ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Phạm Tùng Thủy

NỀN TẢNG XÂY DỰNG TƯ VẤN NGHIỆP VỤ ẢO

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công nghệ thông tin

HÀ NỘI - 2024

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Phạm Tùng Thủy

NỀN TẢNG XÂY DỰNG TƯ VẤN NGHIỆP VỤ ẢO

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS. Nguyễn Văn Sơn

HÀ NỘI - 2024

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Pham Tung Thuy

VIRTUAL CONSULTING BUILDER PLATFORM

BACHELOR'S THESIS

Major: Information technology

Supervisor: Dr. Nguyen Van Son

HANOI - 2024

Lời cảm ơn

Lời đầu tiên, tôi xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc tới thầy TS. Nguyễn Văn Sơn vì đã hướng dẫn và hỗ trợ tôi về kiến thức chuyên môn vô cùng tận tình trong quá trình nghiên cứu và thực hiện khóa luận này.

Tôi cũng xin gửi lời cảm ơn đến các thầy cô, các anh chị, các bạn trong phòng thí nghiệm của bộ môn Công nghệ phần mềm đã hỗ trợ tôi rất nhiều về kiến thức chuyên môn trong quá trình làm khóa luận.

Cuối cùng, tôi gửi lời cảm ơn chân thành tới các thành tới các thầy, cô hiện đang công tác và giảng dạy tại trường Đại học Công Nghệ - Đại học Quốc gia Hà Nội đã tạo mọi điều kiện thuận lợi cho tôi học tập và nghiên cứu.

Lời cam đoan

Tôi là Phạm Tùng Thủy, sinh viên lớp QH-2021-I/CQ-I-IT3 khóa K66 theo học ngành Công nghệ thông tin tại trường Đại học Công Nghệ - Đại học Quốc gia Hà Nội. Tôi xin cam đoan khóa luận "*Nền tảng xây dựng tư vấn nghiệp vụ ảo*" là công trình nghiên cứu do bản thân tôi thực hiện. Các nội dung nghiên cứu, kết quả trong báo cáo là xác thực.

Các thông tin sử dụng trong báo cáo là có cơ sở và không có nội dung nào sao chép từ các tài liệu mà không ghi rõ trích dẫn tham khảo. Tôi xin chịu trách nhiệm về lời cam đoan này.

Hà Nội, ngày 26 tháng 9 năm 2024

Sinh viên

Phạm Tùng Thủy

Tóm tắt

Dịch vụ tư vấn nghiệp vụ truyền thống đang đối mặt với nhiều thách thức lớn: chi phí cao, khó tiếp cận và chất lượng không đồng đều, đặc biệt đối với các doanh nghiệp vừa và nhỏ. Các nền tảng tư vấn ảo dựa trên mô hình ngôn ngữ lớn (LLMs) và công nghệ kết hợp truy xuất dữ liệu (RAG) đã được phát triển để giải quyết những vấn đề này, giúp tự động hóa tư vấn, mở rộng khả năng tiếp cận và giảm chi phí. Tuy nhiên, các giải pháp hiện tại như Langchain, Ragflow và LlamaIndex vẫn còn hạn chế. Cụ thể, chúng gặp khó khăn trong xử lý các tình huống suy luận phức tạp, như tích hợp nhiều nguồn dữ liệu hoặc phân tích trường hợp ngoại lệ, và chưa cá nhân hóa tốt câu trả lời để phù hợp với từng ngữ cảnh cụ thể. Những yếu điểm này làm giảm giá trị thực tiễn của các nền tảng này trong các lĩnh vực đòi hỏi tính chính xác cao như pháp lý, kinh tế, giáo dục, v.v.

Khóa luận này giới thiệu Questin, một nền tảng tư vấn nghiệp vụ ảo tích hợp mô hình LLM chuyên biệt cho Tiếng Việt và công nghệ RAG, với mục tiêu cung cấp giải pháp tư vấn hiệu quả, phù hợp ngữ cảnh và chi phí hợp lý. Điểm nổi bật của Questin là khả năng xử lý các tình huống phức tạp thông qua việc giảm thiểu sự mơ hồ trong câu hỏi của người dùng, truy xuất và tổng hợp dữ liệu chính xác hơn, và linh hoạt tùy chỉnh để phù hợp với từng nhu cầu cụ thể. Nền tảng này cũng giúp nâng cao khả năng sử dụng các mô hình AI để xử lý cả dữ liệu văn bản và số liệu phức tạp, đáp ứng tốt các bài toán chuyên sâu. Kết quả thực nghiệm trên ba lĩnh vực pháp lý, kinh tế và tư vấn tuyển sinh cho thấy Questin có hiệu suất cải thiện hơn so với các giải pháp hiện tại. Cụ thể, trong lĩnh vực pháp lý, Questin đạt *Context Precision* 0.96, cao nhất trong các nền tảng, với *Context Recall* 0.79, vượt trội so với các giải pháp khác. Trong lĩnh vực kinh tế, Questin dẫn đầu với *Context Precision* 0.78 và *Context Recall* 0.79, vượt qua Langchain và Ragflow. Trong tư vấn tuyển sinh, Questin cũng đạt kết quả xuất sắc với 49/55 câu trả lời đúng và *Context Precision* 0.85, vượt qua các giải pháp còn lại.

Từ khóa: *tư vấn ảo*, *mô hình ngôn ngữ lớn*, *truy xuất kết hợp (RAG)*, *tính chính xác*, *tùy chỉnh ngữ cảnh*, *phân tích dữ liệu phức tạp*.

Abstract

Traditional business consulting services face significant challenges: high costs, limited accessibility, and inconsistent quality, especially for small and medium enterprises (SMEs). Virtual consulting platforms leveraging large language models (LLMs) and retrieval-augmented generation (RAG) technology have been developed to address these issues, enabling automated advice, broader accessibility, and cost reduction. However, current solutions like Langchain, Ragflow, and LlamaIndex still have limitations. Specifically, they struggle with handling complex reasoning tasks, such as integrating multiple data sources or analyzing edge cases, and lack personalized responses tailored to specific contexts. These shortcomings reduce the practical value of such platforms in high-precision domains like legal, economic, and educational consulting.

This research introduces QUESTIN, a virtual business consulting platform integrating a Vietnamese-specialized LLM and RAG technology, aimed at providing efficient, context-aware, and cost-effective advisory solutions. QUESTIN excels in handling complex scenarios by reducing ambiguity in user queries, retrieving and synthesizing data more accurately, and offering flexible customization for diverse needs. The platform also enhances the use of AI models for processing both textual and complex numerical data, catering effectively to specialized tasks. Experimental results across three domains: legal, economic, and admissions consulting-demonstrate QUESTIN's superior performance compared to existing solutions. Specifically, in the legal field, QUESTIN achieved a *Context Precision* of 0.96, the highest among the platforms, with a *Context Recall* of 0.79, outperforming other solutions. In the economic field, QUESTIN leads with a *Context Precision* of 0.78 and a *Context Recall* of 0.79, surpassing Langchain and Ragflow. In educational consulting, QUESTIN also delivered excellent results with 49/55 correct answers and a *Context Precision* of 0.85, exceeding the other solutions. These results demonstrate QUESTIN's superior ability to provide accurate, effective, and contextually relevant advisory solutions across different domains.

Keywords: *virtual consulting*, *large language models (LLMs)*, *retrieval-augmented generation (RAG)*, *accuracy*, *context customization*, *complex data analysis*.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh mục các từ viết tắt

STT	**Từ viết tắt**	**Cụm từ đầy đủ**	**Cụm từ tiếng Việt**
1	AI	Artificial Intelligence	Trí tuệ nhân tạo
2	LLM	Large Language Model	Mô hình ngôn ngữ lớn
3	RAG	Retrieval Augmented Generation	Tạo tăng cường truy xuất
4	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
5	NER	Named Entity Recognition	Nhận dạng thực thể có tên
6	POSTAG	Parts of Speech Tagging	Gán nhãn từ loại
7	TF-IDF	Term Frequency - Inverse Document Frequency	Tần suất từ - nghịch đảo tài liệu
8	GMM	Gaussian Mixture Model	Mô hình hỗn hợp Gaussian

Chương 1

Đặt vấn đề

Trong bối cảnh công nghệ đang ngày càng phát triển mạnh mẽ, trí tuệ nhân tạo (AI) đã và đang tạo ra những thay đổi đáng kể trong hầu hết các lĩnh vực đời sống, đặc biệt là trong việc hỗ trợ nghiệp vụ và ra quyết định. Trên thế giới, các nền tảng như ChatGPT, Gemini, v.v. đã chứng minh khả năng vượt trội trong việc tương tác và xử lý ngôn ngữ tự nhiên, giúp người dùng tìm kiếm thông tin nhanh chóng, tối ưu hóa quy trình làm việc, và hỗ trợ đưa ra các quyết định phù hợp.

Tuy nhiên, các nền tảng tư vấn ảo hiện nay phần lớn được thiết kế để giải quyết các vấn đề tổng quát, thiếu tính chuyên sâu cho từng lĩnh vực cụ thể. Điều này dẫn đến những hạn chế đáng kể khi triển khai các ứng dụng vào các ngành nghề đặc thù, đặc biệt trong ngữ cảnh ngôn ngữ Tiếng Việt. Tiếng Việt có đặc điểm ngữ nghĩa phong phú, cú pháp phức tạp, và yêu cầu xử lý ngôn ngữ chính xác để đảm bảo thông tin được diễn đạt một cách phù hợp với ngữ cảnh chuyên môn. Bên cạnh đó, nhiều tổ chức/doanh nghiệp có nhu cầu sử dụng hệ thống tư vấn tự động hóa nhưng lại không có giải pháp phù hợp để tùy chỉnh dữ liệu hoặc thuật toán đáp ứng nhu cầu thực tiễn.

Từ thực tế này, dự án "*Nền tảng xây dựng tư vấn nghiệp vụ ảo*" được phát triển với mục tiêu xây dựng một hệ thống AI chuyên sâu, đáp ứng nhu cầu tư vấn nghiệp vụ trong từng lĩnh vực cụ thể với ngôn ngữ Tiếng Việt. Nền tảng này được thiết kế để cung cấp giải pháp tư vấn nghiệp vụ chuyên sâu sử dụng các mô hình AI tiên tiến kết hợp với dữ liệu nghiệp vụ được tùy chỉnh để giải quyết các vấn đề cụ thể của từng ngành nghề; hỗ trợ ngôn ngữ Tiếng Việt chuyên nghiệp giúp đảm bảo độ chính xác trong ngữ nghĩa và ngữ cảnh khi giao tiếp người dùng; tối ưu hóa quy trình nghiệp vụ để giảm thiểu thời gian, chi phí và công sức cho doanh nghiệp thông qua việc tự động hóa các bước xử lý thông tin và cung cấp câu trả lời.

1.1 Vấn đề về chi phí và chất lượng của các dịch vụ tư vấn nghiệp vụ

Trong thực tế, chi phí sử dụng dịch vụ tư vấn nghiệp vụ từ các chuyên gia thường rất cao, đặc biệt trong các lĩnh vực phức tạp như tài chính, quản lý nhân sự, và công nghệ thông tin. Nghiên cứu chỉ ra rằng giá trị của dịch vụ tư vấn phụ thuộc lớn vào năng lực chuyên môn của chuyên gia cũng như mối quan hệ hợp tác giữa họ và khách hàng [1]. Tuy nhiên, sự thiếu tiêu chuẩn hóa trong quy trình quản lý và tính cạnh tranh cao đã làm gia tăng chi phí, đặc biệt đối với các doanh nghiệp vừa và nhỏ [2].

Không chỉ vấn đề chi phí, khả năng tiếp cận các dịch vụ tư vấn nghiệp vụ cũng gặp nhiều hạn chế. Các doanh nghiệp ở vùng xa thường thiếu khả năng tiếp cận nguồn lực chuyên môn, trong khi các doanh nghiệp nhỏ đối mặt với khó khăn trong việc lựa chọn nhà tư vấn phù hợp và phương pháp làm việc hiệu quả [3, 4]. Thậm chí, chất lượng dịch vụ tư vấn có thể bị ảnh hưởng bởi các yếu tố như điều kiện kinh tế, nhu cầu thị trường và năng lực quản lý nội bộ [5, 6].

Hiện nay, một số nền tảng tư vấn ảo đã được phát triển để giảm thiểu chi phí và cải thiện khả năng tiếp cận. Tuy nhiên, các nền tảng này vẫn gặp nhiều hạn chế về mặt chất lượng. Các nghiên cứu cho thấy rằng việc không cập nhật dữ liệu kịp thời dẫn đến việc cung cấp thông tin tư vấn thiếu chính xác hoặc lỗi thời, gây rủi ro cho doanh nghiệp nếu áp dụng sai lệch [7]. Ngoài ra, việc thiếu hệ thống kế toán chi phí nâng cao cũng làm hạn chế hiệu quả và khả năng tối ưu hóa chi phí của các dịch vụ này [8].

Các vấn đề về chi phí cao, chất lượng không đồng đều và sự thiếu hụt trong khả năng tiếp cận dịch vụ tư vấn nghiệp vụ đã đặt ra thách thức lớn trong việc hỗ trợ sự phát triển bền vững của các doanh nghiệp. Vì vậy, việc phát triển một nền tảng tư vấn nghiệp vụ ảo với chi phí hợp lý và thông tin chất lượng cao là một nhu cầu cấp thiết. Những nền tảng này không chỉ thu hẹp khoảng cách giữa doanh nghiệp lớn và nhỏ mà còn tạo động lực thúc đẩy đổi mới và hiệu quả trong việc đưa ra các quyết định chiến lược.

1.2 Các nghiên cứu liên quan

Quá trình phát triển các nền tảng tư vấn nghiệp vụ ảo đã trải qua nhiều giai đoạn, với sự đóng góp của ba nhóm giải pháp chính: (1) Hệ thống dựa trên quy tắc (Rule-based Systems) [9], (2) Hệ thống dựa trên học máy (Machine Learning-Based Systems) [9], và (3) Các giải pháp học sâu kết hợp mô hình ngôn ngữ lớn (Deep Learning with Large Language Models - LLMs) [10]. Mỗi nhóm giải pháp đại diện cho một bước tiến trong công nghệ, đồng thời cũng bộc lộ những hạn chế đặt ra nhu cầu cho các giải pháp tiên tiến hơn.

Các hệ thống dựa trên quy tắc là một trong những phương pháp đầu tiên được áp dụng để xây dựng hệ thống tư vấn tự động. Chúng hoạt động dựa trên các tập hợp quy tắc logic được thiết kế trước để xử lý dữ liệu đầu vào và tạo ra phản hồi [11]. Một ví dụ kinh điển là ELIZA [12], hệ thống sử dụng các quy tắc mẫu và quy tắc chuyển đổi để tạo ra các phản hồi dựa trên đầu vào của người dùng. Sau đó, PARRY [13] ra đời như một bước cải tiến bằng cách bổ sung các trạng thái cảm xúc, giúp tạo ra phản hồi tự nhiên và phù hợp hơn với bối cảnh hội thoại. Tuy nhiên, những hệ thống này phụ thuộc hoàn toàn vào các quy tắc được định nghĩa sẵn, khiến chúng thiếu khả năng linh hoạt khi xử lý các tình huống chưa được lập trình trước. Việc mở rộng hoặc duy trì hệ thống cũng trở nên khó khăn khi dữ liệu đầu vào ngày càng phức tạp và đa dạng. Ngoài ra, các hệ thống này không có khả năng học hỏi và cải thiện dựa trên tương tác, dẫn đến hiệu quả tư vấn bị giới hạn.

Với sự phát triển của học máy, các hệ thống dựa trên mô hình học chuỗi sang chuỗi (seq2seq) [14] đã tạo ra một bước tiến đáng kể. Phương pháp seq2seq, thường sử dụng mạng nơ-ron hồi quy (RNN) để giải quyết các bài toán phức tạp liên quan đến dự đoán chuỗi sang chuỗi như dịch máy, chú thích hình ảnh [15], nhận diện giọng nói[16], tóm tắt văn bản [17, 18] và hỏi đáp. Các nghiên cứu như của Vinyals và cộng sự [19] đã chứng minh khả năng huấn luyện mô hình seq2seq trên tập dữ liệu lớn để cải thiện chất lượng trả lời. Một cải tiến đáng chú ý là mô hình của Hu và cộng sự [20], cho phép chatbot trả lời câu hỏi cảm xúc và thực hiện suy luận cơ bản dựa trên dữ liệu lớn. Bên cạnh đó, phương pháp học tăng cường (Reinforcement Learning) [21, 22] cũng được áp dụng rộng rãi, trong đó hệ thống được mô hình hóa dựa trên các quy trình quyết định Markov quan sát một phần (POMDPs) [21–25] trong việc cải thiện khả năng học hỏi và tối ưu hóa của các

hệ thống hội thoại. Thay vì chỉ đơn giản dựa vào các dữ liệu huấn luyện được gán nhãn, học tăng cường cho phép hệ thống tự học thông qua quá trình tương tác với môi trường và nhận phản hồi từ các hành động của nó. Cách tiếp cận này giúp chatbot học hỏi được từ các cuộc hội thoại thực tế, qua đó cải thiện khả năng đưa ra các phản hồi phù hợp và chính xác hơn trong các tình huống chưa được định nghĩa trước. Một điểm mạnh của học tăng cường là khả năng xử lý các tình huống phức tạp và động, nơi mà quy tắc hay dữ liệu huấn luyện truyền thống không đủ khả năng đáp ứng. Tuy nhiên, các hệ thống học tăng cường vẫn phụ thuộc vào chất lượng và quy mô của dữ liệu huấn luyện, đặc biệt là dữ liệu có thể phản ánh chính xác các tình huống thực tế. Mặc dù học tăng cường có thể giúp cải thiện tính linh hoạt và khả năng tự thích ứng của hệ thống, nhưng các hệ thống này vẫn gặp khó khăn khi phải đối mặt với các câu hỏi có ngữ nghĩa phức tạp hoặc những tình huống chưa được chuẩn bị trong dữ liệu huấn luyện.

Dưới sự phát triển mạnh mẽ của các mô hình ngôn ngữ lớn (LLMs) như ChatGPT, Microsoft Copilot và Gemini, cộng đồng nghiên cứu đã dần nhận thức được tiềm năng to lớn của những mô hình này trong việc xử lý ngôn ngữ tự nhiên [26]. Sự bùng nổ của các mô hình này đã dẫn đến một làn sóng đổi mới mạnh mẽ trong việc phát triển các mô hình và ứng dụng liên quan, mở ra cơ hội to lớn trong việc tự động hóa và tối ưu hóa quy trình nghiệp vụ trên quy mô toàn cầu [27, 28]. Đặc biệt, các nghiên cứu đã tập trung vào việc khai thác sức mạnh của LLMs để nâng cao chất lượng tư vấn nghiệp vụ ảo, mở ra cơ hội thay đổi mang tính cách mạng cho ngành này [29]. Các mô hình LLMs gần đây đã được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Chẳng hạn, FinGPT [30], một mô hình được phát triển đặc biệt cho ngành tài chính, đã chứng minh khả năng tự động hóa quy trình và cung cấp các phân tích dữ liệu tài chính toàn cầu. Trong lĩnh vực pháp luật, một số mô hình như LawGPT_zh [31], Lawyer LLaMA [32] và ChatLaw [33] đã được thử nghiệm để xây dựng các hệ thống tư vấn pháp lý [34, 35]. Các mô hình này đã chứng minh khả năng hỗ trợ hiệu quả trong việc ra quyết định và cung cấp giải pháp nghiệp vụ nhanh chóng. Tuy nhiên, những hệ thống này vẫn gặp phải những thách thức liên quan đến độ chính xác và khả năng cá nhân hóa trong từng tình huống cụ thể. Dù mang lại những tiến bộ đáng kể, các giải pháp này vẫn tồn tại một số hạn chế. Trước hết, việc triển khai và duy trì các mô hình ngôn ngữ lớn đòi hỏi tài nguyên tính toán và lưu trữ rất lớn, điều này là một rào cản đối với nhiều tổ chức có nguồn lực hạn chế. Thứ hai, hiệu quả của các hệ thống này phụ

thuộc rất nhiều vào chất lượng và sự sẵn có của dữ liệu đầu vào, điều này có thể ảnh hưởng đến khả năng đáp ứng yêu cầu thực tế của người dùng.

Mặc dù các phương pháp trên đã đạt được những tiến bộ đáng kể trong xây dựng hệ thống tư vấn nghiệp vụ, một vấn đề lớn vẫn chưa được giải quyết triệt để: hiện tượng hallucination. Đây là khi hệ thống tạo ra câu trả lời sai lệch hoặc không chính xác do thiếu thông tin đáng tin cậy. Để khắc phục tình trạng này, công nghệ Retrieval-Augmented Generation (RAG) [36] đã ra đời, kết hợp khả năng xử lý ngôn ngữ tự nhiên của mô hình LLM với việc truy xuất thông tin từ các nguồn bên ngoài, từ đó cải thiện độ chính xác và giảm chi phí triển khai nhờ tối ưu hóa tài nguyên. Hiện nay, các nền tảng như Langchain, LlamaIndex, và Ragflow đang hỗ trợ triển khai RAG một cách hiệu quả, giúp xây dựng quy trình RAG nhanh chóng và dễ dàng. Tuy nhiên, các hệ thống này vẫn gặp phải một số hạn chế. Đầu tiên, khi câu hỏi của người dùng mơ hồ hoặc thiếu thông tin rõ ràng, các hệ thống gặp khó khăn trong việc làm rõ ngữ cảnh để đưa ra câu trả lời chính xác. Thứ hai, khi dữ liệu yêu cầu xử lý phức tạp như bảng số liệu, các nền tảng hiện tại chưa thực sự tối ưu. Thứ ba, khả năng tùy chỉnh hệ thống để phù hợp với các yêu cầu cụ thể và bối cảnh riêng biệt còn hạn chế, đặc biệt trong các lĩnh vực đòi hỏi độ chính xác cao như pháp lý, tài chính và giáo dục [37]. Những vấn đề này đòi hỏi cần có các giải pháp mới, không chỉ giảm thiểu hallucination mà còn nâng cao khả năng xử lý ngữ cảnh phức tạp và cải thiện tính linh hoạt của hệ thống.

1.3 Mục tiêu và đóng góp của nghiên cứu

Nhận thức được sự cần thiết của một nền tảng tư vấn nghiệp vụ ảo hỗ trợ đa lĩnh vực và tối ưu hóa hiệu suất bằng Tiếng Việt, khóa luận này giới thiệu phương pháp Questin, một giải pháp dựa trên sự kết hợp giữa mô hình ngôn ngữ lớn (Large Language Model - LLM) chuyên biệt cho Tiếng Việt, gọi là Vistral, và công nghệ Retrieval-Augmented Generation (RAG) [36]. Bằng cách tận dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) tiên tiến, Questin đặt mục tiêu giải quyết các thách thức về chi phí, chất lượng và khả năng tiếp cận trong tư vấn nghiệp vụ.

Ý tưởng chính của phương pháp Questin là kết hợp sức mạnh tổng quát và khả năng hiểu ngữ nghĩa của các mô hình ngôn ngữ lớn với dữ liệu nghiệp vụ chuyên sâu, được truy xuất theo thời gian thực. Công nghệ RAG cho phép Questin trích

xuất và sử dụng dữ liệu liên quan trực tiếp từ các nguồn thông tin tin cậy, sau đó kết hợp với khả năng sinh ngôn ngữ của mô hình để tạo ra các tư vấn mang tính chuyên biệt, chính xác và phù hợp với bối cảnh người dùng. Mục tiêu cao nhất của nghiên cứu là xây dựng một nền tảng có thể tối ưu hóa khả năng cá nhân hóa và chuyên sâu như cung cấp thông tin phù hợp với từng lĩnh vực nghiệp vụ, từng tình huống cụ thể; hỗ trợ ngôn ngữ Tiếng Việt chuyên nghiệp giúp đảm bảo độ chính xác và tự nhiên trong giao tiếp bằng Tiếng Việt, đáp ứng tốt về mặt ngữ nghĩa và cú pháp; tăng khả năng tiếp cận dịch vụ tư vấn giúp giảm thiểu rào cản chi phí.

Để đánh giá hiệu quả của QUESTIN, khóa luận xây dựng bộ câu hỏi chuẩn hóa bao gồm 156 câu hỏi, tập trung vào ba lĩnh vực chính: luật pháp, kinh tế, và tư vấn tuyển sinh. Các câu hỏi được phân loại theo ba mức độ: truy vấn trực tiếp, tích hợp dữ liệu phức tạp, và suy luận. Điều này đảm bảo đánh giá toàn diện hiệu suất của hệ thống trong các ngữ cảnh và tình huống khác nhau. Kết quả thử nghiệm cho thấy câu trả lời của mô hình Vistral cải thiện đáng kể sau tinh chỉnh, với điểm Bleu tăng từ 0.04 lên 0.13 và Semantic Similarity tăng từ 0.71 lên 0.78. So với Langchain, LlamaIndex, và Ragflow, QUESTIN đạt hiệu quả cao nhất về độ phù hợp, thường dẫn đầu hoặc đạt thứ hạng cao ở cả bốn tiêu chí: Context Precision, Context Recall, Faithfulness, và Answer Relevancy trên cả ba lĩnh vực. Đặc biệt, QUESTIN thể hiện khả năng xử lý hiệu quả với các câu hỏi truy vấn và tích hợp dữ liệu, đạt tỷ lệ trả lời thành công ấn tượng so với các giải pháp khác.

Khóa luận này đóng góp một số giá trị chính trong lĩnh vực nghiên cứu, bao gồm:

- **Xây dựng nền tảng tư vấn viên ảo tự động hóa**: Khóa luận này đóng góp vào việc xây dựng một nền tảng tự động hóa quy trình phát triển tư vấn viên ảo, tận dụng công nghệ Retrieval-Augmented Generation (RAG) kết hợp với các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP).
- **Phát triển bộ benchmark tiếng Việt**: Cung cấp một bộ câu hỏi chuẩn hóa đa lĩnh vực với hơn 150 câu hỏi, hỗ trợ đánh giá hiệu quả và toàn diện các hệ thống tư vấn sử dụng tiếng Việt.
- **Cải thiện phương pháp RAG**: Đề xuất các cải tiến nhằm giảm thiểu sự mơ hồ trong truy vấn của người dùng, nâng cao độ chính xác trong truy xuất dữ liệu và tăng hiệu quả xử lý câu trả lời.
- **Hướng tới tính ứng dụng đa ngành**: Khóa luận mở ra tiềm năng áp dụng

giải pháp trong các lĩnh vực như luật pháp, kinh tế và giáo dục, với sự hỗ trợ hiệu quả cho tiếng Việt.

Phần còn lại của báo cáo sẽ đi sâu vào các chủ đề sau:

- Phần 2 sẽ cung cấp những kiến thức cơ bản về mô hình ngôn ngữ lớn, công nghệ RAG và những công nghệ khác được áp dụng trong khóa luận này.
- Phần 3 sẽ giới thiệu ý tưởng và cách thức hoạt động của phương pháp Questin, bao gồm quy trình tích hợp RAG và mô hình Vistral để tạo ra các tư vấn chất lượng cao.
- Phần 4 mô tả chi tiết quá trình triển khai, bao gồm thiết kế hệ thống, các thuật toán quan trọng, và các thí nghiệm đánh giá hiệu quả của phương pháp.
- Cuối cùng, chương này sẽ tóm tắt những kết quả chính, đánh giá những hạn chế còn tồn tại, và đề xuất hướng phát triển trong tương lai để mở rộng và cải thiện giải pháp Questin.

Chương 2

Kiến thức cơ sở

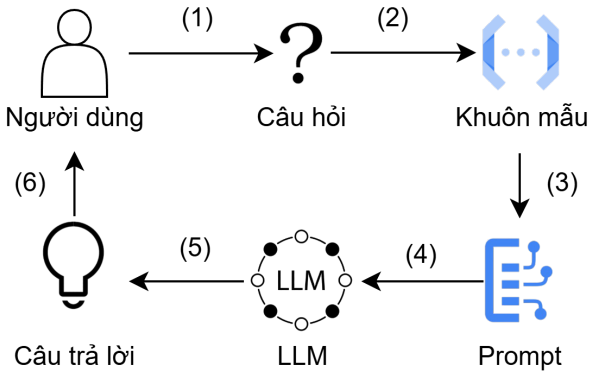
Trong chương này, chúng ta sẽ tìm hiểu những kiến thức nền tảng cần thiết để xây dựng và phát triển "*Nền tảng xây dựng tư vấn nghiệp vụ ảo*". Hiểu rõ các yếu tố cơ bản này sẽ giúp đảm bảo hệ thống đạt được hiệu quả và độ chính xác cao trong quá trình tư vấn và hỗ trợ nghiệp vụ.

2.1 Mô hình ngôn ngữ lớn

Mô hình ngôn ngữ lớn (Large Language Model - LLM) [38, 39] là một công nghệ tiên tiến trong lĩnh vực trí tuệ nhân tạo, được thiết kế để xử lý, phân tích và tạo ra ngôn ngữ tự nhiên với độ chính xác và tinh vi cao. Dựa trên các kiến trúc học sâu, đặc biệt là Transformer, LLM được huấn luyện trên khối lượng dữ liệu văn bản khổng lồ, cho phép mô hình hiểu được không chỉ ngữ nghĩa mà còn cả ngữ cảnh, cấu trúc và mối quan hệ giữa các từ, cụm từ trong ngôn ngữ [40]. Với khả năng này, LLM có thể thực hiện đa dạng các tác vụ ngôn ngữ như trả lời câu hỏi, tóm tắt nội dung, dịch thuật, phân loại văn bản, phân tích ngữ nghĩa, và thậm chí tạo ra nội dung mới mang tính sáng tạo.

Một điểm đặc trưng của LLM chính là quy mô tham số cực lớn, thường lên tới hàng tỷ hoặc thậm chí hàng nghìn tỷ tham số. Quy mô này cho phép mô hình nắm bắt sâu sắc mối quan hệ ngữ nghĩa và ngữ cảnh trong ngôn ngữ, từ đó cung cấp các phản hồi logic, tự nhiên và có tính mạch lạc cao. Đồng thời, LLM được thiết kế để có tính linh hoạt vượt trội, cho phép tùy chỉnh theo các lĩnh vực chuyên biệt, từ y tế, giáo dục, tài chính đến công nghệ, giúp tối ưu hóa hiệu suất trong từng ứng dụng cụ thể.

Luồng hoạt động cơ bản của LLM được minh họa trong Hình 2.1. Đầu tiên, người dùng đưa ra một truy vấn hoặc yêu cầu bằng ngôn ngữ tự nhiên (1). Truy vấn này sau đó được chuyển đổi thành một dạng chuẩn hóa thông qua các khuôn mẫu hoặc cấu trúc định sẵn (2), nhằm tạo ra một định dạng đầu vào đặc biệt, gọi là prompt [41](3). Prompt này đóng vai trò như một câu hỏi rõ ràng và được mô hình



Hình 2.1: Luồng hoạt động cơ bản của LLM

sử dụng để phân tích và xử lý [42]. Khi nhận được prompt (4), LLM dựa vào lượng dữ liệu khổng lồ đã được huấn luyện và sử dụng các cơ chế như attention để hiểu ngữ cảnh, mối quan hệ giữa các từ, ý nghĩa tổng thể của truy vấn và sinh ra phản hồi phù hợp (5). Câu trả lời được mô hình tạo ra sau đó sẽ được gửi trả lại người dùng dưới dạng văn bản tự nhiên (6), với cách diễn đạt mạch lạc, dễ hiểu và mang tính logic cao. Quy trình này đảm bảo rằng LLM có thể đáp ứng được nhiều nhu cầu khác nhau từ người dùng, giúp nó trở thành một công cụ hữu ích và linh hoạt.

Mặc dù có khả năng mạnh mẽ, LLM vẫn tồn tại một số hạn chế đáng chú ý. Một trong những vấn đề lớn nhất là hiện tượng "ảo giác thông tin" (hallucination) [43–45], khi mô hình tạo ra những phản hồi không chính xác hoặc thông tin không tồn tại trong thực tế. Điều này thường xảy ra do mô hình quá phụ thuộc vào dữ liệu huấn luyện mà không có cơ chế xác minh tính chính xác của thông tin. Ngoài ra, do được huấn luyện trên dữ liệu tổng quát, LLM đôi khi thiếu chiều sâu chuyên môn trong các lĩnh vực đặc thù, dẫn đến các câu trả lời tuy đúng về mặt ngữ pháp nhưng lại không phù hợp với ngữ cảnh thực tế. Để khắc phục hạn chế này, công nghệ Retrieval-Augmented Generation (RAG) được giới thiệu như một giải pháp hiệu quả. Thay vì chỉ dựa vào kiến thức có sẵn trong mô hình, RAG kết hợp khả năng sinh ngôn ngữ của LLM với một hệ thống truy xuất thông tin mạnh mẽ. Khi nhận được truy vấn, hệ thống RAG sẽ tìm kiếm thông tin từ các cơ sở dữ liệu hoặc tài liệu đáng tin cậy, sau đó tích hợp dữ liệu này vào phản hồi do LLM tạo ra. Sự kết hợp này không chỉ làm tăng độ chính xác của câu trả lời mà còn giảm thiểu hiện tượng hallucination. RAG đã được ứng dụng thành công trong các hệ thống chatbot, trợ lý ảo và các nền tảng hỗ trợ nghiệp vụ.

2.2 Tạo tăng cường truy xuất (Retrieval Augmented Generation)

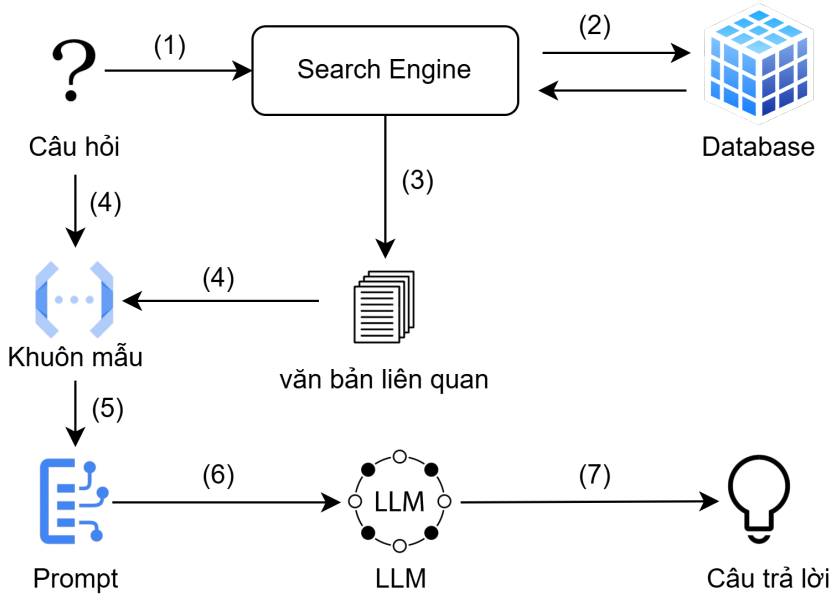
2.2.1 Giới thiệu

Tạo tăng cường truy xuất (Retrieval-Augmented Generation - RAG) [36] là một phương pháp kết hợp giữa mô hình ngôn ngữ lớn (LLM) và các nguồn tri thức bên ngoài để tạo ra các phản hồi chất lượng cao hơn. RAG được thiết kế nhằm khắc phục hạn chế của LLM trong việc cung cấp thông tin chính xác, đặc biệt khi mô hình đối mặt với những câu hỏi ngoài phạm vi dữ liệu huấn luyện hoặc đòi hỏi thông tin cập nhật. LLM thường hoạt động dựa trên dữ liệu tĩnh, dễ dẫn đến lỗi khi dữ liệu không đủ chính xác hoặc lỗi thời. RAG giải quyết vấn đề này bằng cách cho phép LLM truy xuất và tham khảo các nguồn tri thức động, giúp cải thiện đáng kể chất lượng và độ tin cậy của các phản hồi [46]. Nhờ tích hợp các cơ chế truy xuất thông tin, RAG mang lại nhiều lợi ích như giảm chi phí tùy chỉnh mô hình, cung cấp thông tin cập nhật theo thời gian thực và tăng cường tính minh bạch bằng cách cho phép người dùng kiểm tra các nguồn dữ liệu. Phương pháp này cũng giúp nhà phát triển kiểm soát tốt hơn quy trình tạo nội dung, đặc biệt trong các ứng dụng cần độ chính xác cao.

2.2.2 Luồng hoạt động

Luồng hoạt động của Retrieval-Augmented Generation (RAG) được thiết kế nhằm kết hợp hiệu quả các thành phần truy xuất thông tin và khả năng xử lý ngôn ngữ tự nhiên (NLP) của mô hình ngôn ngữ lớn (LLM). Quá trình này đảm bảo các phản hồi được tạo ra có độ chính xác cao bằng cách tận dụng các nguồn tri thức bên ngoài, giúp bổ sung ngữ cảnh cần thiết cho LLM. Hình 2.2 minh họa chi tiết quy trình hoạt động của RAG.

Bước đầu tiên của hệ thống là nhận truy vấn hoặc câu hỏi từ người dùng (1). Truy vấn này được nhập dưới dạng văn bản tự nhiên và có thể bao gồm các yêu cầu cụ thể hoặc tổng quát. Đây là đầu vào khởi động toàn bộ hệ thống. Hệ thống tiếp theo sử dụng các công cụ tìm kiếm như Elasticsearch hoặc FAISS để truy xuất thông tin từ cơ sở dữ liệu [47] (2). Các thuật toán như TF-IDF, BM25 hoặc tìm kiếm dựa trên vector embedding được triển khai để tìm kiếm các tài liệu có liên



Hình 2.2: Luồng hoạt động cơ bản của RAG

quan cao nhất. Dữ liệu lưu trữ trong cơ sở dữ liệu thường bao gồm các bộ sưu tập văn bản hoặc các kho tri thức được tổ chức chặt chẽ. Kết quả truy xuất là một danh sách các tài liệu được xếp hạng theo mức độ liên quan đến truy vấn ban đầu. Các tài liệu này được sàng lọc để giữ lại nội dung phù hợp nhất, đồng thời đảm bảo tính chính xác và giá trị thông tin (3).

Câu hỏi của người dùng và các tài liệu liên quan được chuẩn hóa thông qua một quy trình định dạng có cấu trúc (4). Quá trình này sử dụng các mẫu cấu trúc định sẵn để tạo ra một "prompt" (5) tích hợp, bao gồm cả truy vấn và các đoạn văn bản liên quan. Prompt đóng vai trò cung cấp ngữ cảnh đầy đủ cho LLM và được thiết kế nhằm tối ưu hóa quá trình xử lý thông tin, giúp giảm thiểu nguy cơ sinh ra các phản hồi không chính xác hoặc không phù hợp (hiện tượng hallucination).

Sau khi prompt được xây dựng, nó được chuyển vào mô hình ngôn ngữ lớn (6). LLM phân tích prompt và sử dụng cả kiến thức đã được huấn luyện sẵn cùng với thông tin bổ sung từ các tài liệu truy xuất để tạo ra câu trả lời. Phản hồi được xây dựng dựa trên sự tích hợp giữa ngữ cảnh hiện tại và các dữ liệu liên quan, giúp nâng cao độ chính xác và độ tin cậy của kết quả.

Kết quả cuối cùng được gửi lại cho người dùng (7). Phản hồi này có thể bao gồm nội dung trả lời chi tiết và các trích dẫn từ nguồn dữ liệu được sử dụng, cho phép người dùng kiểm chứng và hiểu rõ quá trình hình thành kết quả.

Luồng hoạt động này của RAG giúp khai thác hiệu quả các tài nguyên dữ liệu bên ngoài để bổ sung cho khả năng của LLM. Hệ thống được thiết kế nhằm giải quyết các hạn chế của LLM truyền thống, bao gồm các vấn đề về độ chính xác, thông tin lỗi thời, và khả năng xử lý các câu hỏi cụ thể. RAG thể hiện sự phù hợp cao đối với các ứng dụng yêu cầu thông tin chuyên sâu, có cấu trúc, và cập nhật liên tục, góp phần mở rộng khả năng ứng dụng của trí tuệ nhân tạo trong nhiều lĩnh vực.

2.2.3 Hạn chế

Retrieval Augmented Generation (RAG) mang lại nhiều lợi ích trong việc cải thiện độ chính xác và chất lượng của phản hồi từ mô hình ngôn ngữ lớn, nhưng cũng tồn tại một số hạn chế đáng chú ý. Một trong những khó khăn chính là khả năng xử lý các câu hỏi mơ hồ hoặc không rõ ràng. Khi truy vấn của người dùng thiếu ngữ cảnh hoặc được biểu đạt một cách không cụ thể, hệ thống truy xuất có thể lấy về các tài liệu không thực sự liên quan hoặc hữu ích, dẫn đến prompt kém chất lượng và phản hồi không chính xác từ LLM.

RAG cũng gặp khó khăn trong việc xử lý các dữ liệu phức tạp như bảng số liệu, biểu đồ, hoặc các tệp PDF chứa nội dung phi cấu trúc hoặc định dạng không tiêu chuẩn. Mặc dù các thuật toán tìm kiếm hiện đại có thể truy xuất các đoạn văn bản liên quan, chúng thường không đủ khả năng để trích xuất và diễn giải thông tin chi tiết từ các loại dữ liệu này. Điều này ảnh hưởng đến khả năng của LLM trong việc tạo ra câu trả lời có ý nghĩa và chính xác từ các nguồn dữ liệu đa dạng.

Hơn nữa, độ chính xác của mô-đun truy xuất vẫn còn là một thách thức. Các thuật toán tìm kiếm, dù tiên tiến, không phải lúc nào cũng xếp hạng đúng mức độ liên quan của tài liệu hoặc bỏ qua các nguồn thông tin quan trọng [48]. Điều này đặc biệt rõ ràng khi cơ sở dữ liệu lớn và không đồng nhất, hoặc khi dữ liệu bị thiếu hụt về chất lượng và tính cập nhật. Sự không chính xác trong bước này có thể làm giảm đáng kể hiệu quả tổng thể của hệ thống RAG [49].

Mặc dù vậy, các nỗ lực cải tiến, chẳng hạn như cải thiện kỹ thuật truy xuất để xử lý tốt hơn các dạng dữ liệu không chuẩn và phát triển các mô hình ngôn ngữ chuyên biệt hơn, hứa hẹn sẽ giải quyết được các hạn chế này. Những tiến bộ như vậy sẽ làm tăng khả năng ứng dụng và tính hiệu quả của RAG trong tương lai.

Hình 2.3: Nhận dạng thực thể

2.3 Xử lý ngôn ngữ tự nhiên (Natural Language Processing)

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là một lĩnh vực quan trọng trong việc xây dựng các hệ thống chatbot, đặc biệt là những hệ thống xử lý lượng lớn dữ liệu văn bản. Kỹ thuật này giúp hệ thống có khả năng hiểu, phân tích và truy xuất thông tin một cách hiệu quả, từ đó cung cấp các phản hồi phù hợp và chính xác. NLP bao gồm nhiều thuật toán quan trọng như Nhận dạng thực thể (Named Entity Recognition - NER), Gán nhãn từ loại (Parts of Speech Tagging - POSTAG), và Tính tần suất nghịch đảo tài liệu (Term Frequency-Inverse Document Frequency - TF-IDF).

2.3.1 Nhận dạng thực thể (Named Entity Recommendation)

Nhận dạng thực thể (NER) là một tác vụ quan trọng trong xử lý ngôn ngữ tự nhiên, được thiết kế để xác định và phân loại các thực thể được đề cập trong văn bản vào các nhóm định nghĩa trước như tên người, tổ chức, địa điểm, thời gian, số lượng, và nhiều loại khác [50]. Ví dụ, như trong Hình 2.3, câu "Anh Lâm làm việc ở tập đoàn WeWork ở thành phố Hà Nội" sẽ được hệ thống NER phân tích như sau: "Anh Lâm" là một thực thể chỉ người (PERSON), "WeWork" là một tổ chức (ORGANIZATION), và "thành phố Hà Nội" là một địa điểm (LOCATION). Tùy thuộc vào ngữ cảnh và loại thực thể, hệ thống sẽ ưu tiên các từ hoặc cụm từ có tầm quan trọng cao hơn trong việc xử lý thông tin.

NER không chỉ nâng cao hiệu suất trong các ứng dụng như trả lời câu hỏi, truy xuất thông tin, hay dịch máy, mà còn hỗ trợ các tác vụ NLP phức tạp như phân tích cú pháp và gán nhãn từ loại. Quá trình nhận dạng thực thể bao gồm hai giai đoạn chính: xác định các thực thể trong văn bản và phân loại chúng vào các nhóm phù hợp. Các phương pháp phổ biến để xây dựng hệ thống NER bao gồm:

- **Phương pháp dựa trên từ điển (Lexicon Based Method)**: Phương pháp này sử dụng một từ điển chứa danh sách các từ hoặc cụm từ được định nghĩa trước. Quá trình thực hiện dựa trên việc kiểm tra xem những từ này có xuất

hiện trong văn bản hay không. Tuy nhiên, phương pháp này ít được sử dụng vì đòi hỏi phải liên tục cập nhật và quản lý từ điển để đảm bảo tính chính xác và hiệu quả.

- **Phương pháp dựa trên quy tắc (Rule Based Method)**: Phương pháp này sử dụng các quy tắc được thiết kế thủ công, dựa trên mẫu ngữ pháp hoặc ngữ cảnh trong văn bản. Các quy tắc có thể tập trung vào cấu trúc từ (mẫu hình) hoặc bối cảnh xung quanh từ cần phân tích. Việc kết hợp cả hai loại quy tắc này giúp nâng cao độ chính xác của hệ thống. Tuy nhiên, tính cứng nhắc của các quy tắc đôi khi không đáp ứng được những tình huống phức tạp hoặc ngữ cảnh mới.

- **Phương pháp dựa trên học máy (Machine Learning-Based Method)**: Phương pháp này bao gồm hai phương pháp con như:

  - **Phân loại đa lớp (Multi-Class Classification)**: Một cách tiếp cận là huấn luyện mô hình học máy để phân loại đa lớp, nhưng phương pháp này yêu cầu rất nhiều dữ liệu được gán nhãn. Ngoài việc gán nhãn, mô hình còn cần hiểu rõ ngữ cảnh để xử lý các câu có tính mơ hồ. Đây là một nhiệm vụ phức tạp đối với các thuật toán học máy đơn giản.

  - **Trường ngẫu nhiên có điều kiện (Conditional Random Field - CRF)**: Trường ngẫu nhiên có điều kiện là một mô hình xác suất được áp dụng rộng rãi trong xử lý ngôn ngữ tự nhiên. CRF có khả năng nắm bắt ngữ cảnh của câu một cách sâu sắc, bằng cách sử dụng dữ liệu đầu vào $X = x_1, x_2, x_3, ..., x_T$ và xác suất $p(y \mid \mathbf{x}) = \frac{1}{z(\vec{x})} \prod_{t=1}^{T} \exp\left\{\sum_{k=1}^{K} \omega_k f_k(y_t, y_{t-1}, \vec{x}_t)\right\}$.

- **Phương pháp dựa trên học sâu (Deep Learning Based Method)**: Hệ thống NER dựa trên học sâu có độ chính xác cao hơn đáng kể so với các phương pháp trước đó nhờ khả năng xử lý và tổ hợp từ ngữ. Điều này đạt được nhờ việc sử dụng phương pháp nhúng từ (word embedding), giúp mô hình hiểu được mối quan hệ ngữ nghĩa và cú pháp giữa các từ. Học sâu còn có thể tự động phân tích các từ mang tính chuyên ngành và các từ phổ biến ở cấp độ cao, giúp mở rộng khả năng ứng dụng vào nhiều tác vụ khác nhau. Nhờ tính tự động hóa cao, các hệ thống học sâu không chỉ tiết kiệm thời gian mà còn hỗ trợ nhà nghiên cứu tập trung vào những nhiệm vụ quan trọng hơn.

2.3.2 Gán nhãn từ loại (Parts of Speech Tagging)

Gán nhãn từ loại (Parts of Speech Tagging - POSTAG) [51] là một hoạt động trong xử lý ngôn ngữ tự nhiên (NLP), nơi mỗi từ trong tài liệu được gán một từ loại cụ thể (trạng từ, tính từ, động từ, v.v.) hoặc một doanh mục ngữ pháp. Quá trình này thêm một lớp thông tin cú pháp và ngữ nghĩa vào các từ, giúp dễ dàng hơn trong việc hiểu cấu trúc và ý nghĩa của câu. Trong các ứng dụng NLP đặc biệt là các chatbot, gán nhãn từ loại rất hữu ích trong việc truy xuất thông tin.

Nhờ việc gán nhãn chính xác, hệ thống có thể dễ dàng phân tích cấu trúc câu, nhận diện mối quan hệ giữa các từ và tối ưu hóa quá trình truy xuất thông tin. Bảng 2.1 cung cấp một số nhãn phổ biến thường được sử dụng trong quá trình gán nhãn từ loại, bao gồm n dành cho danh từ, v cho động từ, a cho tính từ và r cho trạng từ. Bên cạnh các nhãn chính này, hệ thống còn sử dụng những nhãn chi tiết hơn để mô tả rõ ràng chức năng của từ trong ngữ cảnh cụ thể. Chẳng hạn, NN biểu thị danh từ số ít hoặc danh từ thông thường như "con chó", DT là nhãn dành cho mạo từ như "một" hoặc "cái", JJ được sử dụng để đánh dấu các tính từ mô tả đặc điểm hoặc tính chất như "lười biếng", và VBZ là nhãn cho động từ ngôi thứ ba ở thì hiện tại như "chạy". Việc phân loại này không chỉ đảm bảo tính chính xác mà còn cung cấp cái nhìn sâu sắc hơn về ngữ pháp và ngữ nghĩa, giúp hệ thống xử lý hiệu quả cả những câu phức tạp và đa nghĩa.

Cụ thể, gán nhãn từ loại cho phép hệ thống xác định được chức năng của từng từ trong ngữ cảnh câu. Chẳng hạn, trong ví dụ "Một con chó lười biếng nhảy qua hàng rào", hệ thống sẽ thực hiện các bước như sau:

- "Một" được gán nhãn DT (mạo từ), biểu thị số lượng hoặc định lượng của đối tượng trong câu, giúp hệ thống xác định rằng đối tượng "chó" ở đây không mang nghĩa chung chung mà cụ thể là một cá thể.
- "Con chó" được gán nhãn NN (Danh từ), đánh dấu đây là chủ thể chính trong câu và đóng vai trò là trung tâm của hành động.
- "Lười biếng" được gán nhãn JJ (Tính từ), mô tả đặc tính của danh từ.
- "Nhảy qua" được gán nhãn VBZ (Động từ), thể hiện hành động.
- "Hàng rào" được gán nhãn N (Danh từ), chỉ đối tượng mà hành động tác động.

Bảng 2.1: Bảng gán nhãn từ loại

Part of Speech	**Tag (Nhãn)**
Noun (Danh từ)	n
Verb (Động từ)	v
Adjective (Tính từ)	a
Adverb (Trạng từ)	r

2.3.3 Tần suất từ - nghịch đảo tài liệu (Term Frequency-Inverse Document Frequency)

Tần suất nghịch đảo tài liệu (Term Frequency-Inverse Document Frequency - TF-IDF) [52] là một phương pháp phổ biến được sử dụng để đánh giá mức độ quan trọng của một từ trong một tập hợp văn bản (corpus) [53]. Mục tiêu của TF-IDF là xác định các từ đặc trưng và có tính phân biệt cao trong một văn bản, qua đó giúp mô hình nhận diện các từ khóa có ý nghĩa nhất. Mức độ quan trọng của một từ trong văn bản không chỉ phụ thuộc vào số lần xuất hiện của nó mà còn được điều chỉnh bởi độ phổ biến của từ đó trong toàn bộ tập dữ liệu.

Công thức tính tần suất từ (Term Frequency - TF) [54] đo lường số lần từ t xuất hiện trong một văn bản d, và thể hiện mức độ xuất hiện của từ trong tài liệu đó. Công thức tính như sau:

$$tf(t,d) = \frac{\text{Số lần từ } t \text{ xuất hiện trong } d}{\text{Số từ trong văn bản } d} \tag{2.1}$$

Trong đó, TF giúp đánh giá mức độ phổ biến của từ t trong một văn bản cụ thể.

Tiếp theo, tần suất tài liệu (Document Frequency - DF) đo lường sự xuất hiện của từ t trong toàn bộ tập hợp các tài liệu. Khác với TF, DF không đếm số lần từ xuất hiện trong một văn bản, mà là số tài liệu mà từ t xuất hiện. Công thức tính DF:

$$df(t) = \text{Số tài liệu chứa từ } t \tag{2.2}$$

Đây là yếu tố quan trọng giúp xác định mức độ phân biệt của từ trong tập dữ liệu: từ nào xuất hiện trong nhiều tài liệu sẽ có ý nghĩa ít hơn trong việc phân biệt giữa các tài liệu.

Tần suất nghịch đảo tài liệu (Inverse Document Frequency - IDF) là một phép đo dùng để điều chỉnh tần suất từ, phản ánh mức độ quan trọng của từ t trong toàn

bộ tập dữ liệu. Từ nào xuất hiện phổ biến trong nhiều tài liệu sẽ bị giảm trọng số, còn từ ít xuất hiện sẽ có trọng số cao hơn. Công thức tính IDF là:

$$idf(t) = \frac{N}{df(t)} = \frac{N}{N(t)} \tag{2.3}$$

Trong đó, N là tổng tài liệu trong tập dữ liệu và $df(t)$ là số tài liệu chứa từ t. Lượng logarit được áp dụng để giảm ảnh hưởng của các từ xuất hiện thường xuyên.

Cuối cùng, chỉ số TF-IDF được tính bằng cách nhân tần suất từ (TF) và tần suất nghịch đảo tài liệu (IDF), từ đó đánh giá mức độ quan trọng của từ t trong một văn bản d. Công thức tính TF-IDF như sau:

$$tf - idf(t, d) = tf(t, d) \times idf(t) \tag{2.4}$$

Chỉ số TF-IDF cao cho thấy từ đó vừa xuất hiện nhiều trong văn bản, lại vừa ít xuất hiện trong các văn bản khác, từ đó xác định được những từ khóa quan trọng và đặc trưng nhất trong văn bản. Phương pháp TF-IDF rất hữu ích trong các ứng dụng tìm kiếm, phân loại văn bản và hệ thống gợi ý, giúp nâng cao hiệu quả trong việc truy xuất thông tin.

Chương 3

Nền tảng tư vấn nghiệp vụ ảo

Chương 3 sẽ bắt đầu bằng việc phân tích các vấn đề cụ thể đang tồn tại trong các nền tảng tư vấn nghiệp vụ ảo hiện nay. Qua việc đánh giá các hạn chế và thách thức của những hệ thống này, khóa luận rút ra những điểm mấu chốt cần cải thiện. Từ đó, các ý tưởng chủ đạo sẽ được hình thành để phát triển một giải pháp cải thiện hơn, hướng đến việc giải quyết từng vấn đề một cách hiệu quả và toàn diện.

3.1 Các vấn đề hiện tại của nền tảng tư vấn nghiệp vụ ảo

Các nền tảng tư vấn nghiệp vụ ảo đã có những bước tiến đáng kể trong việc hỗ trợ người dùng và cung cấp thông tin, nhưng vẫn tồn tại nhiều hạn chế cần được khắc phục để đáp ứng tốt hơn nhu cầu ngày càng cao. Những hạn chế này không chỉ làm giảm độ tin cậy của hệ thống mà còn ảnh hưởng trực tiếp đến hiệu quả khi xử lý các tình huống phức tạp. Dưới đây là những vấn đề chính và nguyên nhân dẫn đến những hạn chế đó:

- **Vấn đề 1.** Một trong những thách thức lớn nhất là việc xử lý câu hỏi từ phía người dùng. Người dùng thường gặp khó khăn trong việc đặt câu hỏi một cách rõ ràng và cung cấp đầy đủ ngữ cảnh cho hệ thống. Điều này dẫn đến nhiều tình huống như câu hỏi mơ hồ, thiếu thông tin cụ thể, ví dụ: "Doanh thu thương mại điện tử" nhưng không chỉ rõ thời gian, khu vực hay ngành nghề cụ thể. Ngoài ra, người dùng có thể đặt câu hỏi không liên quan đến phạm vi mà hệ thống được thiết kế để xử lý, chẳng hạn hỏi về lập trình trên một nền tảng chỉ hỗ trợ các lĩnh vực kinh tế hoặc pháp luật. Những vấn đề này làm giảm khả năng của hệ thống trong việc cung cấp câu trả lời phù hợp.
- **Vấn đề 2.** Bên cạnh đó, khả năng tìm kiếm và truy xuất thông tin từ cơ sở tri thức cũng là một điểm yếu đáng kể. Việc tổ chức và quản lý dữ liệu không hiệu quả, chẳng hạn như sử dụng cấu trúc đánh chỉ mục (indexing) không phù hợp, dẫn đến tốc độ tìm kiếm chậm và kết quả kém chính xác. Hơn nữa, nhiều hệ thống vẫn dựa vào các thuật toán tìm kiếm truyền thống như TF-IDF

hoặc BM25, thay vì áp dụng các phương pháp hiện đại như tìm kiếm hỗn hợp (hybrid search), kết hợp giữa tìm kiếm theo từ khóa và vector embedding để nắm bắt ý nghĩa sâu hơn của truy vấn. Điều này khiến hệ thống khó xử lý các câu hỏi yêu cầu ngữ nghĩa phức tạp. Ngoài ra, dữ liệu không đầy đủ hoặc đã lỗi thời cũng là nguyên nhân khiến hệ thống không thể cung cấp các câu trả lời phù hợp với bối cảnh hiện tại.

- **Vấn đề 3.** Một vấn đề khác đến từ việc lựa chọn và vận hành mô hình ngôn ngữ lớn (LLM). Mặc dù các mô hình LLM đang không ngừng cải tiến, nhưng việc sử dụng các mô hình không được tùy chỉnh hoặc không phù hợp với nhu cầu cụ thể của nền tảng có thể dẫn đến hiệu quả kém. Ví dụ, các mô hình ngôn ngữ được cập nhật thường xuyên có thể không tương thích với nền tảng cũ, trong khi việc chuyển đổi hoặc nâng cấp lại tốn nhiều chi phí và công sức. Hơn nữa, nhiều hệ thống thiếu tính linh hoạt trong việc kết hợp các mô hình khác nhau hoặc sử dụng mô hình tùy chỉnh, điều này hạn chế khả năng mở rộng và tối ưu hóa hiệu năng của nền tảng.

- **Vấn đề 4.** Ngoài ra, việc xử lý các dạng dữ liệu phức tạp, đặc biệt là dữ liệu dạng bảng hoặc số liệu, cũng là một thách thức lớn. Khi dữ liệu dạng bảng quá nhiều và không được quản lý tốt, hệ thống gặp khó khăn trong việc tìm kiếm và truy xuất chính xác thông tin phù hợp với câu hỏi của người dùng. Dữ liệu như vậy thường yêu cầu các phương pháp lưu trữ, index và truy vấn chuyên biệt, trong khi nhiều nền tảng chưa được trang bị đầy đủ để xử lý hiệu quả loại dữ liệu này. Điều này khiến các câu trả lời liên quan đến bảng số liệu thường thiếu chính xác hoặc không đáp ứng được kỳ vọng của người dùng.

Những hạn chế kể trên chỉ ra rằng các nền tảng tư vấn nghiệp vụ ảo hiện nay cần được cải thiện toàn diện, từ khả năng xử lý câu hỏi của người dùng, truy xuất dữ liệu hiệu quả, đến vận hành linh hoạt các mô hình ngôn ngữ và xử lý dữ liệu phức tạp. Đây là cơ sở để đề xuất những giải pháp cải tiến, giúp các nền tảng này đạt được tính chính xác, hiệu quả và khả năng mở rộng cao hơn trong tương lai.

3.2 Hướng tiếp cận

Từ những vấn đề được đề cập ở Mục 3.1, có thể nhận thấy sự cần thiết phải phát triển một giải pháp toàn diện nhằm khắc phục các hạn chế hiện tại của nền tảng

tư vấn nghiệp vụ ảo. Giải pháp này cần hướng tới việc giảm thiểu sự mơ hồ trong câu hỏi của người dùng, cải thiện hiệu suất truy xuất dữ liệu, tối ưu hóa vận hành và lựa chọn mô hình ngôn ngữ lớn, đồng thời xử lý hiệu quả các dạng dữ liệu phức tạp như bảng và số liệu. Trong khóa luận này, giải pháp QUESTIN được thiết kế dựa trên bốn nguyên tắc cơ bản như sau:

- **Nguyên tắc 1. Xây dựng quy trình kiểm tra và đề xuất câu hỏi**: Hệ thống cần triển khai quy trình kiểm tra tính phù hợp của các câu hỏi, đảm bảo rằng truy vấn của người dùng liên quan trực tiếp đến chủ đề mà nền tảng đang xử lý. Điều này giúp giảm thiểu những truy vấn không liên quan, từ đó nâng cao chất lượng phản hồi. Ngoài ra, hệ thống cũng cần sinh các câu hỏi gợi ý dựa trên cơ sở tri thức hiện có, giúp người dùng dễ dàng định hình được câu hỏi của mình, tránh sự mơ hồ hoặc không biết phải hỏi gì. Một bước tiến nữa là sử dụng các mô hình ngôn ngữ lớn để tạo ra tài liệu giả lập (synthetic documents) dựa trên câu hỏi của người dùng, từ đó tăng cường khả năng tìm kiếm và truy vấn thông tin phù hợp.
- **Nguyên tắc 2. Cải thiện hiệu suất truy xuất dữ liệu**: Hiệu suất truy vấn dữ liệu là yếu tố quan trọng quyết định tính hiệu quả của một nền tảng tư vấn nghiệp vụ. Giải pháp QUESTIN sử dụng phương pháp tìm kiếm kết hợp (hybrid search), tích hợp tìm kiếm toàn văn (full-text search) và truy vấn vector để khai thác cả khía cạnh từ khóa lẫn ngữ nghĩa trong dữ liệu. Elasticsearch được lựa chọn làm công cụ hỗ trợ để đảm bảo tốc độ và hiệu quả truy vấn. Hệ thống cũng xây dựng cấu trúc dữ liệu dạng cây (theo phương pháp Raptor), tạo ra nền tảng linh hoạt hơn cho việc truy vấn các dữ liệu phức tạp. Sau khi truy vấn, các kết quả sẽ được xếp hạng lại bằng mô hình đánh giá mức độ tương đồng, đảm bảo rằng chỉ những câu trả lời có mức độ khớp cao hơn ngưỡng (threshold) được trả về.
- **Nguyên tắc 3. Cải thiện vận hành và linh hoạt sử dụng mô hình ngôn ngữ lớn**: Hệ thống cần đảm bảo khả năng linh hoạt trong việc tích hợp và chuyển đổi giữa các mô hình ngôn ngữ lớn, đáp ứng nhu cầu thay đổi nhanh chóng và đa dạng của các lĩnh vực khác nhau. Điều này được thực hiện thông qua việc cho phép thay đổi mô hình dễ dàng chỉ bằng cách cung cấp API key và Base URL từ các nền tảng khác nhau như OpenAI, TogetherAI, Gemini, v.v. Đối với các mô hình chạy trên máy chủ cục bộ, các thư viện như vLLM,

SGLaM, hoặc llama.cpp sẽ được sử dụng để giảm chi phí vận hành so với việc phụ thuộc hoàn toàn vào dịch vụ API đám mây. Bên cạnh đó, mô hình có thể được cải thiện bằng cách áp dụng kỹ thuật tinh chỉnh (fine-tuning) để tối cải thiện hiệu suất cho các tác vụ tiếng Việt hoặc lĩnh vực cụ thể, cùng với kỹ thuật nén mô hình (quantization) để giảm tài nguyên và tăng tốc độ xử lý.

- **Nguyên tắc 4. Xử lý hiệu quả dữ liệu bảng số liệu**: Đối với các nền tảng cần làm việc với dữ liệu bảng hoặc số liệu, hệ thống cần có cách lưu trữ và xử lý chuyên biệt. Các trường dữ liệu như tiêu đề (header), mô tả bảng và nội dung từng dòng được lưu trữ riêng biệt để dễ dàng index và truy vấn. Khi nhận được truy vấn, hệ thống sử dụng cơ chế lọc và truy xuất dựa trên SQL, đảm bảo khả năng xử lý nhanh chóng và chính xác các yêu cầu liên quan đến dữ liệu bảng, biểu đồ hoặc so sánh số liệu.

Bốn nguyên tắc trên đặt nền tảng cho giải pháp QUESTIN, đảm bảo rằng hệ thống không chỉ khắc phục được các hạn chế hiện tại mà còn đạt được mục tiêu xây dựng một nền tảng tư vấn nghiệp vụ ảo chính xác, linh hoạt và hiệu quả, đáp ứng tốt hơn nhu cầu ngày càng cao từ người dùng.

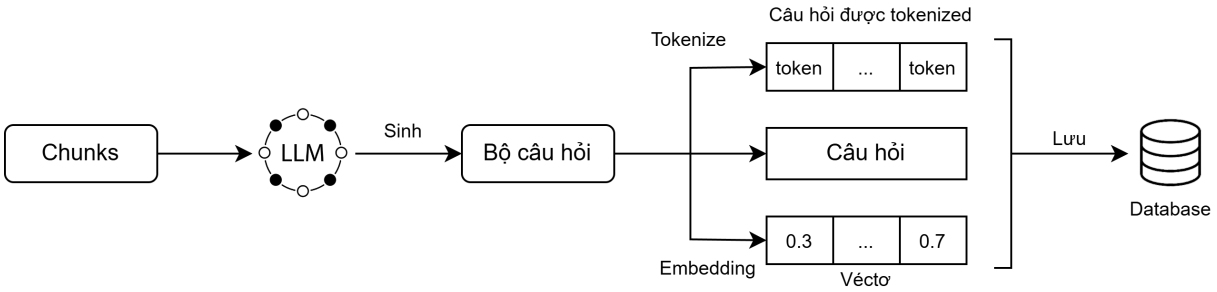
3.3 Thiết kế giải pháp

3.3.1 Xây dựng quy trình kiểm tra và đề xuất câu hỏi người dùng

Để giảm thiểu sự mơ hồ trong câu hỏi của người dùng và nâng cao tính truy vấn của thông tin, giải pháp trong khóa luận đề xuất ba giai đoạn chính: (1) Sinh bộ câu hỏi gợi ý dựa trên cơ sở tri thức, (2) Kiểm tra tính hợp lệ của câu hỏi và gợi ý các câu hỏi phù hợp, (3) Sử dụng mô hình ngôn ngữ lớn để sinh tài liệu giả lập (hypothesis document) nhằm bổ sung thông tin cho quá trình truy vấn.

Xây dựng bộ câu hỏi gợi ý

Trong các tình huống tương tác, đặc biệt là khi người dùng lần đầu sử dụng hệ thống hoặc chưa xác định được rõ ràng câu hỏi của mình, họ có thể cảm thấy lúng túng hoặc không biết phải bắt đầu từ đâu. Điều này có thể dẫn đến việc đặt các câu hỏi không liên quan, không rõ ràng hoặc làm giảm sự hứng thú trong trải

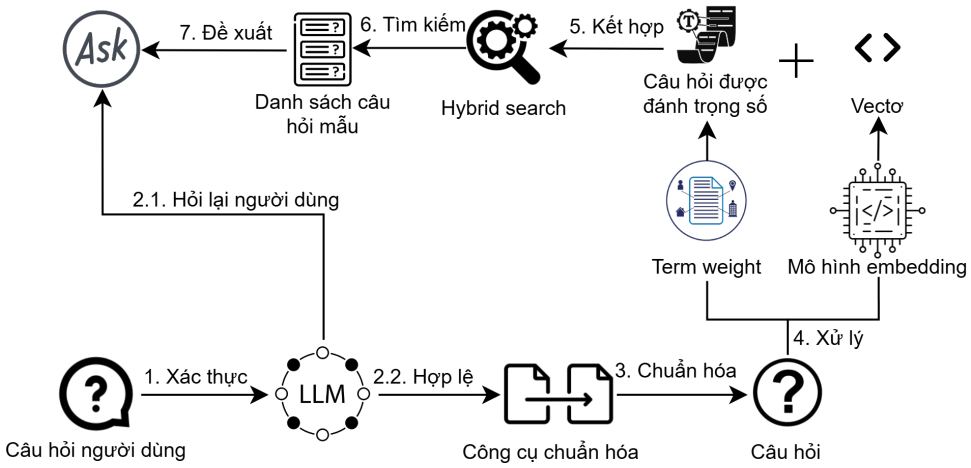


Hình 3.1: Thành phần hệ thống gợi ý và xác thực

nghiệm với hệ thống. Để giải quyết vấn đề này, hệ thống được thiết kế để tự động sinh ra bộ câu hỏi gợi ý dựa trên cơ sở tri thức sẵn có. Những câu hỏi này cung cấp cho người dùng một danh sách các lựa chọn phù hợp, giúp hướng dẫn và khuyến khích họ khai thác hiệu quả thông tin từ hệ thống. Như minh họa trong hình 3.1, hệ thống sử dụng các đoạn văn bản (chunks) đã được phân tích từ tài liệu làm ngữ cảnh cho mô hình ngôn ngữ lớn để sinh ra các câu hỏi gợi ý. Bộ câu hỏi được sinh ra sẽ được xử lý qua hai phiên bản bổ sung như sau:

- **Phiên bản 1**: Bộ câu hỏi được token hóa, phân tích từ và cụm từ để tạo ra các câu hỏi mới, mang lại sự kết nối rõ ràng giữa các từ khóa chính. Việc này giúp hệ thống dễ dàng hiểu và xử lý các cụm từ quan trọng trong câu hỏi của người dùng, tăng khả năng chính xác trong truy vấn.
- **Phiên bản 2**: Bộ câu hỏi được đưa vào mô hình embedding để chuyển đổi thành các vectơ số hóa. Các vectơ này lưu trữ thông tin về ngữ nghĩa, cho phép hệ thống so sánh và tìm kiếm các câu hỏi tương tự dựa trên nội dung thay vì chỉ dựa vào từ khóa.

Cả hai phiên bản của bộ câu hỏi sau khi xử lý, cùng với bản gốc, được lưu trữ trong cơ sở dữ liệu của hệ thống. Bộ câu hỏi ban đầu đóng vai trò là nguồn gợi ý trực tiếp cho người dùng, trong khi các phiên bản đã qua xử lý hỗ trợ các chức năng tìm kiếm và truy vấn ngữ nghĩa nâng cao. Bằng cách kết hợp giữa sinh câu hỏi, tối ưu hóa ngữ nghĩa, và xây dựng cơ sở dữ liệu hỗ trợ mạnh mẽ, hệ thống không chỉ giúp người dùng tránh được sự mơ hồ khi đặt câu hỏi mà còn cải thiện đáng kể trải nghiệm tương tác và hiệu quả truy vấn thông tin.



Hình 3.2: Luồng hoạt động của hệ thống gợi ý và xác thực

Luồng hoạt động kiểm tra và gợi ý câu hỏi

Trong hình 3.2, quy trình làm việc của hệ thống được thiết kế để tối ưu hóa việc xử lý và phản hồi các câu hỏi từ người dùng một cách hiệu quả và chính xác. Mỗi khi người dùng đưa ra một câu hỏi, hệ thống sẽ thực hiện một loạt các bước xử lý nhằm đảm bảo rằng câu hỏi có liên quan và phù hợp với mục đích của hệ thống, đồng thời cung cấp những gợi ý hữu ích để người dùng có thể dễ dàng tìm kiếm câu trả lời mong muốn.

Quá trình này bắt đầu với việc sử dụng một mô hình ngôn ngữ lớn (LLM) để phân tích câu hỏi mà người dùng đã nhập vào (1). LLM giúp xác định xem câu hỏi hiện tại có thuộc phạm vi hoặc lĩnh vực mà hệ thống hỗ trợ hay không. Điều này rất quan trọng vì hệ thống cần đảm bảo rằng câu hỏi của người dùng có liên quan đến nội dung mà nó có thể xử lý. Nếu mô hình ngôn ngữ xác định rằng câu hỏi không liên quan, hệ thống sẽ ngay lập tức yêu cầu người dùng xác nhận lại xem họ có muốn tiếp tục với câu hỏi hiện tại hay không (2.1). Ở đây, người dùng có thể chọn điều chỉnh lại câu hỏi hoặc tiếp tục tiến hành với câu hỏi ban đầu nếu họ cho rằng nó vẫn phù hợp. Ngược lại, nếu câu hỏi được xác định là liên quan, hệ thống sẽ chuyển sang bước tiếp theo là sử dụng kỹ thuật chuẩn hóa để loại bỏ những ký tự hoặc yếu tố không cần thiết có trong câu hỏi (3). Việc này bao gồm việc loại bỏ các dấu chấm câu không cần thiết, các ký tự đặc biệt hoặc các từ không có ý nghĩa quan trọng trong ngữ cảnh câu hỏi. Quá trình chuẩn hóa giúp làm cho câu hỏi trở nên ngắn gọn và dễ dàng phân tích hơn, đồng thời cũng giúp giảm thiểu

sai sót khi hệ thống xử lý.

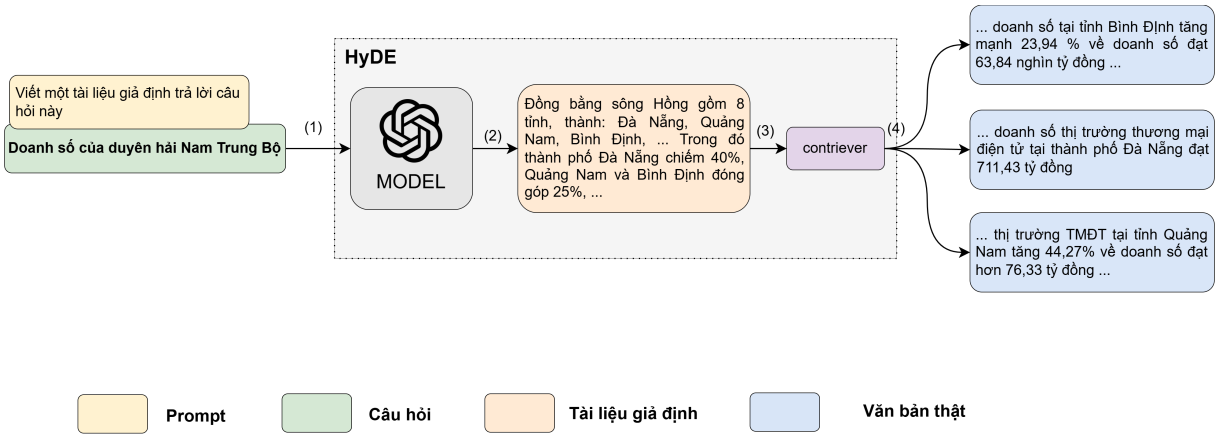
Tiếp theo, câu hỏi đã được chuẩn hóa sẽ được chuyển qua giai đoạn tokenize để phân tách thành các token (4). Trong giai đoạn này, hệ thống sử dụng các công cụ như NER (Named Entity Recognition), POSTAG (Part-of-Speech Tagging), và FREQ (Frequency Analysis) kết hợp với IDF (Inverse Document Frequency) để tính toán trọng số cho các từ khóa. Việc này giúp nhận diện và đánh dấu các thành phần quan trọng trong câu hỏi như tên riêng, địa danh, tổ chức hoặc các thuật ngữ chuyên ngành, từ đó hệ thống có thể hiểu rõ hơn ngữ cảnh và ý nghĩa của câu hỏi. Bên cạnh đó, câu hỏi cũng sẽ được đưa vào một mô hình nhúng (embedding model) để chuyển đổi thành các vectơ (4). Quá trình nhúng này biểu diễn câu hỏi dưới dạng tọa độ trong không gian vectơ, giúp hệ thống so sánh, đối chiếu với các câu hỏi khác một cách hiệu quả và tìm kiếm các câu hỏi tương tự.

Sau khi câu hỏi đã được vectơ hóa và các từ khóa đã được trích xuất, hệ thống sẽ chuyển sang bước tiếp theo, nơi công cụ tìm kiếm của hệ thống sẽ kết hợp các vectơ và câu hỏi sau khi được đánh trọng số này để tạo thành một truy vấn kết hợp để thực hiện việc tra cứu các câu hỏi liên quan trong cơ sở dữ liệu (5). Công cụ tìm kiếm này hoạt động dựa trên nguyên tắc tìm ra các câu hỏi có ngữ nghĩa hoặc cấu trúc tương tự với câu hỏi của người dùng hiện tại, nhằm cung cấp những gợi ý hữu ích cho họ.

Sau khi công cụ tìm kiếm hoàn tất việc truy xuất dữ liệu, hệ thống sẽ hiển thị danh sách các câu hỏi tương tự hoặc liên quan đã tìm thấy từ cơ sở dữ liệu cho người dùng (6). Điều này giúp người dùng có thêm thông tin để tham khảo và có thể điều chỉnh câu hỏi của mình nếu họ thấy rằng một trong các câu hỏi liên quan đã chứa câu trả lời mà họ đang tìm kiếm. Nếu không, người dùng có thể tiếp tục với câu hỏi của mình và nhận câu trả lời từ hệ thống (7). Bằng cách cung cấp các câu hỏi tương tự, hệ thống không chỉ giúp tiết kiệm thời gian tìm kiếm của người dùng mà còn cải thiện độ chính xác của câu trả lời, đặc biệt trong các trường hợp người dùng không chắc chắn về cách diễn đạt câu hỏi của mình.

Sinh tài liệu giả định

Như minh họa trong Hình 3.3, khi hệ thống gặp phải các câu hỏi có tính mơ hồ hoặc thiếu thông tin cần thiết để thực hiện quá trình truy xuất hiệu quả, một



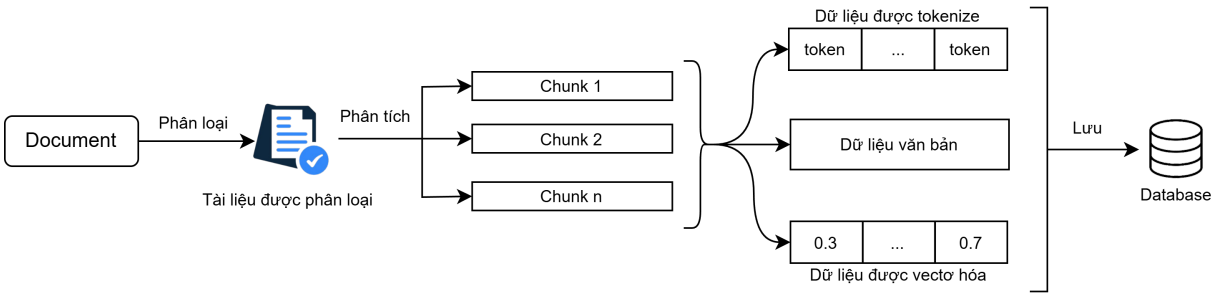
Hình 3.3: Quy trình xây dựng tài liệu giả định

phương pháp hỗ trợ là sử dụng mô hình ngôn ngữ lớn để tạo ra các tài liệu giả định. Quy trình này bao gồm các bước sau:

Trước hết, hệ thống tạo một prompt dựa trên câu hỏi đầu vào của người dùng. Prompt này đóng vai trò như một hướng dẫn cụ thể cho mô hình ngôn ngữ lớn, giúp định hướng quá trình sinh tài liệu giả định sao cho liên quan đến câu hỏi gốc (1). Mô hình ngôn ngữ lớn sau đó sinh ra một văn bản giả định chứa các thông tin mà câu hỏi của người dùng yêu cầu (2). Trong văn bản này, mặc dù các số liệu và nội dung cụ thể có thể không hoàn toàn chính xác, nhưng chúng sẽ cung cấp một bối cảnh hoặc giả thuyết để hỗ trợ quá trình tìm kiếm tiếp theo.

Văn bản giả định này được sử dụng làm truy vấn đầu vào cho công cụ tìm kiếm (3). Dựa trên tài liệu giả định này, hệ thống sẽ thực hiện quá trình tìm kiếm các văn bản liên quan trong cơ sở dữ liệu. Các công cụ tìm kiếm được sử dụng sẽ áp dụng thuật toán tìm kiếm hỗn hợp (hybrid search) đã được mô tả trong Mục 3.3.2 để đảm bảo rằng các kết quả trả về có mức độ liên quan cao với nội dung giả định. Cuối cùng, các kết quả này được tổng hợp và trả về cho người dùng (4).

Phương pháp này có hai ưu điểm chính. Thứ nhất, việc sinh tài liệu giả định giúp hệ thống mở rộng ngữ cảnh truy vấn, đặc biệt khi câu hỏi đầu vào của người dùng không đủ chi tiết hoặc rõ ràng. Thứ hai, phương pháp này tăng khả năng hệ thống tìm kiếm được các thông tin liên quan, ngay cả khi câu hỏi ban đầu không trùng khớp hoàn toàn với nội dung trong cơ sở dữ liệu. Bằng cách sử dụng các tài liệu giả định, hệ thống có thể tiếp cận dữ liệu một cách linh hoạt hơn, từ đó cải thiện đáng kể hiệu suất truy vấn và khả năng trả lời chính xác câu hỏi của người dùng.



Hình 3.4: Quá trình xử lý dữ liệu

3.3.2 Cải thiện hiệu suất truy xuất thông tin

Để nâng cao hiệu quả trong việc truy xuất thông tin, nền tảng QUESTIN áp dụng các kỹ thuật giúp cải thiện khả năng truy vấn và sử dụng cấu trúc dữ liệu phù hợp nhằm tăng cường độ chính xác, tốc độ và khả năng xử lý. Giải pháp được triển khai qua bốn phần chính: (1) Thực hiện quá trình chunking và index dữ liệu để đảm bảo tính linh hoạt và hiệu quả trong việc quản lý dữ liệu, (2) Áp dụng thuật toán tìm kiếm kết hợp (hybrid search) để cải thiện độ chính xác trong truy vấn, (3) Xây dựng cấu trúc dữ liệu dạng cây theo phương pháp Raptor để nâng cao khả năng tìm kiếm đối với dữ liệu phức tạp, (4) Xếp hạng lại các kết quả tìm kiếm.

Quá trình xử lý văn bản

Để xử lý tài liệu văn bản một cách toàn diện và linh hoạt, việc phát triển một hệ thống xử lý văn bản chuyên sâu là điều không thể thiếu. Như minh họa trong Hình 3.4, khi một tài liệu được thêm vào trong hệ thống, hệ thống sẽ phân loại xem đó là loại tài liệu nào PDF, Excel, TXT, v.v. Sau đó hệ thống sẽ phân tích tài liệu đó và phân tách thành các đoạn văn bản (chunk) nhỏ hơn để lưu trong hệ thống. Với mỗi chunk hệ thống cũng tạo thêm hai phiên bản của nó tương tự như việc xử lý câu hỏi ở Mục 3.3.1 như sau:

- **Phiên bản 1**: Đoạn văn bản được chuẩn hóa bằng cách loại bỏ các ký tự không cần thiết, sau đó được tokenize thành các từ hoặc cụm từ mang ý nghĩa rõ ràng. Quá trình này giúp hệ thống nhận diện và xử lý chính xác các cụm từ quan trọng trong văn bản. Phiên bản này được thiết kế để hỗ trợ các phương pháp tìm kiếm dựa trên toàn văn (full-text search), trong đó các token được

gán trọng số dựa trên các yếu tố như tần suất xuất hiện (TF-IDF), loại từ (POS tagging), và ý nghĩa của thực thể (NER). Trọng số này đóng vai trò quan trọng trong việc cải thiện độ chính xác của tìm kiếm toàn văn, sẽ được trình bày trong Mục 3.3.2.

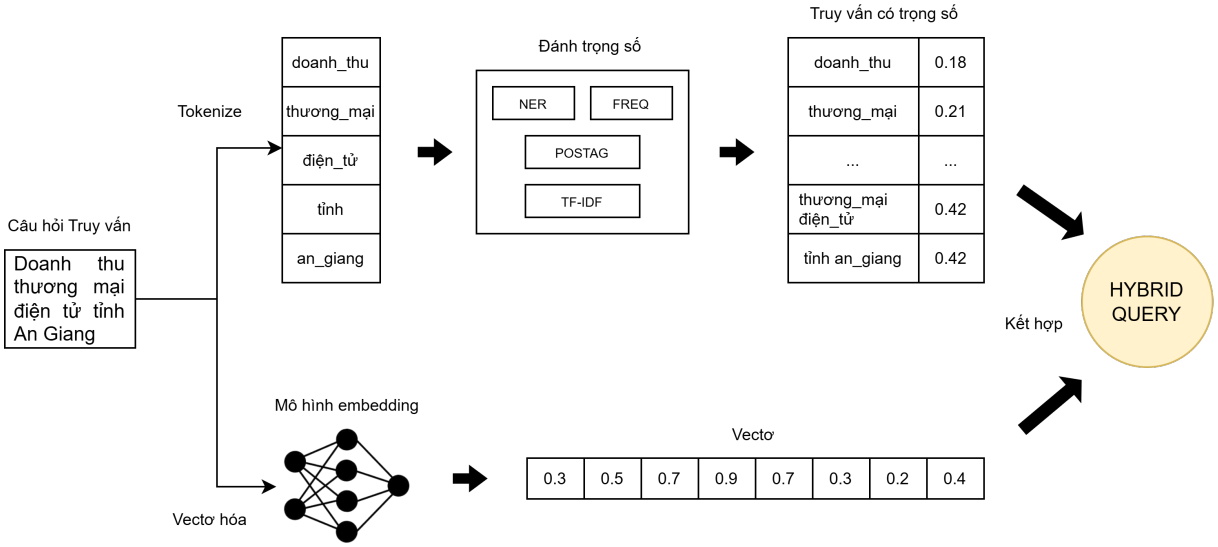
- **Phiên bản 2**: Đoạn văn bản được đưa vào một mô hình embedding để chuyển đổi thành các vectơ biểu diễn số hóa. Các vectơ này lưu trữ thông tin về ngữ nghĩa của văn bản, giúp hệ thống thực hiện các phép so sánh độ tương đồng dựa trên ngữ nghĩa thay vì chỉ dựa trên từ khóa. Điều này đặc biệt quan trọng đối với các câu hỏi phức tạp hoặc khi các từ khóa trong câu hỏi không khớp hoàn toàn với nội dung văn bản.

Cả hai phiên bản của đoạn văn bản, cùng với bản gốc, được lưu trữ trong một cấu trúc dữ liệu duy nhất tại công cụ tìm kiếm. Đoạn văn bản ban đầu đóng vai trò là ngữ cảnh hỗ trợ trực tiếp cho quá trình trả lời câu hỏi, trong khi hai phiên bản bổ sung được sử dụng để nâng cao hiệu quả và độ chính xác của các phương pháp truy vấn khác nhau. Quy trình xử lý tài liệu này không chỉ đảm bảo dữ liệu được chuẩn bị một cách toàn diện, mà còn cải thiện khả năng truy xuất của hệ thống. Bằng cách sử dụng các kỹ thuật xử lý văn bản này, hệ thống có thể hỗ trợ người dùng tìm kiếm và truy vấn dữ liệu một cách hiệu quả hơn, đáp ứng được cả các câu hỏi đơn giản và phức tạp với mức độ chính xác cao hơn.

Thuật toán tìm kiếm hỗn hợp (hybrid search)

Thuật toán tìm kiếm hỗn hợp kết hợp sức mạnh của các phương pháp tìm kiếm truyền thống và hiện đại. Như Hình 3.5 minh họa, thuật toán chia thành hai quá trình chính:

- **Quá trình 1**: Truy vấn đầu vào được xử lý thông qua quá trình tiền xử lý (tokenization), trong đó các từ và cụm từ mang ý nghĩa quan trọng được phân tách. Quá trình này giúp hệ thống xác định rõ các điểm nhấn chính trong truy vấn, chẳng hạn như cụm từ "tỉnh An Giang" hoặc "thương mại điện tử". Tiếp theo, hệ thống áp dụng kỹ thuật nhận diện thực thể tên (NER) để gắn nhãn và đánh dấu các token phù hợp. Các token quan trọng như địa danh, tên tổ chức, và các thực thể đặc biệt sẽ được gán trọng số cao hơn nhằm tăng mức



Hình 3.5: Thuật toán tìm kiếm hỗn hợp

độ ưu tiên trong tìm kiếm. Cùng lúc đó, POSTAG (gán nhãn loại từ) được sử dụng để xác định loại từ, chẳng hạn như danh từ riêng hoặc địa danh, với các trọng số được điều chỉnh tương ứng nhằm nhấn mạnh các thành phần này. Sau khi xác định loại từ, hệ thống tính toán tần suất xuất hiện của từng token (FREQ) và sử dụng hàm nghịch đảo tần suất xuất hiện tài liệu (IDF) để giảm thiểu ảnh hưởng của những token phổ biến không mang ý nghĩa phân biệt cao. Quá trình này cho phép hệ thống cân bằng giữa việc tìm kiếm chính xác và việc lọc nhiễu từ những token ít giá trị. Chuỗi truy vấn sau khi được tính trọng số sẽ được chuyển vào giai đoạn tìm kiếm toàn văn bằng thuật toán BM25. Trọng số của mỗi token trong chuỗi được tính toán theo công thức sau:

$$w_t^{'} = \frac{[(0.3 \cdot IDF(freq(t), N) + 0.7 \cdot IDF(df(t), N)) \cdot NER(t) \cdot POS(t)]}{\sum_{t \in tokens} w_t} \quad (3.1)$$

Trong đó:

- $w_t^{'}$ là trọng số chuẩn hóa cuối cùng của từ t trong chuỗi truy vấn.
- $IDF(freq(t), N)$ là hàm nghịch đảo tần suất tài liệu dựa trên tần suất xuất hiện của từ t trong tập dữ liệu N.
- $IDF(df(t), N)$ là hàm IDF tính dựa trên độ phổ biến tài liệu (document frequency) của từ t trên tập dữ liệu. Thành phần này bổ sung thêm thông tin về mức độ phân tán của từ trong toàn bộ tập dữ liệu N.

- $NER(t)$ là trọng số dựa trên loại thực thể của từ t, được xác định thông qua kỹ thuật nhận diện thực thể tên. Các từ thuộc các loại thực thể đặc biệt như địa danh, tên tổ chức, hoặc tên riêng thường được gán trọng số cao hơn để phản ánh tầm quan trọng của chúng.
- $POS(t)$ là trọng số dựa trên nhãn loại từ của t, xác định bởi POSTAG. Các loại từ như danh từ riêng, địa danh, hoặc từ mang tính thông tin thường được ưu tiên cao hơn.
- $\sum_{t \in tokens^{w_t}}$ là tổng trọng số của tất cả các token trong chuỗi truy vấn, dùng để chuẩn hóa trọng số của từng token, đảm bảo tổng trọng số cuối cùng của toàn bộ truy vấn luôn bằng 1.

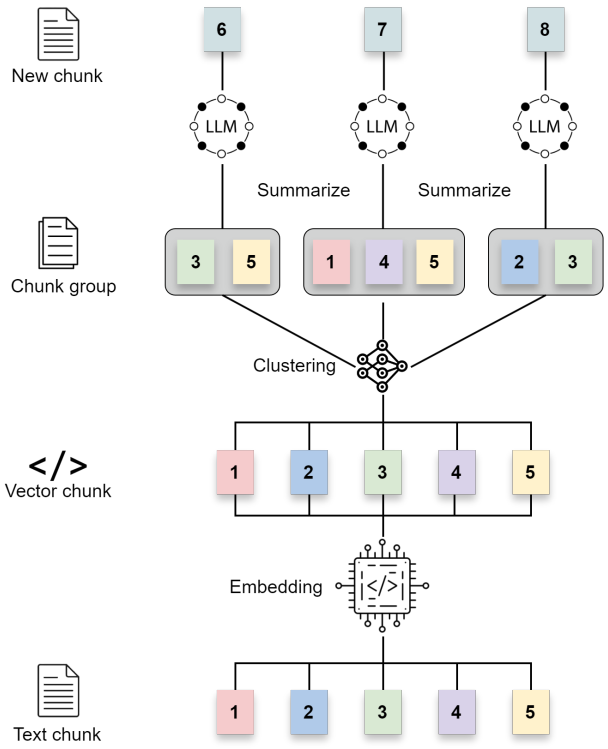
- **Quá trình 2**: Truy vấn đầu vào được chuyển đổi thành vector bằng cách sử dụng mô hình embedding tiên tiến, đảm bảo biểu diễn được ngữ nghĩa sâu sắc và mối quan hệ ngữ cảnh giữa các từ. Quá trình này giúp hệ thống không chỉ dừng lại ở việc hiểu các từ đơn lẻ mà còn nắm bắt được ý nghĩa toàn diện của cả câu truy vấn, kể cả khi có các cấu trúc ngữ pháp phức tạp hoặc ý nghĩa ngầm định. Kỹ thuật embedding có khả năng ánh xạ các từ và cụm từ vào một không gian vector có chiều cao, nơi các vector gần nhau đại diện cho các từ hoặc ngữ cảnh có liên quan về mặt ngữ nghĩa. Điều này cho phép hệ thống tìm kiếm không chỉ dựa trên sự khớp từ khóa mà còn dựa trên sự tương đồng ý nghĩa giữa truy vấn và tài liệu trong cơ sở dữ liệu. Việc sử dụng embedding làm tăng đáng kể độ chính xác của kết quả, đặc biệt trong các trường hợp truy vấn không rõ ràng hoặc có sự đa nghĩa.

Cuối cùng, hệ thống kết hợp kết quả tìm kiếm toàn văn và tìm kiếm vector thành truy vấn lai tổng hợp, giúp tìm kiếm các đoạn văn bản liên quan một cách nhanh chóng và chính xác, đảm bảo hiểu được ngữ cảnh và phong phú trong kết quả.

Xây dựng cây tìm kiếm

Trong Hình 3.6, dữ liệu được tổ chức dưới dạng cây nhằm nắm bắt ý nghĩa của đoạn văn bản ở cả cấp độ khái quát và chi tiết. Cụ thể, cây dữ liệu được xây dựng dựa trên các bước sau đây:

- **Bước 1.** Sau khi các đoạn văn bản được chia nhỏ thông qua kỹ thuật chunking,

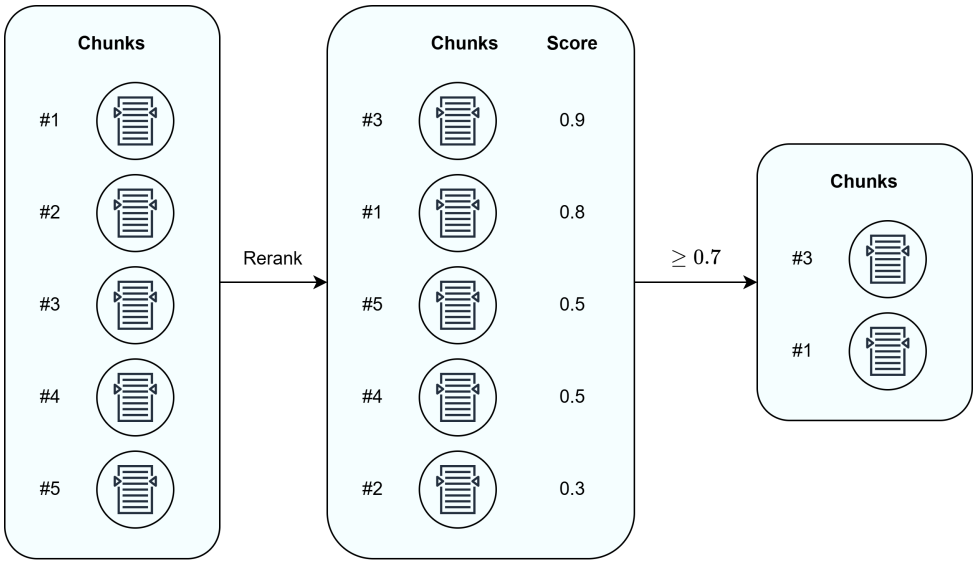


Hình 3.6: Quá trình xây dựng cây dữ liệu

chúng vẫn được duy trì dưới dạng văn bản gốc. Tuy nhiên, để các mô hình học máy có thể phân loại và nhóm các đoạn văn bản này vào các nhóm ngữ nghĩa phù hợp, đầu vào phải được chuyển hóa sang không gian vectơ. Điều này đòi hỏi QUESTIN thực hiện việc chuẩn hóa dữ liệu và biểu diễn chúng dưới dạng cấu trúc vectơ, phù hợp với yêu cầu của các mô hình học sâu. Để thực hiện quá trình này, QUESTIN sử dụng mô hình sentence transformer tiếng Việt nhằm đảm bảo các đoạn văn bản đầu vào được chuyển đổi một cách chính xác và giữ nguyên tính ngữ nghĩa của chúng.

- **Bước 2.** Để có thể hiểu rõ ngữ nghĩa của từng đoạn văn bản và phân chúng thành các nhóm tương ứng theo mức độ tương đồng ngữ nghĩa, QUESTIN áp dụng mô hình Gaussian Mixture Model (GMM) [55]. Mô hình này phân phối các đoạn văn bản vào các cụm khác nhau, mỗi cụm bao gồm các đoạn có mức độ tương đồng nhất định về mặt ngữ nghĩa. Bằng cách này, QUESTIN có thể tổ chức dữ liệu thành các nhóm có ý nghĩa, giúp việc xử lý và truy xuất thông tin trở nên chính xác và hiệu quả hơn.

- **Bước 3.** Sau khi các đoạn văn bản được phân cụm, từng cụm được sử dụng



Hình 3.7: Xếp hạng kết quả tìm kiếm

như một ngữ cảnh liên kết. Tại bước này, mỗi cụm đóng vai trò cung cấp ngữ cảnh để đưa vào xử lý bằng một LLM. Mô hình này có nhiệm vụ tóm tắt nội dung của các đoạn văn bản trong từng cụm. Sau khi tóm tắt, văn bản mới (kết quả của quá trình tóm tắt) sẽ được sử dụng như đoạn văn bản mới.

Quá trình này tiếp tục được lặp lại, tuân theo chuỗi các bước từ Bước 1, đảm bảo rằng mỗi lớp trong cây dữ liệu phản ánh nội dung ngày càng khái quát và cô đọng hơn. Vòng lặp này sẽ tiếp diễn cho đến khi đạt được độ sâu quy định trong cấu trúc cây hoặc khi mỗi cụm chỉ còn lại một đoạn văn bản duy nhất. Nhờ vậy, cây dữ liệu không chỉ tổ chức nội dung một cách rõ ràng mà còn tối ưu hóa về mặt thông tin, giúp lưu giữ ý nghĩa sâu sắc mà không bị mất mát ngữ nghĩa quan trọng trong từng lớp của cây. Nhờ vào những nguyên tắc này, QUESTIN có thể xây dựng một cấu trúc cây dữ liệu mạnh mẽ và chặt chẽ, vừa đáp ứng được yêu cầu phân loại, vừa đảm bảo rằng thông tin từ cấp độ chi tiết đến tổng quát được xử lý một cách logic và hiệu quả.

Xếp hạng lại kết quả tìm kiếm

Hình 3.7 mô tả quá trình xếp hạng lại các đoạn văn bản tìm được sau quá trình truy vấn. Trong quá trình truy vấn, hệ thống có thể tìm thấy nhiều đoạn văn bản liên quan, nhưng không phải tất cả đều có mức độ liên quan cao đến câu hỏi của

người dùng. Việc sử dụng tất cả các đoạn văn bản này làm ngữ cảnh cho mô hình ngôn ngữ lớn có thể gây tốn thêm tài nguyên và làm giảm hiệu quả xử lý của mô hình, đặc biệt khi có các đoạn văn bản không phù hợp. Do đó, bước xếp hạng lại là cần thiết để lọc ra những đoạn văn bản thực sự có giá trị và giảm thiểu tài nguyên cần thiết cho quá trình truy vấn.

Sau khi kết quả tìm kiếm được trả về, mỗi đoạn văn bản sẽ được tính toán lại điểm số dựa trên độ tương đồng với câu truy vấn. Quá trình xếp hạng lại có thể được thực hiện bằng cách sử dụng mô hình xếp hạng (reranking model) hoặc thông qua công thức tính toán độ tương đồng như sau:

$$sim[i] = (CosineSimilarity(\vec{a}, b\vec{[i]}) \cdot vtweight) + (tokenSimilarity(atks, btks[i]) \cdot tkweight) \tag{3.2}$$

Trong đó:

- $sim[i]$ là độ tương đồng của đoạn văn bản thứ i so với câu truy vấn.
- $\vec{a}$ là véctơ số học (embedding) của câu truy vấn.
- $b\vec{[i]}$ là véctơ số học (embedding) của đoạn văn bản thứ i.
- $atks$ là các token trong câu truy vấn.
- $btks[i]$ là các token trong đoạn văn bản thứ i.
- $CosineSimilarity$ và $tokenSimilarity$ lần lượt là độ tương đồng giữa các véctơ (tính bằng cosine similarity) và độ tương đồng giữa các token trong câu truy vấn và đoạn văn bản.
- $vtweight$ và $tkweight$ là trọng số ứng với độ tương đồng giữa các véctơ và độ tương đồng giữa các token.

Sau khi tính toán độ tương đồng, hệ thống sẽ xếp hạng các đoạn văn bản dựa trên điểm số $sim[i]$ tính được. Những đoạn văn bản có điểm số vượt qua ngưỡng nhất định (threshold) sẽ được chọn làm ngữ cảnh để trả lời câu hỏi truy vấn. Ngưỡng này có thể được điều chỉnh tùy theo yêu cầu của hệ thống hoặc mức độ chính xác mong muốn. Việc xếp hạng lại kết quả tìm kiếm giúp đảm bảo rằng chỉ những

đoạn văn bản có liên quan cao nhất được đưa vào quá trình xử lý tiếp theo, từ đó nâng cao hiệu quả và chất lượng của các câu trả lời.

3.3.3 Cải thiện vận hành và lựa chọn mô hình ngôn ngữ lớn

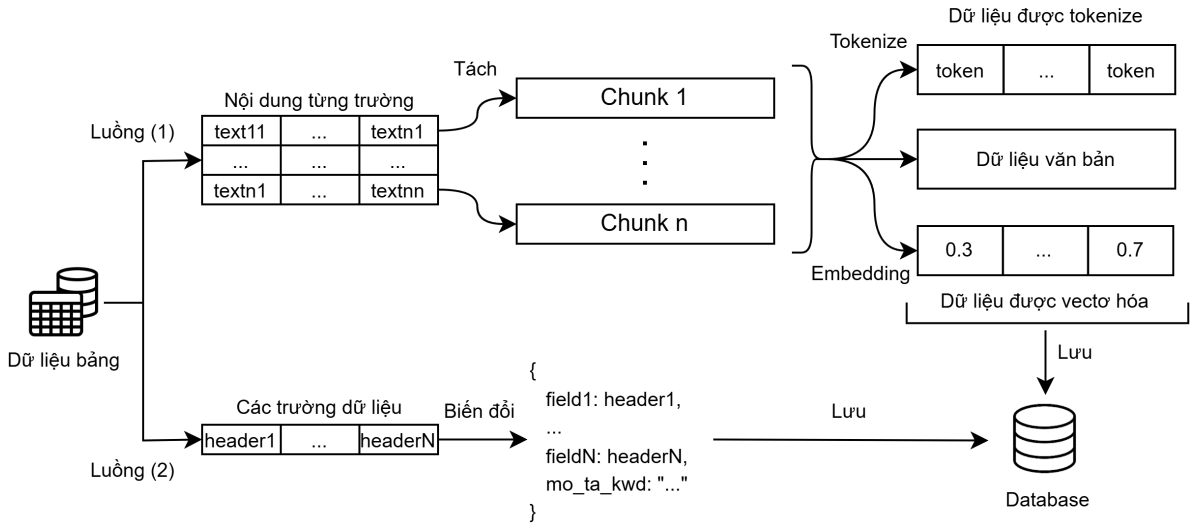
Để đảm bảo hệ thống hoạt động ổn định, đáp ứng nhu cầu đa dạng và thay đổi nhanh chóng từ người dùng, việc cải thiện khả năng vận hành và lựa chọn mô hình ngôn ngữ lớn là yếu tố then chốt. Giải pháp tập trung vào hai mục tiêu chính: tối ưu hóa sự linh hoạt trong việc tích hợp mô hình và giảm thiểu chi phí vận hành mà vẫn đảm bảo hiệu suất cao.

Trước tiên, hệ thống được thiết kế để dễ dàng tích hợp các mô hình ngôn ngữ lớn (LLMs) từ nhiều nhà cung cấp khác nhau. Người quản trị có thể cấu hình hệ thống bằng cách cung cấp thông tin cơ bản như API key và Base URL từ các nền tảng phổ biến như OpenAI, TogetherAI, Gemini, v.v. Điều này cho phép hệ thống chuyển đổi linh hoạt giữa các mô hình mà không cần thay đổi cấu trúc hệ thống, hỗ trợ tối ưu việc lựa chọn mô hình phù hợp với yêu cầu cụ thể của từng lĩnh vực.

Đối với các tổ chức có nhu cầu bảo mật cao hoặc mong muốn tiết kiệm chi phí vận hành, hệ thống hỗ trợ tích hợp các mô hình ngôn ngữ lớn chạy trên máy chủ cục bộ. Các thư viện như vLLM, SGLaM, llama.cpp được sử dụng để triển khai các mô hình trên phần cứng nội bộ, giảm sự phụ thuộc vào dịch vụ đám mây. Cách tiếp cận này không chỉ giảm đáng kể chi phí mà còn đảm bảo quyền kiểm soát và bảo mật dữ liệu tốt hơn.

Ngoài ra, để nâng cao hiệu suất của các mô hình ngôn ngữ trong các ngữ cảnh hoặc lĩnh vực cụ thể, hệ thống có thể áp dụng các kỹ thuật tinh chỉnh (fine-tuning). Điều này cho phép mô hình học thêm các đặc điểm riêng biệt của tiếng Việt hoặc các lĩnh vực chuyên môn như tài chính, y tế, giáo dục, v.v. Quá trình tinh chỉnh được thực hiện trên các bộ dữ liệu có chất lượng cao, đảm bảo mô hình không chỉ hiểu ngữ nghĩa tổng quát mà còn phản hồi chính xác và phù hợp hơn với ngữ cảnh.

Hơn nữa, hệ thống áp dụng kỹ thuật nén mô hình (quantization) để giảm kích thước mô hình và tối ưu hóa hiệu quả sử dụng tài nguyên. Kỹ thuật này không chỉ giúp giảm tải cho phần cứng mà còn cải thiện tốc độ xử lý, đặc biệt hữu ích trong các ứng dụng yêu cầu phản hồi thời gian thực hoặc vận hành trên các thiết bị có giới hạn tài nguyên như máy chủ nhỏ hoặc edge devices.



Hình 3.8: Quá trình xử lý dữ liệu bảng số liệu

Bảng 3.1: Ví dụ bảng chỉ số sản xuất công nghiệp

Tỉnh	**2019**	**2020**	**2021**
An Giang	109,9	103,3	103,1
Bình Dương	109,0	106,1	103,0
TP.Hồ Chí Minh	107,3	95,4	95,4

Tổng thể, việc linh hoạt trong tích hợp, khả năng tùy chỉnh, và tối ưu hóa mô hình giúp hệ thống không chỉ đáp ứng các yêu cầu hiện tại mà còn sẵn sàng mở rộng và thích nghi với các nhu cầu mới trong tương lai, đảm bảo vận hành hiệu quả và tối ưu chi phí.

3.3.4 Xử lý dữ liệu bảng số liệu

Để có thể xử lý một cách phù hợp bảng số liệu, khóa luận này đã đề xuất hai giai đoạn chính: (1) Indexing dữ liệu dạng bảng vào cơ sở trí thức sao cho phù hợp và (2) Truy vấn bảng bằng SQL sao cho hợp lệ.

Quá trình xử lý dữ liệu bảng số liệu

Để có thể xử lý một cách phù hợp bảng số liệu, khóa luận này đã đề xuất hai giai đoạn chính: (1) Indexing dữ liệu dạng bảng vào cơ sở trí thức sao cho phù hợp và

(2) Truy vấn bảng bằng SQL sao cho hợp lệ. Dữ liệu bảng số liệu thường chứa các thông tin có cấu trúc rõ ràng nhưng đa dạng về định dạng, đòi hỏi một quy trình xử lý chuyên biệt để đảm bảo hệ thống có thể truy xuất thông tin một cách chính xác và hiệu quả. Như minh họa trong Hình 3.8, quá trình xử lý dữ liệu bảng số liệu được thiết kế thành hai luồng chính, tập trung vào xử lý các dòng dữ liệu và phân tích các trường dữ liệu, nhằm tối ưu hóa khả năng lưu trữ và truy vấn.

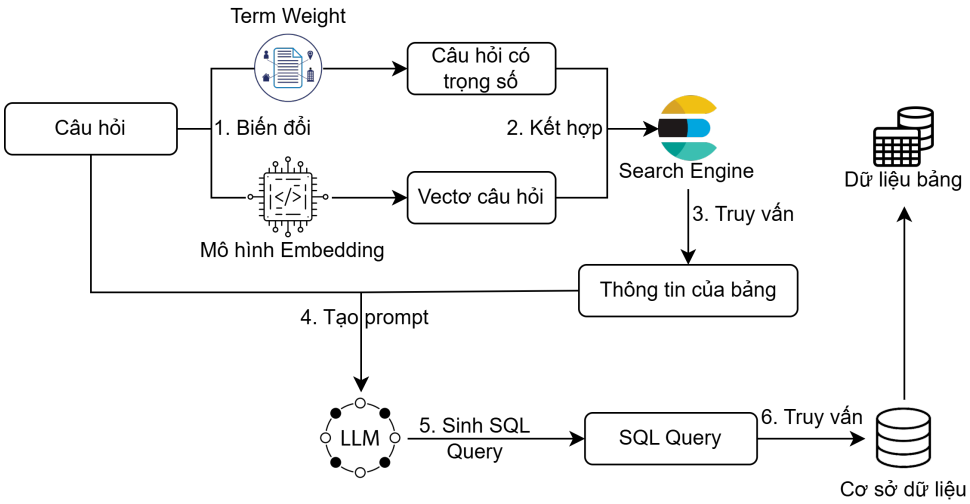
Đầu tiên, với các dòng dữ liệu, mỗi dòng được nối thành một đoạn văn bản liền mạch, đóng vai trò tương tự như một chunk. Các đoạn này sau đó được xử lý theo cách tương tự như đã mô tả trong Mục 3.3.2, bao gồm chuẩn hóa, tạo các phiên bản bổ sung, và chuyển đổi thành các vectơ số hóa để hỗ trợ tìm kiếm toàn văn và tìm kiếm ngữ nghĩa. Việc này giúp hệ thống tích hợp hiệu quả dữ liệu bảng số liệu vào quy trình tìm kiếm chung, tăng cường khả năng truy xuất với các câu hỏi liên quan đến nội dung trong từng dòng của bảng.

Đối với các trường dữ liệu (header), hệ thống tách riêng phần tiêu đề của bảng và phân tích từng cột để xác định loại dữ liệu mà nó chứa. Ví dụ, trong Bảng 3.1 có các trường như "Tỉnh", "2019", "2020", và "2021" thì trường "Tỉnh" sẽ được xác định là dữ liệu dạng văn bản (text), trong khi các trường "2019", "2020", và "2021" sẽ được xác định là dữ liệu dạng số (float). Kết quả của quá trình phân tích này được lưu trữ dưới dạng cấu trúc JSON như sau:

```
{
    "tinh_tks": "Tỉnh",
    "2019_float": "2019",
    "2020_float": "2020",
    "2021_float": "2021",
}
```

Trong đó, hậu tố "_tks" biểu thị trường có dữ liệu kiểu văn bản (text), và "_float" biểu thị trường có dữ liệu kiểu số (float). Cách lưu trữ này giúp hệ thống nhận diện rõ ràng loại dữ liệu của từng trường, hỗ trợ hiệu quả cho các truy vấn SQL. Những trường tiêu đề này đóng vai trò làm chỉ mục chính trong cơ sở dữ liệu, đảm bảo việc truy xuất dữ liệu theo các tiêu chí cụ thể được thực hiện một cách nhanh chóng và chính xác.

Sau khi hoàn thành hai luồng xử lý, toàn bộ dữ liệu được lưu trữ trong cơ sở dữ



Hình 3.9: Luồng hoạt động truy xuất dữ liệu dạng bảng

liệu theo cách tổ chức phù hợp với mục tiêu sử dụng. Các dòng dữ liệu được lưu dưới dạng các chunk để hỗ trợ truy vấn dựa trên văn bản, trong khi các trường tiêu đề được sử dụng như chỉ mục chính cho các truy vấn SQL. Cách tiếp cận này không chỉ đảm bảo hệ thống có thể xử lý hiệu quả dữ liệu bảng số liệu mà còn tăng cường khả năng trả lời các câu hỏi liên quan đến số liệu, so sánh, hoặc các truy vấn mang tính chất phân tích cao hơn.

Luồng truy vấn dữ liệu dạng bảng

Hình 3.9 minh họa chi tiết quy trình hoạt động của hệ thống trong việc truy xuất dữ liệu dạng bảng. Khi người dùng nhập câu hỏi vào hệ thống, một loạt các bước xử lý được thực hiện để đảm bảo rằng thông tin trả về không chỉ chính xác mà còn đáp ứng ngữ cảnh cụ thể của câu hỏi.

Đầu tiên, câu hỏi của người dùng được xử lý thành hai phiên bản song song (1). Phiên bản đầu tiên trải qua quá trình tokenization và đánh trọng số (term weighting), sử dụng các phương pháp như IDF (Inverse Document Frequency) để xác định các từ và cụm từ quan trọng. Việc này cho phép hệ thống tập trung vào các yếu tố cốt lõi, phản ánh đúng ý định của người dùng.

Phiên bản thứ hai của câu hỏi được chuyển đổi thành vector thông qua một mô hình nhúng ngữ nghĩa (embedding model), giúp biểu diễn câu hỏi dưới dạng tọa độ trong không gian ngữ nghĩa. Quá trình này không chỉ dựa vào từ vựng mà còn

xem xét ngữ cảnh và ý nghĩa tổng thể, cho phép hệ thống hiểu câu hỏi sâu hơn.

Hai phiên bản này sau đó được kết hợp và sử dụng làm truy vấn để tìm kiếm trong Elasticsearch (2). Công cụ này xác định các bảng dữ liệu có thông tin mô tả có mức độ tương đồng cao nhất với truy vấn đã tạo. Nhờ đó, hệ thống không chỉ tìm thấy các bảng phù hợp mà còn định vị các nguồn dữ liệu có tiềm năng cao nhất để giải đáp câu hỏi (3).

Khi bảng liên quan được xác định, thông tin về bảng, bao gồm các trường dữ liệu, sẽ được sử dụng để xây dựng một prompt đặc biệt (4). Prompt này, kết hợp với câu hỏi gốc của người dùng, được đưa vào một mô hình ngôn ngữ lớn (LLM), nơi câu truy vấn SQL phù hợp được tự động sinh ra (5). Câu lệnh SQL sau đó được thực thi để trích xuất dữ liệu từ cơ sở dữ liệu, và kết quả cuối cùng được trả về dưới dạng bảng cụ thể, trực quan cho người dùng (6).

Quy trình này đảm bảo rằng hệ thống không chỉ tự động hóa toàn bộ quá trình tạo truy vấn mà còn tối ưu hóa việc khai thác dữ liệu, mang lại câu trả lời chính xác và có giá trị.

Chương 4

Thực nghiệm và đánh giá

Chương này sẽ tập trung trình bày về quá trình thực nghiệm và đánh giá giải pháp Questin dựa trên thiết kế đã mô tả chi tiết trong phần trước. Phần thực nghiệm sẽ đi sâu vào việc áp dụng Questin để đánh giá hiệu quả và tính chính xác của phương pháp này, cùng với đó là đi sâu vào đánh giá từng thành phần trong Questin về sự đóng góp về hiệu suất tổng thể.

4.1 Dữ liệu

Để phát triển và đánh giá mô hình Vistral, khóa luận đã xây dựng một bộ dữ liệu phong phú và đa dạng, đảm bảo phục vụ hiệu quả cho cả quá trình huấn luyện và kiểm thử hệ thống. Bộ dữ liệu bao gồm 3.150 câu hỏi từ nhiều lĩnh vực dành cho quá trình tinh chỉnh mô hình ngôn ngữ lớn Vistral, được thiết kế nhằm tối ưu hóa khả năng hiểu ngữ cảnh và xử lý câu hỏi của mô hình. Đặc biệt, bộ kiểm thử bao gồm 156 câu hỏi đại diện, được chọn lọc kỹ lưỡng từ ba lĩnh vực chính: pháp luật, kinh tế, và tư vấn tuyển sinh. Các câu hỏi này được phân chia thành ba mức độ truy vấn, tích hợp và suy luận để đánh giá khả năng của hệ thống trong việc xử lý các tình huống khác nhau (Bảng 4.1). Song song với bộ câu hỏi, hàng trăm nghìn văn bản từ nhiều nguồn tin cậy cũng được sử dụng làm ngữ liệu nền, đảm bảo rằng hệ thống không chỉ trả lời trực tiếp mà còn có thể đưa ra phản hồi sâu sắc, có tính thuyết phục cao. Cụ thể, quy trình xây dựng bộ dữ liệu cho phương pháp Questin được thực hiện theo ba bước chính sau đây:

Bảng 4.1: Bộ câu hỏi đánh giá hệ thống Questin

#	**Lĩnh vực**	**Câu hỏi**			
		Truy vấn	**Tích hợp**	**Suy luận**	**Tổng**
1	Luật pháp	30	10	11	**51**
2	Kinh tế	24	13	13	**50**
3	Tư vấn tuyển sinh	29	14	12	**55**

- **Bước 1 - Thu thập dữ liệu**:
  - **Thu thập tài liệu chuyên môn**: Quy trình xây dựng bộ dữ liệu bắt đầu từ việc thu thập tài liệu chuyên ngành từ các nguồn uy tín và đáng tin cậy. Cụ thể, các văn bản pháp luật được lấy từ Trung tâm Dữ liệu Quốc gia và các nghị định, thông tư mới nhất. Đối với kinh tế, dữ liệu đến từ các báo cáo thương mại của Metric, trang web VnEconomy và các số liệu thống kê từ Tổng cục Thống kê Việt Nam. Trong lĩnh vực tư vấn tuyển sinh, nguồn dữ liệu chủ yếu là thông tin cập nhật từ các trang tuyển sinh chính thức của các trường đại học, chẳng hạn như Đại học Công nghệ. Những tài liệu này đều được chọn lọc kỹ lưỡng nhằm đảm bảo độ tin cậy và phù hợp với ngữ cảnh thực tế tại Việt Nam.
  - **Thu thập câu hỏi chuyên ngành**: Ngoài tài liệu, bộ câu hỏi cũng được xây dựng từ các nguồn đáng tin cậy. Trong lĩnh vực pháp luật, nền tảng sẽ thu thập các câu hỏi từ các diễn đàn nơi các chuyên gia pháp lý trực tiếp giải đáp. Với kinh tế và tư vấn tuyển sinh, do số lượng câu hỏi thực tế còn hạn chế, nền tảng dựa vào tài liệu chuyên môn và sử dụng GPT-4o để sinh ra các câu hỏi, đảm bảo sự đa dạng và độ phức tạp cần thiết.
- **Bước 2 - Gán nhãn dữ liệu**: Sau khi lấy và tạo ra các bộ câu hỏi cho việc kiểm thử hệ thống, các câu hỏi này cần được phân thành ba mức độ khác nhau để đánh giá được độ phức tạp của hệ thống ở từng cấp độ. Như trong Bảng 4.1, bộ câu hỏi sẽ có ba mức độ chính:
  - **Truy vấn**: Câu hỏi yêu cầu hệ thống chỉ cần tìm kiếm và truy xuất chính xác đoạn văn bản liên quan để trả lời.
  - **Tổng hợp**: Đòi hỏi hệ thống kết hợp thông tin từ nhiều nguồn để đưa ra câu trả lời đầy đủ. Ví dụ, trong lĩnh vực pháp luật, một câu hỏi có thể yêu cầu tham chiếu đến nhiều bộ luật hoặc nghị định; trong lĩnh vực kinh tế, có thể cần tổng hợp dữ liệu từ nhiều khu vực địa lý khác nhau để đưa ra câu trả lời chính xác.
  - **Suy luận**: Đòi hỏi hệ thống phải tư duy logic hoặc kết hợp nhiều yếu tố để suy ra câu trả lời. Ví dụ, trong tuyển sinh, hệ thống phải đánh giá liệu các tiêu chí nhập học có phù hợp; trong pháp luật, phải xác định tình huống có vi phạm luật hay không.

- **Bước 3 - Tiền xử lý dữ liệu**:
  - **Dữ liệu văn bản**: Đối với dữ liệu văn bản, quá trình làm sạch bao gồm việc loại bỏ các ký tự đặc biệt, dấu câu thừa và những yếu tố không cần thiết khác có thể làm giảm hiệu suất của mô hình. Sau khi làm sạch, văn bản được phân tách thành các đơn vị nhỏ hơn, chẳng hạn như từ hoặc câu, thông qua kỹ thuật tokenization. Bước này giúp hệ thống dễ dàng gán trọng số cho các từ hoặc cụm từ quan trọng, hỗ trợ quá trình truy vấn và xử lý thông tin hiệu quả hơn. Việc xử lý văn bản ở cấp độ chi tiết này cũng đảm bảo rằng dữ liệu đầu vào được tối ưu hóa để mô hình có thể học hỏi một cách chính xác và toàn diện.
  - **Câu hỏi kiểm thử**: Các câu trả lời do mô hình GPT-4o tạo ra được kiểm duyệt thủ công để loại bỏ những lỗi không phù hợp, đảm bảo tính chính xác và chất lượng của bộ kiểm thử.

4.2 Quy trình thực nghiệm

Để đánh giá một cách toàn diện và chi tiết hiệu quả của nền tảng Questin, khóa luận đã triển khai một loạt các thử nghiệm nhằm so sánh và phân tích các yếu tố quan trọng ảnh hưởng đến hiệu suất hoạt động của hệ thống. Các thử nghiệm tập trung vào những khía cạnh then chốt, bao gồm tác động của mô hình Vistral, hiệu quả của quá trình tinh chỉnh mô hình, và vai trò của công nghệ RAG trong việc khai thác dữ liệu ngoài để nâng cao chất lượng trả lời.

Các thử nghiệm được thực hiện trên một hệ thống máy chủ sử dụng hệ điều hành Linux, với cấu hình phần cứng mạnh mẽ gồm GPU A4000 và bộ nhớ 16GB RAM. Quá trình đã tối ưu hóa các siêu tham số quan trọng của mô hình, bao gồm dropout, batch_size, số lượng epoch, và tốc độ học (learning rate), nhằm đạt được hiệu suất tốt nhất trên tập dữ liệu huấn luyện. Việc tinh chỉnh này không chỉ giúp cải thiện khả năng trả lời của mô hình mà còn đảm bảo tính ổn định khi áp dụng trên các bộ dữ liệu khác nhau.

Quá trình thực nghiệm trong khóa luận được triển khai theo hai kịch bản chính, nhằm đánh giá hiệu quả của việc tinh chỉnh mô hình và ứng dụng hệ thống Questin trong thực tế. Trong kịch bản đầu tiên, mục tiêu là kiểm tra tác động của việc tinh chỉnh mô hình Vistral. Tập dữ liệu được chia thành hai phần rõ ràng:

phần đầu gồm 3.150 câu hỏi thuộc nhiều lĩnh vực khác nhau, được sử dụng để thực hiện quá trình tinh chỉnh mô hình; phần còn lại bao gồm 100 câu hỏi cơ bản, nhằm kiểm tra khả năng cải thiện của mô hình sau tinh chỉnh. Các chỉ số đánh giá tập trung vào độ chính xác ngữ nghĩa và khả năng diễn đạt, qua đó xác định mức độ hiệu quả của việc tinh chỉnh trong việc nâng cao chất lượng trả lời.

Kịch bản thứ hai được thiết kế để đánh giá khả năng ứng dụng của hệ thống Questin trong các tình huống thực tế. Bộ dữ liệu thử nghiệm bao gồm 156 câu hỏi thuộc ba lĩnh vực chính: luật pháp, kinh tế, và tư vấn tuyển sinh. Những câu hỏi này được lựa chọn với tiêu chí đảm bảo sự đa dạng về nội dung cũng như mức độ phức tạp, nhằm cung cấp một đánh giá toàn diện về năng lực truy xuất thông tin và chất lượng trả lời của hệ thống. Về quy trình thực nghiệm, các điều kiện được thiết lập để đảm bảo tính khách quan và công bằng khi so sánh Questin với các nền tảng Langchain, LlamaIndex, và Ragflow. Cụ thể, các nền tảng được cấu hình sử dụng chung các thành phần sau:

- **Mô hình embedding**: `keepitreal/vietnamese-sbert`, một mô hình chuyển đổi câu đặc biệt tối ưu cho Tiếng Việt.
- **Mô hình tạo sinh**: `gpt-4o-mini`, được lựa chọn để đảm bảo tính đồng nhất và hiệu quả trong việc sinh câu trả lời.
- **Thuật toán tìm kiếm**: Hybrid Search, kết hợp giữa tìm kiếm dựa trên từ khóa và tìm kiếm ngữ nghĩa để cải thiện độ chính xác.
- **Chunking dữ liệu**: Toàn bộ văn bản được chia thành các đoạn (chunk) với giới hạn tối đa là 256 token mỗi đoạn nhằm tối ưu hóa khả năng xử lý của các mô hình.

Tất cả các nền tảng được thử nghiệm trên cùng một bộ dữ liệu và được chuẩn hóa với các cài đặt nêu trên. Việc thống nhất này nhằm loại bỏ các yếu tố gây nhiễu không liên quan, từ đó cung cấp một đánh giá chính xác về năng lực của từng hệ thống trong việc xử lý dữ liệu và đưa ra câu trả lời phù hợp.

Quy trình thực nghiệm được thiết kế để mang lại cái nhìn sâu sắc về hiệu suất của hệ thống Questin. Thử nghiệm không chỉ đo lường chất lượng trả lời mà còn phân tích khả năng tích hợp công nghệ RAG vào quy trình xử lý. Công nghệ này

giúp tận dụng các nguồn dữ liệu từ bên ngoài, từ đó cải thiện độ chính xác và tính phù hợp của câu trả lời.

Kết quả từ các thử nghiệm không chỉ cung cấp bằng chứng về hiệu quả vượt trội của QUESTIN so với các mô hình truyền thống mà còn xác định những điểm mạnh và hạn chế của hệ thống. Điều này tạo cơ sở để tiếp tục phát triển và hoàn thiện, đảm bảo rằng hệ thống không chỉ đáp ứng mà còn vượt qua kỳ vọng của người dùng trong nhiều ngữ cảnh khác nhau.

4.3 Độ đo đánh giá

Để đánh giá chính xác hiệu quả của nền tảng QUESTIN trong việc hỗ trợ tư vấn đa lĩnh vực bằng nghiệp vụ ảo, khóa luận thiết kế hai kịch bản thử nghiệm cụ thể:

- **Kịch bản 1**: Đánh giá hiệu suất của mô hình Vistral sau quá trình tinh chỉnh (finetuning), nhằm xác định mức độ cải thiện sau khi điều chỉnh các tham số.
- **Kịch bản 2**: Đánh giá khả năng truy xuất thông tin và chất lượng câu trả lời mà hệ thống cung cấp.

4.3.1 Độ đo đánh giá mô hình

Để đánh giá mô hình sau khi tinh chỉnh, khóa luận đã sử dụng hai độ đo chính là *Bleu Score* và *Semantic Similarity* [56].

Bleu Score

Bleu Score là độ đo được sử dụng để đánh giá mức độ tương đồng giữa câu trả lời được mô hình tạo ra và câu trả lời mục tiêu. Công thức được tính như sau:

$$BLEU = \underbrace{min(1, e^{1-\frac{reference-length}{output-length}})}_{brevity\ penalty} \cdot \underbrace{\left(\prod_{i=1}^{4} precision_i\right)^{1/4}}_{n-gram\ overlap} \tag{4.1}$$

Trong đó:

- $precision_i$ đại diện cho độ chính xác của n-gram và được xác định bởi:

$$precision_i = \frac{\sum_{n \in \text{Candidate}} \min(m^n_{cand}, m^n_{ref})}{\sum_{n \in \text{Candidate}} m^n_{cand}} \tag{4.2}$$

- m^n_{cand} là số lượng n-gram trong câu trả lời của mô hình trùng khớp với câu trả lời tham chiếu.

- m^n_{ref} là số lượng n-gram tương ứng trong câu trả lời tham chiếu.

Công thức trên bao gồm hai thành phần chính:

- **Brevity Penalty**: Giảm mức ưu tiên cho các câu trả lời quá ngắn, đảm bảo sự cân bằng giữa độ dài và tính chính xác của đầu ra.

- **N-gram Overlap**: Đánh giá mức độ trùng khớp giữa các n-gram từ câu trả lời của mô hình và câu tham chiếu. Thành phần này không chỉ kiểm tra tính phù hợp với các unigram mà còn đánh giá sự lưu loát qua các bigram, trigram, và four-gram.

Semantic Similarity

Khác với *Bleu Score*, *Semantic Similarity* được sử dụng để đánh giá mức độ tương đồng về mặt ý nghĩa giữa câu trả lời từ mô hình và câu trả lời mong đợi. Để tính toán, đầu tiên câu trả lời được biểu diễn dưới dạng vector thông qua mô hình *sentence-transformer*. Sau đó, độ tương đồng giữa hai vector được tính bằng *cosine similarity* theo công thức:

$$similarity(A, B) = \cos(\theta) = \frac{A \cdot B}{|A||B|} \tag{4.3}$$

Trong đó:

- θ là góc giữa hai vector biểu diễn.

- $A \cdot B$ là tích vô hướng của hai vector, được tính bởi $A \cdot B = \sum_{i=1}^{n} A_i B_i$.

- $|A|$ là độ dài của vector A, được tính theo công thức $|A| = \sqrt{\sum_{i=1}^{n} A_i^2}$.

4.3.2 Độ đo đánh giá hiệu suất hệ thống

Để đánh giá chính xác hiệu quả của hệ thống QUESTIN, khóa luận đã áp dụng bốn thước đo chính: *Faithfulness*, *Answer Relevancy*, *Context Precision*, và *Context Recall* [57]. Những thước đo này giúp đánh giá toàn diện chất lượng và tính chính xác của hệ thống trong việc cung cấp thông tin và hỗ trợ người dùng.

Faithfulness

Tiêu chí *Faithfulness* đo lường mức độ nhất quán giữa câu trả lời do hệ thống sinh ra và ngữ cảnh đã được truy xuất. Chỉ số này phản ánh khả năng của hệ thống trong việc đưa ra câu trả lời dựa trên thông tin chính xác và phù hợp với ngữ cảnh. Giá trị của độ đo nằm trong khoảng từ 0 đến 1, với giá trị càng cao thể hiện tính chính xác và độ tin cậy của câu trả lời càng tốt.

$$Faithfulness = \frac{\text{Number of claims in the answer can be inferred from given context}}{\text{Total number of claims in the generated answer}} \tag{4.4}$$

Để tính toán *Faithfulness Score*, trước tiên nền tảng tiến hành trích xuất một tập các nhận định (*claims*) từ câu trả lời mà hệ thống đã tạo ra. Sau đó, từng nhận định sẽ được đối chiếu với ngữ cảnh đã truy xuất để kiểm tra xem liệu nó có thể được suy ra từ ngữ cảnh đó hay không. Quy trình này giúp đánh giá tính xác thực và độ chính xác của câu trả lời dựa trên thông tin có sẵn.

Answer Relevancy

Độ đo *Answer Relevancy* tập trung vào việc đánh giá mức độ phù hợp của câu trả lời mà hệ thống sinh ra so với câu hỏi của người dùng. Điểm số thấp cho thấy câu trả lời không đầy đủ hoặc chứa thông tin không cần thiết, trong khi điểm số cao phản ánh mức độ liên quan tốt và câu trả lời chính xác, tập trung vào yêu cầu của người dùng. Chỉ số này được tính dựa trên mối liên hệ giữa câu hỏi, ngữ cảnh truy vấn và câu trả lời.

Answer Relevancy được tính toán bằng cách sử dụng độ tương đồng cosin trung bình giữa câu hỏi ban đầu và một số câu hỏi được sinh ra từ mô hình. Công thức

được biểu diễn như sau:

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N} cos(E_{g_i}, E_o) = \frac{1}{N}\sum_{i=1}^{N} \frac{E_{g_i} \cdot E_o}{\|E_{g_i}\|\|E_o\|} \tag{4.5}$$

Trong đó:

- E_{g_i} là vector embedding của câu hỏi được sinh ra thứ i.
- E_o là vector embedding của câu hỏi ban đầu.
- N là số lượng câu hỏi được sinh ra.

Quá trình tính toán điểm *Answer Relevancy* bao gồm việc sử dụng mô hình ngôn ngữ lớn (LLM) để tạo ra các câu hỏi dựa trên câu trả lời do hệ thống cung cấp. Sau đó, độ tương tự cosin trung bình giữa các câu hỏi này và câu hỏi ban đầu dùng để xác định mức độ phù hợp của câu trả lời với yêu cầu gốc của người dùng.

Context Precision

Context Precision là thước đo dùng để đánh giá mức độ chính xác trong việc sắp xếp các mục liên quan đến câu trả lời chính xác (ground truth) trong ngữ cảnh được truy xuất. Tiêu chí này kiểm tra xem các thông tin cần thiết cho câu trả lời có được hệ thống xếp hạng cao trong kết quả trả về hay không. Giá trị của *Context Precision* nằm trong khoảng từ 0 đến 1, trong đó điểm số càng cao cho thấy hệ thống có khả năng truy xuất các thông tin quan trọng một cách chính xác hơn. Công thức tính *Context Precision* như sau:

$$\text{Context Precision@}K = \frac{\sum_{k=1}^{K}(Precision@k \times v_k)}{\text{Total number of relevant items in the top K results}} \tag{4.6}$$

$$Precision@k = \frac{\text{true positives@}k}{(\text{true positives@}k + \text{false positives@}k)} \tag{4.7}$$

Trong đó:

- K là số lượng mục (chunks) trong ngữ cảnh được xếp hạng.
- v_k là chỉ số liên quan tại vị trí thứ k (với giá trị 1 nếu mục đó liên quan và 0 nếu không).

Công thức này tính toán độ chính xác bằng cách xem xét các mục quan trọng có được hệ thống truy xuất và xếp hạng cao trong danh sách kết quả không. *Precision@k* đo lường tỷ lệ các mục chính xác trong top k kết quả, với "true positives" là các mục liên quan và "false positives" là các mục không liên quan.

Context Recall

Context Recall đo lường mức độ ngữ cảnh được truy xuất có thể bao quát đầy đủ các thông tin liên quan để trả lời chính xác câu hỏi (ground truth). Chỉ số này phản ánh khả năng của hệ thống trong việc truy xuất ngữ cảnh phù hợp với câu trả lời chính xác, với giá trị nằm trong khoảng từ 0 đến 1. Điểm số càng cao thể hiện khả năng truy xuất thông tin toàn diện và hiệu suất tốt hơn.

Để tính toán *Context Recall*, mỗi thông tin trong câu trả lời chính xác (ground truth) sẽ được phân tích và kiểm tra xem liệu nó có thể được suy ra từ ngữ cảnh được hệ thống truy xuất hay không. Công thức tính *Context Recall* như sau:

$$\text{context recall} = \frac{\text{GT claims that can be attributed to context}}{\text{Number of claims in GT}} \tag{4.8}$$

Chỉ số này cho biết mức độ đầy đủ của ngữ cảnh trong việc cung cấp thông tin cần thiết cho câu trả lời chính xác. Một hệ thống với *Context Recall* cao đảm bảo rằng mọi yếu tố quan trọng trong câu trả lời đều có thể được truy xuất từ ngữ cảnh.

4.4 Kết quả thực nghiệm

4.4.1 Ảnh hưởng của việc tinh chỉnh mô hình

Trong quá trình tinh chỉnh mô hình ngôn ngữ Vistral-7b, khóa luận đã áp dụng các kỹ thuật và cấu hình đặc biệt để đảm bảo mô hình có thể đưa ra các câu trả lời một cách chính xác, mạch lạc và chứa đựng thông tin đáng tin cậy. Mục tiêu của việc này là nâng cao khả năng xử lý ngôn ngữ của mô hình, giúp nó hiểu sâu hơn và cải thiện chất lượng sinh văn bản trong nhiều ngữ cảnh khác nhau.

Tập dữ liệu dùng để tinh chỉnh gồm 3.150 câu hỏi từ các lĩnh vực đa dạng, được lựa chọn cẩn thận nhằm tránh hiện tượng overfitting. Điều này giúp mô hình không chỉ thành thạo trong một lĩnh vực cụ thể mà còn có khả năng chuyển giao kiến thức giữa các lĩnh vực khác nhau. Các tham số được điều chỉnh bao gồm:

- $num\_train\_epochs = 1$: Chỉ thực hiện một epoch để giảm thời gian huấn luyện và giữ cho mô hình linh hoạt.
- $per\_device\_train\_batch\_size = 4$: Batch size là 4 để đảm bảo sử dụng tốt nguồn tài nguyên và bộ nhớ GPU.
- $gradient\_accumulation\_steps = 4$: Tích hợp gradient qua 4 bước để giảm áp lực về bộ nhớ.
- $optim = paged\_adamw\_8bit$: Sử dụng trình tối ưu hóa "paged_adamw_8bit" để tối ưu hóa quá trình học.

Sau khi tinh chỉnh, nền tảng đã đánh giá hiệu suất của mô hình finetuned-Vistral-7b so với phiên bản gốc bằng cách sử dụng chỉ số *Bleu Score* và *Semantic Similarity*.

Bảng 4.2: Đánh giá hiệu suất của mô hình sau khi tinh chỉnh

Mô hình	**Bleu Score**	**Semantic Similarity**
Vistral-7b	0.04	0.71
finetuned-Vistral-7b	0.13	0.78

Kết quả trong Bảng 4.2 cho thấy quá trình tinh chỉnh đã mang lại sự cải thiện rõ rệt cho mô hình. Cụ thể:

- *Bleu Score* của mô hình finetuned-Vistral-7b tăng từ 0.04 lên 0.13. Đây là sự cải thiện đáng kể, cho thấy khả năng sinh văn bản của mô hình đã trở nên sát với câu trả lời chính xác hơn. *Bleu Score* phản ánh mức độ tương đồng giữa câu trả lời của mô hình và câu trả lời chuẩn, do đó, sự tăng trưởng này đồng nghĩa với việc mô hình đã học được cách sinh ra các câu trả lời có tính liên kết và chính xác cao hơn sau khi tinh chỉnh.
- *Semantic Semilarity* tăng từ 0.71 lên 0.78, biểu thị khả năng hiểu ngữ nghĩa và tính nhất quán của mô hình đã được cải thiện. Điều này đặc biệt quan trọng vì nó cho thấy mô hình không chỉ sinh ra các câu trả lời đúng về mặt hình thức, mà còn phù hợp về ngữ nghĩa và bối cảnh với câu hỏi được đưa ra.

Nhìn chung, việc tinh chỉnh mô hình với tập dữ liệu đa dạng và cấu hình tối ưu đã giúp cải thiện đáng kể cả về khả năng sinh văn bản chính xác và mức độ phù

hợp về mặt ngữ nghĩa. Điều này minh chứng cho hiệu quả của việc tinh chỉnh và mở ra tiềm năng áp dụng mô hình trong nhiều bối cảnh tư vấn nghiệp vụ thực tế.

4.4.2 Hiệu suất của hệ thống

Để đánh giá tổng thể hiệu suất của hệ thống QUESTIN, khóa luận đã tiến hành các thử nghiệm và thu thập số liệu thông qua ba khía cạnh chính: chất lượng câu trả lời của hệ thống, khả năng truy xuất thông tin và khả năng bao phủ câu hỏi. Kết quả đánh giá được so sánh với các nền tảng RAG tiên tiến khác là LangChain, LlamaIndex và nền tảng Ragflow, và được trình bày trong Bảng 4.3, 4.4 và 4.5.

Khả năng trả lời của hệ thống

Bảng 4.3 trình bày kết quả so sánh khả năng trả lời của các giải pháp RAG, bao gồm Langchain, LlamaIndex, Ragflow và QUESTIN, khi được thử nghiệm trên ba lĩnh vực chính: luật pháp, kinh tế và tư vấn tuyển sinh. Dựa trên hai tiêu chí chính là *Failfulness* và *Answer Relevancy*, kết quả này cho thấy sự khác biệt đáng kể về hiệu suất giữa các giải pháp.

Trong lĩnh vực luật pháp, nơi yêu cầu độ chính xác cao và khả năng cung cấp thông tin phù hợp với ngữ cảnh phức tạp, QUESTIN đạt chỉ số *Failfulness* và *Answer Relevancy* lần lượt là 0.6 và 0.48. Kết quả này nhỉnh hơn so với Langchain (0.55, 0.42) và tương đương với Ragflow (0.6, 0.48), đồng thời cao hơn LlamaIndex (0.56, 0.46). Những chỉ số này cho thấy QUESTIN có khả năng duy trì hiệu suất ổn định, đảm bảo độ chính xác và tính liên quan khi xử lý các câu hỏi trong lĩnh vực pháp lý. Mặc dù không hoàn toàn vượt trội trong mọi tiêu chí, QUESTIN đã chứng minh được tính cạnh tranh và khả năng đáp ứng hiệu quả các yêu cầu chuyên môn của lĩnh vực này.

Đối với lĩnh vực kinh tế, nơi đòi hỏi sự tích hợp thông tin đa nguồn và khả năng phân tích ngữ cảnh chi tiết, QUESTIN đã thể hiện hiệu suất nổi bật với chỉ số *Failfulness* đạt 0.61 và *Answer Relevancy* đạt 0.56, cao nhất trong tất cả các giải pháp được thử nghiệm. Những con số này nhấn mạnh năng lực của QUESTIN trong việc cung cấp các câu trả lời không chỉ chính xác mà còn phù hợp với ngữ cảnh đa chiều của các câu hỏi kinh tế. LlamaIndex có kết quả khả quan với *Failfulness* đạt 0.58 và *Answer Relevancy* đạt 0.48, nhưng vẫn thấp hơn so với QUESTIN.

Bảng 4.3: Đánh giá chất lượng câu trả lời của hệ thống

Lĩnh vực	**Giải pháp**	**Khả năng trả lời của hệ thống**	
		Failfulness	**Answer Relevancy**
Luật pháp	Langchain	0.55	0.42
	LlamaIndex	0.56	0.46
	Ragflow	**0.6**	**0.48**
	Questin	**0.6**	**0.48**
Kinh tế	Langchain	0.51	0.39
	LlamaIndex	0.58	0.48
	Ragflow	0.43	0.4
	Questin	**0.61**	**0.56**
Tư vấn tuyển sinh	Langchain	**0.83**	0.72
	LlamaIndex	0.79	0.69
	Ragflow	0.7	0.61
	Questin	**0.83**	**0.73**

Ragflow và Langchain có hiệu suất hạn chế hơn, đặc biệt về độ chính xác, khi chỉ đạt *Failfulness* lần lượt là 0.43 và 0.51. Những kết quả này khẳng định tiềm năng của Questin trong việc xử lý các yêu cầu thông tin kinh tế phức tạp.

Trong lĩnh vực tư vấn tuyển sinh, nơi tính liên quan và cá nhân hóa của câu trả lời đóng vai trò quan trọng, Questin và Langchain cùng đạt chỉ số *Failfulness* cao nhất là 0.83, cho thấy khả năng cung cấp thông tin chính xác vượt trội. Tuy nhiên, Questin đã đạt chỉ số *Answer Relevancy* cao nhất là 0.73, nhỉnh hơn Langchain (0.72) và vượt qua các giải pháp khác như LlamaIndex (0.69) và Ragflow (0.61). Kết quả này khẳng định rằng Questin không chỉ chính xác mà còn phù hợp với nhu cầu cá nhân hóa thông tin trong lĩnh vực tư vấn tuyển sinh, nơi đòi hỏi các câu trả lời cần sát với ngữ cảnh và yêu cầu cụ thể của người dùng.

Nhìn chung, các kết quả thực nghiệm cho thấy rằng mỗi giải pháp đều có những thế mạnh riêng tùy thuộc vào lĩnh vực và tiêu chí đánh giá. Questin không phải lúc nào cũng vượt trội tuyệt đối, nhưng đã chứng minh khả năng toàn diện và nổi bật trong các lĩnh vực kinh tế và tư vấn tuyển sinh. Trong lĩnh vực luật pháp, hệ thống vẫn duy trì tính cạnh tranh với các giải pháp hàng đầu khác, đồng thời thể hiện tiềm năng phát triển để cải thiện hơn nữa. Các kết quả này cung cấp một nền

Bảng 4.4: Đánh giá khả năng truy xuất của hệ thống

Lĩnh vực	**Giải pháp**	**Khả năng truy xuất của hệ thống**	
		Context Precision	**Context Recall**
Luật pháp	Langchain	0.94	0.69
	LlamaIndex	0.95	0.75
	Ragflow	0.94	0.77
	Questin	**0.96**	**0.79**
Kinh tế	Langchain	0.71	0.73
	LlamaIndex	0.77	0.73
	Ragflow	0.75	0.7
	Questin	**0.78**	**0.79**
Tư vấn tuyển sinh	Langchain	0.81	0.7
	LlamaIndex	0.84	0.74
	Ragflow	0.77	0.63
	Questin	**0.85**	**0.76**

tảng vững chắc để định hướng các ứng dụng trong tương lai, cũng như chỉ ra các lĩnh vực cần tập trung để tiếp tục nâng cao hiệu suất của hệ thống.

Khả năng truy xuất thông tin của hệ thống

Bảng 4.4 trình bày đánh giá khả năng truy xuất thông tin của các giải pháp, bao gồm Langchain, LlamaIndex, Ragflow và QUESTIN, dựa trên hai tiêu chí chính: *Context Precision* (độ chính xác bối cảnh) và *Context Recall* (khả năng tái hiện bối cảnh). Kết quả này phản ánh hiệu suất của các hệ thống trong việc truy xuất và tái hiện các thông tin liên quan đến ngữ cảnh được yêu cầu.

Trong lĩnh vực luật pháp, QUESTIN đạt chỉ số cao nhất với *Context Precision* là 0.96 và *Context Recall* là 0.79. Các giá trị này nhỉnh hơn so với LlamaIndex (0.95, 0.75), Ragflow (0.94, 0.77) và Langchain (0.94, 0.69). Điều này chứng minh rằng QUESTIN không chỉ có khả năng xác định thông tin phù hợp nhất với bối cảnh yêu cầu mà còn tái hiện thông tin một cách đầy đủ hơn so với các giải pháp khác. Dù mức chênh lệch không lớn, kết quả này cho thấy sự tối ưu hóa đáng kể trong mô hình truy xuất của QUESTIN đối với các ngữ cảnh pháp lý phức tạp.

Trong lĩnh vực kinh tế, vốn đòi hỏi khả năng tổng hợp thông tin từ nhiều nguồn và phân tích ngữ cảnh đa chiều, QUESTIN tiếp tục đạt kết quả tốt nhất với *Context Precision* đạt 0.78 và *Context Recall* đạt 0.79. So với các hệ thống khác như LlamaIndex (0.77, 0.73), Ragflow (0.75, 0.70) và Langchain (0.71, 0.73), QUESTIN vượt trội cả về độ chính xác bối cảnh lẫn khả năng tái hiện đầy đủ thông tin. Kết quả này nhấn mạnh tính ưu việt của QUESTIN trong việc xử lý các yêu cầu thông tin kinh tế, đặc biệt khi ngữ cảnh phức tạp và cần mức độ chính xác cao.

Đối với lĩnh vực tư vấn tuyển sinh, QUESTIN tiếp tục thể hiện hiệu suất cao nhất với *Context Precision* đạt 0.85 và *Context Recall* đạt 0.76. Kết quả này cao hơn LlamaIndex (0.84, 0.74), Langchain (0.81, 0.70) và Ragflow (0.77, 0.63). Mặc dù các giải pháp như LlamaIndex cũng đạt hiệu suất khá, QUESTIN đã chứng minh khả năng vượt trội trong việc cung cấp thông tin liên quan và đầy đủ, đáp ứng hiệu quả các yêu cầu cá nhân hóa và tính liên quan cao trong lĩnh vực này.

Nhìn chung, kết quả thực nghiệm trong ba lĩnh vực cho thấy sự khác biệt đáng kể giữa các giải pháp. Mặc dù các hệ thống khác đều có những ưu điểm riêng, QUESTIN đã chứng minh được năng lực nổi bật với khả năng truy xuất thông tin vừa chính xác vừa đầy đủ trong mọi ngữ cảnh được thử nghiệm. Kết quả này cũng phản ánh sự ổn định của QUESTIN trong các tình huống yêu cầu tính phức tạp cao, từ ngữ cảnh pháp lý chặt chẽ đến các vấn đề kinh tế và tư vấn mang tính cá nhân hóa. Tuy nhiên, để đạt được hiệu suất toàn diện hơn, các kết quả này đồng thời gợi mở rằng vẫn còn nhiều không gian để tối ưu hóa và cải thiện hơn nữa các chỉ số đánh giá trong tương lai.

Tuy nhiên, những kết quả đạt được vẫn chưa cho thấy sự vượt trội rõ rệt trong mọi khía cạnh khi so sánh với các giải pháp khác. Ví dụ, trong lĩnh vực luật pháp, QUESTIN tuy dẫn đầu về *Context Precision* với giá trị 0.96 và *Context Recall* 0.79, nhưng sự chênh lệch so với các giải pháp khác như LlamaIndex (0.95, 0.75) và Ragflow (0.94, 0.77) là không quá nhiều. Điều này chỉ ra rằng mặc dù hệ thống thể hiện khả năng truy xuất mạnh mẽ, vẫn có điều kiện để cải thiện hơn nữa khả năng tái hiện đầy đủ bối cảnh trong các trường hợp phức tạp. Một trong những hạn chế có thể nằm ở việc đánh trọng số phụ thuộc nhiều vào các kỹ thuật NLP hiện tại. Mặc dù việc sử dụng NER và POS tagging giúp tăng khả năng nhận diện các yếu tố quan trọng trong văn bản, nhưng các yếu tố ngữ nghĩa phức tạp hoặc ngữ cảnh phi cấu trúc đôi khi chưa được xử lý đầy đủ. Ngoài ra, phương pháp IDF

cũng có thể chưa phản ánh chính xác mối liên quan của các token trong các trường hợp có sự mơ hồ hoặc ngữ cảnh đa chiều. Để cải thiện, QUESTIN có thể mở rộng phương pháp hybrid search bằng cách tích hợp thêm các mô hình học sâu (deep learning) để tăng cường khả năng hiểu ngữ cảnh và xử lý ngữ nghĩa. Cụ thể, việc kết hợp với các mô hình ngôn ngữ lớn (LLMs) có thể giúp giải quyết tốt hơn các trường hợp ngữ cảnh phức tạp hoặc phi cấu trúc. Bên cạnh đó, cần tiếp tục tối ưu hóa cách đánh trọng số bằng cách sử dụng các thuật toán tiên tiến hơn để xử lý các mối quan hệ ngữ nghĩa không chỉ dựa trên tần suất mà còn cả ngữ cảnh rộng hơn và phức tạp hơn.

Khả năng bao phủ câu hỏi

Bảng 4.5 minh họa chi tiết về khả năng bao phủ câu hỏi của các giải pháp khác nhau khi thử nghiệm trên ba lĩnh vực chính: luật pháp, kinh tế, và tư vấn tuyển sinh. Để đánh giá khả năng này, các câu hỏi được chia thành ba nhóm chính: truy vấn (câu hỏi yêu cầu truy xuất thông tin từ dữ liệu), tích hợp (câu hỏi đòi hỏi kết hợp thông tin từ nhiều nguồn), và suy luận (câu hỏi yêu cầu phân tích sâu và áp dụng suy luận). Kết quả phân tích cho thấy rằng hệ thống QUESTIN không chỉ cao hơn về tổng số lượng câu hỏi được bao phủ mà còn thể hiện khả năng xử lý hiệu quả trong từng nhóm câu hỏi.

Trong lĩnh vực luật pháp, QUESTIN đạt tổng cộng 32/51 câu hỏi, ngang bằng với LlamaIndex và Ragflow, và vượt qua Langchain (29/51). Đặc biệt, QUESTIN thể hiện sự xuất sắc ở nhóm câu hỏi tích hợp với tỷ lệ 10/10, bằng với Ragflow nhưng vượt qua các giải pháp khác. Mặc dù vậy, nhóm câu hỏi truy vấn và suy luận vẫn có thể được cải thiện, khi QUESTIN lần lượt đạt 17/30 và 5/11, thấp hơn LlamaIndex (19/30) ở nhóm truy vấn. Điều này chỉ ra rằng hệ thống có thể cần tập trung hơn vào việc nâng cao khả năng xử lý các câu hỏi yêu cầu truy xuất thông tin cụ thể và suy luận chi tiết.

Trong lĩnh vực kinh tế, QUESTIN thể hiện sự vượt trội với tổng số 32/50 câu hỏi được bao phủ, cao hơn đáng kể so với LlamaIndex (26/50), Ragflow (22/50), và Langchain (21/50). Hệ thống đạt kết quả ấn tượng ở nhóm câu hỏi truy vấn với tỷ lệ 22/24, cao nhất trong số các giải pháp. Đối với nhóm câu hỏi tích hợp, QUESTIN đạt 9/13, vượt qua LlamaIndex (7/13) và Ragflow (8/13). Tuy nhiên, ở nhóm câu hỏi suy luận, QUESTIN chỉ đạt 1/13, cho thấy một điểm yếu đáng kể trong việc xử

Bảng 4.5: Đánh giá khả năng bao phủ câu hỏi của hệ thống

Lĩnh vực	**Giải pháp**	**Mức độ câu hỏi**			
		Truy vấn	**Tích hợp**	**Suy luận**	**Tổng**
Luật pháp	Langchain	18/30	6/10	**5/11**	29/51
	LlamaIndex	**19/30**	8/10	**5/11**	**32/51**
	Ragflow	18/30	**10/10**	4/11	**32/51**
	Questin	17/30	**10/10**	**5/11**	**32/51**
Kinh tế	Langchain	16/24	3/13	**2/13**	21/50
	LlamaIndex	19/24	7/13	0/13	26/50
	Ragflow	13/24	8/13	1/13	22/50
	Questin	**22/24**	**9/13**	1/13	**32/50**
Tư vấn tuyển sinh	Langchain	24/29	**14/14**	11/12	**49/55**
	LlamaIndex	22/29	**14/14**	**12/12**	48/55
	Ragflow	21/29	13/14	10/12	44/55
	Questin	**25/29**	**14/14**	10/12	**49/55**

lý các câu hỏi đòi hỏi khả năng suy luận phức tạp trong lĩnh vực này. Kết quả này gợi ý rằng việc tích hợp thêm các kỹ thuật nâng cao để có thể cải thiện đáng kể hiệu suất của hệ thống trong nhóm câu hỏi này.

Lĩnh vực tư vấn tuyển sinh là nơi Questin đạt hiệu suất tốt nhất với tổng số 49/55 câu hỏi, ngang bằng với Langchain và vượt qua LlamaIndex (48/55) cũng như Ragflow (44/55). Đáng chú ý, Questin dẫn đầu ở nhóm câu hỏi truy vấn với 25/29, vượt qua tất cả các giải pháp khác. Ở nhóm câu hỏi tích hợp, hệ thống đạt 14/14, ngang bằng với Langchain và LlamaIndex, cho thấy sự ổn định trong việc xử lý các câu hỏi yêu cầu kết hợp thông tin. Tuy nhiên, ở nhóm câu hỏi suy luận, Questin đạt 10/12, thấp hơn LlamaIndex (12/12) và Langchain (11/12), điều này chỉ ra một cơ hội để cải thiện khả năng phân tích sâu trong lĩnh vực này.

Hệ thống Questin đạt được khả năng bao phủ câu hỏi tốt trong các thử nghiệm nhờ sự cải tiến về khả năng truy xuất và tích hợp thông tin. Đối với nhóm câu hỏi truy vấn và tích hợp, hệ thống tận dụng hiệu quả các thuật toán tìm kiếm ngữ nghĩa như vector search và semantic retrieval để truy xuất chính xác các thông tin liên quan. Điều này giúp hệ thống xử lý tốt các câu hỏi yêu cầu tìm kiếm thông tin cụ thể từ cơ sở dữ liệu lớn. Ngoài ra, quá trình xếp hạng lại (reranking) các kết

quả tìm kiếm đã tăng cường độ chính xác, đảm bảo rằng chỉ những đoạn văn bản liên quan nhất được đưa vào làm ngữ cảnh đầu vào, từ đó cải thiện đáng kể chất lượng của câu trả lời.

Tuy nhiên, đối với nhóm câu hỏi suy luận, khả năng xử lý của hệ thống vẫn còn hạn chế. Các câu hỏi thuộc nhóm này thường đòi hỏi phân tích sâu hoặc suy luận qua nhiều bước logic phức tạp, điều mà các mô hình hiện tại chưa tối ưu hoàn toàn. Hạn chế này có thể xuất phát từ việc nền tảng chưa hiểu rõ câu hỏi phức tạp của người dùng dẫn đến không truy xuất được các đoạn văn bản thực sự liên quan. Kết quả cho thấy rằng, mặc dù Questin có khả năng xử lý một phần các yêu cầu suy luận, nhưng hiệu suất ở nhóm này chưa đạt mức tương đương với các nhóm câu hỏi truy vấn và tích hợp.

Nhằm cải thiện khả năng xử lý các câu hỏi suy luận, các định hướng phát triển tương lai có thể bao gồm việc tích hợp các cơ chế suy luận tri thức (Knowledge Reasoning) để tăng cường khả năng khai thác mối quan hệ logic trong dữ liệu. Đồng thời, việc áp dụng các kiến trúc đa tác nhân (Multi-Agent Systems) cũng có tiềm năng nâng cao hiệu suất bằng cách cho phép các tác nhân chuyên biệt phối hợp xử lý thông tin và suy luận. Những giải pháp này có thể giúp hệ thống đạt được sự cân bằng tốt hơn trong xử lý các nhóm câu hỏi khác nhau, mở rộng khả năng ứng dụng trong nhiều lĩnh vực phức tạp.

Kết luận và hướng phát triển

Trong bối cảnh nhu cầu tư vấn nghiệp vụ ngày càng gia tăng ở các lĩnh vực như tài chính, quản lý nhân sự, kế toán, và công nghệ thông tin, việc tiếp cận các dịch vụ tư vấn chuyên nghiệp vẫn là thách thức lớn với nhiều doanh nghiệp, đặc biệt là các doanh nghiệp nhỏ và vừa, do chi phí cao và hạn chế trong khả năng tiếp cận. Chính vì thế, mục tiêu của khóa luận này là thiết kế một nền tảng tư vấn nghiệp vụ ảo, giúp cung cấp các giải pháp tư vấn chuyên sâu, tiết kiệm chi phí và dễ dàng tiếp cận, hướng tới việc trở thành một công cụ hỗ trợ đắc lực cho doanh nghiệp trong việc ra quyết định.

Hệ thống được phát triển dựa trên sự kết hợp giữa trí tuệ nhân tạo (AI), mô hình ngôn ngữ lớn (LLM), và cơ chế RAG (retrieval-augmented generation). LLM được sử dụng để hiểu ngữ cảnh và phân tích các câu hỏi của người dùng, từ đó dự đoán nhu cầu tư vấn cụ thể. Trong khi đó, cơ chế RAG cho phép truy xuất thông tin chuyên sâu từ các nguồn dữ liệu nghiệp vụ đáng tin cậy, giúp giải đáp cả những câu hỏi phức tạp và đưa ra khuyến nghị phù hợp với tình huống thực tế. Đồng thời, nền tảng chú trọng đến việc cải thiện trải nghiệm người dùng thông qua cá nhân hóa và liên tục cập nhật hệ thống dựa trên phản hồi thực tế.

Kết quả đánh giá cho thấy Questin đạt hiệu suất vượt trội hoặc tương đương với các giải pháp hiện có như Langchain, LlamaIndex, và Ragflow. Trong lĩnh vực luật pháp, Questin đứng đầu về *Context Precision* với điểm số 0.96, và có khả năng truy xuất thông tin tốt (*Context Recall* 0.79), vượt qua các giải pháp khác. Đối với lĩnh vực kinh tế, Questin đạt *Answer Relevancy* cao nhất (0.56) và bao phủ 32/50 câu hỏi, vượt trội hơn Langchain (21/50) và Ragflow (22/50). Trong lĩnh vực tư vấn tuyển sinh, Questin tiếp tục duy trì hiệu suất mạnh mẽ với 49/55 câu trả lời đúng và *Answer Relevancy* đạt 0.73, ngang bằng hoặc vượt qua các hệ thống khác. Những kết quả này cho thấy Questin không chỉ đáp ứng mà còn vượt qua kỳ vọng trong việc cung cấp các giải pháp tư vấn chính xác và toàn diện, phục vụ hiệu quả cho nhiều lĩnh vực khác nhau.

Tuy nhiên, hệ thống vẫn tồn tại một số hạn chế. Đầu tiên, hệ thống gặp khó khăn khi xử lý các câu hỏi phức tạp, đa lớp hoặc đòi hỏi suy luận sâu, dẫn đến một số câu trả lời chưa đầy đủ. Để khắc phục, cần nâng cấp mô hình ngôn ngữ lớn

nhằm tăng cường khả năng suy luận, đồng thời tích hợp các cơ chế xử lý tri thức (knowledge reasoning). Thứ hai, độ tin cậy và tính đầy đủ của cơ sở dữ liệu nghiệp vụ là một thách thức, đặc biệt khi dữ liệu không được cập nhật kịp thời. Để giải quyết, cần xây dựng cơ chế tự động cập nhật dữ liệu và triển khai quy trình kiểm tra chất lượng định kỳ. Cuối cùng, khả năng hỗ trợ đa ngữ của hệ thống còn hạn chế, khiến việc sử dụng hệ thống với các ngôn ngữ khác ngoài tiếng Anh và tiếng Việt chưa tối ưu. Hướng phát triển trong tương lai là mở rộng các mô-đun đa ngữ để phục vụ người dùng trên toàn cầu.

Tôi kỳ vọng rằng nền tảng tư vấn nghiệp vụ ảo này sẽ trở thành công cụ đắc lực, hỗ trợ các doanh nghiệp và cá nhân tiếp cận thông tin chuyên sâu một cách hiệu quả và tiết kiệm. Qua đó, góp phần nâng cao khả năng ra quyết định, quản lý, và vận hành doanh nghiệp trong một thế giới ngày càng cạnh tranh và đổi mới.

Tài liệu tham khảo

[1] Hrupska Khrystyna. Trends and challenges in the field of consulting and business processes management. *Scial & Legal Studios*, page 152.

[2] Liming Xia, Nan Lin, and Yue Li. Research on business process management of engineering cost consulting enterprise. In *Proceedings of the 2nd International Conference on Economics and Management, Education, Humanities and Social Sciences (EMEHSS 2018)*, pages 491–496. Atlantis Press, 2018/03.

[3] Shande Feng. Research on success factors of management consulting projects in smes. In *Proceedings of the 4th Africa-Asia Dialogue Network (AADN) International Conference on Advances in Business Management and Electronic Commerce Research*, pages 1–4, 2022.

[4] Hye-Joo Song, Yen-Yoo You, and Hyun-Sung Park. A study on the influence of consulting execution characteristics on result quality and business utilization: Focusing on companies participating in government-supported consulting projects. *Cognitive Computing for Risk Management*, pages 91–103, 2022.

[5] Matias Bronnenmayer, Bernd W Wirtz, and Vincent Göttel. Success factors of management consulting. *Review of Managerial Science*, 10:1–34, 2016.

[6] Albina Borysivna KOVALENKO Oksana Mykhailivna KRAVCHENKO. Factors of influence on the consulting services development. 2020.

[7] Sumit Kumar Dam, Choong Seon Hong, Yu Qiao, and Chaoning Zhang. A complete survey on llm-based ai chatbots, 2024.

[8] Wojciech Sadkowski. Calculating quality costs in a selected service enterprise. *Krakow Review of Economics and Management/Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie*, (1(985)):103–119, Sep. 2020.

[9] Maali Mnasri. Recent advances in conversational nlp: Towards the standardization of chatbot building. *arXiv preprint arXiv:1903.09025*, 2019.

[10] Richard Csaky. Deep learning based chatbot models, 2019.

[11] Kiran Ramesh, Surya Ravishankaran, Abhishek Joshi, and K Chandrasekaran. A survey of design techniques for conversational agents. In *International con-*

ference on information, communication and computing technology, pages 336–350. Springer, 2017.

[12] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[13] Kenneth M Colby. Artificial paranoia: A computer simulation model of paranoid processes, 1975.

[14] I Sutskever. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[15] Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Attend to you: Personalized image captioning with context sequence memory networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 895–903, 2017.

[16] Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*, 2016.

[17] A Rush. A neural attention model for abstractive sentence summarization. *arXiv Preprint, CoRR, abs/1509.00685*, 2015.

[18] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[19] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[20] Tianran Hu, Anbang Xu, Zhe Liu, Quanzeng You, Yufan Guo, Vibha Sinha, Jiebo Luo, and Rama Akkiraju. Touch your heart: A tone-aware chatbot for customer care on social media. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[21] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 93–100, 2000.

[22] Steve J Young. Talking to machines (statistically speaking). In *INTERSPEECH*, pages 9–16, 2002.

[23] Jason D Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

[24] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

[25] Milica Gašić, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve Young. On-line policy optimisation of bayesian spoken dialogue systems via human interaction. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8367–8371. IEEE, 2013.

[26] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2024.

[27] Hyungjin Ko and Jaewook Lee. Can chatgpt improve investment decision? from a portfolio management perspective. In *From a Portfolio Management Perspective*, 2023.

[28] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: A large language model, instruction data and evaluation benchmark for finance, 2023.

[29] Haochun Wang, Sendong Zhao, Zewen Qiang, Nuwa Xi, Bing Qin, and Ting Liu. Beyond direct diagnosis: Llm-based multi-specialist agent consultation for automatic diagnosis. *arXiv preprint arXiv:2401.16107*, 2024.

[30] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models, 2023.

[31] Yutong Meng Yuhao Wang Hongcheng Liu, Yusheng Liao. Xiezhi chinese law large language model. `https://github.com/LiuHC0428/LAW_GPT`, 2023.

[32] Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. Lawyer llama technical report, 2023.

[33] Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model, 2024.

[34] Yi Feng, Chuanyi Li, and Vincent Ng. Legal judgment prediction: A survey of the state of the art. In *IJCAI. ijcai. org*, pages 5641–9, 2022.

[35] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. How does nlp benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, 2020.

[36] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[37] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[38] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[39] Zichong Wang, Zhibo Chu, Thang Viet Doan, Shiwen Ni, Min Yang, and Wenbin Zhang. History, development, and principles of large language models: an introductory survey. *AI and Ethics*, pages 1–17, 2024.

[40] Nadeen Fathallah, Arunav Das, Stefano De Giorgis, Andrea Poltronieri, Peter Haase, and Liubov Kovriguina. Neon-gpt: A large language model-powered pipeline for ontology learning. In *The Extended Semantic Web Conference*, 2024.

[41] Mohamad Diab, Julian Herrera, Bob Chernow, and Coco Mao. Stable diffusion prompt book. Technical report, Technical Report, 2022.

[42] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[43] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023.

[44] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March 2023.

[45] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55:1 – 38, 2022.

[46] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[47] Jiazhan Feng, Chongyang Tao, Xiubo Geng, Tao Shen, Can Xu, Guodong Long, Dongyan Zhao, and Daxin Jiang. Synergistic interplay between search and large language models for information retrieval. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9571–9583, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[48] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729, 2024.

[49] Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K. Qiu, and Lili Qiu. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely, 2024.

[50] Kalyani Pakhale. Comprehensive overview of named entity recognition: Models, domain-specific applications and challenges, 2023.

[51] Michele Banko and Robert C Moore. Part-of-speech tagging in context. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 556–561, 2004.

[52] Lukáš Havrlant and Vladik Kreinovich. A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *International Journal of General Systems*, 46(1):27–36, 2017.

[53] Anand Rajaraman and Jeffrey David Ullman. *Data Mining*, page 1–17. Cambridge University Press, 2011.

[54] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Scoring, term weighting, and the vector space model*, page 100–123. Cambridge University Press, 2008.

[55] Cinzia Viroli and Geoffrey J. McLachlan. Deep gaussian mixture models, 2017.

[56] Anique Tahir, Lu Cheng, Manuel Sandoval, Yasin N Silva, Deborah L Hall, and Huan Liu. Evaluating llms capabilities towards understanding social dynamics. *arXiv preprint arXiv:2411.13008*, 2024.

[57] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation, 2023.