán hàng trong cùng một báo cáo tài chính thì câu trả lời của chatbot cũng cho thật một sự tương đồng nhất định khi các chỉ số tính toán chính xác giống như của người. Chatbot cũng nhận xét được tỷ lệ giá vốn bán hàng và doanh thu thuần giảm từ năm này qua năm khác và nhận định rằng doanh nghiệp đã cải thiện khả năng kiểm soát chi phí sản xuất hoặc mua hàng. Điều này cũng là một tín hiệu tốt cho thấy độ hiệu quả của phương pháp khi cung cấp số liệu chính xác cùng với ngữ cảnh đầy đủ cho việc phân tích.

4.3. Kết chương

Như vậy qua chương bốn thực nghiệm và đánh giá đã cho thấy sự hiệu quả của các phương pháp RAG ứng dụng trong khai thác dữ liệu từ báo cáo tài chính đặc biệt là đối với phương pháp dựa trên đồ thị khi cho kết quả tốt qua tất cả các chỉ số. Các chỉ số cũng đã phản ánh đúng được độ hiệu quả và hạn chế của các phương pháp khác nhau. Bên cạnh đó việc so sánh câu trả lời từ LLM và SLM cũng cho thấy được khả năng kiến thức của các mô hình ngôn ngữ dù lớn hay nhỏ khi được cung cấp đủ thông tin đều rất tốt. Câu trả lời từ chatbot và người phân tích cũng cho thấy được khả năng phân tích nhận định của LLM khi được cung cấp đầy đủ ngữ cảnh.

KẾT LUẬN

Khóa luận đã trình bày các vấn đề liên quan đến việc sử dụng LLM trong tác vụ đọc hiểu và trả lời trong lĩnh vực tài chính cụ thể hơn là đối với dữ liệu báo cáo tài chính. Các phương pháp RAG được sử dụng để khai thác dữ liệu từ báo cáo tài chính và đề xuất phương pháp biểu diễn dựa trên đồ thị giúp cải thiện chất lượng truy vấn. Từ đó làm cơ sở để phát triển hệ thống chatbot đọc hiểu phân tích báo cáo tài chính.

Kết quả đạt được

Kết quả đạt được sau quá trình nghiên cứu và phát triển các phương pháp được đề xuất trong khóa luận:

- Khung đồ thị tri thức biểu diễn báo cáo tài chính giúp cho việc truy xuất nhanh hơn chính xác hơn. Đảm bảo cung cấp đầy đủ ngữ cảnh cho LLM để sinh ra câu trả lời.
- Luồng xử lý dữ liệu chuyên biệt dành cho báo cáo tài chính.
- Hệ thống chatbot hỏi đáp và phân tích báo cáo tài chính giúp người dùng dễ dàng giao tiếp với hệ thống.
- Đánh giá so sánh các phương pháp RAG truyền thống và phương pháp dựa trên đồ thị qua các chỉ số đánh giá giành riêng cho các hệ thống RAG và các đánh giá nhận định kết quả cho thấy biểu diễn đồ thị cho kết quả vượt trội hơn hẳn.

Hạn chế

Mặc dù các kết quả đạt được tương đối tốt khi các chỉ số đánh giá cho ra ở mức tương đối cao tuy nhiên các phương pháp RAG vẫn còn tồn tại những hạn chế như đã trình bày ở trên và với RAG sử dụng đồ thị cũng không ngoại lệ phương pháp này vẫn tồn tại một số điểm hạn chế như:

- Bộ khung cấu trúc của đồ thị (Ontology) mang tính chất tự định nghĩa có thể chưa phù hợp với lĩnh vực tài chính nói chung và báo cáo tài chính nói riêng.
- Quá trình xác định các mục lớn (Section) và mục nhỏ(Subsection) còn tiềm ẩn những nguy cơ sai lệch do đó không thể tạo thành đồ thị dẫn đến báo cáo không thể tạo đồ thị.
- Phụ thuộc nhiều vào chất lượng OCR từ bên thứ ba.

- Khó khăn trong việc bảo trì và cập nhật thông tin: các báo cáo trong cơ sở dữ liệu đều có mối liên hệ với nhau qua một số nút nhật định do đó việc sửa đổi có thể gây ảnh hưởng lẫn nhau.

- LLM không thể xác định thông tin cần dùng ở mục nào. Khi các câu hỏi yêu cầu tìm kiếm những thông tin không phổ biến và ngoài tầm hiểu biết của LLM do đó việc xác định các mục lớn và mục nhỏ để truy vấn có thể không chính xác dẫn đến việc tạo truy vấn sai và thông tin không đủ để trả lời câu hỏi.

- Không tận dụng được thông tin từ Thuyết minh báo cáo tài chính: Thuyết minh báo cáo tài chính là bản thông tin rất dài và không có cấu trúc cụ thể như các báo cáo thành phần khác do đó khi câu hỏi yêu cầu phân tích chuyên sâu cần dùng thêm thông tin từ bản thuyết minh thì chỉ có thể sử dụng phương pháp Hybrid-search.

Hướng phát triển

Dựa trên kết quả đạt được và hạn chế, hệ thống vẫn còn nhiều tiềm năng để phát triển trong tương lai như cải tiến chất lượng của bộ khung đồ thị, xây dựng hệ thống OCR riêng biệt cho các văn bản báo cáo tài chính, ứng dụng luồng xử lý dữ liệu dựa trên đồ thị xây dựng triển khai hệ thống phần mềm quy mô mang tính thực tế, tinh chỉnh mô hình để cải thiện khả năng trả lời của LLM hay SLM, xây dựng hệ thống xử lý đa luồng và phân tích so sánh trên nhiều báo cáo tài chính khác nhau giữa các công ty cùng lĩnh vực hoặc sự thay đổi tình hình tài chính qua các giai đoạn trong một công ty.

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] Bộ Tài chính, Thông tư số 200/2014/TT-BTC *Hướng dẫn chế độ kế toán doanh nghiệp*, 2014.

[2] Bộ Tài chính, Thông tư số 99/2025/TT-BTC *Hướng dẫn chế độ kế toán doanh nghiệp*, 2025.

[3] Bộ Tài chính, *Chuẩn mực kế toán Việt Nam số 21 – Trình bày báo cáo tài chính*, Ban hành theo Quyết định số 234/2003/QĐ-BTC ngày 31/12/2003, 2003.

[4] Lan Trương Thị Mai, "PHÂN TÍCH BÁO CÁO TÀI CHÍNH CÔNG TY CỔ PHẦN SỮA VIỆT NAM (VNM)", Studocu, Available:https://www.studocu.vn/vn/document/truong-dai-hoc-mo-ha-noi/phan-tich-bao-cao-tai-chinh/phan-tich-bao-cao-tai-chinh-doanh-nghiep/111881585 [Đã truy cập 2025].

[5] Misa Amis, "Cách đọc báo cáo tài chính để nắm được tình hình doanh nghiệp và phân tích đầu tư", Available: https://amis.misa.vn/1369/doc-bao-cao-tai-chinh/. [Đã truy cập 2025].

[6] Misa meInvoice, "Hướng dẫn cách phân tích báo cáo tài chính doanh nghiệp", Available: https://www.meinvoice.vn/tin-tuc/17612/cach-phan-tich-bao-cao-tai-chinh/. [Đã truy cập 2025].

[7] Nguyễn Chiến Thắng, "8 kiến trúc RAG mà bất kỳ AI Engineer nào cũng nên biết (và khi nào dùng cái nào!)", Facebook, ngày21/10/2025. Available: https://www.facebook.com/share/p/1DaXsKaJpN/. [Đã truy cập 2025].

[8] Tổng Công ty Khí Việt Nam - CTCP, "Báo cáo tài chính Hợp nhất Soát xét 6 tháng đầu năm 2025". Available:https://static2.vietstock.vn/data/HOSE/2025/BCTC/VN/QUY%202/GAS_Baocaotaichinh_6T_2025_Soatxet_Hopnhat.pdf [Đã truy cập 2025].

[9] Uỷ ban Chứng khoán Nhà nước, "Chuẩn phân ngành GICS", Available: https://www.ssc.gov.vn/webcenter/portal/ubck/pages_r/l/chitit?dDocName=APPSSCGOVVN162099773 [Đã truy cập 2025].

Tiếng anh

[10] Alex G. Kim, Maximilian Muhn, Valeri V. Nikolaev, "Financial Statement Analysis with Large Language Models", *arXiv preprint arXiv:2407.17866*, 2024.

[11] AWS, "What is LLM (Large Language Model)?",
Available: https://aws.amazon.com/what-is/large-language-model/?nc1=h_ls [Đã truy cập 2025].

[12] AWS, "What is OCR (Optical Character Recognition)?",
Available: https://aws.amazon.com/what-is/ocr/ [Đã truy cập 2025].

[13] Christian Sarmiento, Eitel J. M. Lauría, "Investigating Flavors of RAG for Applications
in College Chatbots", *Proceedings of the 17th International Conference on Computer Supported Education – CSEDU 2025*, 2025, pp.421–428.

[14] Emmanuel Ameisen / Parlance Labs, "Fine-Tuning is Dead: Long Live Fine-Tuning", Blog / Talk, 2024, Available: https://parlance-labs.com/education/fine_tuning/emmanuel.html [Đã truy cập 2025].

[15] Hongyang Yang, Xiao-Yang Liu, Christina Dan Wang, "FinGPT: Open-Source Financial Large Language Models", *arXiv preprint arXiv:2306.06031*, 2023.

[16] LlamaIndex, "Welcome to LlamaCloud",
Available: https://developers.llamaindex.ai/python/cloud/ [Đã truy cập 2025].

[17] Neo4j, "What Is a Knowledge Graph?", Available: https://neo4j.com/blog/knowledge-graph/what-is-knowledge-graph/ [Đã truy cập 2025].

[18] Paolo Pedinotti, Peter Baumann, Nathan Jessurun, Leslie Barrett, Enrico Santus, "Prompting the Market? A Large-Scale Meta-Analysis of GenAI in Finance NLP (2022–2025)", *CoRR arXiv:2509.09544*, 2025.

[19] pmdartus, "How LLMs Generate Text for the Rest of Us", Published on Jun 20 2025, Available: https://pm.dartus.fr/posts/2025/how-llm-generate-text/ [Đã truy cập 2025].

[20] Prompt Engineering Guide, "Retrieval Augmented Generation (RAG) for LLMs", Available: https://www.promptingguide.ai/research/rag [Đã truy cập 2025].

[21] Shahul Es, Jithin James, Luis Espinosa-Anke, Steven Schockaert, “RAGAS: Automated Evaluation of Retrieval Augmented Generation”, *arXiv preprint arXiv:2309.15217*, 2023.

[22] Vaswani Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, “Attention Is All You Need”, *Proceedings of NeurIPS 2017*, 2017.

[23] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, Liang Zhao, “Graph Retrieval-Augmented Generation”, *Findings of the ACL: NAACL 2025*, 2025, pp. 4145–4157.