

ĐẠI HỌC HUẾ
KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO



LÊ THANH HÙNG

**KHUNG KIẾN TRÚC RAG CẢI TIẾN
DỰA TRÊN PHÂN RÃ NGŨ NGHĨA VÀ
TRUY XUẤT ĐA MÔ HÌNH CHO
XÂY DỰNG TRI THỨC NỘI BỘ**

**KHÓA LUẬN TỐT NGHIỆP
KHOA HỌC DỮ LIỆU VÀ
TRÍ TUỆ NHÂN TẠO**

THÀNH PHỐ HUẾ, NĂM 2024

HUE UNIVERSITY
SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF DATA SCIENCE AND ARTIFICIAL INTELLIGENCE



LE THANH HUNG

**IMPROVED RAG ARCHITECTURAL
FRAMEWORK BASED ON
SEMANTIC DECOMPOSITION AND
MULTI-MODEL RETRIEVAL FOR
INTERNAL KNOWLEDGE
CONSTRUCTION**

**UNDERGRADUATE THESIS OF
DATA SCIENCE AND
ARTIFICIAL INTELLIGENCE**

HUE CITY, YEAR 2024

LỜI CAM DOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS Hồ Quốc Dũng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính xác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Khóa luận tốt nghiệp còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Khóa luận/Đồ án tốt nghiệp của mình. Khoa Kỹ thuật và Công nghệ - Đại học Huế không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP Hué, ngày ... tháng.... năm...

NGƯỜI HƯỚNG DẪN

(ký và ghi rõ họ tên)

TÁC GIẢ

(Tác giả DATN ký ghi rõ họ tên)

DECLARATION OF AUTHORSHIP

I hereby declare that this thesis was carried out by myself under the guidance and supervision of Dr. Ho Quoc Dung and that the work and the results contained in it are original and have not been submitted anywhere for any previous purposes. The data and figures presented in this thesis are for analysis, comments, and evaluations from various resources by my own work and have been duly acknowledged in the reference part.

In addition, other comments, reviews and data used by other authors, and organizations have been acknowledged, and explicitly cited.

I will take full responsibility for any fraud detected in my thesis. School of Engineering and Technology, Hue University is unrelated to any copyright infringement caused on my work (if any).

Hue City, day...month... year...

INSTRUCTOR

(Signature and full name)

AUTHOR

(Signature and full name)

LỜI CẢM ƠN

Trước hết, em xin chân thành gửi lời cảm ơn đến Khoa Kỹ thuật và Công nghệ, nơi đã cung cấp cho em một môi trường học tập năng động và cùng nhiều cơ hội được tiếp xúc với các tri thức, là phần quan trọng trong sự phát triển của em trong thời gian học tập và phát triển bản thân.

Tiếp đến, em xin bày lòng cảm ơn sâu sắc đến tiến sĩ Hồ Quốc Dũng, người thầy đã tận tâm hướng dẫn em không chỉ trong thời gian thực hiện khóa luận mà còn trong suốt quá trình học tập và phát triển tại khoa. Sự nhiệt tình và chỉ dẫn của thầy đóng góp to lớn vào những kết quả mà em đạt được.

Em cũng xin bày tỏ lòng biết ơn đến tất cả các thầy cô bộ môn trong và ngoài khoa đã giảng dạy, cung cấp kiến thức bổ ích cho em trong suốt quá trình học tập và định hình bản thân, các thầy cô không chỉ cung cấp cho em kiến thức chuyên ngành cần thiết mà là thế giới quan mới và các quan điểm tiên tiến.

Em cũng xin chân thành cảm ơn các thầy cô trong Hội đồng chấm khóa luận đã dành thời gian đọc và đóng góp những ý kiến quý báu để em có thể hoàn thiện khóa luận này tốt hơn.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn ở bên động viên, khích lệ và hỗ trợ em trong suốt quá trình học tập và nghiên cứu tại trường.

Mặc dù đã có nhiều cố gắng trong quá trình thực hiện, song không thể tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp từ quý thầy cô, ban hội đồng để khóa luận được hoàn thiện hơn.

Em xin chân thành cảm ơn!

(Tác giả ĐATN ký ghi rõ họ tên)

Lê Thanh Hùng

ACKNOWLEDGMENT

First and foremost, I would like to extend my profound gratitude to the Faculty of Engineering and Technology, which has provided me with a dynamic and intellectually stimulating academic environment. This institution has been instrumental in my intellectual and personal development, offering invaluable opportunities for academic and professional growth.

I am deeply appreciative of Dr. Ho Quoc Dung, my esteemed supervisor, who has demonstrated exceptional dedication not only during the preparation of this thesis but throughout my entire academic journey. His meticulous guidance, insightful mentorship, and unwavering support have been pivotal in shaping my academic achievements and research capabilities.

My sincere appreciation extends to all faculty members within and beyond the department who have imparted comprehensive knowledge and facilitated my academic and intellectual formation. These distinguished educators have not merely transmitted disciplinary expertise but have also expanded my worldview and introduced progressive perspectives that transcend traditional academic boundaries.

I am particularly grateful to the distinguished members of the thesis evaluation committee who generously allocated their time to meticulously review this work and provide constructive feedback. Their scholarly insights have been instrumental in refining and enhancing the quality of this research.

Lastly, I wish to express my heartfelt thanks to my family and colleagues who have consistently provided emotional support, encouragement, and academic motivation throughout my scholarly endeavors.

While I have endeavored to maintain the highest standards of academic rigor, I acknowledge that this work may have inherent limitations. I would be most appreciative of any constructive criticism and scholarly recommendations that could further improve the academic merit of this thesis.

Sincerely,

Le Thanh Hung

TÓM TẮT

Đề tài nghiên cứu xây dựng khung kiến trúc tăng cường truy xuất thông tin (RAG) cải tiến dựa trên phân rã ngữ nghĩa và xây dựng chatbot nội bộ là một đề tài đề xuất các phương án giải quyết hạn chế của mô hình RAG truyền thống. Trước những hạn chế trong việc truy xuất dữ liệu cho câu hỏi của người dùng, khi cần phải tổng hợp dữ liệu từ nhiều vị trí khác nhau để đưa ra phản hồi chính xác, các mô hình hiện tại dễ bị bỏ lỡ, nghiên cứu đã thành công trong việc sử dụng mô hình T5 và kiến trúc Sequence-to-Sequence để phân rã các yêu cầu người dùng thành nhiều yêu cầu nhỏ hơn và tiến hành truy xuất dữ liệu trên đó rồi tổng hợp, bằng cách đó đề tài nghiên cứu đã đưa ra phương pháp làm hạn chế việc truy xuất thiếu, không đầy đủ thông tin của các mô hình truyền thống.

Không dừng lại ở đó, đề tài đưa ra phương pháp kết hợp mô hình ColBERT và BM25 để tăng cường hiệu suất truy xuất thông tin. Với khối lượng lớn về tài nguyên hệ thống khi yêu cầu lưu trữ thông tin được mã hoá ở mức ColBERT tồn quá nhiều tài nguyên khiến cho phương pháp chỉ dừng lại ở mức nghiên cứu. Tuy nhiên, đề tài đã đề xuất phương pháp xây dựng mô hình chủ đề với FASTopic và cách tối ưu các chủ đề, từ đó thực hiện truy xuất dữ liệu từ các cụm với tốc độ nhanh hơn, đưa ColBERT từ giải pháp nghiên cứu đi đến thực tế.

Cuối cùng nghiên cứu đã kết hợp tất cả các thành phần lại với nhau và xây dựng hệ thống Chatbot dựa trên tri thức nội bộ, chứng minh tính thực tế của nghiên cứu và đáp ứng nhu cầu cấp bách của các doanh nghiệp về một hệ thống tri thức nội bộ được doanh nghiệp toàn quyền quản lý và lưu trữ cục bộ.

ABSTRACT

This research addresses critical limitations in traditional Retrieval-Augmented Generation (RAG) architectures by proposing an innovative approach to internal knowledge-based chatbot development. The study introduces a novel semantic decomposition framework that significantly enhances information retrieval and response generation capabilities.

The research successfully employs the T5 model and Sequence-to-Sequence architecture to decompose complex user queries into more granular sub-queries, enabling more comprehensive and precise information retrieval. By breaking down complex queries, the proposed method mitigates the information retrieval shortcomings inherent in traditional RAG models, ensuring more accurate and contextually relevant responses.

A key contribution of the study is the innovative hybrid retrieval method combining ColBERT and BM25 techniques to optimize information retrieval performance. Recognizing the computational challenges associated with ColBERT's high-resource encoding requirements, the research develops a novel topic modeling approach using FASTopic. This optimization strategy transforms ColBERT from a purely theoretical solution to a practical implementation, enabling efficient data retrieval through intelligent topic clustering.

The culmination of the research is the development of an internal knowledge-based chatbot system. This implementation not only demonstrates the practical feasibility of the proposed approach but also addresses the urgent enterprise need for locally managed and secured knowledge management systems.

By integrating advanced natural language processing techniques with intelligent information retrieval strategies, this research provides a significant advancement in developing more responsive, accurate, and context-aware conversational AI systems for enterprise environments.

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT VÀ KÝ HIỆU	I
DANH MỤC CÁC HÌNH, ĐỒ THỊ, BIỂU ĐỒ	II
DANH MỤC CÁC BẢNG	III
PHẦN I: MỞ ĐẦU.....	1
1. <i>Tính cấp thiết của đề tài khóa luận</i>	1
2. <i>Mục tiêu của nghiên cứu</i>	3
3. <i>Phương pháp nghiên cứu</i>	5
4. <i>Cấu trúc của Đồ án Tốt nghiệp.....</i>	7
PHẦN II: NỘI DUNG NGHIÊN CỨU	8
<i>Chương 1: Tổng quan về nghiên cứu.....</i>	8
1.1. <i>Tình hình nghiên cứu trong và ngoài nước</i>	8
1.1.1. <i>Tổng quan về Chatbot</i>	8
1.1.2. <i>Tổng quan về truy xuất thông tin</i>	10
1.1.3. <i>Tổng quan về RAG (Retrieval-Augmented Generation)</i>	11
1.1.4. <i>Hạn chế của các hệ thống RAG ở thời điểm hiện tại</i>	15
1.1.5. <i>Đánh giá kết quả hệ thống RAG</i>	16
1.1.6. <i>Tổng quan về phân rã câu hỏi</i>	17
1.2. <i>Cơ sở lý thuyết</i>	18
1.2.1. <i>Long Short-Term Memory (LSTM).....</i>	18
1.2.2. <i>Cơ chế Tự Chú Ý</i>	20
1.2.3. <i>Mô hình BERT</i>	21
1.2.4. <i>Mô hình Seq2Seq (Sequence to Sequence)</i>	24
1.2.5. <i>Mô hình T5 (Text to Text transfer Transformer).....</i>	26

1.2.6. BM25 (Best Match 25)	27
1.2.7. ColBERT (Contextualized Late Interaction over BERT)	28
1.2.8. Mô hình chủ đề FASTopic	32
1.2.9. Cơ sở dữ liệu vector Neo4j	33
1.3. <i>Tiểu kết chương 1</i>	34
Chương 2: Mô tả bài toán và khung nghiên cứu.....	35
2.1. <i>Mô tả bài toán</i>	36
2.2. <i>Khung nghiên cứu</i>	38
2.2.1. Giai đoạn 1: Nghiên cứu xây dựng mô hình phân rã câu hỏi	39
2.2.2. Giai đoạn 2: Nghiên cứu mô hình truy xuất dữ liệu	41
2.2.3. Giai đoạn 3: Tổng hợp nghiên cứu và xây dựng Chatbot nội bộ	42
2.3. <i>Tiểu kết chương 2</i>	44
Chương 3: Thu thập và tiền xử lý dữ liệu	46
3.1. <i>Bộ dữ liệu hỗ trợ phân rã câu hỏi</i>	46
3.1.1. Chuyển đổi và tái cấu trúc dữ liệu	47
3.1.2. Tiền xử lý dữ liệu	49
3.2. <i>Bộ dữ liệu MS Macro cho hệ thống truy xuất thông tin</i>	50
3.2.1. Tiền xử lý dữ liệu Macro.....	51
3.2.2. Xây dựng Cơ sở dữ liệu vector ở mức Token	51
3.3. <i>Tiểu kết chương 3</i>	52
Chương 4: Thiết kế, lựa chọn kỹ thuật phân tích, trích xuất đặc trưng và triển khai các mô hình.....	54
4.1. <i>Mô hình phân rã câu hỏi dựa trên Seq2Seq và T5.....</i>	54
4.2. <i>Mô hình truy xuất thông tin với ColBERT và BM25.....</i>	56

<i>4.3. Mô hình chủ đề với FASTopic.....</i>	59
<i>4.4. Xây dựng hệ thống chatbot sử dụng tri thức nội bộ.....</i>	61
4.4.1. Kỹ thuật chia nhỏ văn bản theo ngữ nghĩa	61
4.4.2. Kết hợp và hoàn thiện	62
<i>Chương 5: Đánh giá kết quả và ứng dụng thực tế.....</i>	64
<i>5.1. Đánh giá kết quả tổng hợp</i>	64
5.1.1. Đánh giá hiệu xuất mô hình phân rã câu hỏi	64
5.1.2. Đánh giá hiệu xuất mô hình truy xuất.....	66
5.1.3. Kết quả xây dựng giao diện người dùng cho chatbot nội bộ	67
<i>5.2. Khả năng ứng dụng thực tế</i>	69
5.2.1. Tính thực khả thi của đề tài khi sử dụng trong thực tế.	69
5.2.2. Rủi ro trong thực tế.	70
<i>5.3. Hướng phát triển trong tương lai.....</i>	70
KẾT LUẬN VÀ KIẾN NGHỊ.....	72
<i>1. Kết luận</i>	72
<i>2. Kiến nghị.....</i>	73
2.1 Hướng phát triển nghiên cứu	73
2.2 Kiến nghị ứng dụng thực tiễn	75
TÀI LIỆU THAM KHẢO.....	77

DANH MỤC CÁC CHỮ VIẾT TẮT VÀ KÝ HIỆU

LLM	Large language model
GenAI	Generative Artificial Intelligence
IR	Information Retrieve
NLP	Natural Language Processing
RAG	Retrieval-Augmented Generation
BERT	Bidirectional Encoder Representations from Transformers
T5	Text-to-Text Transfer Transformer
Seq2Seq	Sequence-to-Sequence
DPR	Dense Passage Retrieval
ColBERT	Contextualized Late Interaction over BERT
MRR	Mean Reciprocal Rank
MAP	Mean Average Precision
NDCG	Normalized Discounted Cumulative Gain
ROUGE-1	Recall-Oriented Understudy for Gisting Evaluation (Unigram Overlap)
ROUGE-2	Recall-Oriented Understudy for Gisting Evaluation (Bigram Overlap)
ROUGE-3	Recall-Oriented Understudy for Gisting Evaluation (Trigram Overlap)
BLEU	Bilingual Evaluation Understudy

DANH MỤC CÁC HÌNH, ĐỒ THỊ, BIỂU ĐỒ

Số hiệu hình	Tên hình, đồ thị, biểu đồ	Trang
Hình 1.1	Mô hình RAG truyền thông	12
Hình 1.2	Dense Passage Retrieve (DPR) và kết quả mô hình	14
Hình 1.3	Cơ chế LSTM	19
Hình 1.4	Cơ chế tự chú ý	21
Hình 1.5	Mô hình BERT	22
Hình 1.6	Mask language model	23
Hình 1.7	Mô hình T5	26
Hình 1.8	Mô hình BM25 trong truy xuất thông tin	28
Hình 1.9	Mô hình ColBERT	31
Hình 1.10	Hiệu suất mô hình FASTopic	32
Hình 1.11	Lưu trữ và truy xuất dữ liệu trong Neo4j	34
Hình 2.1	Khung nghiên cứu xây dựng mô hình phân rã câu hỏi	39
Hình 2.2	Khung nghiên cứu xây dựng mô hình truy xuất dữ liệu	41
Hình 2.3	Khung nghiên cứu tổng hợp và xây dựng Chatbot nội bộ	42
Hình 4.1	Mô hình Seq2Seq sử dụng huấn luyện	55
Hình 4.2	Phương pháp tính độ tương đồng của ColBERT	57
Hình 4.3	Kiến trúc hệ thống Chatbot nội bộ	63
Hình 5.1	Quá trình huấn luyện mô hình phân rã câu hỏi	65
Hình 5.2	Dữ liệu được biểu diễn trong Neo4j	68

DANH MỤC CÁC BẢNG

<i>Số hiệu bảng</i>	<i>Tên bảng</i>	<i>Trang</i>
Bảng 5.1	Kết quả mô hình phân rã câu hỏi	64
Bảng 5.2	Thông tin cấu hình thiết bị huấn luyện mô hình ColBER	66
Bảng 5.3	Đánh giá hiệu xuất mô hình truy xuất thông tin	67
Bảng 5.4	Tốc độ truy xuất dữ liệu	68

PHẦN I: MỞ ĐẦU

1. Tính cấp thiết của đề tài khóa luận

Ngày nay, với sự phát triển mạnh mẽ của các mô hình ngôn ngữ lớn (LLM) và trí tuệ nhân tạo sinh thành (GenAI), chúng ta đang chứng kiến một cuộc cách mạng trong cách con người tương tác với máy tính và truy cập thông tin. Đặc biệt, sự phát triển của các mô hình ngôn ngữ đang cực kỳ được chú trọng khai thác và phát triển bởi tiềm năng vô cùng to lớn trong nhiều tự động hóa và cá nhân hóa nhiều công đoạn. Các mô hình ngôn ngữ lớn hiện nay có khả năng hiểu ngữ nghĩa và phân tích sâu sắc văn bản, có thể thấy được sự mạnh mẽ của các mô hình ngôn ngữ khi thang đo năng lực của chúng hiện tại không chỉ dừng ở độ chính xác mà còn phản ánh sự tự nhiên khi phản hồi và tính thực tế.

Tuy nhiên, bên cạnh những tiến bộ đáng kể, các mô hình tiên tiến hiện nay vẫn phải đối mặt với nhiều thách thức về độ tin cậy của thông tin. Một trong những vấn đề quan trọng nhất là khả năng xác định nguồn gốc của các câu trả lời từ mô hình ngôn ngữ. Việc thiếu cơ sở rõ ràng để đánh giá và kiểm chứng thông tin được cung cấp làm giảm độ tin cậy của các phản hồi. Đặc biệt đáng ngại là xu hướng của các mô hình ngôn ngữ trong việc thể hiện sự tự tin quá mức khi đưa ra phản hồi cho mọi yêu cầu của người dùng, bất kể mức độ hiểu biết thực sự của chúng về vấn đề được đề cập.

Thực tế cho thấy, mặc dù các mô hình ngôn ngữ lớn đáp ứng tốt các nhu cầu sử dụng chung, việc ứng dụng chúng trong các lĩnh vực chuyên sâu vẫn gặp nhiều trở ngại. Hiện tượng "ảo giác" khi mô hình tạo ra những thông tin không có thực hoặc đưa ra những câu trả lời sai lệch vẫn là một thách thức lớn cần được giải quyết. Điều này đặt ra yêu cầu cấp thiết trong việc nghiên cứu và phát triển các giải pháp nhằm nâng cao độ tin cậy và khả năng kiểm chứng của các câu trả lời từ mô hình ngôn ngữ lớn.

Thêm vào đó, trong thời đại ngày nay, có hàng nghìn, hàng triệu thông tin mới được tạo ra mỗi ngày, làm thế nào các mô hình ngôn ngữ có thể cập nhập được

những kiến thức này sẽ là một vấn đề lớn. Việc đào tạo lại một mô hình ngôn ngữ lớn rất đắt đỏ và tốn nhiều nguồn lực, hơn nữa, ta không thể huấn luyện mô hình với thông tin mới mỗi ngày được.

Trước những thách thức nêu trên, Retrieval-Augmented Generation (RAG) nổi lên như một giải pháp đầy triển vọng, được đề xuất lần đầu bởi Lewis et al. RAG là một kiến trúc kết hợp giữa hệ thống truy xuất thông tin và mô hình ngôn ngữ lớn, nhằm tăng cường độ tin cậy và tính chính xác của các câu trả lời. Bản chất của RAG là việc bổ sung thông tin từ nguồn dữ liệu đáng tin cậy vào quá trình sinh câu trả lời của mô hình ngôn ngữ lớn. Có thể thấy, RAG ra đời như một giải pháp cho vấn đề độ chính xác và cập nhật kiến thức. Bằng cách tích hợp một cơ sở dữ liệu tri thức, RAG cho phép mô hình ngôn ngữ lớn truy cập và tìm kiếm thông tin từ nguồn đáng tin cậy. Nếu không tìm thấy thông tin, mô hình có thể thừa nhận là không biết, từ đó tăng độ tin cậy của câu trả lời. Hơn nữa, việc cập nhật cơ sở dữ liệu tri thức hàng ngày giúp các mô hình ngôn ngữ lớn có được kiến thức mới mà không cần phải đào tạo lại toàn bộ mô hình [1].

Mặc dù RAG đã chứng minh được hiệu quả trong việc nâng cao độ tin cậy của câu trả lời từ mô hình ngôn ngữ lớn, kiến trúc RAG truyền thống vẫn còn những hạn chế cần được cải thiện:

Thứ nhất, cơ chế truy xuất thông tin hiện tại thường xử lý câu truy vấn như một thể thống nhất, chưa xét đến tính đa dạng và phức tạp trong cấu trúc ngữ nghĩa của câu hỏi. Câu hỏi của người dùng thường phức tạp và lồng ghép nhiều yêu cầu. Khi truy vấn trực tiếp từ câu hỏi người dùng có thể dẫn đến việc bỏ sót thông tin quan trọng hoặc không tổng hợp đầy đủ thông tin khi phản hồi trong khi câu trả lời có đầy đủ trong cơ sở dữ liệu.

Thứ hai, các phương pháp truy xuất thông tin hiện được sử dụng phổ biến trong RAG thường chỉ tập trung vào một khía cạnh của sự tương đồng - hoặc là khớp từ vựng hoặc là khớp ngữ nghĩa. Điều này tạo ra một sự đánh đổi: hoặc là có độ chính xác cao trong việc tìm kiếm các thuật ngữ cụ thể nhưng thiếu khả năng hiểu ngữ cảnh rộng hơn, hoặc là có khả năng nắm bắt tốt mối quan hệ ngữ nghĩa nhưng đôi khi bỏ qua những chi tiết quan trọng trong câu hỏi.

Ngoài ra, một xu thế trong thời đại hiện nay đó là người dùng thường muốn sở hữu toàn bộ kiến trúc hệ thống, có thể thấy hiện nay có rất nhiều nền tảng hỗ trợ xây dựng RAG. Tuy nhiên các tổ chức thường yêu cầu được kiểm soát và sở hữu toàn diện kiến trúc hệ thống, đặc biệt là phần cơ sở dữ liệu và việc lưu trữ thông tin. Điều này xuất phát từ mối quan ngại chính đáng về rủi ro bảo mật khi để một bên thứ ba nắm giữ dữ liệu nội bộ, đặc biệt trong bối cảnh xây dựng chatbot và hệ thống hỗ trợ trí thức cho tổ chức.

Những hạn chế này trở nên đặc biệt rõ rệt trong bối cảnh xây dựng tri thức nội bộ, nơi câu hỏi thường mang tính chuyên ngành cao và đòi hỏi sự kết hợp nhiều loại thông tin từ các tài liệu khác nhau. Việc không thể nắm bắt đầy đủ các khía cạnh của câu hỏi và hạn chế trong phương pháp truy xuất có thể dẫn đến việc bỏ sót những tài liệu quan trọng hoặc truy xuất thông tin không phù hợp với ngữ cảnh thực tế của tổ chức.

Qua đó, có thể thấy được nhu cầu, sự cần thiết về một hệ thống Chatbot toàn diện dựa trên dữ liệu cá nhân, việc nghiên cứu giải quyết các vấn đề hiện tại sẽ đóng góp vào sự phát triển của các hệ truy xuất thông tin nói chung, cung cấp khung xây dựng, làm giảm hạn chế của các mô hình hiện tại và cung cấp giải pháp nói chung cho các tổ chức có nhu cầu. Giải quyết được các thách thức này sẽ mở ra tiềm năng to lớn trong việc ứng dụng công nghệ AI vào việc xây dựng và khai thác tri thức nội bộ của các tổ chức một cách hiệu quả và an toàn. Đồng thời góp phần thúc đẩy sự phát triển chung của công nghệ xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo.

2. Mục tiêu của nghiên cứu

Từ việc phân tích thực trạng hiện nay, khóa luận đặt ra mục tiêu xây dựng một hệ thống chatbot thông minh đáp ứng các yêu cầu đặc thù của tổ chức. Không đơn thuần là một chatbot với tri thức tổng quát, hệ thống được thiết kế để trở thành một giải pháp toàn diện cho việc quản lý và khai thác tri thức nội bộ của tổ chức.

Hệ thống hướng đến ba đặc tính quan trọng:

- Thứ nhất, có khả năng hiểu sâu hơn về câu hỏi người dùng và phân rã các yêu cầu từ người dùng không phải tìm kiếm một cách mù quáng.

- Thứ hai, khả năng cung cấp câu trả lời có độ tin cậy cao thông qua việc trích dẫn và dẫn chiếu đến nguồn thông tin cụ thể.
- Thứ ba, tính linh hoạt và khả năng thích ứng cao thông qua khả năng tuỳ biến, cập nhật và quản lý tri thức một cách chủ động, dễ dàng cập nhập thông tin mới cũng như loại bỏ đi các dữ liệu đã lỗi thời.
- Thứ tư, có tính tự chủ cao khi tổ chức có thể làm chủ hoàn toàn hệ thống, từ cơ sở dữ liệu, mô hình ngôn ngữ lớn đến cơ chế truy xuất thông tin mà không phụ thuộc vào bên thứ ba.

Với những mục tiêu này, khóa luận hướng đến việc tạo ra một giải pháp thực tiễn, đáp ứng nhu cầu cấp thiết của các tổ chức trong việc xây dựng và vận hành hệ thống quản lý tri thức thông minh trong kỷ nguyên số. Để có thể đi vào chi tiết hơn, hệ thống đề tài đưa ra các mục tiêu cụ thể như sau:

- Xây dựng hệ thống phân rã yêu cầu từ người dùng: Khi tương tác với Chatbot, người dùng thường lòng ghép các yêu cầu với nhau, ta cần phải tách biệt được các yêu cầu này ra và tiến hành xử lý lần lược các yêu cầu, không phải câu hỏi ban đầu của người dùng trực tiếp vào tiến trình xử lý.
- Xây dựng hệ thống truy xuất được từng yêu cầu của người dùng và đưa ra câu trả lời đầy đủ nhất. Hệ thống không chỉ có khả năng truy xuất dựa trên các từ khoá mà còn phải hiểu được các từ đồng nghĩa, có nghĩa gần nhau, hiểu được đoạn dữ liệu nào chứa câu trả lời mà người dùng mong muốn.
- Hệ thống sẽ được xây dựng cục bộ khi mà tổ chức có thể nắm giữ từ đầu đến cuối mà không cần thông qua một bên thứ ba trong bất kỳ công đoạn nào.
- Hệ thống có khả năng tương tác dễ dàng, doanh nghiệp có thể dễ dàng cập nhập hay xoá tri thức, có thể dễ dàng xem và kiểm tra các tri thức lỗi thời.
- Xây dựng một giao diện trực quan, thân thiện với người dùng.

- Đánh giá hiệu quả của hệ thống từng phần của hệ thống với các thước đo cụ thể và phù hợp.
- Nghiên cứu và đề xuất giải pháp cho các vấn đề liên quan đến độ chính xác, cập nhật kiến thức, và tính linh hoạt của Chatbot.

Thông qua việc thực hiện các mục tiêu này, nghiên cứu đặt kỳ vọng sẽ tạo ra một sản phẩm có giá trị thực tiễn cao, có thể triển khai và áp dụng hiệu quả trong môi trường doanh nghiệp. Đồng thời, kết quả nghiên cứu cũng sẽ đóng góp vào việc phát triển và hoàn thiện các giải pháp chatbot thông minh, đặc biệt trong bối cảnh chuyển đổi số và nhu cầu ngày càng cao về quản lý tri thức hiệu quả của các tổ chức.

3. Phương pháp nghiên cứu

Đối tượng nghiên cứu: Kiến trúc và chức năng truy xuất thông tin của hệ thống chatbot dựa trên mô hình RAG

Các đối tượng nghiên cứu cụ thể:

- Phương pháp phân rã và xử lý đa yêu cầu từ người dùng khi tương tác với chatbot.
- Cơ chế truy xuất thông tin thông minh kết hợp cả khớp từ vựng và khớp ngữ nghĩa.
- Quy trình xây dựng, cập nhật và quản lý cơ sở tri thức nội bộ.
- Giải pháp triển khai hệ thống chatbot hoàn toàn cục bộ đảm bảo tính tự chủ cho tổ chức.
- Phương pháp đánh giá hiệu quả và độ tin cậy của hệ thống chatbot trong môi trường thực tế. Bao gồm: Đánh giá khả năng phân rã, khả năng truy xuất, và tốc độ phản hồi của hệ thống.

Phương pháp nghiên cứu lý thuyết:

- Tổng hợp và phân tích các công trình nghiên cứu liên quan đến kiến trúc RAG, các phương pháp truy xuất thông tin và các mô hình ngôn ngữ lớn.

- Nghiên cứu các phương pháp phân rã câu hỏi và xử lý ngôn ngữ tự nhiên hiện đại.
- Tìm hiểu các kỹ thuật đánh giá độ chính xác và hiệu quả của hệ thống chatbot.
- Phân tích các giải pháp triển khai hệ thống cục bộ.

Phương pháp nghiên cứu thực nghiệm:

- Thiết kế và xây dựng nguyên mẫu của hệ thống chatbot theo kiến trúc đề xuất.
- Thực hiện các thử nghiệm về khả năng phân rã yêu cầu người dùng và độ chính xác của hệ thống truy xuất thông tin.
- Đánh giá hiệu quả của hệ thống thông qua các phương pháp đo cụ thể như độ chính xác của câu trả lời, thời gian phản hồi, và khả năng xử lý các câu hỏi phức tạp.

Phạm Vi Nghiên Cứu:

- Về mặt kỹ thuật:
 - Tập trung vào phát triển phần phân rã câu hỏi người dùng và hệ thống truy xuất thông tin thông minh.
 - Nghiên cứu và triển khai các giải pháp lưu trữ và quản lý tri thức nội bộ một cách hiệu quả.
 - Xây dựng cơ chế cập nhật và duy trì cơ sở tri thức một cách linh hoạt.
 - Phát triển giao diện người dùng trực quan và dễ sử dụng.
 - Tối ưu hóa hiệu năng hệ thống để đảm bảo thời gian phản hồi nhanh chóng.
- Về mặt ứng dụng:
 - Tập trung vào việc xây dựng chatbot phục vụ nhu cầu quản lý và khai thác tri thức nội bộ của tổ chức.

- Ưu tiên các tính năng thiết yếu như khả năng tìm kiếm thông tin chính xác, cập nhật tri thức dễ dàng, và bảo mật thông tin.
- Đặt trọng tâm vào tính ứng dụng thực tế và khả năng triển khai trong môi trường doanh nghiệp.
- Xây dựng các tính năng quản trị hệ thống để doanh nghiệp có thể tự chủ trong việc vận hành và bảo trì.

4. Cấu trúc của Đề án Tốt nghiệp

Khóa luận được chia thành các chương với nội dung chính như sau:

PHẦN MỞ ĐẦU

- *Tính cấp thiết của đề tài khóa luận*
- *Mục tiêu của khóa luận*
- *Phương pháp nghiên cứu*
- *Cấu trúc của khóa luận*

NỘI DUNG KHÓA LUẬN

- *Chương 1: Tổng quan về nghiên cứu*
- *Chương 2: Mô tả bài toán và khung nghiên cứu*
- *Chương 3: Thu thập và tiền xử lý dữ liệu*
- *Chương 4: Thiết kế, lựa chọn kỹ thuật phân tích, trích xuất đặc trưng và triển khai các mô hình*
- *Chương 5: Đánh giá kết quả và ứng dụng thực tế*

KẾT LUẬN VÀ KIẾN NGHỊ

TÀI LIỆU THAM KHẢO

PHẦN II: NỘI DUNG NGHIÊN CỨU

Chương 1: Tổng quan về nghiên cứu

1.1. Tình hình nghiên cứu trong và ngoài nước

1.1.1. Tổng quan về Chatbot

Trong bối cảnh bùng nổ trước sự phát triển của trí tuệ nhân tạo, đặc biệt trong lĩnh vực xử lý ngôn ngữ tự nhiên, Chatbot đang được nhận về nhiều sự quan tâm bởi tiềm năng to lớn mà nó có thể mang lại trong tương lai. Việc ứng dụng chatbot trong các lĩnh vực như Tiếp thị, Hệ thống hỗ trợ, Giáo dục, Chăm sóc sức khỏe, Di sản văn hóa và Giải trí đã đẩy mạnh tốc độ phát triển của các lĩnh vực trên cũng như tiết kiệm nhiều công sức, lược bỏ nhiều giai đoạn lặp lại gây tiêu tốn nguồn lực. Chatbot, về bản chất, là một hệ thống phần mềm thông minh được thiết kế để mô phỏng cuộc hội thoại với con người thông qua giao diện đối thoại tự nhiên. Tuy nhiên, chúng không chỉ được xây dựng với mục đích đơn giản như vậy, ứng dụng của chatbot cực kỳ rộng rãi khi được tích hợp vào các nền tảng giáo dục, tìm kiếm thông tin, kinh doanh và thương mại điện tử. Chatbot có thể được phân loại thành nhiều loại khác nhau, bao gồm Task-oriented Dialogue Systems (TODs), Intelligent Personal Assistants (IPAs) và Chit-chat Dialogue Systems (CCDs). Mỗi loại chatbot có những đặc điểm và ứng dụng riêng, nhưng mục tiêu chung của chúng là cải thiện trải nghiệm người dùng thông qua việc cung cấp thông tin và hỗ trợ một cách nhanh chóng và hiệu quả [2].

Trong thời đại mà lượng thông tin quá nhiều, chúng ta cần đọc một lượng lớn văn bản chỉ để tìm ra câu trả lời cho vấn đề của mình, Chatbot có thể thay chúng ta làm việc đó và đưa ra câu trả lời cụ thể cho vấn đề mà người dùng đang tìm kiếm, như vậy sẽ tiết kiệm được rất nhiều thời gian và nâng cao năng suất cho công việc của người dùng [3].

Các Chatbot mạnh mẽ hiện nay như ChatGPT, Gemini, Claude,... là kết quả của các mô hình ngôn ngữ lớn. Mô hình ngôn ngữ, khái niệm nổi lên trong những năm gần đây thật ra đã xuất hiện từ rất lâu, một trong những bài báo lâu đời về

ngôn ngữ nhất có thể nhắc đến phương pháp thống kê xử lý phân tán được chia sẻ từ những năm 1991 [4]. Mặc dù với ý tưởng đơn giản, mang nơ ron này cho thấy bước đầu tiên trong nhiệm vụ sinh ngôn ngữ. Có thể nói đây là một trong những bài báo đặt nền tảng cho các mô hình ngôn ngữ lớn sau này.

Ngày nay, Chatbot mạnh mẽ hơn bao giờ hết tuy nhiên vấn đề của chatbot vẫn tồn tại, nhiều mô hình ý tưởng về sinh ngôn ngữ đầu tiên gặp vấn đề về khác biệt giữa cách mà máy nhìn nhận ngôn ngữ và con người, mặc dù ngày nay nó đã được tối ưu đáng kể và người dùng biết cách để điều hướng AI sinh văn bản theo ý muốn của mình. Người ta đã xây dựng các mô hình ngôn ngữ lớn hơn với hàng tỷ tham số, dung lượng dữ liệu cũng được mở rộng đáng kể và tăng độ chính xác cũng như tính tự nhiên khi sinh văn bản, tuy nhiên vấn đề lại tiếp tục xảy ra đối với các LLM khi:

- Các LLM thường quá tự tin vào câu trả lời của mình, và thường tạo ra những câu trả lời không chính xác hay mọi người thường nói là nó “biết” ra các câu trả lời.
- LLM không cung cấp được bằng chứng cho câu trả lời của mình dẫn đến thiếu tin tưởng về chất lượng [5].
- Việc xây dựng và huấn luyện LLM tốn rất nhiều thời gian và tài nguyên, nên huấn luyện dữ liệu mới cho LLM cần được xem xét rất kỹ lưỡng.
- Các LLM chỉ trả lời dựa trên dữ liệu đã được huấn luyện nên thông tin thường khá cũ, trong thời đại sản sinh thông tin liên tục ngày nay thì đây thực sự là một vấn đề.
- Do khó khăn trong việc huấn luyện lại và tinh chỉnh, việc sử dụng LLM trong một lĩnh vực riêng biệt cũng gặp nhiều khó khăn và chưa thể tận dụng triệt để các LLM sẵn có.

Trước sự phát triển mạnh mẽ của Chatbot, nhu cầu về việc sở hữu Chatbot của riêng mình là rất lớn với mục đích đa dạng, trong học tập, Đại học Kinh tế Quốc dân đã xây dựng Chatbot nội bộ hỗ trợ tuyển sinh cho riêng mình, tuy không sử dụng LLM nhưng cũng thành công trong việc đưa các ứng dụng NLP vào phục

vụ tuyển sinh [6].

Việc sở hữu Chatbot cho riêng mình là một điều tuyệt vời có thể tối ưu hoá nhiều quy trình và giảm tải lượng công việc cho doanh nghiệp. Tuy nhiên, huấn luyện lại LLM cho các ứng dụng cụ thể là một thách thức lớn do yêu cầu tài nguyên cao và thời gian kéo dài. Điều này không chỉ ảnh hưởng đến hiệu suất mà còn làm giảm tính khả thi trong việc áp dụng LLM cho nhiều lĩnh vực khác nhau. Cần có những phương pháp mới để cải thiện khả năng tinh chỉnh và cập nhật LLM nhằm đáp ứng nhu cầu thay đổi nhanh chóng của thị trường [7].

Như vậy, có thể thấy Chatbot đang là mối quan tâm và mang lại tiềm năng to lớn, được ứng dụng rộng rãi đa lĩnh vực, tuy nhiên vẫn còn nhiều bất cập và hạn chế cần cải thiện. Để cải thiện những vấn đề này, truy xuất thông tin (IR - Information Retrieval) là một trong những vấn đề cần được nghiên cứu để cung cấp thông tin cho người dùng hiệu quả hơn, đặc biệt là khi ta sở hữu lượng lớn dữ liệu.

1.1.2. Tổng quan về truy xuất thông tin

Truy xuất thông tin (Information Retrieval - IR) là quá trình tìm kiếm và lấy thông tin có liên quan từ một tập hợp lớn dữ liệu, thường là văn bản. Mục tiêu của IR là giúp người dùng tìm thấy thông tin mà họ cần một cách nhanh chóng và hiệu quả. IR có ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm tìm kiếm web, quản lý tài liệu, và các hệ thống hỗ trợ quyết định.

Trong bối cảnh hiện nay, với sự gia tăng nhanh chóng của dữ liệu và thông tin trên mạng, các hệ thống IR đang phải đối mặt với nhiều thách thức. Một trong những vấn đề chính là khả năng xử lý khôi lượng lớn dữ liệu mà vẫn đảm bảo độ chính xác và độ tin cậy của kết quả tìm kiếm. Theo Manning et al. (2008) trong cuốn sách "Introduction to Information Retrieval", các hệ thống IR truyền thống thường dựa vào các mô hình thống kê để đánh giá mức độ liên quan của tài liệu với truy vấn của người dùng [8]. Tuy nhiên, những mô hình này đôi khi không đủ mạnh để xử lý các truy vấn phức tạp hoặc ngữ nghĩa sâu sắc.

Hiện nay, IR vẫn gặp nhiều thách thức, có thể kể đến như:

- **Khả năng hiểu ngữ nghĩa:** Một trong những thách thức lớn nhất trong IR là

khả năng hiểu ngữ nghĩa của câu truy vấn và tài liệu. Nhiều hệ thống hiện tại vẫn gặp khó khăn trong việc nhận diện ý nghĩa thực sự của từ ngữ trong các ngữ cảnh khác nhau.

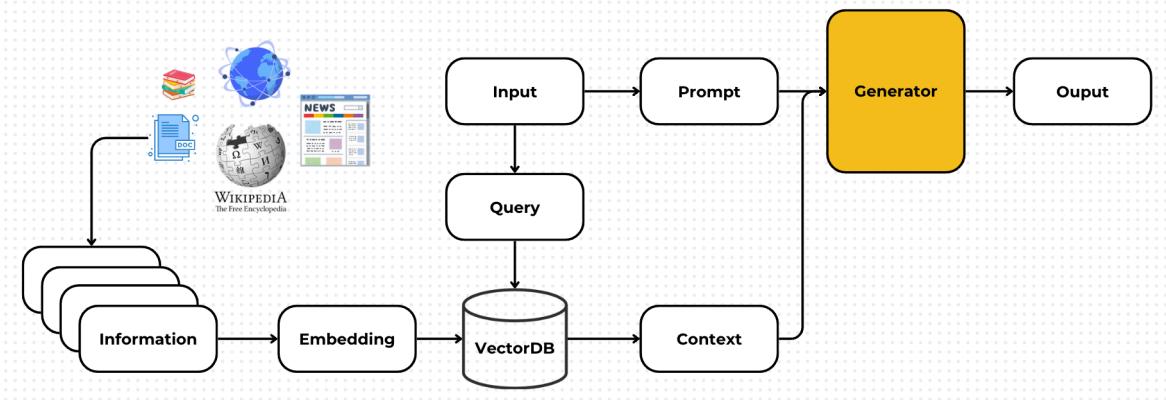
- Độ chính xác và độ bao phủ: Để đánh giá hiệu suất của một hệ thống IR, hai chỉ số quan trọng là độ chính xác và độ bao phủ thường được sử dụng. Độ chính xác đo lường tỷ lệ kết quả trả về là liên quan so với tổng số kết quả trả về, trong khi độ bao phủ đo lường tỷ lệ kết quả liên quan được tìm thấy so với tổng số kết quả liên quan có sẵn. Theo Baeza-Yates & Ribeiro-Neto (2011) trong cuốn sách "Modern Information Retrieval", việc cân bằng giữa hai chỉ số này là rất khó khăn, đặc biệt khi số lượng dữ liệu lớn [9].
- Cập nhật dữ liệu: Với sự phát triển nhanh chóng của thông tin, việc cập nhật dữ liệu trong các hệ thống IR cũng trở thành một thách thức lớn. Các mô hình hiện tại thường dựa vào dữ liệu đã được huấn luyện trước đó, dẫn đến việc thông tin có thể trở nên lỗi thời hoặc không chính xác theo thời gian.

Những thách thức mà các hệ thống IR hiện tại đang phải đối mặt cho thấy sự cần thiết phải tiếp tục nghiên cứu và phát triển công nghệ để đáp ứng nhu cầu ngày càng cao về truy xuất thông tin chính xác và kịp thời trong thế giới ngày nay.

1.1.3. Tổng quan về RAG (Retrieval-Augmented Generation)

Đối với Chatbot thông truyền thống, chúng ta cần huấn luyện trên dữ liệu để Chatbot có thể phản hồi đến người dùng, vấn đề cũng đến từ đó khi Chatbot phản hồi tốt trên dữ liệu được huấn luyện nhưng lại thường cố gắng tạo một câu trả lời “ảo” với các thông tin chưa được huấn luyện [1].

Tuy nhiên, chúng ta hoàn toàn có thể tự cung cấp thông tin mới cho Chatbot, các LLM hoàn toàn có thể hiểu được và đưa ra câu trả lời chính xác dựa trên dữ liệu đó. Nếu có một phương pháp tự động cung cấp thông tin cho LLM và nếu không tìm thấy thì LLM sẽ phản hồi là không có thông tin về vấn đề được hỏi thì sẽ giải quyết được vấn đề này.



Hình 1.1: Mô hình RAG truyền thống

RAG là một kỹ thuật mới nổi trong lĩnh vực xử lý ngôn ngữ tự nhiên, kết hợp giữa việc truy xuất thông tin và sinh văn bản. Lewis và cộng sự (2020) đã chứng minh rằng RAG có thể cải thiện đáng kể chất lượng và độ chính xác của câu trả lời trong các hệ thống hỏi đáp.

RAG giải quyết hai vấn đề chính của LLM truyền thống:

- Độ chính xác và nguồn gốc thông tin: Bằng cách truy xuất thông tin từ một cơ sở dữ liệu tri thức đáng tin cậy, RAG có thể cung cấp câu trả lời với nguồn gốc rõ ràng và có thể kiểm chứng.
- Cập nhật kiến thức: RAG cho phép cập nhật cơ sở dữ liệu tri thức mà không cần đào tạo lại toàn bộ mô hình, giúp LLM luôn có thông tin mới nhất.

RAG gồm 2 phần chính là quá trình xây dựng cơ sở dữ liệu vector và quá trình truy xuất dữ liệu. Trong mỗi phần gồm nhiều bước mà mỗi bước lại có thể thực hiện theo nhiều cách khác nhau từ cơ bản đến nâng cao.

Xây dựng cơ sở dữ liệu Vector

- Trích xuất văn bản (từ tệp, ảnh, các đường dẫn internet)

Quá trình này liên quan đến việc thu thập và chuẩn hóa dữ liệu từ nhiều nguồn khác nhau. Các phương pháp có thể bao gồm kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để trích xuất thông tin có cấu trúc từ văn bản phi cấu trúc, và các thuật toán nhận dạng ký tự quang học (OCR) cho dữ liệu hình ảnh. Ngoài ra còn có thể sử dụng các kỹ thuật scrapping để lấy thông tin từ internet

- *Chia nhỏ văn bản*

Phân đoạn văn bản là một bước quan trọng để tối ưu hóa việc biểu diễn và truy xuất thông tin. Trong việc phân đoạn, xem xét số lượng token là điều quan trọng bởi vì trong mỗi khung chat, mô hình chỉ có xử lý một số lượng văn bản giới hạn.

Trong khi mỗi token chiếm khoảng 4 ký tự, nhiều chiến lược cắt đoạn được đưa ra dựa trên số lượng từ, lượng token, ... Tuy nhiên cũng có các phương pháp tiên tiến hơn bao gồm chia đoạn dựa trên ngữ nghĩa và phân tích cấu trúc đoạn văn.

- *Mã hóa văn bản*

Ở phần này, thường sử dụng các mô hình biểu diễn ngôn ngữ tiên tiến như Transformers, BERT để chuyển đổi văn bản thành biểu diễn vector mật độ cao. Nghiên cứu về các phương pháp nhúng từ ngữ cảnh và biểu diễn ngôn ngữ đa phương thức.

Tuy nhiên, đối với RAG, việc mã hóa văn bản không chỉ cần xem xét xem có mã hoá đủ thông tin không mà còn phải xem xét về kích thước vector đầu ra, khi vector quá lớn có thể dẫn đến làm chậm quá trình truy xuất.

- *Xây dựng quan hệ giữa các vector (nâng cao)*

Bước này chỉ thực hiện đối với cơ sở dữ liệu đồ thị, khi đó, ta cần phải xác định quan hệ giữa các vector với nhau. Áp dụng các kỹ thuật học sâu như graph neural networks để nắm bắt mối quan hệ phức tạp giữa các đơn vị thông tin.

Tích hợp kiến thức từ ontology và knowledge graphs để tăng cường ngữ nghĩa của biểu diễn vector.

- *Lưu vào cơ sở dữ liệu*

Lưu các Vector được mã hóa và quan hệ giữa các Vector vào cơ sở dữ liệu

Truy xuất dữ liệu

- *Mã hóa câu truy xuất*

Việc mã hóa câu truy xuất phải chọn phương pháp giống với mô hình đã mã

hoá thông tin để đảm bảo đúng chiều và kích thước vector.

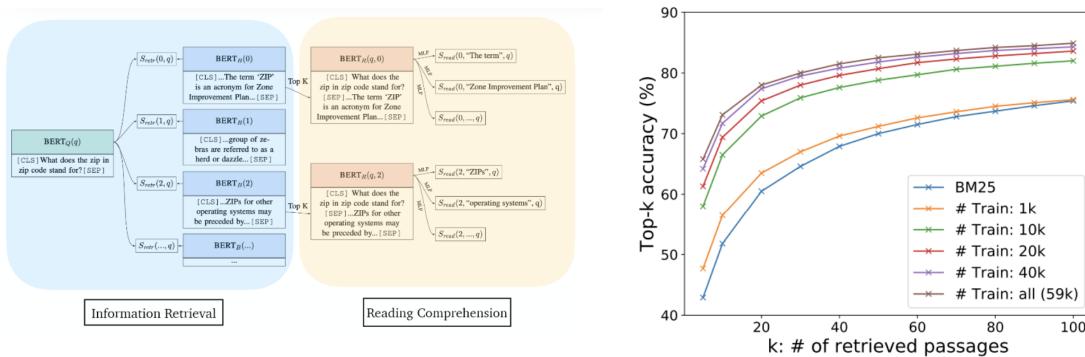
- *Tìm kiếm ứng viên trên cơ sở dữ liệu*

Bằng các phương pháp xử lý ngôn ngữ tự nhiên để truy xuất các Vector trong cơ sở dữ liệu từ Vector truy xuất, các phương pháp đơn giản có thể bắt đầu từ việc tính cosine similarity giữa các Vector.

- *Thứ hạng các ứng viên*

Các ứng viên được tìm thấy từ bước trên có thể không hoàn toàn đúng hay giá trị cosine cao không có nghĩa là Vector đó chứa câu trả lời. Bằng cách xem xét lại các từ khoá, cấu trúc và ngữ nghĩa câu một lần nữa để tìm ra vector thật sự chứa câu trả lời. Có thể đào sâu hơn với các phương pháp như TF-IDF, BM25, PDR, FAISS của Meta hay ColBERT.

Trong việc truy xuất dữ liệu, ý tưởng của nhiệm vụ này chính là so sánh văn bản để tìm kiếm những văn bản giống với yêu cầu của người dùng nhất có thể, cho nhiệm vụ này, rất nhiều hướng giải quyết đã được đưa ra từ lâu, TF-IDF là một trong những phương pháp tiếp cận đơn giản nhất có vấn đề này, tiếp đến là BM25 một phương pháp cải tiến từ TF-IDF để có thể xử lý được những tài liệu lớn hơn khi xem xét thêm thông tin về độ dài văn bản [10]. BM25 cho thấy vượt trội về tốc độ và độ xử lý, dù được xuất bản từ những năm 2009, hiện nay BM25 vẫn là phương pháp được nhiều người lựa chọn.



Hình 1.2: Dense Passage Retrieve (DPR) và kết quả mô hình.

Một trong những phương pháp nâng cao hơn, đào sâu hơn vào vấn đề này đó chính là Dense Passage Retrieve (DPR) được đề xuất vào 2020 khi xây dựng kiến trúc với hai thành phần, truy xuất và phản hồi. Mô hình ban đầu sẽ được huấn

luyện để tìm ra các đoạn văn bản phù hợp nhất dựa trên tập dữ liệu sau đó tiến hành phản hồi câu hỏi dựa trên các văn bản này. Từ kết quả cho thấy, nó vượt trội hơn BM25 về mặt hiệu năng (xem hình 1.2) [11]. Tuy nhiên ở trong phần này, ta sẽ chỉ tập trung vào cách mà DPR truy xuất dữ liệu.

1.1.4. Hạn chế của các hệ thống RAG ở thời điểm hiện tại.

Các phương pháp truy xuất thông tin hiện nay đang đối mặt với hai thách thức chính:

Thứ nhất, về tính chính xác của việc so khớp vector. Mặc dù các phương pháp hiện tại có khả năng biểu diễn thông tin khá tốt, nhưng vẫn tồn tại sai lệch do vector đại diện cho câu hỏi không thể hoàn toàn trùng khớp với vector đại diện cho câu trả lời trong cơ sở dữ liệu.

Thứ hai, về giới hạn của vector biểu diễn cố định. Các phương pháp mã hóa hiện tại thường tạo ra vector với số chiều cố định, bất kể độ dài hay mức độ phức tạp của văn bản đầu vào. Điều này dẫn đến hai vấn đề:

- Việc nén thông tin vào một vector kích thước cố định có thể làm mất mát thông tin chi tiết của văn bản gốc.
- Trong khi đó, nếu tăng số chiều của vector để lưu trữ nhiều thông tin hơn sẽ làm tăng đáng kể chi phí về tài nguyên, cả trong việc lưu trữ lẫn trong quá trình tính toán khi thực hiện truy xuất thông tin.

Rõ ràng việc truy xuất dữ liệu với đa số các hướng tiếp cận đều theo ý tưởng mã hóa câu truy vấn, tài liệu và so sánh hai vector này với nhau, tuy nhiên làm như vậy có thể sẽ bị bỏ qua nhiều từ khoá và thông tin giữa các văn bản với nhau, do đó Khattab phát triển ColBERT với ý tưởng có thể đào sâu hơn việc so sánh các từ trong tài liệu và câu truy vấn với nhau [12]. Cuối cùng các ứng viên thông tin tìm được từ cơ sở dữ liệu sẽ đưa vào LLM để cung cấp thông tin cho câu trả lời.

Thêm vào đó, các hệ thống truy xuất thường sử dụng trực tiếp vector đại diện câu hỏi của người dùng để truy xuất mà không qua xử lý, khi người dùng đưa ra một câu hỏi cần phải tổng hợp từ nhiều nguồn, việc truy xuất và lấy một lượng

thông tin cố định sẽ dẫn đến thiếu sót và không đảm bảo toàn vẹn câu trả lời.

1.1.5. *Dánh giá kết quả hệ thống RAG*

Trong hệ thống Retrieval-Augmented Generation (RAG), quá trình đánh giá kết quả đóng vai trò quan trọng và cần được thực hiện một cách toàn diện trên hai khía cạnh chính: phần truy xuất thông tin và phần sinh kết quả phản hồi. Mỗi khía cạnh này đòi hỏi một bộ tiêu chí đánh giá riêng biệt, phản ánh đặc thù và mục tiêu cụ thể của từng giai đoạn trong quy trình RAG.

Đối với phần truy xuất thông tin, các chỉ số đánh giá tập trung vào khả năng hệ thống tìm kiếm và lựa chọn những thông tin có liên quan và chính xác từ cơ sở dữ liệu. Các metric phổ biến trong nhóm này bao gồm các chỉ số không dựa trên thứ hạng như độ chính xác, độ bao phủ ở top k kết quả. Những phương pháp này đánh giá khả năng hệ thống trong việc xác định và trích xuất thông tin phù hợp mà không quan tâm đến thứ tự xuất hiện của chúng. Bên cạnh đó, các metric dựa trên thứ hạng như MRR và MAP được sử dụng để đánh giá không chỉ sự phù hợp của thông tin được truy xuất mà còn cả thứ tự ưu tiên của chúng trong danh sách kết quả. Ngoài ra, các chỉ số đặc thù như Misleading Rate, Mistake Reappearance Rate, và Error Detection Rate cũng được áp dụng để đánh giá sâu hơn về độ tin cậy và khả năng phát hiện lỗi của hệ thống truy xuất.

Về phần đánh giá kết quả phản hồi, tức là phần sinh nội dung, các metric tập trung vào việc đánh giá chất lượng văn bản được tạo ra. ROUGE là một trong những metric quan trọng, đánh giá chất lượng tóm tắt bằng cách so sánh với các tham chiếu do con người tạo ra. BLEU được sử dụng rộng rãi trong đánh giá chất lượng dịch máy, nhưng cũng có thể áp dụng cho việc đánh giá nội dung được sinh ra. BertScore là một metric tiên tiến hơn, sử dụng embeddings từ các mô hình transformer để đánh giá sự tương đồng ngữ nghĩa giữa văn bản được tạo ra và văn bản tham chiếu. Một hướng tiếp cận mới và đầy hứa hẹn là việc sử dụng các LLM để đánh giá chất lượng văn bản, có khả năng xem xét các khía cạnh phức tạp như tính mạch lạc, sự liên quan, và tính trôi chảy của nội dung.

Ngoài các phương pháp tự động, đánh giá bởi con người vẫn đóng vai trò

quan trọng trong việc xác định chất lượng và hiệu quả tổng thể của hệ thống RAG. Đánh giá của con người có thể cung cấp những hiểu biết sâu sắc về tính hữu ích, sự phù hợp và khả năng đáp ứng nhu cầu thực tế của người dùng, những yếu tố mà các metric tự động có thể chưa nắm bắt được một cách toàn diện.

Việc kết hợp đa dạng các phương pháp và metric đánh giá này không chỉ giúp đánh giá toàn diện hiệu suất của hệ thống RAG mà còn cung cấp cái nhìn sâu sắc về các khía cạnh cần cải thiện. Điều này đặc biệt quan trọng trong bối cảnh ứng dụng RAG vào lĩnh vực giáo dục, nơi mà độ chính xác, tính cập nhật và sự phù hợp của thông tin là yếu tố then chốt để đảm bảo chất lượng học tập và sự tin cậy của hệ thống hỗ trợ học tập tự động [13].

1.1.6. Tổng quan về phân rã câu hỏi

Phân rã câu hỏi là một kỹ thuật tiên tiến trong lĩnh vực xử lý ngôn ngữ tự nhiên. Phương pháp này được thiết kế nhằm chuyển đổi các câu hỏi phức tạp, đa chiều thành các câu hỏi con nhỏ gọn, dễ xử lý, qua đó nâng cao đáng kể hiệu suất và độ chính xác của các hệ thống xử lý thông tin.

Thông thường, quá trình phân rã câu hỏi bao gồm các bước chính sau:

- Nhận Diện Cấu Trúc: Phân tích cấu trúc ngữ nghĩa và cú pháp của câu hỏi gốc.
- Xác Định Thành Phần: Tách biệt các yếu tố thông tin quan trọng trong câu hỏi.
- Tạo Câu Hỏi Con: Xây dựng các câu hỏi nhỏ hơn, tập trung và cụ thể.
- Tích Hợp Kết Quả: Tổng hợp các câu trả lời của từng câu hỏi con để đưa ra phản hồi toàn diện.

Ở thời điểm hiện tại phân rã câu hỏi chủ yếu được sử dụng để cải thiện hiệu suất của các mô hình text-to-sql được đề cập trong bài báo như “*Semantic Decomposition of Question and SQL for Text-to-SQL Parsing*” cho thấy hiệu suất của mô hình tốt hơn đáng kể khi sử dụng kỹ thuật này. Khi áp dụng kỹ thuật này, các hệ thống có thể chuyển đổi các truy vấn phức tạp bằng ngôn ngữ tự nhiên thành

những câu truy vấn SQL chính xác, mở ra những khả năng mới trong việc tương tác với cơ sở dữ liệu. Điều này không chỉ là một bước tiến về mặt kỹ thuật, mà còn là một bước đột phá trong cách con người tương tác với hệ thống thông tin [14].

Đa số các phương pháp hiện tại đều sử dụng LLM để phân rã, tất nhiên, việc sử dụng LLM phân rã mang lại hiệu quả cực kỳ lớn nhưng cũng khiến cho thời gian xử lý tăng lên đáng kể và làm chậm toàn bộ tiến trình [15]. Một số ít bài báo tìm ra giải pháp khác để phân rã dựa trên ngữ nghĩa nhưng cũng phụ thuộc nhiều vào phương pháp xác định thực thể, ta cần có mô hình xác định thực thể hoạt động tốt đối với những dạng này, được đề cập trong “*Complex Question Decomposition for Semantic Parsing*” [16].

Phân rã câu hỏi không chỉ là một công nghệ, mà còn là một cánh cửa mở ra những khả năng vô tận. Tưởng tượng một tương lai où các hệ thống AI có thể hiểu và xử lý thông tin một cách tinh vi, gần như như con người, đó chính là hướng phát triển mà các nhà nghiên cứu đang không ngừng theo đuổi.

Từ việc hỗ trợ tra cứu thông tin, phân tích dữ liệu, cho đến việc phát triển các trợ lý ảo thông minh, phân rã câu hỏi sẽ là một trong những công nghệ then chốt định hình tương lai của trí tuệ nhân tạo. Phân rã câu hỏi chính là minh chứng cho sự tiến hóa không ngừng của trí tuệ nhân tạo. Không phải là một cuộc cách mạng ồn ào, mà là một sự thay đổi âm thầm, sâu sắc, đang từng bước định hình lại cách chúng ta tương tác và hiểu thông tin.

Chúng ta đang chứng kiến một kỷ nguyên mới, nơi máy móc không chỉ xử lý thông tin, mà còn có khả năng hiểu được bản chất phức tạp của ngôn ngữ con người. Và phân rã câu hỏi, với tất cả sự tinh tế và sức mạnh của mình, sẽ tiếp tục là một trong những chìa khóa quan trọng mở ra các khả năng vô tận này.

1.2. Cơ sở lý thuyết

1.2.1. Long Short-Term Memory (LSTM)

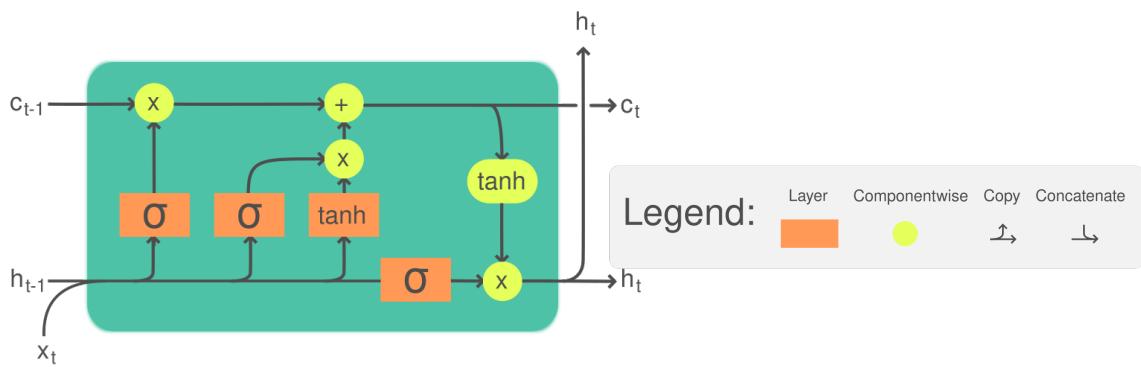
LSTM (Long Short-Term Memory) là một kiến trúc mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) được phát triển để giải quyết các hạn chế của các mạng RNN truyền thống, đặc biệt là vấn đề mất mát gradient và khả năng ghi

nhớ thông tin dài hạn [17].

Động Lực Phát Triển: Các mạng RNN truyền thống gặp phải hai vấn đề chính:

- **Vấn Đề Mất Mát Gradient:** Trong quá trình lan truyền ngược, các gradient trở nên rất nhỏ hoặc rất lớn, dẫn đến việc mất mát thông tin quan trọng hoặc không thể huấn luyện hiệu quả.
- **Khó Khăn Trong Ghi Nhớ Thông Tin Dài Hạn:** Các mạng RNN không thể lưu trữ hiệu quả thông tin từ các bước thời gian xa.

LSTM được thiết kế với cấu trúc phức tạp hơn so với RNN truyền thống, bao gồm ba cổng chính và một ô nhớ (cell state):



Hình 1.3: Cơ chế LSTM

Cổng Quên (forget gate): Cổng này quyết định thông tin nào sẽ bị loại bỏ khỏi ô nhớ. Sử dụng hàm sigmoid để tạo ra các giá trị từ 0 đến 1, trong đó:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f).$$

Như vậy, giá trị càng nhỏ thì loại bỏ hoàn toàn thông tin càng nhiều và ngược lại khi giá trị càng tiệm cận đến 1.

Cổng Đầu Vào (Input Gate): Cổng này kiểm soát việc thêm thông tin mới vào ô nhớ.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C).$$

Trong đó:

- i_t xác định mức độ cập nhật.
- \tilde{C}_t tạo ra các giá trị ứng viên để thêm vào.

Cổng Đầu Ra (Output Gate): Cổng này quyết định thông tin nào sẽ được xuất ra từ trạng thái hiện tại.

$$o_t = \sigma \left(W_o \cdot [h_{t-1}, x_t] + b_o \right); h_t = o_t \cdot \tanh(C_t).$$

Quá trình cập nhật ô nhớ: theo công thức sau, trạng thái ô nhớ sẽ được cập nhập bằng cách nhân trạng thái cũ với cổng quên và thêm thông tin mới từ cổng đầu vào.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

LSTM giải quyết hiệu quả hai vấn đề then chốt trong học sâu. Về mặt gradient, kiến trúc này cho phép gradient luân chuyển dễ dàng hơn so với các mạng RNN truyền thống, nhờ cơ chế cổng tinh vi. Điều này giúp mô hình học được các phụ thuộc dài hạn, vượt qua giới hạn của các mạng nơ-ron hồi quy cổ điển.

Tính linh hoạt trong lưu trữ thông tin là ưu điểm nổi bật tiếp theo. LSTM có khả năng lựa chọn thông tin một cách có chủ đích - có thể lưu trữ, ghi nhớ hoặc loại bỏ dữ liệu một cách có chọn lọc. Đặc tính này làm cho LSTM đặc biệt hiệu quả khi xử lý các chuỗi dài và phức tạp.

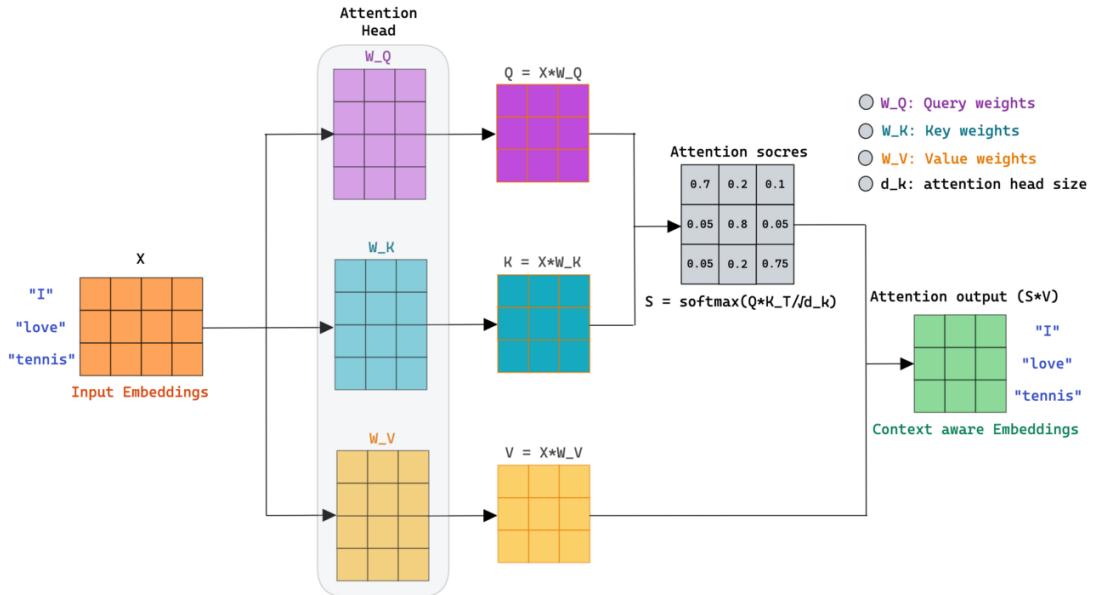
1.2.2. Cơ chế Tự Chú Ý

Cơ chế Tự Chú Ý được giới thiệu lần đầu tiên trong bài báo mang tính bước ngoặt "Attention Is All You Need". Trong khi các mô hình truyền thống gặp phải các hạn chế quan trọng: Xử lý tuần tự, không thể song song hóa tính toán, khó nắm bắt các phụ thuộc dài hạn và mất mát thông tin trong quá trình lan truyền. Cơ chế tự chú ý được đề cập trong kiến trúc Transformers là nền tảng quan trọng cho nhiều kiến trúc xử lý ngôn ngữ tự nhiên sau này [18].

Cơ chế Self-Attention được định nghĩa thông qua ba phép biến đổi chính:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V.$$

Trong đó Q, K, V lần lượt là Query matrix, Key matrix và Value matrix, với d_k là dimension của key vectors. Đầu vào để tính attention sẽ bao gồm ma trận Q, K là hai ma trận được dùng để tính mức độ chú ý (attention) mà các từ trong câu trả về cho một từ cụ thể. Vector này sẽ được tính dựa trên trung bình trọng số trong ma trận V với trọng số attention.



Hình 1.4: Cơ chế tự chú ý

Ý nghĩa sâu xa của phương trình này nằm ở khả năng tính toán sự tương quan giữa các phần tử trong chuỗi. Việc sử dụng hàm softmax và chia cho $\sqrt{d_k}$ giúp ổn định quá trình huấn luyện và ngăn chặn sự suy giảm gradient, tránh tràn luồng nếu số mũ quá lớn.

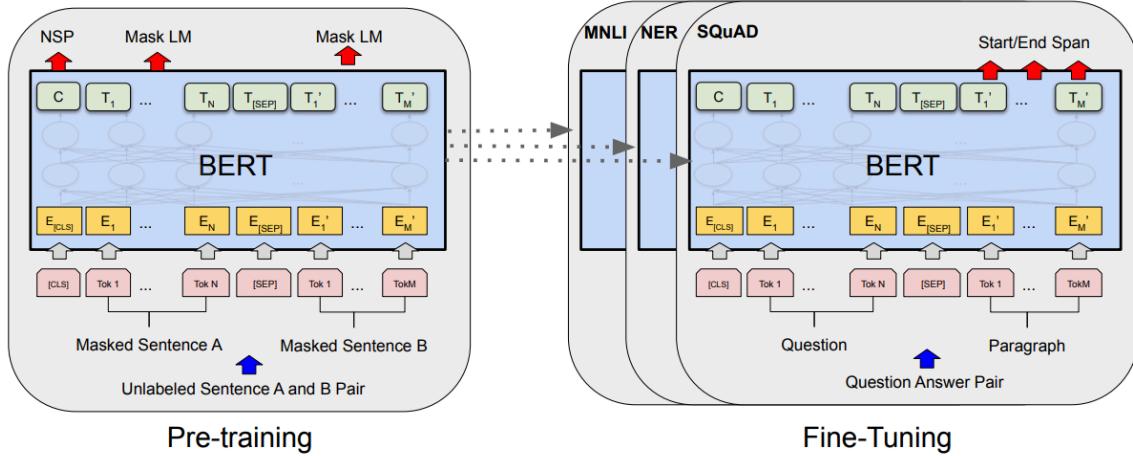
Đóng góp quan trọng nhất của Self-Attention không chỉ nằm ở tính toán học thuận tay, mà còn ở khả năng mở rộng và linh hoạt. Khái niệm Multi-Head Attention cho phép mô hình học các biểu diễn đa chiều, mỗi "đầu" có thể nắm bắt các khía cạnh khác nhau của dữ liệu.

1.2.3. Mô hình BERT

Trong xử lý ngôn ngữ tự nhiên, mã hoá thông tin một cách đầy đủ, thể hiện nhiều mặt của câu bằng vector là nhiệm vụ đặc biệt quan trọng, ảnh hưởng đến chất lượng mô hình cuối cùng. Mã hoá thông tin văn bản (text embedding) gồm nhiều công đoạn, một vector được xem là biểu diễn tốt một đoạn văn bản cần phải đạt

được các yêu cầu: thể hiện được vị trí các từ trong câu, thể hiện được ý nghĩa của các từ trong câu và thể hiện mối quan hệ giữa các từ trong câu [19].

BERT (Bidirectional Encoder Representations from Transformers) là một kiến trúc học sâu được phát triển bởi Google Research vào năm 2018, đánh dấu một bước tiến quan trọng trong việc áp dụng pre-trained language models cho các tác vụ xử lý ngôn ngữ tự nhiên. Kiến trúc này dựa trên transformer encoders và được thiết kế để học biểu diễn ngữ cảnh hai chiều của văn bản thông qua cơ chế tự chú ý (self-attention) đa lớp. BERT sử dụng kiến trúc Transformer, cùng với một số cải tiến để đáp ứng yêu cầu của bài toán học máy dựa trên ngôn ngữ. Các lớp Encoder của Transformer được xếp chồng lên nhau để tạo thành kiến trúc mạng nơ-ron sâu. Sự độc đáo của BERT nằm ở khả năng xử lý cả ngữ cảnh của từ trong câu bằng cách sử dụng cả hai hướng (từ trái sang phải và từ phải sang trái) thay vì chỉ một hướng như các mô hình trước đó [19].



Hình 1.5: Mô hình BERT [19]

BERT sử dụng WordPiece tokenization để phân tách văn bản thành các token cơ bản. Mỗi token được biểu diễn bởi tổng của ba loại embedding: Token Embeddings biểu diễn ngữ nghĩa của token, Segment Embeddings phân biệt các câu trong cặp câu đầu vào, và Position Embeddings mã hóa vị trí tương đối của token trong câu. Như đã trình bày trong phần trước về Cơ chế Tự Chú Ý, phép toán attention được định nghĩa thông qua các ma trận Q , K , V . Trong bối cảnh của BERT, khái niệm này được mở rộng thành Multi-Head Attention, cho phép mô hình học

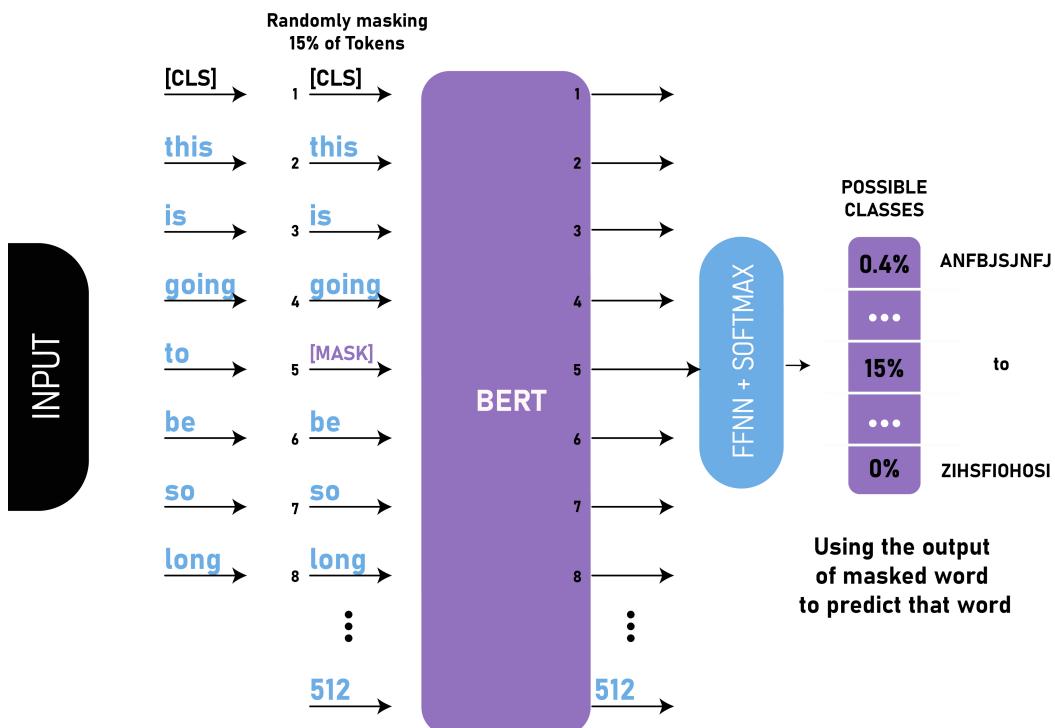
các biểu diễn đa chiều. Mỗi "head" trong Multi-Head Attention sẽ được tính toán như sau:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O.$$

Trong đó mỗi head được tính bằng:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V).$$

Khi áp dụng vào kiến trúc BERT, Multi-Head Attention không chỉ là một phép toán thuần túy, Khi áp dụng vào kiến trúc BERT, Multi-Head Attention không chỉ là một phép toán thuần túy, mà còn là cơ chế quan trọng giúp mô hình nắm bắt được các mối quan hệ phức tạp giữa các từ trong văn bản.



Hình 1.6: Mask language model

Quay trở lại quá trình pre-training, BERT sử dụng hai nhiệm vụ chính: Masked Language Modeling (MLM) và Next Sentence Prediction (NSP). Trong nhiệm vụ MLM, 15% tokens trong văn bản đầu vào được che ngẫu nhiên với:

- 80% được thay thế bằng token [MASK].
- 10% được thay thế bằng một token ngẫu nhiên.

- 10% giữ nguyên.

Mục tiêu của MLM là dự đoán các token bị che dựa trên ngữ cảnh hai chiều, tận dụng tối đa khả năng của cơ chế self-attention trong việc học biểu diễn ngữ nghĩa và mối quan hệ giữa các từ.

Như vậy, từ cơ chế self-attention ban đầu, BERT đã phát triển và mở rộng để trở thành một mô hình pre-training mạnh mẽ, có khả năng học biểu diễn ngôn ngữ sâu sắc và linh hoạt.

Điểm mạnh của BERT:

- **Hiểu Ngữ Cảnh Toàn Diện:** BERT có khả năng hiểu ngữ cảnh của từ trong một câu một cách toàn diện, giúp cải thiện hiệu suất cho nhiều tác vụ NLP.
- **Sử Dụng Được Cho Nhiều Nhiệm Vụ:** BERT không chỉ được sử dụng cho các tác vụ như phân loại văn bản hay dự đoán từ tiếp theo, mà còn có thể được fine-tune cho các nhiệm vụ cụ thể nhận dạng thực thể, phân tích ngữ nghĩa, và nhiều hơn nữa.
- **Mã Nguồn Mở:** Việc Google AI cung cấp mã nguồn mở của BERT đã tạo điều kiện thuận lợi cho cộng đồng nghiên cứu và phát triển trong lĩnh vực NLP.

1.2.4. Mô hình Seq2Seq (Sequence to Sequence)

Seq2Seq là một kiến trúc mô hình học sâu được thiết kế để chuyển đổi các chuỗi đầu vào thành các chuỗi đầu ra, đặc biệt hiệu quả trong các bài toán dịch máy, tóm tắt văn bản, và sinh văn bản. Mô hình Seq2Seq bao gồm hai thành phần chính: Encoder và Decoder. Encoder được sử dụng để mã hóa chuỗi đầu vào thành một biểu diễn vector ngữ cảnh, trong khi Decoder sử dụng vector này để sinh ra chuỗi đầu ra [20].

Công thức toán học mô tả quá trình ánh xạ từ chuỗi đầu vào X sang chuỗi đầu ra Y được biểu diễn như sau:

$$P(Y | X) = \prod_{t=1}^{|Y|} P(y_t | y_{<t}, c).$$

Trong đó:

- c là vector ngữ cảnh được sinh ra bởi encoder.
- $y_{<t}$ là các token được sinh ra trước đó

Kiến trúc Encoder của Seq2Seq thường sử dụng các mạng nơ-ron hồi quy như LSTM hoặc GRU để xử lý chuỗi đầu vào. So với RNN truyền thống, LSTM giải quyết được vấn đề mất mát gradient và khả năng lưu trữ thông tin dài hạn, làm tăng hiệu quả của encoder.

Quá trình mã hóa được thực hiện như sau:

$$h_t = \text{LSTM}(x_t, h_{t-1}).$$

Trong đó:

- h_t : là trạng thái tại thời điểm t .
- x_t : Đầu vào tại thời điểm t .
- h_{t-1} : Trạng thái ẩn tại thời điểm trước đó.
- $LSTM$: Biểu diễn cơ chế của một LSTM.

Ở giai đoạn decoder, vector ngữ cảnh được sử dụng như điểm khởi đầu cho việc sinh từng token đầu ra. Quá trình này có thể được mô tả bằng công thức:

$$y_t = \text{Decoder}(y_{t-1}, c, h_{t-1}).$$

- y_t : Đầu ra tại thời điểm t .
- y_{t-1} : Đầu vào tại thời điểm t .
- c : Vector ngữ cảnh (context vector) từ encoder.
- h_{t-1} : Trạng thái ẩn tại thời điểm trước đó.
- Decoder: Hàm biểu diễn cơ chế của bộ giải mã (decoder).

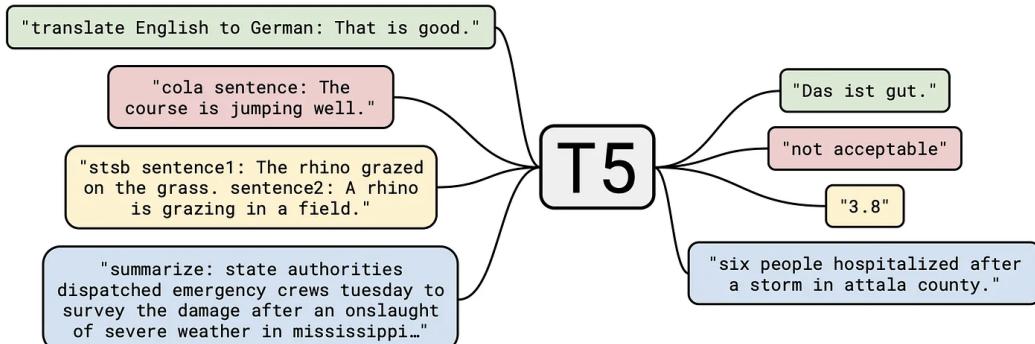
Một trong những cải tiến quan trọng của Seq2Seq là việc áp dụng cơ chế

Attention, cho phép mô hình tập trung vào các phần khác nhau của chuỗi đầu vào khi sinh từng token đầu ra. Sự kết hợp giữa LSTM và Attention đã làm tăng đáng kể hiệu quả của mô hình Seq2Seq trong các bài toán.

Điểm mạnh của kiến trúc này nằm ở khả năng: Xử lý các chuỗi đầu vào có độ dài thay đổi, nắm bắt các mối quan hệ ngữ cảnh phức tạp, linh hoạt trong việc ánh xạ giữa các không gian ngôn ngữ khác nhau.

1.2.5. Mô hình T5 (Text to Text transfer Transformer)

Mô hình T5, được phát triển bởi nhóm nghiên cứu của Google, đại diện cho một paradigm mới trong lĩnh vực xử lý ngôn ngữ tự nhiên. Nghiên cứu này xuất phát từ nhận thức về sự hạn chế của các mô hình tiền nhiệm trong việc xử lý đa dạng các nhiệm vụ ngôn ngữ. Đóng góp cơ bản của T5 nằm ở việc thiết lập một khuôn khổ thống nhất cho các bài toán xử lý ngôn ngữ. Khác biệt then chốt so với các mô hình trước đây là T5 chuyển đổi mọi nhiệm vụ NLP thành một bài toán chuyển đổi văn bản (text-to-text), sử dụng kiến trúc Transformer encoder-decoder hoàn chỉnh [21].



Hình 1.7: Mô hình T5 [21]

Có thể xem T5 chính là nhiều mô hình Seq2Seq kết hợp với nhau và sử dụng từ khoá bắt đầu để xác định nhiệm vụ tạo chuỗi tiếp theo từ đầu vào. Phương pháp pre-training của T5 mang tính đột phá với mục tiêu khử nhiễu. Quá trình này liên quan đến việc che hoặc nhiễu một phần tokens trong văn bản (khoảng 15-20%), với mục tiêu khôi phục văn bản gốc. Điểm khác biệt so với BERT là phương pháp này được áp dụng đồng thời cho cả encoder và decoder.

Tính linh hoạt của T5 thể hiện qua khả năng chuyển đổi các nhiệm vụ NLP

khác nhau thành một dạng thức thống nhất. Ví dụ điển hình bao gồm việc chuyển đổi các tác vụ như dịch máy, phân loại văn bản, hay tóm tắt văn bản thành một quy trình chuyển đổi văn bản nhất quán.

Mặc dù mang lại nhiều tiến bộ, T5 vẫn đối mặt với một số thách thức. Nhu cầu về tài nguyên tính toán lớn, khả năng thiên lệch do dữ liệu huấn luyện, và độ phức tạp trong việc tinh chỉnh cho từng nhiệm vụ cụ thể là những hạn chế đáng lưu ý. Trong bối cảnh phát triển của các mô hình học sâu, T5 đánh dấu một bước chuyển quan trọng. Mô hình không chỉ kế thừa mà còn tổng hợp và mở rộng các ý tưởng từ các nghiên cứu tiền nhiệm như BERT, Transformer, góp phần định hình xu hướng nghiên cứu trong lĩnh vực xử lý ngôn ngữ tự nhiên. Ý nghĩa sâu sắc của T5 nằm ở việc đề xuất một framework thống nhất, cho phép các nhà nghiên cứu tiếp cận các bài toán NLP một cách linh hoạt và toàn diện hơn, vượt qua các giới hạn của các mô hình truyền thống.

1.2.6. BM25 (Best Match 25)

BM25 là thuật toán tìm kiếm dựa trên mô hình xác suất, có thể xem BM25 là phiên bản nâng cao của TF-IDF khi sử dụng ý tưởng tương tự nhưng lại có cải tiến và khắc phục hạn chế về số lượng và trọng số của các từ khoá khi xử lý lượng lớn thông tin [10].

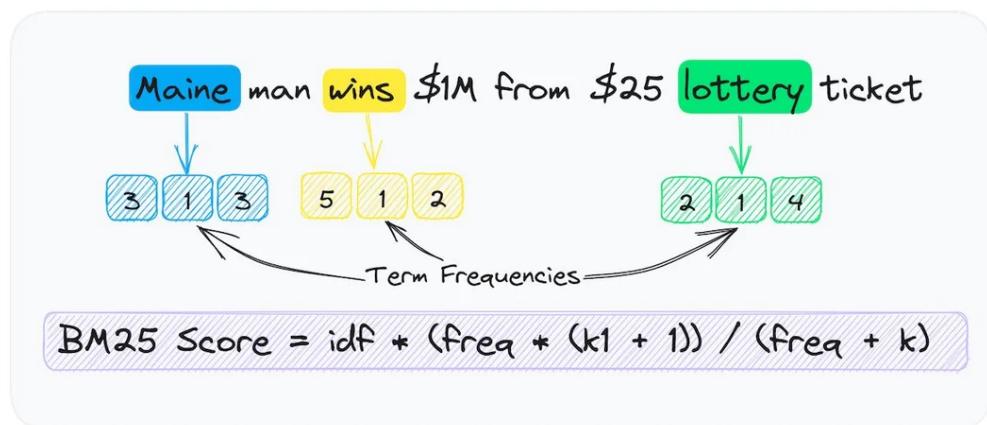
Để tìm kiếm một tài liệu liên quan với yêu cầu người dùng, chúng ta có công thức BM25 như sau:

$$\text{BM25}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i, D) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{d_{\text{avg}}})}.$$

Trong đó:

- $f(q_i, D)$: là số lần mà term q_i xuất hiện trong tất cả các tài liệu D .
- $|D|$: là số từ trong tất cả các tài liệu D .
- d_{avg} : là số lượng từ trung bình trong mỗi tài liệu.
- b và k_1 là các tham số của BM25.

Tham số k_1 trong BM25 đóng vai trò then chốt trong việc điều chỉnh ảnh hưởng của tần suất từ khóa đối với độ liên quan của tài liệu. Bản chất của tham số này nằm ở khả năng kiểm soát quá trình bão hòa tuyến tính của trọng số từ. Khi giá trị k_1 tăng lên, đường cong ảnh hưởng của từ khóa trở nên phẳng hơn, có nghĩa là mỗi lần xuất hiện của từ sẽ có tác động giảm dần đến điểm số của tài liệu. Điều này ngăn chặn hiện tượng những từ xuất hiện quá nhiều sẽ áp đảo hoàn toàn điểm số, đảm bảo tính cân bằng trong việc đánh giá độ liên quan.



Hình 1.8: Mô hình BM25 trong truy xuất thông tin

Ví dụ, với k_1 thấp, sự xuất hiện lần thứ 10 của một từ khóa sẽ có tác động rất nhỏ so với lần xuất hiện đầu tiên. Ngược lại, với k_1 cao, ảnh hưởng của những lần xuất hiện sau vẫn còn đáng kể, giúp các tài liệu có nhiều từ khóa không bị phạt quá mức. Bản chất toán học của k_1 cho phép BM25 tạo ra một cơ chế trọng số tinh vi, vượt trội so với TF-IDF truyền thống. Thuật toán này không chỉ đơn thuần đếm số lần xuất hiện, mà còn xem xét bối cảnh và mức độ ảnh hưởng của từng lần xuất hiện.

Trong thực tế, việc lựa chọn giá trị k_1 phù hợp đòi hỏi sự tinh chỉnh và thử nghiệm. Các nhà nghiên cứu thường sử dụng các kỹ thuật như kiểm định chéo để tìm ra giá trị tối ưu cho từng tập dữ liệu cụ thể, đảm bảo thuật toán BM25 hoạt động hiệu quả nhất.

1.2.7. ColBERT (Contextualized Late Interaction over BERT)

ColBERT là một mô hình đột phá trong việc tối ưu hóa truy vấn và xử lý thông tin, được phát triển dựa trên kiến trúc ngôn ngữ BERT [12]. Mô hình này tập

trung vào việc mã hóa ngữ cảnh và tương tác chi tiết giữa các từ trong truy vấn và tài liệu. Trong bối cảnh phát triển của các hệ thống truy xuất thông tin, phương pháp DPR truyền thống tồn tại những hạn chế đáng kể. Nguyên nhân chính đến từ hạn chế của việc mã hóa thông tin khi các mô hình thông thường nén toàn bộ văn bản thành một vector với số chiều cố định, dẫn đến việc:

- Mất đi chi tiết ngữ nghĩa phong phú.
- Không thể nắm bắt được mối quan hệ tinh tế giữa các từ.
- Giảm khả năng phân biệt các ngữ cảnh phức tạp.

ColBERT đã giải quyết các hạn chế trên thông qua phương pháp mã hóa token độc đáo, thay vì nén văn bản thành một vector duy nhất ColBERT thực hiện mã hóa từng token thành các vector riêng biệt, việc này đảm bảo duy trì thông tin ngữ cảnh cho mỗi từ và cho phép tương tác chi tiết giữa các token của truy vấn và tài liệu.

$$\text{Relevance}(Q, D) = \sum_{i=1}^n \max (\text{sim}(q_i, D)).$$

Trong đó:

- Q_i : Vector của từng token trong truy vấn.
- D_j : Vector của từng token trong tài liệu.
- $\text{sim}()$: Hàm tính độ tương đồng (thường là cosine similarity).

Quá trình xử lý trong ColBERT trải qua nhiều bước, đầu tiên cần phải chuyển văn bản thành các token, tiếp đến chúng ta mã hóa các token này thay vì mã hóa toàn bộ câu, như vậy mỗi token sẽ được đại diện với một vector với số chiều tương ứng với phương pháp mã hóa, ví dụ nếu mã hóa bằng BERT-base, mỗi token sẽ được biểu diễn bằng một vector 768 chiều với đầy đủ ngữ nghĩa, thông tin chi tiết và khả năng hiểu sâu, phân biệt cao.

Khi thực hiện truy vấn, ColBERT thực hiện so sánh từng vector token của câu truy vấn với các token trong kho tài liệu sau đó. Xem ví dụ mô tả chi tiết

cách ColBERT xử lý truy vấn (ở đây tôi biểu diễn cách tính điểm tương đồng giữa hai câu):

Câu truy vấn: “When did the Transformers cartoon series come out”.

Văn bản: “The animate Transformers was released in August 1986”.

Đầu tiên, ta cần phân tách câu thành các token, bước này có thể sử dụng phương pháp WordPiece, tuy nhiên để đơn giản hóa ví dụ, tôi sẽ thực hiện tách token theo các từ. Như vậy ta sẽ có:

Token cho câu truy vấn:

- [CLS]
- when
- did
- ...
- out
- [SEP]

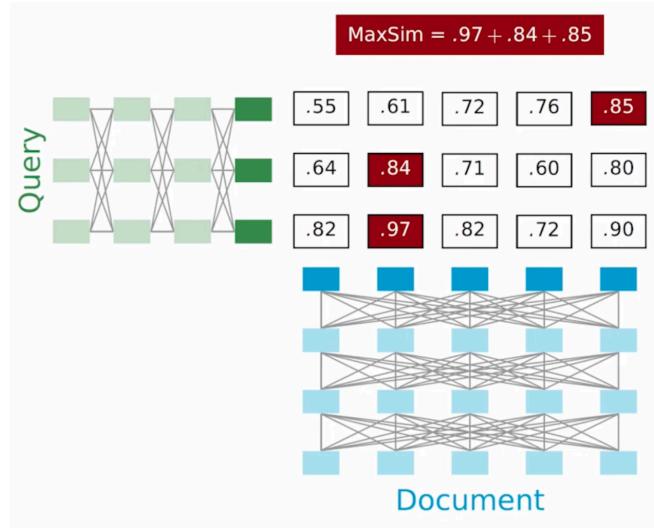
Token cho văn bản tương tự:

- [CLS]
- the
- ...
- [SEP]

Mỗi token trên sẽ được mã hoá bằng một phương pháp bất kỳ, ở đây ví dụ với mã hoá bằng BERT ta sẽ có:

$$\text{Embedding}(x_i) = \text{BERT}(\theta)(x_i) \in \mathbb{R}^{768}.$$

Sau khi mã hoá ta sẽ có tập hợp các vector đại diện cho một câu truy vấn hay tập hợp vector đại diện cho đoạn văn bản chứa thông tin, mỗi vector này có 768 chiều và sẽ được lưu trữ vào cơ sở dữ liệu.



Hình 1.9: Mô hình ColBERT

Tiếp theo, ta cần tính độ tương đồng giữa các vector token và lấy kết quả cao nhất, mỗi token trong câu truy vấn sẽ được so sánh với các token trong văn bản và lấy điểm số cao nhất cho mỗi token, sau đó ta tính tổng các giá trị này để được kết quả cuối cùng. Tương tự thực hiện với rẽn toàn bộ cơ sở dữ liệu, ta có được kết quả cuối cùng khi văn bản có điểm cao nhất sẽ là văn bản liên quan nhất.

Tính điểm số tổng thể:

$$\begin{aligned}
 \text{Relevance}(Q, D) = & \max (\text{sim}(\text{When}, D)) \\
 & + \max (\text{sim}(\text{did}, D)) \\
 & + \max (\text{sim}(\text{the}, D)) \\
 & + \max (\text{sim}(\text{Transformers}, D)) \\
 & + \max (\text{sim}(\text{cartoon}, D)) \\
 & + \max (\text{sim}(\text{series}, D)) \\
 & + \max (\text{sim}(\text{come}, D)) \\
 & + \max (\text{sim}(\text{out}, D)) .
 \end{aligned}$$

Ta thấy, với phương pháp xử lý như vậy mô hình có thể nhận thấy được điểm tương đồng rất cao giữa các cặp từ như “*cartoon*” và “*animate*” hay từ “*when*” với “*August*” hay “*1986*” và từ “*come*” với “*released*”. Như vậy, có thể thấy ColBERT là một phương pháp tinh vi, nắm bắt sát ý nghĩa và giảm thiểu mất mát thông tin khi xử lý độc lập từng token và không bị ràng buộc bởi thứ tự các từ trong câu, tuy nhiên việc xem xét và mã hóa ở cấp độ token thật sự tiêu tốn nhiều tài

nguyên trong lưu trữ và tính toán cũng như tốc độ truy vấn của mô hình.

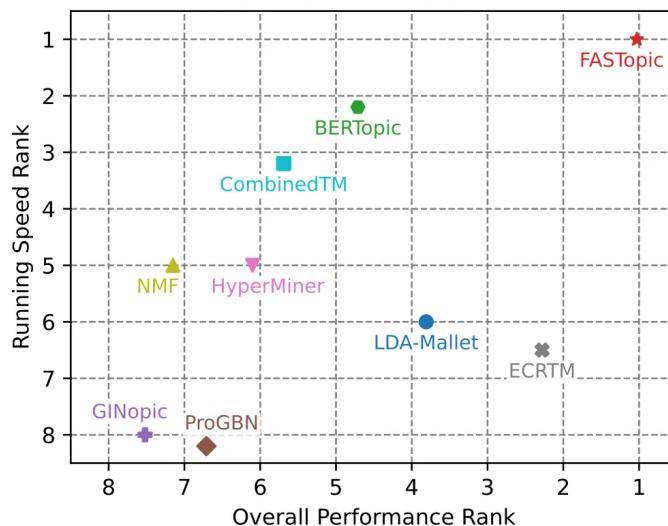
1.2.8. Mô hình chủ đề FASTopic

Việc phân tích và rút trích chủ đề từ tập văn bản lớn luôn là một thách thức then chốt trong lĩnh vực xử lý ngôn ngữ tự nhiên. Các phương pháp truyền thống như Latent Dirichlet Allocation (LDA) thường gặp khó khăn về hiệu suất tính toán và độ chính xác khi xử lý các tập dữ liệu phức tạp [22].

FASTopic (Fast, Adaptive, Stable, and Transferable topic model) ra đời như một giải pháp đột phá, kế thừa và cải tiến các phương pháp học máy không giám sát. Mô hình này tận dụng kỹ thuật phân rã ma trận không âm để xác định các chủ đề một cách nhanh chóng và chính xác. Về mặt toán học, FASTopic được biểu diễn thông qua phép phân rã ma trận:

$$V \approx WH.$$

Trong đó, V là ma trận véc-tơ văn bản gốc, W biểu diễn ma trận trọng số chủ đề, và H là ma trận phân bố chủ đề. Mục tiêu của quá trình học là tìm ra các ma trận W và H sao cho sự khác biệt giữa V và tích WH là nhỏ nhất. Điểm khác biệt cốt lõi của FASTopic nằm ở khả năng xử lý tối ưu các tập văn bản lớn. So với LDA, phương pháp này giảm thiểu đáng kể chi phí tính toán, đồng thời duy trì độ chính xác cao trong việc rút trích chủ đề.



Hình 1.10: Hiệu suất mô hình FASTopic [23]

Quá trình tiền xử lý văn bản trong FASTopic được thực hiện một cách tinh vi, bao gồm các kỹ thuật như loại bỏ từ dừng, chuẩn hóa văn bản và trích xuất đặc trưng ngữ nghĩa. Điều này giúp mô hình có khả năng phân tích sâu sắc và chính xác các tập văn bản phức tạp. Tuy nhiên, FASTopic vẫn tồn tại những thách thức nhất định. Độ phức tạp tính toán gia tăng với quy mô dữ liệu, và chất lượng kết quả phụ thuộc rất nhiều vào khâu tiền xử lý và chuẩn bị dữ liệu ban đầu. Ý nghĩa thực tiễn của FASTopic được chứng minh qua các ứng dụng quan trọng như phân tích văn bản học thuật, hệ thống gợi ý nội dung, và quản trị tri thức trong các tổ chức. Mô hình này mở ra những triển vọng mới trong việc khai thác và chuyển hóa thông tin từ nguồn dữ liệu phi cấu trúc [23].

1.2.9. Cơ sở dữ liệu vector Neo4j

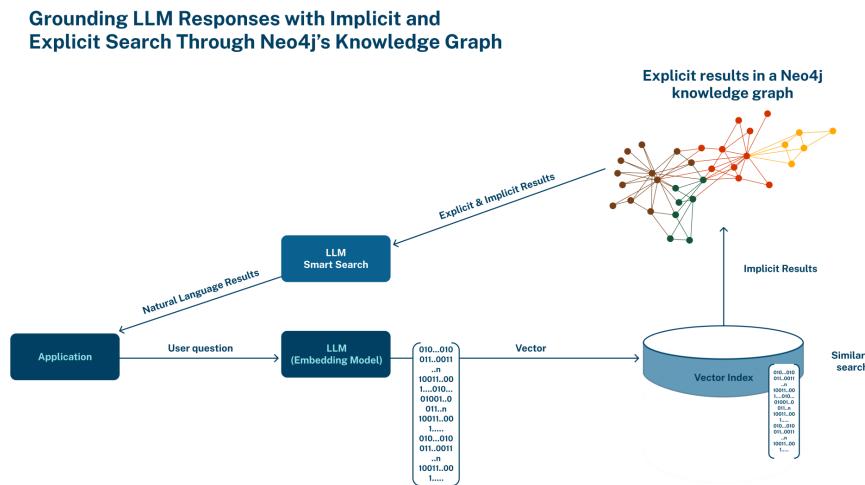
Neo4j là một trong những cơ sở dữ liệu đồ thị phổ biến nhất hiện nay, được thiết kế để xử lý và quản lý các mối quan hệ phức tạp giữa dữ liệu. Khác với các cơ sở dữ liệu quan hệ truyền thống, Neo4j tập trung vào việc lưu trữ và truy vấn thông tin dựa trên các nút và các mối quan hệ giữa chúng. Điều này cho phép Neo4j thực hiện các truy vấn phức tạp một cách hiệu quả, giúp người dùng dễ dàng khám phá và phân tích dữ liệu liên kết.

Neo4j sử dụng ngôn ngữ truy vấn Cypher, một ngôn ngữ mạnh mẽ và dễ sử dụng, cho phép người dùng thực hiện các truy vấn đồ thị một cách trực quan. Hệ thống này cũng hỗ trợ việc lưu trữ dữ liệu phi cấu trúc thông qua các thuộc tính của nút và mối quan hệ, giúp tối ưu hóa khả năng tìm kiếm và phân tích thông tin trong các ứng dụng phức tạp.

Một trong những tính năng nổi bật mới được tích hợp vào Neo4j là khả năng truy xuất thông tin vector. Tính năng này cho phép người dùng thực hiện tìm kiếm tương tự dựa trên các vector, giúp phát hiện các mối quan hệ ẩn giữa các nút trong cơ sở dữ liệu. Neo4j sử dụng thuật toán HNSW (Hierarchical Navigable Small World) để tối ưu hóa quá trình tìm kiếm k-nearest neighbors, từ đó cải thiện hiệu suất và độ chính xác của việc truy vấn thông tin.

Neo4j đã được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ quản lý

chuỗi cung ứng đến phân tích mạng xã hội. Ví dụ, trong ngành dược phẩm, một số công ty đã tích hợp tìm kiếm vector vào đồ thị tri thức của họ, giúp giảm thời gian tự động hóa báo cáo quy định lên đến 75% nhờ vào khả năng liên kết ngữ nghĩa giữa các thực thể. Điều này cho thấy tiềm năng lớn của Neo4j trong việc cung cấp thông tin chính xác và kịp thời, đặc biệt trong bối cảnh sản sinh thông tin liên tục ngày nay.



Hình 1.11: Lưu trữ và truy xuất dữ liệu trong Neo4j

Với sự phát triển không ngừng của công nghệ và nhu cầu ngày càng cao về việc quản lý và phân tích dữ liệu phức tạp, Neo4j đã khẳng định vị thế của mình như một giải pháp lý tưởng cho các ứng dụng cần truy xuất thông tin hiệu quả. Việc kết hợp giữa cơ sở dữ liệu đồ thị và khả năng tìm kiếm vector không chỉ giúp cải thiện độ chính xác mà còn mở ra nhiều cơ hội mới cho việc khai thác dữ liệu trong tương lai.

1.3. Tiêu kết chương 1

Chương 1 đã thực hiện một phân tích toàn diện về bối cảnh công nghệ và các xu hướng nghiên cứu trong lĩnh vực chatbot, truy xuất thông tin và kỹ thuật RAG. Qua tổng quan nghiên cứu, một số nhận định quan trọng có thể được rút ra:

Thứ nhất, chatbot và hệ thống trí tuệ nhân tạo đang chứng kiến sự bùng nổ về ứng dụng, với tiềm năng lan tỏa trong nhiều lĩnh vực như giáo dục, chăm sóc sức

khỏe, tiếp thị và di sản văn hóa. Tuy nhiên, các hệ thống này vẫn tồn tại những hạn chế đáng kể về tính chính xác, khả năng cập nhật kiến thức và quản lý thông tin.

Thứ hai, kỹ thuật RAG được xác định như một giải pháp đột phá để giải quyết các thách thức của các LLM truyền thống. Bằng cách tích hợp khả năng truy xuất thông tin từ các cơ sở dữ liệu tri thức, RAG mở ra những khả năng mới trong việc cung cấp các câu trả lời chính xác, có nguồn gốc rõ ràng và luôn được cập nhật.

Thứ ba, các phương pháp truy xuất thông tin hiện đại đang không ngừng phát triển, từ các phương pháp truyền thống như TF-IDF, BM25 đến các kỹ thuật tiên tiến như DPR và ColBERT. Mỗi phương pháp đều hướng tới việc cải thiện khả năng hiểu ngữ nghĩa, xử lý các truy vấn phức tạp và tối ưu hóa quá trình trích chọn thông tin.

Các thách thức chính được xác định bao gồm:

- Giới hạn của việc mã hóa vector với kích thước cố định.
- Khả năng hiểu ngữ nghĩa sâu sắc của các câu truy vấn.
- Tính chính xác trong việc so khớp vector và trích xuất thông tin.

Ngoài ra, Chương 1 còn cung cấp kiến thức về các kỹ thuật xử lý ngôn ngữ tự nhiên, các mô hình học sâu tiên tiến mà dự định sẽ được sử dụng để giải quyết các vấn đề hiện tại

Với bối cảnh công nghệ đang phát triển nhanh chóng, nghiên cứu này hướng tới việc đóng góp các giải pháp mới trong việc cải thiện hệ thống truy xuất và tổng hợp thông tin, góp phần nâng cao trải nghiệm người dùng với các hệ thống trí tuệ nhân tạo.

Các nội dung nghiên cứu lý thuyết được trình bày trong chương này sẽ là nền tảng quan trọng để định hướng các phương pháp nghiên cứu và triển khai trong các chương tiếp theo của luận văn.

Chương 2: Mô tả bài toán và khung nghiên cứu

2.1. Mô tả bài toán

Trong đề tài này, tôi xây dựng một Chatbot nội bộ hoạt động trên dữ liệu cá nhân bằng cách tận dụng lại các mô hình ngôn ngữ lớn đã có sẵn mà không cần phải huấn luyện lại với dữ liệu của tôi bằng cách sử dụng kỹ thuật truy xuất thông tin để cung cấp dữ liệu nội bộ cho mô hình ngôn ngữ lớn.

Nghiên cứu này tập trung phát triển kỹ thuật truy xuất thông tin, cải thiện điểm hạn chế của các mô hình cũ như truy xuất cố định số lượng dữ liệu trên mỗi câu truy vấn hay hạn chế về việc biểu diễn thông tin thông qua vector.

Không dừng lại ở đó, đề tài tìm ra giải pháp đáp ứng nhu cầu thực tiễn của các doanh nghiệp về giải pháp nắm giữ toàn bộ hệ thống mà không cần sử dụng dịch vụ của bên thứ ba, đảm bảo an toàn và bảo mật thông tin cho doanh nghiệp.

Để đạt được mục tiêu trên, tôi xác định được các mục tiêu cụ thể cần phải thực hiện được như sau:

- Xây dựng được mô hình phân rã câu hỏi từ những dùng thành các câu hỏi nhỏ hơn, đảm bảo truy xuất toàn bộ thông tin mà người dùng yêu cầu.
- Xây dựng phương pháp biểu diễn thông tin tốt hơn, có gắng biểu diễn đầy đủ ý nghĩa của thông tin nhiều nhất có thể.
- Xây dựng mô hình phân cụm thông tin thành các nhóm, như vậy khi truy xuất có thể đạt tốc độ và hiệu quả tốt hơn.
- Xây dựng giao diện sử dụng mô hình trực quan và dễ tiếp cận.
- Toàn bộ giải pháp phải đều có khả năng tự quản trị và chạy cục bộ, không thông qua bất cứ ai đảm bảo dữ liệu không bị rò rỉ ra ngoài.

Trong đó, đề tài xác định những thách thức phải đổi mới có thể ảnh hưởng đến chất lượng mô hình như:

- Tài nguyên hệ thống: Đây một yếu tố quan trọng. Nếu hệ thống không đủ mạnh để xử lý các truy vấn phức tạp hoặc khối lượng dữ

liệu lớn, điều này có thể dẫn đến độ trễ trong phản hồi hoặc thậm chí lỗi trong quá trình truy xuất thông tin.

- Chất lượng mô hình LLM: Chất lượng của mô hình ngôn ngữ lớn được sử dụng cũng ảnh hưởng trực tiếp đến hiệu suất của chatbot. Nếu mô hình không hiểu tốt ngữ cảnh, dù đã truy xuất đúng thông tin cần thiết nhưng vẫn sẽ không nhận được câu trả lời chính xác.
- Độ phức tạp và quá chuyên sâu của câu truy vấn: Khi câu truy vấn quá chuyên ngành, mô hình ngôn ngữ lớn có thể hiểu được khi cung cấp thông tin cần thiết, tuy nhiên vẫn đề gấp phải nằm ở bước phân rã thông tin khi mô hình có thể không phân rã chính xác.

Đánh giá độ khó của bài toán về thách thức mà dự án phải đối mặt. Một số yếu tố làm tăng tính phức tạp của bài toán bao gồm:

- Kích thước dữ liệu lớn: Khi làm việc với một lượng dữ liệu lớn, việc xử lý và truy xuất thông tin trở nên khó khăn hơn. Hệ thống cần phải tối ưu hóa để đảm bảo rằng nó có thể tìm kiếm và truy xuất thông tin một cách hiệu quả mà không làm giảm hiệu suất.
- Nhiều biến đầu vào: Chatbot cần phải xử lý nhiều loại câu hỏi và yêu cầu khác nhau từ người dùng, điều này tạo ra sự đa dạng trong các biến đầu vào mà hệ thống phải xử lý. Sự đa dạng này làm tăng độ phức tạp trong việc xây dựng và tinh chỉnh mô hình.
- Độ chính xác cao: Để đảm bảo rằng người dùng nhận được thông tin chính xác và hữu ích, hệ thống cần phải đạt được độ chính xác cao trong việc truy xuất và trình bày thông tin. Điều này đòi hỏi các thuật toán phức tạp và khả năng xử lý ngữ nghĩa tốt.

Tuy có nhiều khó khăn nhưng yêu cầu giải quyết bài toán này là rất quan trọng vì nó mang lại giá trị thực tiễn cho doanh nghiệp trong việc tối ưu hóa quy trình làm việc và nâng cao trải nghiệm người dùng. Việc phát triển một chatbot nội bộ hiệu quả sẽ giúp doanh nghiệp:

- Tăng cường hiệu suất làm việc: Chatbot có khả năng tự động hóa

nhiều tác vụ lặp đi lặp lại, giúp nhân viên tiết kiệm thời gian và tập trung vào các nhiệm vụ quan trọng hơn.

- Cải thiện chất lượng dịch vụ khách hàng: Với khả năng cung cấp thông tin nhanh chóng và chính xác, chatbot sẽ nâng cao trải nghiệm của khách hàng khi tương tác với doanh nghiệp.
- Bảo mật thông tin: Giải pháp này cho phép doanh nghiệp kiểm soát hoàn toàn dữ liệu nội bộ mà không phụ thuộc vào bên thứ ba, từ đó giảm thiểu rủi ro mất mát thông tin nhạy cảm.

Để thu hẹp nghiên cứu và tập trung vào việc xử lý truy xuất, đề tài đặt ra các giả định sau:

- Thông tin nội bộ ở đây sẽ linh động hơn khi người dùng có thể tải lên tệp PDF hoặc sử dụng dữ liệu của một trang web bất kỳ.
- Dữ liệu thô trích xuất từ tệp PDF hãy thu thập từ các trang web được giả định là đầy đủ, không bị lỗi hay nhiễu từ.
- Giả định rằng hệ thống có khả năng xử lý đa ngôn ngữ
- Giả định rằng hệ thống có khả năng nhận biết và không trả lời các thông tin độc hại, các vấn đề nhạy cảm về chủ quyền dân tộc quốc gia.

2.2. Khung nghiên cứu

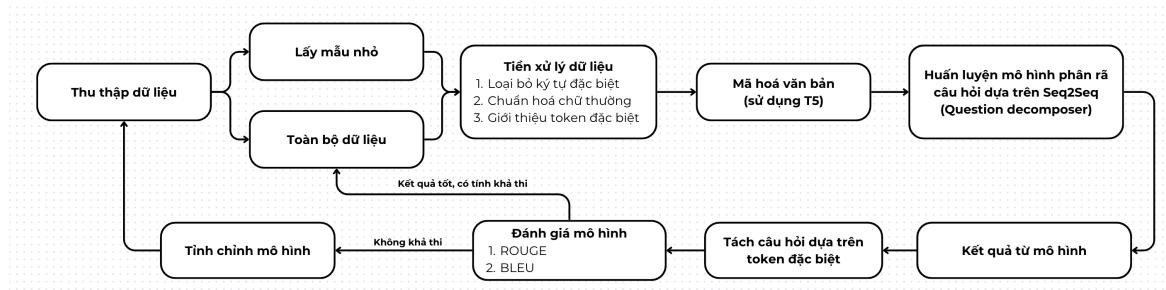
Để giải quyết bài toán và đạt được các mục tiêu nghiên cứu đã đề ra, đề tài được thiết kế theo một khung nghiên cứu có cấu trúc logic và có hệ thống, được chia thành các giai đoạn nghiên cứu độc lập nhưng có mối quan hệ hữu cơ, nhằm đảm bảo tính khoa học và tính khả thi của giải pháp công nghệ.

Khung nghiên cứu được xây dựng dựa trên nguyên tắc phương pháp luận khoa học, với mục tiêu phân tích, thiết kế và triển khai từng thành phần của hệ thống Chatbot nội bộ một cách có hệ thống và chặt chẽ. Mỗi giai đoạn nghiên cứu sẽ tập trung giải quyết những thách thức cụ thể, đồng thời tạo nền tảng cho các giai đoạn tiếp theo.

Cụ thể, khung nghiên cứu được chia thành ba giai đoạn chính, mỗi giai đoạn đều có mục tiêu, phương pháp và kết quả nghiên cứu riêng biệt, nhưng luôn hướng đến mục tiêu tổng thể của đề tài.

Bằng cách tiếp cận nghiên cứu có hệ thống này, tôi kỳ vọng sẽ phát triển được một giải pháp Chatbot nội bộ hiệu quả, đáp ứng các yêu cầu về truy xuất thông tin, xử lý ngôn ngữ và quản trị tri thức số.

2.2.1. Giai đoạn 1: Nghiên cứu xây dựng mô hình phân rã câu hỏi



Hình 2.1: Khung nghiên cứu xây dựng mô hình phân rã câu hỏi

Trong giai đoạn đầu tiên, tôi sẽ tập trung nghiên cứu phương pháp phân rã câu hỏi từ người dùng thành các câu hỏi nhỏ hơn. Bước này được ưu tiên thực hiện trước bởi nó là bước đầu tiên trong hệ thống cuối cùng.

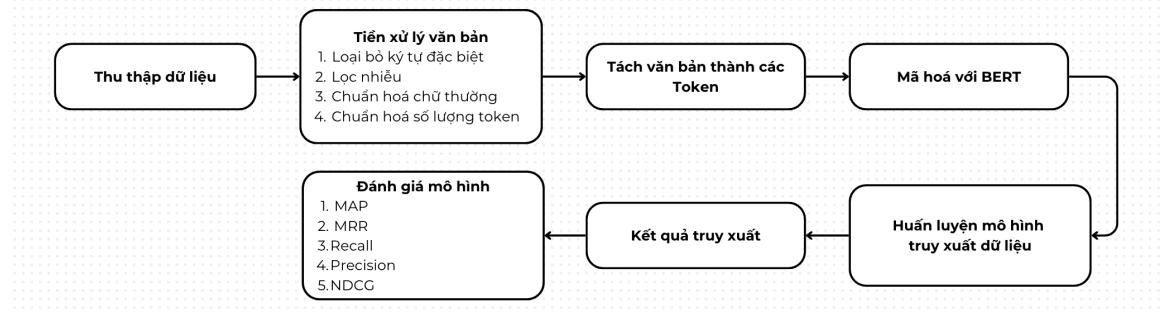
Để xây dựng được mô hình phân rã câu hỏi, tôi dựa trên kiến trúc Seq2Seq và sử dụng T5, mô hình đã được huấn luyện từ trước để đạt được tốc độ hội tụ nhanh hơn.

Các bước thực hiện như sau:

- Bước 1: Thu thập dữ liệu
 - Tìm kiếm và kiểm tra các tập dữ liệu có sẵn hoặc lên kế hoạch xây dựng bộ dữ liệu.
 - Kiểm tra Các nguồn tài nguyên nổi tiếng về dữ liệu và tinh chỉnh lại dữ liệu cho phù hợp với đề tài
 - Dữ liệu mong muốn cần có câu hỏi chính và các câu hỏi đã được phân mảnh từ câu hỏi chính.
- Bước 2: Lấy mẫu

- Lấy mẫu nhỏ dữ liệu, cần xáo trộn và đảm bảo dữ liệu ngẫu nhiên không quá đặc trưng.
- Bước 3: Tiền xử lý dữ liệu
 - Sau khi lấy mẫu, cần tiến hành xử lý dữ liệu để giảm thiểu độ nhiễu khi huấn luyện mô hình.
 - Cần loại bỏ các ký tự đặc biệt như liên kết website, biểu tượng cảm xúc, ký tự toán học lạ,...
 - Vì mô hình Seq2Seq chỉ trả về một câu phản hồi, ta không thể yêu cầu nó phản hồi nhiều câu tách biệt được vậy nên ở đây ta giới thiệu token [SEP] để phân tách các câu hỏi.
- Bước 4: Mã hoá văn bản
 - Tiếp theo cần mã hoá câu hỏi đầu vào và các câu hỏi mong muốn ở đầu ra.
 - Ở đây có thể sử dụng T5 hoặc các biến thể của T5 để mã hoá.
- Bước 5: Huấn luyện mô hình
 - Thực hiện huấn luyện mô hình
 - Huấn luyện thử trên tập dữ liệu nhỏ, kiểm tra các siêu tham số để mô hình tính toán phù hợp với tài nguyên hệ thống mà không bị sự cố.
- Bước 6: Tách câu hỏi phân mảnh từ mô hình
 - Dựa vào token đặc biệt đã giới thiệu, tách kết quả từ mô hình thành các câu hỏi nhỏ
- Bước 7: Đánh giá kết quả
 - Dựa vào kết quả mô hình ta có thể đánh giá kết quả thông qua các phương pháp như ROUGE 1; ROUGE 2; ROUGE3; ROUGE L hay BLEU

2.2.2. Giai đoạn 2: Nghiên cứu mô hình truy xuất dữ liệu



Hình 2.2: Khung nghiên cứu xây dựng mô hình truy xuất dữ liệu

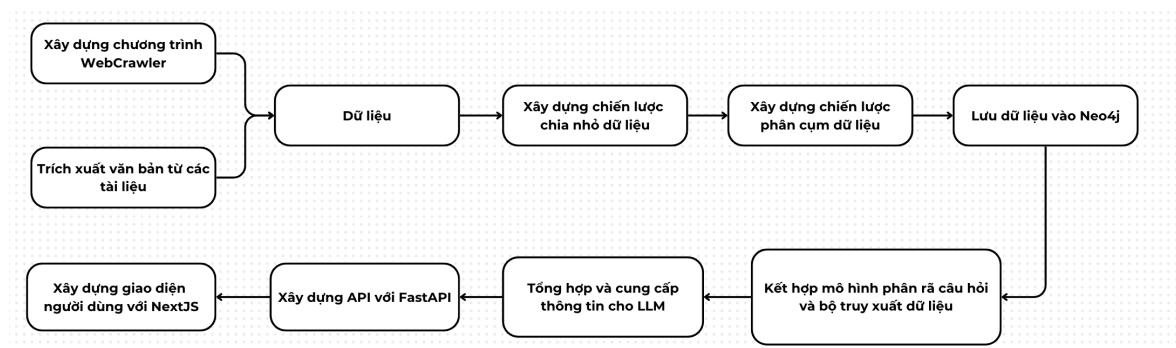
Sau khi xây dựng thành công mô hình phân rã câu hỏi, ta tiếp tục xây dựng bộ truy xuất dữ liệu, ở đây tôi dự định dựa trên nền tảng mô hình ColBERT. Thông thường với các mô hình truy xuất ta cần mã hoá dữ liệu và câu truy vấn rồi so sánh chúng và tìm ra các cặp tương đồng. Với ColBERT cũng vậy, khác biệt là ColBERT so sánh ở mức độ token chứ không phải cả câu.

- Bước 1: Thu thập dữ liệu
 - Đầu tiên cần tiến hành thu thập dữ liệu, dữ liệu mong muốn gồm có câu hỏi và tập hợp các thông tin liên quan được gắn nhãn hoặc sắp xếp theo thứ tự.
 - Câu trả lời của câu truy vấn phải thực sự nằm trong tập hợp các thông tin liên quan.
- Bước 2: Tiền xử lý văn bản
 - Cần tiến hành các bước xử lý văn bản như loại bỏ các ký tự đặc biệt, chuẩn hoá văn bản, đảm bảo đoạn văn không quá dài, lọc trùng lặp.
- Bước 3: Tách văn bản thành token
 - Tiếp theo ta cần tách cả câu truy vấn và tập hợp dữ liệu thành các token.
- Bước 4: Sử dụng BERT mã hoá thông tin
 - Sau khi tách văn bản thành các token, tiếp theo ta mã hoá

các token này với BERT.

- Bước 5: Huấn luyện mô hình truy xuất dữ liệu
 - Lần lượt huấn luyện mô hình với dữ liệu đã qua xử lý, mục tiêu là câu truy vấn phải có vector gần với $P+$ và cách xa các $P-$ trong tập dữ liệu.
- Bước 6: Đánh giá kết quả truy xuất
 - Đánh giá kết quả truy xuất thông qua các phương pháp như MAP, MRR, Precision, Recall, NDCG.

2.2.3. Giai đoạn 3: Tổng hợp nghiên cứu và xây dựng Chatbot nội bộ



Hình 2.3: Khung nghiên cứu tổng hợp và xây dựng Chatbot nội bộ

Mục tiêu cuối cùng của đề tài là xây dựng hệ thống Chatbot nội bộ, sau khi xây dựng mô hình phân rã câu hỏi ở giai đoạn 1 thành công và mô hình truy xuất thông tin ở giai đoạn 2, ta tiến hành kết hợp tất cả và xây dựng hệ thống chatbot nội bộ hoàn chỉnh.

- Bước 1: Xây dựng chương trình có khả năng tiếp nhận thông tin được cung cấp từ người dùng.
 - Xác định thông tin được người dùng cung cấp qua hai dạng, tệp văn bản (tệp docs, pdf, txt,...) hoặc đường link đến một trang web cụ thể.
 - Xây dựng phương pháp lấy văn bản từ tệp dữ liệu, có thể sử dụng các thư viện có sẵn như PyMDF, pdfplumber,... để trích xuất văn bản từ tệp PDF.

- Sử dụng chương trình Craw dữ liệu để thu thập dữ liệu từ trang web khi người dùng cung cấp đường link, ở đây ta sử dụng thư viện markdown-crawler để thu thập và lưu vào tệp markdown.
- Bước 2: Chia nhỏ văn bản theo ngữ nghĩa (semantic chunking)
 - Sử dụng các mô hình mã hoá văn bản để chuyển đổi các câu văn thành không gian vector nhiều chiều. Mỗi câu được biểu diễn bằng một vector embedding, cho phép đo lường mức độ tương đồng ngữ nghĩa.
 - Áp dụng phép đo cosine similarity để xác định mức độ liên quan giữa các câu. Qua đó, hệ thống có thể nhận biết các điểm chuyển ý trong văn bản một cách chính xác.
 - Xác định các khối văn bản (chunk) dựa trên hai tiêu chí: Độ dài và độ tương đồng ngữ nghĩa.
- Bước 3: Mã hoá thông tin
 - Tiến hành tách token từ văn bản.
 - Mã hoá các token đã được tách với BERT.
 - Lưu dữ liệu vào cơ sở dữ liệu đồ thị.
- Bước 4: Phân cụm thông tin
 - Phân nhóm các thông tin đã được chia nhỏ bằng cách sử dụng các mô hình phân cụm.
 - Trước hết cần tách hành mã hoá thông tin, sau khi có các vector đại diện, ta tiến hành sử dụng các mô hình phân cụm hoặc các mô hình phân chủ đề như LDA (Latent Dirichlet Allocation); HDP (Hierarchical Dirichlet Process) hay FASTopic (Fast, Adaptive, Stable, and Transferable Topic Model).
- Bước 5: Kết hợp mô-đun phân ra câu hỏi vào hệ thống

- Kết nối mô-đun phân rã câu hỏi vào hệ thống, khi người dùng đưa yêu cầu thì phân rã thành các câu hỏi nhỏ hơn.
 - Các câu hỏi này tiếp tục được đưa vào xác định chủ đề trước khi tiến hành đưa đến bộ truy xuất.
- Bước 6: Kết hợp mô-đun truy xuất thông tin vào hệ thống
 - Mô hình truy xuất thông tin đã được xây dựng từ trước đó, tiếp theo ta cần kết hợp vào trong hệ thống.
 - Sau khi có được câu hỏi và chủ đề cần truy vấn, ta xác định các node trong cơ sở dữ liệu đồ thị và chỉ tiến hành truy xuất dựa trên các node đó.
- Bước 7: Nhận phản hồi từ LLM
 - Sau khi có được kết quả truy xuất, cần tiến hành xây dựng prompt để mô hình hiểu ngữ cảnh và đảm bảo đưa vào đầy đủ thông tin.
 - Sử dụng nền tảng Ollama để tải mô hình LLM xuống và chạy cục bộ.
 - Đưa prompt vào mô hình LLM và nhận phản hồi.
- Bước 8: Xây dựng giao diện và hoàn thiện hệ thống
 - Tiến hành xây dựng API để kết nối phần xử lý giữa các mô hình với giao diện thông qua FastAPI.
 - Tiến hành xây dựng giao diện chatbot với NextJS.
 - Kết hợp hệ thống kiểm tra và sửa lỗi.

2.3. Tiêu kết chương 2

Như vậy, nội dung Chương 2 đã trình bày chi tiết về bối cảnh, mục tiêu và khung nghiên cứu của đề tài phát triển Chatbot nội bộ sử dụng kỹ thuật truy xuất thông tin. Qua phân tích, nghiên cứu đã xác định rõ các thách thức và giá trị thực tiễn của giải pháp công nghệ này.

Về mặt khoa học, nghiên cứu tập trung giải quyết những hạn chế của các mô hình truy xuất thông tin truyền thống, đặc biệt là vấn đề về biểu diễn và truy xuất thông tin. Các mục tiêu cụ thể bao gồm: xây dựng mô hình phân rã câu hỏi, cải thiện phương pháp biểu diễn thông tin, phát triển kỹ thuật phân cụm thông tin hiệu quả, và xây dựng giao diện trực quan.

Khung nghiên cứu được chia thành ba giai đoạn quan trọng:

1. Giai đoạn nghiên cứu mô hình phân rã câu hỏi: Sử dụng kiến trúc Seq2Seq và mô hình T5, nghiên cứu tập trung vào việc phân tách câu hỏi phức tạp thành các câu hỏi nhỏ hơn và dễ xử lý.
2. Giai đoạn nghiên cứu mô hình truy xuất dữ liệu: Dựa trên nền tảng ColBERT, nghiên cứu phát triển phương pháp truy xuất thông tin ở mức độ token, mở ra khả năng so sánh ngữ nghĩa chi tiết hơn.
3. Giai đoạn tích hợp và xây dựng Chatbot: Kết hợp các mô-đun đã phát triển để tạo ra một hệ thống chatbot nội bộ toàn diện, có khả năng xử lý dữ liệu từ nhiều nguồn và cung cấp trải nghiệm tương tác thông minh.

Về giá trị thực tiễn, giải pháp này hướng đến việc giải quyết các nhu cầu cấp thiết của doanh nghiệp: tăng cường hiệu suất làm việc, cải thiện chất lượng dịch vụ khách hàng, và đảm bảo an toàn thông tin. Đặc biệt, mô hình được thiết kế với khả năng tự quản trị và chạy cục bộ, giảm thiểu rủi ro rò rỉ thông tin.

Các giả định và hạn chế của nghiên cứu cũng được xác định rõ, bao gồm khả năng xử lý đa ngôn ngữ, tính linh động của dữ liệu đầu vào, và năng lực nhận biết các nội dung nhạy cảm.

Với cách tiếp cận khoa học và định hướng ứng dụng thực tế, nghiên cứu này có tiềm năng mở ra những giải pháp đột phá trong lĩnh vực xử lý ngôn ngữ tự nhiên và quản trị tri thức số.

Chương 3: Thu thập và tiền xử lý dữ liệu

3.1. Bộ dữ liệu hỗ trợ phân rã câu hỏi

Trong lĩnh vực xử lý ngôn ngữ tự nhiên, việc lựa chọn và xây dựng bộ dữ liệu phù hợp đóng vai trò then chốt quyết định chất lượng và độ tin cậy của các mô hình nghiên cứu. Nghiên cứu của tôi đã kế thừa và phát triển từ những đóng góp đáng chú ý của các nhà khoa học hàng đầu thế giới, đặc biệt là các nghiên cứu tiên phong từ Facebook Research và các nhóm nghiên cứu quốc tế.

Trong tác vụ phân rã câu hỏi, Facebook research đã nghiên cứu về bài toán này và đưa ra hướng giải quyết với phương pháp “unsupervised seq2seq”, đây là một phương pháp rất mới và có giá trị nghiên cứu lớn lao. Tuy mô hình không thật sự mang lại hiệu quả cao nhưng những gì họ công hiến không chỉ là một phương pháp tiếp cận mới mà còn là bộ dữ liệu phân rã câu hỏi mà họ đã xây dựng. Trong nghiên cứu, Facebook reaseach đã sử dụng bộ dữ liệu HotpotQA và thực hiện một số phương pháp xử lý văn bản, biến đổi để có thể sử dụng bộ dữ liệu này trong nhiệm vụ phân rã câu hỏi, thật tuyệt khi họ đã công khai nguồn tài nguyên này và cho phép người dùng sử dụng bộ dữ liệu của họ. Bộ dữ liệu được thiết kế đặc biệt để hỗ trợ các nhiệm vụ phân rã câu hỏi, trong đó người dùng cần tìm kiếm thông tin từ nhiều nguồn khác nhau để trả lời một câu hỏi phức tạp. Người dùng có thể thực thi mã nguồn được công khai để tải dữ liệu từ định dạng JSON và sử dụng cho các mục đích nghiên cứu [24].

Trong một nghiên cứu khác cũng tập trung giải quyết vấn đề mà đề tài đề cập, Bài báo "Distilling Reasoning Capabilities into Smaller Language Models" của Kumar Shridhar trình bày một phương pháp mới nhằm cải thiện khả năng suy luận của các mô hình ngôn ngữ nhỏ hơn thông qua việc sử dụng kiến thức từ các mô hình lớn hơn. Một trong những điểm nổi bật của nghiên cứu này là bộ dữ liệu mà họ đã sử dụng để kiểm tra và phát triển phương pháp của mình [25]. Nghiên cứu này tập trung vào ba bộ dữ liệu chính phục vụ cho các nhiệm vụ suy luận đa bước:

- **GSM8K:** Đây là một bộ dữ liệu gồm các bài toán toán học được thiết

kế để kiểm tra khả năng suy luận của các mô hình. Mỗi bài toán trong GSM8K yêu cầu người dùng phải thực hiện nhiều bước suy luận để đạt được kết quả cuối cùng.

- **StrategyQA:** Bộ dữ liệu này chứa các câu hỏi liên quan đến chiến lược, yêu cầu người dùng phải suy nghĩ và phân tích để tìm ra câu trả lời. Nó được thiết kế để đánh giá khả năng giải quyết vấn đề phức tạp của mô hình.
- **SVAMP:** SVAMP là một bộ dữ liệu khác cũng tập trung vào các bài toán toán học, nhưng với cấu trúc câu hỏi và câu trả lời phức tạp hơn, nhằm thử thách khả năng suy luận của các mô hình ngôn ngữ.

Mỗi bộ dữ liệu đều bao gồm các bài toán kèm theo câu trả lời đúng, và trong một số trường hợp, có thể có thêm các bước suy luận trung gian dẫn đến câu trả lời. Điều này cho phép mô hình học cách phân rã vấn đề thành các phần nhỏ hơn, từ đó cải thiện khả năng giải quyết vấn đề phức tạp. Trong nghiên cứu, nhóm tác giả đã áp dụng phương pháp SOCRATIC COT để tạo ra các cặp câu hỏi và câu trả lời trung gian từ các bài toán trong ba bộ dữ liệu này. Họ đã sử dụng mô hình lớn để tạo ra các bước suy luận mà sau đó được sử dụng để huấn luyện các mô hình nhỏ hơn. Kết quả cho thấy rằng việc sử dụng các ví dụ phân rã từ mô hình lớn giúp cải thiện hiệu suất của mô hình nhỏ lên tới 40% [25].

Việc công khai và sử dụng những bộ dữ liệu này không chỉ giúp nâng cao chất lượng nghiên cứu mà còn tạo điều kiện cho cộng đồng nghiên cứu phát triển thêm nhiều phương pháp mới trong lĩnh vực xử lý ngôn ngữ tự nhiên.

3.1.1. Chuyển đổi và tái cấu trúc dữ liệu

Đầu tiên, với việc sử dụng hai bộ dữ liệu khác nhau, ta cần tiến hành đồng nhất dữ liệu cho nhiệm vụ cuối cùng. Trước hết tôi định nghĩa cấu trúc tệp JSON sao cho rõ ràng các thành phần dữ liệu và thuận tiện cho việc huấn luyện nhất có thể, tiếp đến ta biến đổi cấu trúc dữ liệu từ và đưa vào cấu trúc JSON đã định nghĩa. Mỗi tệp dữ liệu với cấu trúc khác nhau cần có phương pháp tách khác nhau để chuẩn hóa.

Quá trình chuyển đổi và tái cấu trúc dữ liệu được thực hiện một cách hệ thống và cẩn trọng, nhằm đảm bảo tính nhất quán và khả năng sử dụng tối ưu của bộ dữ liệu. Các bước cụ thể được mô tả như sau:

Định nghĩa cấu trúc JSON: Trước tiên, tôi tiến hành xác định và thiết kế một cấu trúc JSON rõ ràng, bao gồm các trường dữ liệu quan trọng. Mục tiêu là tạo ra một cấu trúc linh hoạt, dễ dàng truy xuất và xử lý trong quá trình huấn luyện mô hình. Cấu trúc này cần đáp ứng các yêu cầu sau:

Tính minh bạch: Mỗi trường dữ liệu được định nghĩa rõ ràng, với mô tả chi tiết về nội dung và ý nghĩa.

- **Tính mở rộng:** Cấu trúc cho phép bổ sung các trường thông tin bổ sung nếu cần thiết.
- **Tính tương thích:** Đảm bảo khả năng tích hợp với các công cụ và thư viện xử lý dữ liệu phổ biến.

Phương pháp chuyển đổi: Đối với mỗi bộ dữ liệu có cấu trúc khác nhau, tôi áp dụng các kỹ thuật tách và chuyển đổi riêng biệt. Quy trình này bao gồm:

- **Phân tích cấu trúc ban đầu:** Nghiên cứu kỹ cấu trúc gốc của từng bộ dữ liệu.
- **Xác định các trường thông tin quan trọng:** Lựa chọn các trường dữ liệu cần thiết cho nhiệm vụ nghiên cứu.
- **Ánh xạ dữ liệu:** Chuyển đổi các trường từ cấu trúc gốc sang cấu trúc JSON đã định nghĩa.
- **Kiểm tra tính toàn vẹn:** Đảm bảo không mất mát thông tin quan trọng trong quá trình chuyển đổi.

Xác thực dữ liệu: Sau khi hoàn tất quá trình chuyển đổi, một bước kiểm tra kỹ lưỡng được thực hiện: Kiểm tra tính toàn vẹn của dữ liệu; Xác minh số lượng bản ghi; Đảm bảo không có lỗi trong cấu trúc JSON; Kết quả của quá trình này là một bộ dữ liệu được chuẩn hóa, sẵn sàng cho các bước xử lý và huấn luyện mô hình tiếp theo. Sau khi đưa dữ liệu về cấu trúc JSON đã định nghĩa, ta tiến hành kiểm tra

dữ liệu, lọc và loại bỏ các dữ liệu bị trùng lặp.

3.1.2. Tiền xử lý dữ liệu

Trong giai đoạn tiền xử lý dữ liệu cho mô hình seq2seq, tôi tập trung vào việc chuẩn hóa văn bản nhằm tối ưu hóa hiệu quả huấn luyện và giảm thiểu chi phí tài nguyên tính toán. Quy trình được thực hiện như sau:

Chuẩn hóa chữ thường: Toàn bộ dữ liệu được chuyển đổi sang chữ thường. Mặc dù các mô hình học sâu có thể huấn luyện được trên cả chữ viết hoa và chữ thường, việc chuyển đổi thông nhất sang chữ thường mang lại một số lợi ích quan trọng:

- Giảm thiểu số lượng token, từ đó tiết kiệm bộ nhớ và thời gian huấn luyện mô hình.
- Loại bỏ sự dư thừa giữa các biến thể chữ cái.
- Tăng tính nhất quán của dữ liệu đầu vào.

Xử lý tối thiểu: Với mô hình seq2seq, việc tiền xử lý được thực hiện một cách tối thiểu. Mục tiêu chính là duy trì nguyên bản chất của văn bản, cho phép mô hình tự học và xử lý các đặc điểm ngôn ngữ. Các kỹ thuật như loại bỏ stopwords, lemmatization hay stemming không được áp dụng để tránh mất mát thông tin quan trọng.

Nguyên tắc chính trong tiền xử lý:

- Bảo toàn cấu trúc và ngữ nghĩa gốc của văn bản.
- Tối ưu hóa hiệu quả huấn luyện.
- Giảm thiểu chi phí tính toán.

Cuối cùng, ta tiến hành một số bước kiểm tra thống kê dữ liệu, như tính độ dài trung bình câu hỏi, số lượng câu hỏi phân rã,... để có cái nhìn trực quan về dữ liệu. Các bước xử lý này nhằm chuẩn bị dữ liệu tốt nhất cho quá trình huấn luyện mô hình seq2seq, tập trung vào việc chia dữ liệu và xác định các tham số phù hợp.

3.2. Bộ dữ liệu MS Macro cho hệ thống truy xuất thông tin

Trong bối cảnh phát triển công nghệ trí tuệ nhân tạo hiện đại, việc xây dựng macro data cho hệ thống Truy Xuất và Sinh Thông Tin Tăng Cường đòi hỏi một phương pháp tiếp cận toàn diện và tinh vi. Khác với các phương thức truyền thống, macro data trong nghiên cứu này được kiến tạo như một hệ sinh thái tri thức phức hợp, tích hợp sâu sắc các nguồn thông tin từ các lĩnh vực khoa học máy tính, công nghệ thông tin và các kho tri thức chuyên ngành.

Quá trình xây dựng macro data không chỉ đơn thuần là việc tập hợp khối lượng lớn thông tin, mà còn là một nghệ thuật tinh tế trong việc lựa chọn, xử lý và chuyển đổi dữ liệu. Các nguồn thông tin được lựa chọn kỹ lưỡng từ các nền tảng học thuật uy tín như Arxiv, Semantic Scholar, và các tập san chuyên ngành quốc tế. Mỗi tài liệu được trải qua quy trình xử lý chuyên sâu, bao gồm việc làm sạch dữ liệu, trích xuất tri thức và chuyển đổi thành các không gian vector nhiều chiều thông qua các mô hình mã hoá tiên tiến [26].

Trọng tâm của phương pháp nghiên cứu nằm ở việc xây dựng một hệ thống linh hoạt và thích ứng. Các kỹ thuật tiền xử lý tiên tiến được áp dụng nhằm đảm bảo tính chính xác, loại bỏ nhiễu và chuẩn hóa cấu trúc thông tin. Đặc biệt, việc sử dụng các thuật toán học máy tiên tiến giúp hệ thống có khả năng tự động học hỏi, điều chỉnh và nâng cao chất lượng truy xuất thông tin một cách liên tục.

Thách thức lớn nhất trong quá trình nghiên cứu chính là quản lý và xử lý khối lượng thông tin khổng lồ. Để vượt qua giới hạn này, nhóm nghiên cứu đã phát triển các giải pháp song song hóa và phân tán dữ liệu, kết hợp với các kỹ thuật học máy hiệu suất cao. Kết quả là một hệ thống RAG có khả năng truy xuất và tổng hợp thông tin một cách nhanh chóng, chính xác từ các nguồn dữ liệu phức tạp và đa dạng.

Có thể thấy, ý nghĩa sâu sắc của bộ dữ liệu macro trong nghiên cứu này không chỉ dừng lại ở việc cung cấp thông tin, mà còn là một bước đột phá trong việc xây dựng các hệ thống trí tuệ nhân tạo có khả năng học hỏi, thích ứng và sáng tạo. Đây được coi như một nền tảng quan trọng cho các nghiên cứu tương lai về xử

lý ngôn ngữ tự nhiên và hệ thống thông minh.

3.2.1. Tiền xử lý dữ liệu Macro

Quá trình tiền xử lý dữ liệu macro được thiết kế như một hệ thống xử lý tinh vi, nhằm chuẩn bị nguồn thông tin một cách tối ưu cho việc huấn luyện hệ thống truy xuất thông tin. Trọng tâm của quá trình này là biến đổi khối lượng lớn dữ liệu thành một nguồn tri thức tinh chế, có cấu trúc và có thể xử lý được.

Giai đoạn làm sạch và chuẩn hóa văn bản được thực hiện một cách hết sức chi tiết. Các văn bản từ các nguồn học thuật như Arxiv và Semantic Scholar trải qua một quy trình tinh lọc toàn diện. Việc loại bỏ các ký tự không mong muốn, chuyển đổi sang chữ thường, và xử lý các ký hiệu đặc biệt được thực hiện một cách hệ thống, nhằm đảm bảo tính nhất quán và chất lượng của dữ liệu.

Kỹ thuật phân đoạn văn bản được áp dụng một cách thông minh, không chỉ đơn thuần là chia nhỏ văn bản mà còn là một quá trình bảo toàn ngữ cảnh và ý nghĩa. Mỗi đoạn văn được đánh giá kỹ lưỡng về mức độ quan trọng, loại bỏ các phần trùng lặp và giữ lại các thông tin có giá trị học thuật cao.

3.2.2. Xây dựng Cơ sở dữ liệu vector ở mức Token

Việc chuyển đổi dữ liệu văn bản sang không gian vector token là một bước then chốt trong kiến trúc hệ thống truy xuất thông tin. Quá trình này được thực hiện thông qua mô hình ColBERT embedding, một phương pháp tiên tiến trong việc mã hóa ngôn ngữ tự nhiên.

Mỗi token được ánh xạ sang một không gian vector đa chiều, trong đó mỗi vector không chỉ đơn thuần là một điểm trong không gian toán học, mà còn mang theo toàn bộ thông tin ngữ nghĩa và cú pháp của từ. Quá trình chuẩn hóa và tinh chỉnh vector được thực hiện một cách tỉ mỉ, nhằm loại bỏ các nhiễu và tối ưu hóa khả năng truy xuất thông tin.

Cấu trúc lưu trữ vector token được thiết kế để đảm bảo hiệu suất cao nhất. Hệ thống không chỉ lưu trữ các vector mà còn tối ưu hóa các phép toán vector, cho phép tra cứu nhanh chóng và chính xác. Điều này đặc biệt quan trọng trong việc hỗ trợ các thuật toán tìm kiếm và so sánh thông minh.

Khác với các phương pháp truyền thống, hệ thống mà tôi nghiên cứu không phụ thuộc vào việc nhập trực tiếp mã hoá các dữ liệu thông tin. Thay vào đó, tôi xây dựng một hệ thống linh hoạt, có khả năng tự động học hỏi và điều chỉnh dựa trên các vector token chi tiết.

Ý nghĩa sâu sắc của phương pháp này nằm ở khả năng nắm bắt và biểu diễn tri thức ở mức độ chi tiết nhất. Mỗi token không chỉ là một phần của văn bản, mà còn là một đơn vị mang theo toàn bộ bối cảnh và ý nghĩa, tạo nên một hệ thống truy xuất thông tin có khả năng hiểu và phân tích văn bản một cách sâu sắc.

3.3. Tiêu kết chương 3

Chương 3 trình bày chi tiết quá trình thu thập, chuẩn bị và xử lý dữ liệu cho nghiên cứu, tập trung vào hai khía cạnh chính: bộ dữ liệu hỗ trợ phân rã câu hỏi và bộ dữ liệu macro cho hệ thống truy xuất thông tin.

Về bộ dữ liệu phân rã câu hỏi, nghiên cứu đã kể thừa và phát triển từ các nguồn dữ liệu quan trọng như HotpotQA, GSM8K, StrategyQA và SVAMP. Quá trình chuyển đổi dữ liệu được thực hiện một cách có hệ thống, bao gồm:

- Xây dựng cấu trúc JSON rõ ràng, linh hoạt và dễ sử dụng.
- Áp dụng các kỹ thuật chuyển đổi và ánh xạ dữ liệu chuyên sâu.
- Kiểm tra tính toàn vẹn và xác thực dữ liệu.

Giai đoạn tiền xử lý dữ liệu tập trung vào việc chuẩn hóa văn bản với các nguyên tắc chính:

- Chuyển đổi thông nhất sang chữ thường.
- Giảm thiểu chi phí tính toán.
- Bảo toàn cấu trúc và ngữ nghĩa gốc của văn bản.

Đối với bộ dữ liệu macro, nghiên cứu đã xây dựng một hệ sinh thái tri thức phức tạp từ các nguồn học thuật uy tín như Arxiv và Semantic Scholar. Điểm nhấn của phương pháp là:

- Áp dụng các kỹ thuật tiền xử lý tiên tiến.

- Phát triển giải pháp song song hóa và phân tán dữ liệu.
- Chuyển đổi dữ liệu sang không gian vector token sử dụng mô hình ColBERT.

Ý nghĩa sâu sắc của việc xây dựng bộ dữ liệu này không chỉ là cung cấp thông tin, mà còn là một bước đột phá trong việc phát triển các hệ thống trí tuệ nhân tạo có khả năng học hỏi, thích ứng và sáng tạo. Qua quá trình thu thập và tiền xử lý, nghiên cứu đã xây dựng được một nền tảng dữ liệu chất lượng, chuẩn bị sẵn sàng cho các bước phân tích và huấn luyện mô hình tiếp theo.

Chương 4: Thiết kế, lựa chọn kỹ thuật phân tích, trích xuất đặc trưng và triển khai các mô hình

4.1. Mô hình phân rã câu hỏi dựa trên Seq2Seq và T5

Xây dựng mô hình phân rã câu hỏi đặt mục tiêu là chuyển đổi một câu hỏi phức tạp thành nhiều câu hỏi con có liên quan. Trong nhiệm vụ huấn luyện mô hình phân rã câu hỏi thành các câu hỏi nhỏ, tôi lựa chọn mô hình chính để huấn luyện là kiến trúc Seq2Seq, đây là kiến trúc đơn giản nhưng lại rất hiệu quả trong các tác vụ xử lý ngôn ngữ tự nhiên. Đầu vào được xác định là câu hỏi chính. Đầu ra của mô hình là các câu hỏi đã được phân rã. Tuy nhiên, mô hình Seq2Seq chỉ có thể tạo một chuỗi ở đầu ra trong khi chúng ta cần có nhiều câu hỏi nh. Trong trường hợp này, nếu sử dụng kiến trúc đa tầng thì có thể tạo ra nhiều chuỗi ở đầu ra, tuy nhiên, các tầng trong kiến trúc này lại độc lập và sẽ không thể biết được câu hỏi đã được phân rã ở tầng khác thế nào để tránh việc hai tầng tạo ra một câu hỏi.

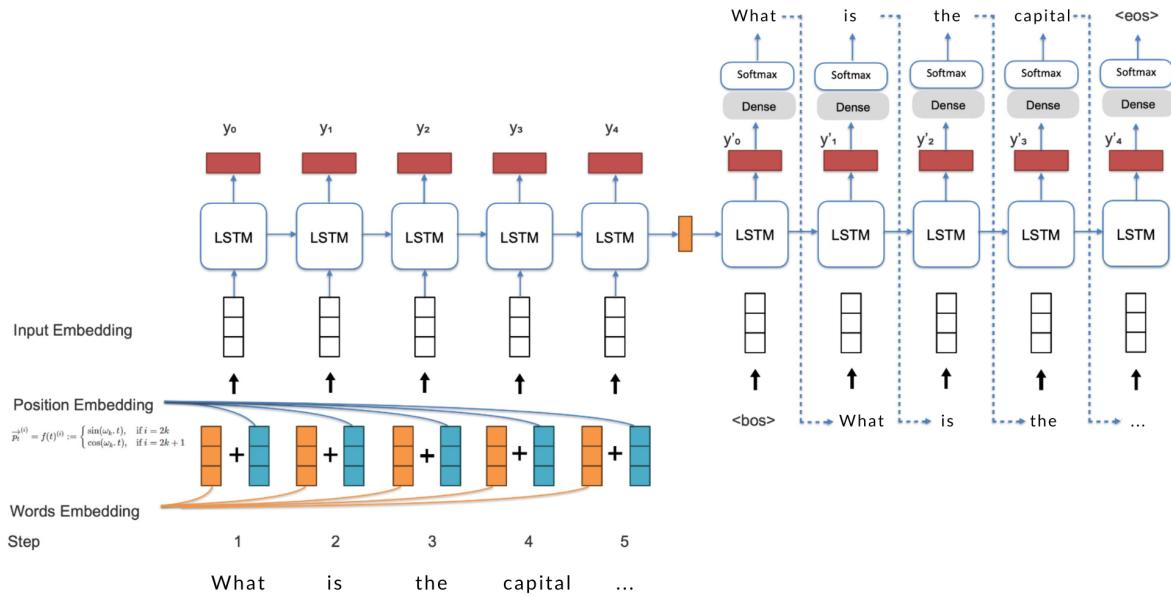
Giải pháp để Seq2Seq có thể tạo ra nhiều câu hỏi nhỏ khi chỉ phản hồi một chuỗi đầu ra duy nhất đó chính là chúng ta biến đổi dữ liệu đầu ra mong muốn. Cụ thể, tôi áp dụng kỹ thuật gộp nhiều câu hỏi con thành một chuỗi duy nhất bằng cách sử dụng một token đặc biệt để phân tách. Ta giới thiệu token đặc biệt [SEP] mang ý nghĩa kết thúc câu. Kỹ thuật này cho phép chuyển đổi một tập hợp các câu hỏi con thành một chuỗi duy nhất, giữ nguyên ngữ nghĩa và mối liên hệ giữa chúng. Cụ thể, với câu hỏi ban đầu "*What is the capital of Vietnam and when was it established ?*", mục tiêu là tách thành hai câu hỏi con:

- "*What is the capital of Vietnam ?*"
- "*When was the capital of Vietnam established ?*"

Sau khi áp dụng phương pháp, chuỗi đầu ra sẽ được biến đổi thành: “[*BOS*] *What is the capital of Vietnam? [SEP]* *When was the capital of Vietnam established? [EOS]*”

Token [SEP] đóng vai trò là điểm phân cách rõ ràng giữa các câu hỏi, cho phép mô hình Seq2Seq xử lý và tạo ra nhiều câu hỏi trong một chuỗi đầu ra duy

nhất. Phương pháp này mở ra khả năng mới trong việc phân rã và mở rộng các câu hỏi phức tạp. Điều này cũng giải quyết được vấn đề về sự thiếu liên kết trong kiến trúc đa tầng. Token [BOS] thể hiện việc bắt đầu sinh câu trả lời trong khi đó token [EOS] đánh dấu việc dừng quá trình sinh câu trả lời, đây là hai token đặc biệt quan trọng đối với mô hình Seq2Seq, tôi sẽ giải thích chi tiết hơn trong phần tiếp theo.



Hình 4.1: Mô hình Seq2Seq sử dụng để huấn luyện

Trong kiến trúc Seq2Seq với hai thành phần chính: Encoder và Decoder mỗi phần gồm nhiều khối mô hình mạng nơ-ron liên kết lại với nhau, ở đây, tôi chọn mạng nơ-ron LSTM để xây dựng mô hình. Đầu vào của mô hình sẽ đi qua lớp Encoder, và quá trình mã hóa bắt đầu bằng việc ánh xạ các token của câu hỏi gốc vào không gian vector liên tục, sử dụng kỹ thuật embedding kết hợp thông tin từ vựng và vị trí. Việc mã hóa vị trí được thực hiện thông qua các hàm sin và cos, cho phép mô hình nắm bắt mối quan hệ tương đối giữa các từ trong chuỗi. Encoder được cấu tạo từ nhiều lớp, mỗi lớp bao gồm cơ chế self-attention đa đầu và mạng kết nối lan truyền hoàn toàn. Kỹ thuật layer normalization và kết nối residual được áp dụng nhằm tăng cường khả năng học của mạng nơ-ron. Tại mỗi bước xử lý, các đơn vị LSTM cập nhật trạng thái ẩn, tích lũy và chuyển đổi thông tin ngữ nghĩa từ câu hỏi gốc. Xét một câu hỏi S với m token: s_1, \dots, s_m trong đó các token $s_i \in V_s$ và V_s là tập hợp các token trong tập từ vựng, token s_m là một token đặc biệt được biểu

diễn bởi [EOS] thể hiện cho việc kết thúc câu và dừng quá trình tính toán. Như vậy mỗi token s_i được ánh xạ vào không gian vector liên tục:

$$\bar{s}_i = W^e \cdot s_i + p_i.$$

- $W^e \in \mathbb{R}^{d \times |V_s|}$ là ma trận mã hoá của các token
- s_i là mã hoá dạng one-hot của các token.
- $p_i \in \mathbb{R}^d$ là vector vị trí của các token tại vị trí i
- d chính là số chiều của lớp ẩn

Trong quá trình phát triển mô hình phân rã câu hỏi, tôi đã lựa chọn sử dụng mô hình T5 thay vì xây dựng một kiến trúc hoàn toàn mới. T5 là một mô hình transformer tiên tiến được huấn luyện trên nền tảng Seq2Seq, với khả năng học từ tập dữ liệu siêu lớn. Ưu điểm chính của việc sử dụng T5 là nền tảng ngôn ngữ sẵn có, khả năng hiểu và diễn giải tiếng Anh một cách chuyên sâu, và tính linh hoạt cao trong việc chuyển đổi nhiệm vụ. Phương pháp transfer learning cho phép ta tinh chỉnh mô hình trên tập dữ liệu đặc thù về phân rã câu hỏi một cách nhanh chóng và hiệu quả. So với việc xây dựng mô hình từ đầu, cách tiếp cận này giảm thiểu đáng kể thời gian và nguồn lực phát triển, đồng thời tận dụng được kiến thức sẵn có của mô hình được huấn luyện trên quy mô lớn.

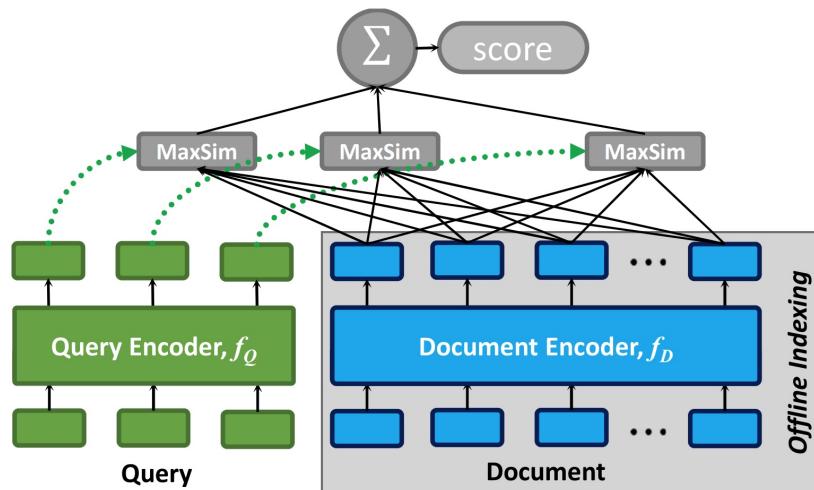
Vậy quá trình huấn luyện mô hình được diễn ra như sau: Câu hỏi đầu vào sẽ được phân tách thành các token, mỗi token sẽ được mã hoá và đi qua các khối LSTM đầu tiên, kết thúc quá trình mã hoá ở khối Encoder. Token đặc biệt đầu tiên được đưa vào [BOS] kết hợp với vector đầu ra từ khối Encoder để tính xác suất của token đầu tiên trong quá trình sinh dữ liệu, chúng ta tiếp tục lấy token vừa sinh ra tiến hành mã hoá và kết hợp với vector trước đó để tính xác suất từ tiếp theo. Việc sinh dữ liệu sẽ diễn ra đến khi gặp được token đặc biệt [EOS] thì quá trình sinh dữ liệu sẽ dừng lại.

4.2. Mô hình truy xuất thông tin với ColBERT và BM25

Trong quá trình xây dựng mô hình truy xuất, tôi kết hợp hai mô hình chính là ColBERT và BM25. Trước hết hãy bắt đầu với quá trình xây dựng ColBERT, do

hạn chế về nguồn lực tính toán cũng như việc huấn luyện ColBERT thực sự tiêu tốn lượng lớn tài nguyên khi thực hiện ở cấp độ token, tôi chia tập dữ liệu ra làm hai phần cho hai đợt huấn luyện với tỉ lệ 50-50. Mô hình cuối cùng sẽ được đánh giá sau khi hoàn tất hai lần huấn luyện.

Trong kiến trúc ColBERT, quá trình mã hóa được chia thành hai thành phần chính: mã hóa câu truy vấn và mã hóa dữ liệu văn bản. Một đặc điểm quan trọng của kiến trúc này là cách quản lý các tham số huấn luyện. Cụ thể, trong quá trình huấn luyện, tham số của bộ mã hóa câu truy vấn sẽ được liên tục cập nhật. Ngược lại, tham số của bộ mã hóa dữ liệu văn bản sẽ được giữ nguyên. Điều này xuất phát từ những xem xét thực tế về tài nguyên tính toán. Việc mã hóa lại toàn bộ dữ liệu văn bản là một quá trình vô cùng tốn kém về mặt tài nguyên máy tính. Nếu chúng ta phải mã hóa lại toàn bộ kho dữ liệu, chẳng hạn như toàn bộ nội dung trên internet, chi phí tính toán sẽ là không tưởng. Do đó, việc giữ nguyên tham số của bộ mã hóa dữ liệu không phải là do không thể cập nhật, mà là một chiến lược tối ưu hóa tài nguyên tính toán.Thêm vào đó, Atlas đã đề cập kết quả ghiên cứu của mình trong bài báo “*Atlas: Few-shot Learning with Retrieval Augmented Language Models*” và chứng minh rằng chỉ cần cập nhập tham số của bộ mã hóa truy xuất là đủ để mô hình hoạt động tốt [27].



Hình 4.2: Phương pháp tính độ tương đồng của ColBERT [12]

Cả hai khối mã hóa đều sử dụng mô hình BERT làm tiền đề và chỉ thực hiện cập nhập tham số trên bộ mã hóa câu truy vấn. Mục tiêu của bộ mã hóa câu truy vấn

là phải làm sao cho vector đại diện câu hỏi càng giống vector đại diện câu trả lời càng tốt. Để tính toán độ tương đồng giữa hai vector, tôi sử dụng hàm cosine similarity với phép toán MaxSim làm cốt lõi, cụ thể phương trình tính toán độ tương đồng được thể hiện như sau:

$$\text{sim}(Q, D) = \sum_{i=1}^{|Q|} \max_{j \in D} (q_i \cdot d_j)$$

Trong đó:

- Q là vector truy vấn.
- D là vector của tài liệu.
- q_i là vector của token thứ i trong câu truy vấn.
- d_j là vector của token thứ j trong văn bản tài liệu.

Phương trình trên trả về điểm số thể hiện mức độ tương đồng của câu truy vấn với các tài liệu. Trên thực tế trong quá trình huấn luyện bởi vì không cần cập nhập lại bộ mã hoá tài liệu, chúng ta có thể xây dựng sẵn một cơ sở dữ liệu vector, như vậy ta sẽ chỉ cần phải thực hiện phép so sánh và tiết kiệm đáng kể thời gian cho việc huấn luyện.

Khi xây dựng hàm mất mát, mục tiêu của hàm mất mát trong mô hình ColBERT là tối ưu hóa khoảng cách ngữ nghĩa giữa các vector đại diện, nhằm đảm bảo rằng các vector của truy vấn và tài liệu liên quan sẽ được ánh xạ gần nhau trong không gian vector, trong khi các vector của truy vấn và tài liệu không liên quan sẽ được đẩy xa nhau. Tôi áp dụng Margin Ranking Loss - một phương pháp học có giám sát hiệu quả trong các bài toán xếp hạng. Phương trình toán học của hàm mất mát được thể hiện như sau:

$$L = \max \left(0, \text{margin} - (\text{score}_{\text{pos}} - \text{score}_{\text{neg}}) \right).$$

Trong đó các thành phần:

- $\text{score}_{\text{pos}}$ là điểm số của cặp truy vấn và tài liệu liên quan.

- $\text{score}_{\text{neg}}$ là điểm số của cặp truy vấn tài liệu không liên quan.
- margin là một hằng số siêu tham số (thường được chọn là 0.1).

Nguyên lý hoạt động của hàm mất mát là Nếu $\text{score}_{\text{pos}} > \text{score}_{\text{neg}}$ một khoảng margin, mất mát sẽ bằng 0, ngược lại, hàm sẽ tạo ra một lực đẩy để điều chỉnh các vector về vị trí phù hợp hơn. Trong quá trình huấn luyện, hàm mất mát sẽ hướng dẫn mô hình điều chỉnh các vector sao cho vector truy vấn gần với các tài liệu liên quan và xa các tài liệu không liên quan. Kỹ thuật này cho phép ColBERT xây dựng được một không gian vector có khả năng phân biệt cao giữa các nội dung khác nhau, qua đó nâng cao chất lượng truy xuất thông tin.

Không chỉ dừng lại ở ColBERT, tôi kết hợp mô hình BM25 để tăng cường độ tin cậy cho mô hình, ColBERT sẽ đảm bảo truy xuất tốt trong ngữ nghĩa và BM25 sẽ truy xuất tốt với các thực thể xác định như tên người, tổ chức hay các sự kiện được nêu rõ trong câu hỏi. Trong phần này thay vì thực hiện BM25 trên các từ vựng, tôi sẽ tận dụng các token từ quá trình xây dựng ColBERT để tính toán BM25, như vậy BM25 cũng sẽ được thực hiện ở cấp độ token. Để hoàn thiện công thức tính điểm tương đồng, tôi sẽ kết hợp điểm số từ hai mô hình ColBERT và BM25 theo một phương pháp tổng hợp hybrid. Công thức cụ thể sẽ như sau:

$$\text{Hybrid Score} = \alpha \cdot \text{ColBERT Score} + \beta \cdot \text{BM25 Score}.$$

- α, β là các hệ số trọng số có thể điều chỉnh.
- ColBERT Score là kết quả từ hàm *MaxSim* thể hiện điểm tương đồng của ColBERT.
- BM25 Score là điểm số từ thuật toán BM25 tại cấp độ token.

Như vậy, phương pháp này không chỉ cải thiện độ chính xác của việc truy xuất thông tin mà còn tăng khả năng xử lý các truy vấn phức tạp với nhiều ngữ cảnh khác nhau

4.3. Mô hình chủ đề với FASTopic

Ta có thể thấy rằng việc tính toán ở cấp độ token trong ColBERT thực sự rất phức tạp tốn tài nguyên và cũng tiêu tốn đáng kể thời gian truy xuất, trên thực tế,

việc truy xuất thông tin chậm khiến cho giải pháp trở nên vô nghĩa đối với việc áp dụng vào bài toán thực. Để có thể thực sự đưa ColBERT từ nghiên cứu đến ứng dụng, ta cần phải tối ưu thời gian mà ColBERT truy xuất thông tin. Giải pháp chính là xây dựng các mô hình chủ đề, chúng ta đã có các vector của dữ liệu cần truy xuất được lưu trong cơ sở dữ liệu, vậy chúng ta có thể tiến hành phân loại các dữ liệu này thành các chủ đề, khi người dùng đặt câu hỏi, ta phân loại câu hỏi và tiến hành truy xuất trên các chủ đề này, như vậy sẽ tiết kiệm nhiều thời gian hơn.

Thách thức chính nằm ở việc xác định số lượng chủ đề phù hợp. Một số lượng chủ đề quá ít sẽ dẫn đến mảnh mai thông tin, trong khi quá nhiều chủ đề lại gây phức tạp hóa quá trình truy xuất. Do đó, việc tìm ra phương pháp tối ưu để xác định số lượng chủ đề trở nên vô cùng quan trọng. Để có thể tìm được K chủ đề cho cơ sở dữ liệu sao cho tối ưu nhất, mà K thay đổi linh động tùy theo dữ liệu được lưu, ta cần thực hiện một số tính toán. Giải pháp cho vấn đề này, chính là phương pháp tính điểm độ gắn kết (coherence score). Phương pháp này cho phép đánh giá chất lượng các chủ đề bằng cách phân tích mức độ liên quan ngữ nghĩa giữa các từ trong mỗi chủ đề. Bằng cách tính toán điểm độ gắn kết cho các số lượng chủ đề khác nhau, chúng ta có thể xác định một cách khoa học số lượng chủ đề tối ưu nhất cho tập dữ liệu. Công thức tính điểm độ gắn kết C_v được định nghĩa như sau:

$$C_v = \frac{1}{|T|} \sum_{t \in T} \frac{\sum_{w_i, w_j \in t} \text{PMI}(w_i, w_j)}{|(w_i, w_j) : w_i, w_j \in t|}.$$

Trong đó:

- T là điểm số từ thuật toán BM25 tại cấp độ token.
- $\text{PMI}(w_i, w_j)$ là điểm thông tin điều kiện tương hỗ (Pointwise Mutual Information) giữa w_i và w_j .
- $|(w_i, w_j) : w_i, w_j \in t|$ là số lượng cặp từ trong chủ đề.

Bằng cách hệ thống tính toán điểm độ gắn kết cho các số lượng chủ đề khác nhau, ta có thể xác định một cách khách quan và chính xác số lượng chủ đề tối ưu cho tập dữ liệu cụ thể. Quá trình này không chỉ là một phép tính thuần túy, mà còn

là một phân tích sâu sắc về cấu trúc và bản chất của thông tin.

Sau khi xác định được số lượng chủ đề cho dữ liệu, tôi tiến hành tạo các mối quan hệ, chủ đề cho dữ liệu, trong việc sử dụng FASTopic với số lượng chủ đề là K , tôi chọn mô hình All-mini-L6-v2 làm mô hình mã hoá cho FASTopic, bởi đây là một mô hình nhẹ và được giới thiệu khuyên dùng trong bài báo của FASTopic.

4.4. Xây dựng hệ thống chatbot sử dụng tri thức nội bộ

Để hoàn thiện toàn bộ hệ thống, chúng ta cần phải xây dựng các phần còn thiếu trong một phần mềm RAG, như đã nêu trước đó về các bước xây dựng RAG, dù đã hoàn thiện bộ truy xuất, ta cần giải pháp để chia nhỏ dữ liệu cũng như thực hiện lưu trữ dữ liệu, ngoài ra còn cần các phần phụ trợ khác.

4.4.1. Kỹ thuật chia nhỏ văn bản theo ngữ nghĩa

Trước hết đối với việc chia nhỏ dữ liệu và lưu vào cơ sở dữ liệu, tôi sử dụng phương pháp chia nhỏ theo ngữ nghĩa, là một phương pháp tiên tiến trong việc phân tách văn bản, vượt trội so với các kỹ thuật truyền thống của việc chia văn bản. Không giống như các phương pháp cắt văn bản theo độ dài cố định hay dựa trên các dấu ngắt đơn giản, phương pháp này sử dụng trí tuệ nhân tạo để hiểu và tách văn bản dựa trên ngữ nghĩa và bối cảnh nội dung.

Đầu tiên, hệ thống sử dụng các mô hình transformer tiên tiến như Sentence Transformers để chuyển đổi các câu văn thành các vector embedding ở đây tôi tái sử dụng mô hình All-MiniLM-L6-v2 [28]. Những vector này không chỉ là biểu diễn số học của từng câu, mà còn mang trong mình toàn bộ bối cảnh ngữ nghĩa, mối quan hệ và ý nghĩa sâu sắc của nội dung.

Tiếp theo, hệ thống tính toán độ tương đồng ngữ nghĩa giữa các câu liền kề bằng cách sử dụng phép đo cosine similarity. Điều này cho phép xác định các điểm chuyển đổi ngữ nghĩa - những nơi mà nội dung chuyển sang một chủ đề hoặc bối cảnh hoàn toàn mới.

Phương pháp chia nhỏ này mang lại những lợi ích đáng kể so với các phương pháp truyền thống:

- Thứ nhất, nó giữ nguyên được bối cảnh và mối liên hệ ngữ nghĩa của văn bản. Mỗi đoạn văn được chia tách không chỉ đảm bảo độ dài phù hợp, mà còn đảm bảo tính liên kết và mẠch lạc về mặt nội dung.
- Thứ hai, phương pháp này rất linh hoạt. Bằng cách điều chỉnh các tham số như ngưỡng tương đồng và kích thước chunk tối đa, hệ thống có thể thích ứng với nhiều loại văn bản và yêu cầu khác nhau.

4.4.2. Kết hợp và hoàn thiện

Sau khi xây dựng và hoàn thiện các phần riêng lẻ, ta tiến đến kết hợp toàn bộ các thành phần lại và xây dựng ứng dụng cuối cùng như mục đích đề tài đặt ra, xây dựng chatbot cho tri thức nội bộ. Các bước kết hợp được thực hiện như sau:

Trước hết, bắt đầu với việc xây dựng cơ sở dữ liệu, dữ liệu được đưa vào ở đây bao gồm tệp văn bản dạng PDF và được link đến các trang web bất kỳ có thể truy cập được. Đối với tệp PDF, ta dễ dàng trích xuất văn bản tệp, còn đối với dữ liệu từ internet, cần xây dựng chương trình thu thập dữ liệu, ta có thể tải dữ liệu từ trang web xuống dạng thô, lưu vào tệp html rồi sử dụng beatifulsoup và markdownify để chuyển html thành markdown, từ đó ta có văn bản thô của trang web.

Tiếp đến, các đoạn văn bản sẽ được cắt nhỏ theo phương pháp chia cắt theo ngữ nghĩa đã được xây dựng ở trên. Các đoạn văn bản nhỏ này tiếp tục được phân tách ở cấp độ token và tiến hành mã hóa các vector này sau đó được lưu vào cơ sở dữ liệu Neo4j. Sau khi mã hóa văn bản, ta tiếp tục mã hóa văn bản với All-MiniLM-L6-v2 và tính độ gắn kết văn bản với đa dạng số lượng chủ đề khác nhau, khi phân cụm chủ đề với FASTopic. Sau khi chọn được số lượng chủ đề thích hợp, tiến hành tạo các “center node” và tạo các mối quan hệ giữa các điểm dữ liệu. Sau bước này ta hoàn thiện bước xây dựng cơ sở dữ liệu tri thức nội bộ.

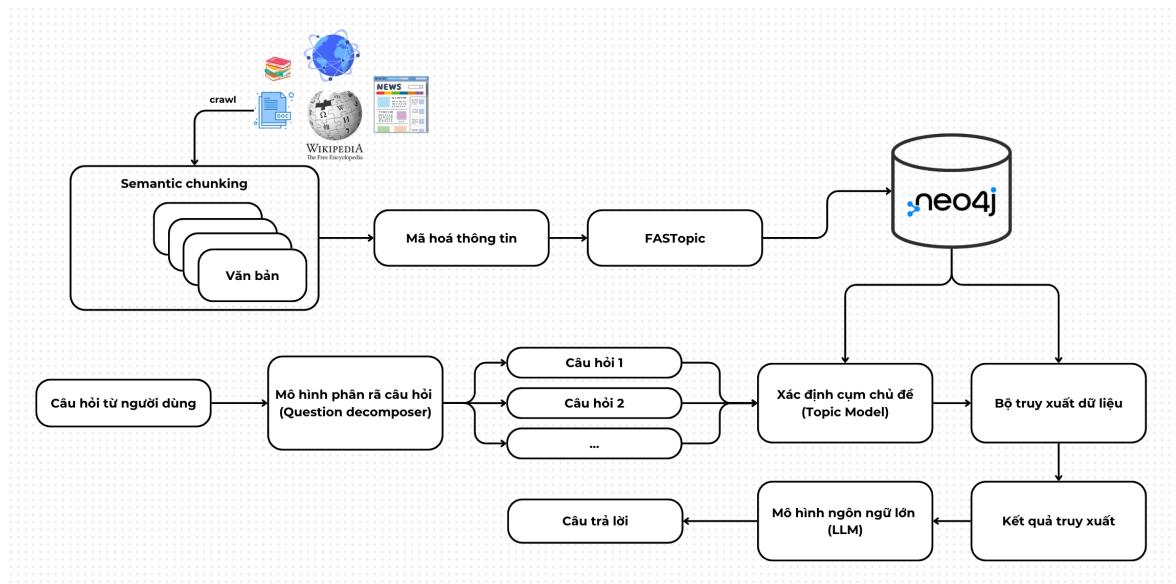
Khi người dùng đặt câu hỏi, ta sử mô hình phân rã câu hỏi tách câu hỏi người dùng thành các câu hỏi nhỏ hơn và tiến hành truy xuất thông tin trên các câu hỏi này. Các câu hỏi này sẽ được phân loại vào các chủ đề và tiến hành truy xuất trên các chủ đề đó để lấy thông tin. Thông tin truy xuất được sẽ được tổng hợp lại

và cung cấp cho LLM để trả lời. Ở đây, tôi sử dụng mô hình Gemma2:27b với dung lượng mô hình 16GB, được cung cấp từ Ollama, đơn giản và dễ tích hợp, sử dụng cục bộ [29].

Trong giai đoạn phát triển hệ thống chatbot tri thức nội bộ, việc xây dựng giao diện người dùng và hệ thống API là một bước quan trọng đòi hỏi sự tích hợp chặt chẽ giữa công nghệ front-end và back-end. Tôi lựa chọn Next.js làm nền tảng phát triển giao diện, kết hợp với FastAPI để xây dựng các điểm cuối (endpoints) phục vụ truy vấn và trao đổi dữ liệu.

Mục tiêu chính của giao diện là tạo ra một trải nghiệm người dùng trực quan, thân thiện và hiệu quả. Next.js được ưu tiên lựa chọn nhờ khả năng render trang động, tối ưu hóa hiệu suất và hỗ trợ mạnh mẽ cho ứng dụng React. Đồng thời, FastAPI sẽ đảm nhiệm việc xử lý các yêu cầu, kết nối với cơ sở dữ liệu Neo4j và mô hình ngôn ngữ, mang lại giải pháp toàn diện và linh hoạt cho hệ thống chatbot tri thức.

Quá trình phát triển sẽ tập trung vào việc thiết kế các API có cấu trúc rõ ràng, an toàn và dễ dàng mở rộng, đáp ứng các yêu cầu về tra cứu, tương tác và quản lý tri thức của người dùng.



Hình 4.3: Kiến trúc hệ thống chatbot nội bộ

Chương 5: Đánh giá kết quả và ứng dụng thực tế

5.1. Đánh giá kết quả tổng hợp

5.1.1. Đánh giá hiệu xuất mô hình phân rã câu hỏi

Nghiên cứu tập trung phát triển các kỹ thuật tiên tiến nhằm phân rã câu hỏi phức tạp thành các truy vấn ngữ nghĩa chính xác và hiệu quả. Quá trình nghiên cứu đã áp dụng những phương pháp kỹ thuật tiên tiến trong lĩnh vực xử lý ngôn ngữ tự nhiên.

Phương pháp phân rã câu hỏi được xây dựng dựa trên ba kỹ thuật cốt lõi: trích xuất từ khóa ngữ nghĩa, phân tích cấu trúc ngữ pháp và ánh xạ quan hệ tri thức. Mỗi kỹ thuật đóng vai trò quan trọng trong việc chuyển đổi câu hỏi gốc thành các truy vấn con mang tính chất học thuật và chuyên sâu.

Phương pháp	Độ chính xác	Nhận xét
ROUGE 1	65.14%	Khả năng nhận diện từ khóa chính xác cao
ROUGE 2	46.57%	Thể hiện khả năng kết nối thông tin phức tạp
ROUGE L	60.11%	Đảm bảo tính toàn vẹn ngữ nghĩa
BLEU	30.38%	Đánh giá thấp với ngữ nghĩa phức tạp và cấu trúc câu đa dạng.

Bảng 5.1: Kết quả mô hình phân rã câu hỏi

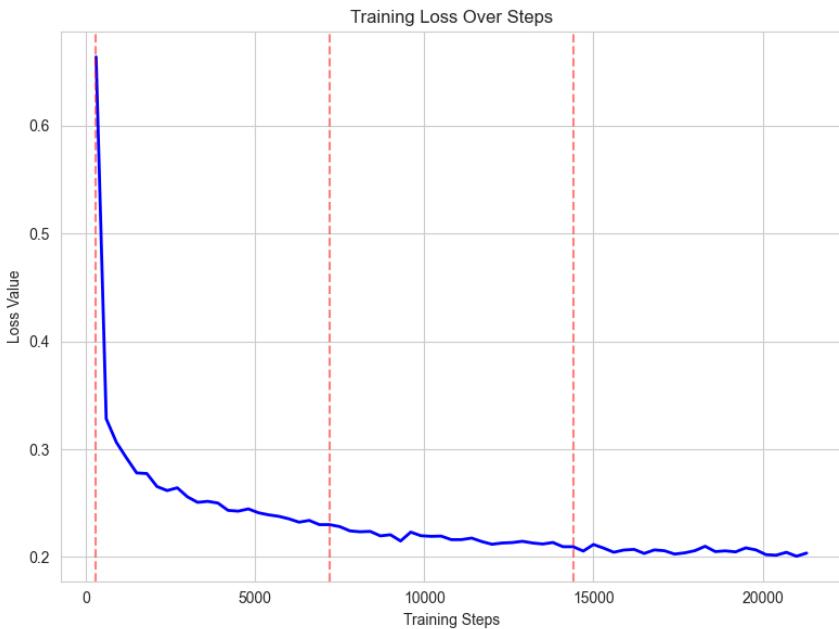
Kết quả nghiên cứu cho thấy các kỹ thuật phân rã câu hỏi đã đạt được những bước tiến đáng kể trong việc xử lý ngôn ngữ tự nhiên. Phương pháp trích xuất từ khóa thể hiện hiệu suất ấn tượng, với khả năng nhận diện các yếu tố then chốt của câu hỏi một cách chính xác.

Quá trình huấn luyện mô hình được thực hiện trên một hệ thống máy tính với cấu hình kỹ thuật cao, nhằm đảm bảo hiệu năng tối ưu và khả năng xử lý các tác vụ phức tạp trong lĩnh vực xử lý ngôn ngữ tự nhiên.

Câu Hình Phần Cứng:

- Card đồ họa: NVIDIA GeForce GTX 3080

- Bộ nhớ GPU: 32GB
- Bộ nhớ RAM hệ thống: 64GB



Hình 5.1: Quá trình huấn luyện mô hình phân rã câu hỏi

Mặc dù đạt được nhiều tiến bộ, nghiên cứu vẫn nhận thức rõ những giới hạn của phương pháp. Độ phức tạp của ngôn ngữ tự nhiên, sự đa dạng về ngữ cảnh và cấu trúc câu hỏi vẫn là những thách thức lớn đối với hệ thống.

Một số hạn chế chính bao gồm:

- Khó khăn trong việc xử lý các câu hỏi có tính chất trừu tượng và phức tạp cao.
- Giới hạn trong việc duy trì nguyên bản ngữ nghĩa khi phân rã.
- Độ chính xác chưa đạt mức quá cao ở các trường hợp đặc thù.

Nghiên cứu về kỹ thuật phân rã câu hỏi đã mở ra những triển vọng đáng khích lệ trong lĩnh vực xử lý ngôn ngữ tự nhiên. Qua việc áp dụng các phương pháp tiên tiến như trích xuất từ khóa ngữ nghĩa, phân tích cấu trúc ngữ pháp và ánh xạ quan hệ tri thức, tôi đã đạt được những kết quả đáng ghi nhận.

Tuy nhiên, nghiên cứu cũng chỉ ra những thách thức quan trọng cần được nghiên cứu sâu hơn. Độ phức tạp của ngôn ngữ tự nhiên, sự đa dạng về ngữ cảnh và cấu trúc câu hỏi vẫn là những rào cản chính cần vượt qua.

5.1.2. Đánh giá hiệu xuất mô hình truy xuất

Trong đánh giá hiệu xuất mô hình truy xuất được thực hiện nhằm phân tích chi tiết về năng lực truy xuất thông tin của phương pháp Hybrid kết hợp giữa ColBERT và BM25. Nghiên cứu tập trung vào việc so sánh hiệu quả của mô hình với các tham số trọng số alpha (α) khác nhau, góp phần làm rõ khả năng tích hợp các phương pháp truy xuất dày đặc (dense retrieval) và thưa (sparse retrieval).

Mô hình được huấn luyện trên thiết bị M3 Max Pro với cấu hình như sau:

Thành phần	Thông số kỹ thuật
CPU speed	2393MHz
CPU cores	14
GPU speed	9008MHz
GPU core	30

Bảng 5.2: Thông tin cấu hình thiết bị huấn luyện mô hình ColBERT

Kết quả đánh giá cho thấy mô hình Hybrid với tham số $\alpha = 0.3$ thể hiện hiệu suất vượt trội nhất trên hầu hết các chỉ số. Ở cấu hình này, mô hình thể hiện khả năng xếp hạng ưu việt, cho phép tìm kiếm và định vị các tài liệu liên quan một cách chính xác và hiệu quả.

Việc kết hợp ColBERT (truy xuất dựa trên học sâu) và BM25 (truy xuất dựa trên từ khóa) đã mang lại kết quả khả quan. Cấu hình $\alpha = 0.3$ cho thấy sự cân bằng tối ưu giữa hai kỹ thuật, giúp cải thiện đáng kể độ chính xác và khả năng thu hồi thông tin so với các cấu hình khác. Kết quả nghiên cứu chỉ ra rằng phương pháp Hybrid có tiềm năng to lớn trong việc nâng cao hiệu quả tìm kiếm thông tin. Bằng cách kết hợp các ưu điểm của hai phương pháp truy xuất khác nhau, mô hình đã chứng minh khả năng vượt trội trong việc xử lý các nhiệm vụ truy xuất thông tin phức tạp.

Phương pháp	ColBERT	Hybrid ($\alpha = 0.3$)	Hybrid ($\alpha = 0.5$)	Hybrid ($\alpha = 0.7$)
MRR@3	59.33%	62.33%	55.33%	55.33%
MAP@3	59.33%	62.33%	55.33%	55.33%
Recall@3	95.11%	96.11%	93.11%	93.11%
NDCG@3	69.24%	72.43%	65.24%	65.24%

Bảng 5.3: *Dánh giá hiệu xuất mô hình truy xuất thông tin*

Tuy nhiên, sự khác biệt giữa các cấu hình không quá lớn, cho thấy tính nhạy cảm của mô hình với việc điều chỉnh trọng số. Điều này gợi mở hướng nghiên cứu tiếp theo về việc tinh chỉnh các tham số để đạt hiệu quả tối ưu. Nhìn chung, nghiên cứu khăng định tính ưu việt của phương pháp Hybrid trong việc cải thiện chất lượng truy xuất thông tin, đồng thời mở ra triển vọng ứng dụng rộng rãi trong các hệ thống tìm kiếm và tra cứu thông tin hiện đại.

5.1.3. Kết quả xây dựng giao diện người dùng cho chatbot nội bộ

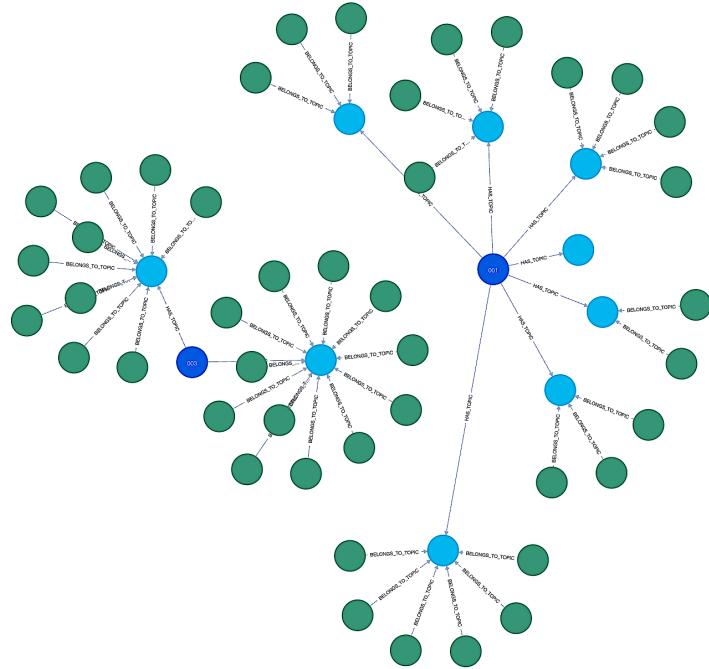
Sau khi kết hợp các thành phần với nhau, ta cũng cần tiến hành kiểm tra, kiểm thử và sửa các lỗi hệ thống. Như vậy, giao diện của hệ thống được thiết kế nhằm mang lại trải nghiệm người dùng trực quan, linh hoạt và thuận tiện. Các chức năng chính bao gồm:

- Tạo mới cuộc trò chuyện.
- Tải tệp tài liệu từ máy tính.
- Cung cấp đường link website để hệ thống tự động thu thập dữ liệu.

Trong đó, một tính năng đặc biệt của hệ thống là khả năng thu thập và xử lý dữ liệu từ các nguồn khác nhau một cách tự động, giúp mở rộng phạm vi tri thức một cách linh hoạt.

Hình : giao diện các phần mềm Chatbot nội bộ

Sau khi thu thập, mã hóa, dữ liệu được lưu trữ và kết nối trong cơ sở dữ liệu đồ thị Neo4j. Hệ thống tự động xây dựng các mối quan hệ phức tạp giữa các điểm dữ liệu, tạo nên một mạng lưới tri thức động và linh hoạt.



Hình 5.2: Dữ liệu được biểu diễn trong Neo4j

Trong hình 5.2 biểu diễn các node dữ liệu được liên kết với các chủ đề, mỗi node dữ liệu chứa nhiều loại thông tin, văn bản, vector, số lượng token, chủ đề, ... Mỗi tài liệu sẽ được tạo các chủ đề thông qua FASTopic và node này sẽ liên kết với các node dữ liệu khác thuộc cùng loại thông tin.

Để đánh giá khả năng vận hành của hệ thống, tôi tiến hành so sánh tốc độ truy xuất dữ liệu qua hai phương thức chính:

Số trang	560	260	83
Số từ	101085	48085	15057
Thời gian mã hóa	27m16.6s	12m16.6s	3m45.2s
Thời gian truy xuất	16.2s	7.9s	2.3s
Số lượng node	3951	1951	754
Số lượng chủ đề	16	12	6
Thời gian truy xuất trên chủ đề	3.6s	2.8s	1.6s

Bảng 5.4: Tốc độ truy xuất dữ liệu

Kết Quả Nhận Định: Phương thức truy xuất theo chủ đề thể hiện ưu thế rõ rệt về tốc độ xử lý. Tuy nhiên, việc đánh giá không chỉ dừng lại ở tốc độ mà còn phải

xem xét đến chất lượng và độ chính xác của thông tin được truy xuất.

Mặc dù đạt được nhiều tiến bộ, hệ thống vẫn còn một số hạn chế cần được nghiên cứu và cải thiện

- Nâng cao độ chính xác của việc truy xuất thông tin
- Tối ưu hóa thuật toán phân loại chủ đề
- Mở rộng khả năng xử lý các nguồn dữ liệu phức tạp

Nghiên cứu đã chứng minh tính khả thi của phương pháp truy xuất thông tin theo chủ đề, mở ra những triển vọng mới trong việc xây dựng các hệ thống tra cứu tri thức thông minh và hiệu quả. Hướng nghiên cứu tiếp theo sẽ tập trung vào việc tinh chỉnh các thuật toán, nâng cao chất lượng truy xuất và mở rộng phạm vi áp dụng của hệ thống.

5.2. *Khả năng ứng dụng thực tế*

5.2.1. *Tính thực khả thi của đề tài khi sử dụng trong thực tế.*

Đề tài có tiềm năng được chào đón, và cải tiến để sử dụng trong thực tế, nhu cầu của thị trường đã sẵn có về yêu cầu của các doanh nghiệp, các dịch vụ chăm sóc khách hàng hay các công ty tư vấn, bất cứ lĩnh vực nào cũng có yêu cầu về một trợ lý thông minh giúp họ nhận được câu trả lời lập tức mà không cần phải đọc quá nhiều văn bản tài liệu để hiểu về các chính sách dịch vụ.

Trên thực tế, mô hình RAG đã rất được quan tâm, các phương pháp ứng dụng RAG rất đa dạng, đơn giản với một trợ lý giúp người dùng hiểu hơn về doanh nghiệp của họ trên website cũng đã khiến khách hàng ấn tượng và tin tưởng vào dịch vụ của nhà cung cấp hơn, việc nhận thông tin nhanh chóng giúp doanh nghiệp xác nhận và thu vào nhiều đơn hàng hơn khi tiếp thị.

Tóm lại, ứng dụng truy xuất thông tin nội bộ và việc nắm toàn bộ hệ thống là một phương án tối ưu và đầy tiềm năng khi mà nhu cầu đã sẵn có thì các ứng dụng tối ưu và bảo mật sẽ càng được quan tâm hơn hết.

5.2.2. Rủi ro trong thực tế.

Trên thực tế, khi xây dựng và quản trị hệ thống chatbot nội bộ, ta cần khá nhiều nguồn tài nguyên để duy trì toàn bộ các thành phần của hệ thống từ bộ truy xuất đến cơ sở dữ liệu.

Việc không tối ưu được dung lượng hệ thống hay tài nguyên không đủ để tự quản trị các mô hình LLM dẫn đến phải sử dụng các mô hình nhỏ có thể ảnh hưởng đến chất lượng câu trả lời khi bộ truy xuất đã tìm kiếm và tổng hợp đúng thông tin cho câu trả lời nhưng LLM lại không hiểu được và đưa ra câu trả lời sai.

Tốc độ phản hồi cũng là một vấn đề cần quan tâm, khác với mặt nghiên cứu, khi xây dựng ứng dụng thì trải nghiệm người dùng là một yếu tố quan trọng, tối ưu và tăng tốc độ phản hồi của hệ thống cần rất nhiều thời gian để nghiên cứu và cải tiến, nếu sử dụng hệ thống lớn thì cũng gây khó khăn cho các doanh nghiệp nhỏ không đủ khả năng chi trả.

5.3. Hướng phát triển trong tương lai

Có thể thấy chúng ta đã hoàn thiện nhiều bước để đi đến kết quả hiện tại, tiêu nhiên, mô hình hiện tại còn nhiều hạn chế cần cải tiến.

Trước hết, để hoàn thiện hệ thống cuối cùng, ta cần phải thực hiện nhiều bước mã hóa khác nhau bằng nhiều phương pháp khác nhau, đầu tiên, ta mã hóa với T5 để phân rã câu hỏi, tiếp đến ta mã hóa ở mức Token với BERT để lưu trữ, rồi ta lại mã hóa bằng All-MiniLM-L6-V2 để xây dựng các chủ đề, nếu có thể tối ưu hóa quá trình này và dùng duy nhất một bộ mã hóa, nó sẽ giảm đáng kể thời gian và tài nguyên hệ thống.

Tiếp đến, chúng ta đang sử dụng Neo4j để lưu trữ nhưng lại chưa hoàn toàn khai thác hết toàn bộ tiềm năng của cơ sở dữ liệu này. Neo4j mạnh mẽ ở việc biểu diễn và truy xuất dựa trên các mối quan hệ, nhưng chúng ta lại chỉ dùng ở các mô hình chủ đề, ta có thể khai thác và tạo quan hệ cho các node thông qua việc xác định các thực thể, như vậy sẽ biết được node này chứa thông tin liên quan đến ai, tổ chức hay sự kiện hoặc bất cứ thực thể nào ta muốn xác định. Như vậy quá trình truy xuất sẽ toàn vẹn hơn.

Một vấn đề khác chính là chúng ta cần một phương xác để xác định được, khi nào thì cần truy xuất. Trong nhiều trường hợp, LLM có thể phản hồi người dùng mà không cần cung cấp thêm dữ liệu, xác định được điều này sẽ giúp hệ thống tối ưu và tiết kiệm đáng kể tài nguyên. Có thể kể đến ý tưởng của mô hình FLARE, khi xem xét độ tự tin của mô hình và tiến hành truy xuất khi cần thiết, tuy nhiên phương pháp này không thực sự khả thi trong thực tế, độ tự tin của mô hình khi sinh văn bản rất khó để đo lường, thêm vào đó, phương pháp này sẽ không giúp LLM tránh hiện tượng "ảo giác" khi đưa phản hồi cho người dùng [30].

Cuối cùng, hệ thống RAG thực hiện truy xuất thông tin trên cơ sở dữ liệu vector, vậy ta có thể hiểu rằng bất cứ loại dữ liệu nào có thể mã hoá thành vector đều có thể đưa vào trong RAG. Nghĩa là việc truy xuất không nên chỉ dừng lại ở văn bản mà còn có thể sử dụng với hình ảnh, âm thanh, ...

Trên đây là một số hướng phát triển đi từ hạn chế của đề tài cũng như quá trình xây dựng giúp phát hiện ra những thiếu sót, những hướng đi mới giúp hệ thống phát triển hơn.

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Đề tài nghiên cứu đã thành công trong việc vượt qua các hạn chế của hệ thống truy xuất thông tin RAG truyền thống, mở ra những định hướng đột phá trong lĩnh vực xử lý ngôn ngữ tự nhiên và quản trị tri thức. Qua quá trình nghiên cứu chuyên sâu, đề tài đã xây dựng được một kiến trúc hệ thống chatbot nội bộ mang tính đột phá, tích hợp các phương pháp tiên tiến trong việc truy xuất và tổng hợp thông tin.

Một trong những đóng góp quan trọng nhất của nghiên cứu là việc phát triển phương pháp phân rã câu hỏi một cách hệ thống và tinh vi. Phương pháp này cho phép hệ thống hiểu sâu hơn về bản chất của truy vấn, từ đó cải thiện đáng kể độ chính xác và phạm vi truy xuất thông tin. Bằng cách áp dụng kỹ thuật phân tích ngữ nghĩa tiên tiến, nghiên cứu đã vượt qua giới hạn của các mô hình truyền thống, mang lại khả năng xử lý thông tin phức tạp một cách linh hoạt và chính xác.

Nghiên cứu còn mang đến một đột phá quan trọng trong việc kết hợp các mô hình truy xuất khác nhau. Bằng cách tích hợp mô hình truy xuất nơ-ron (neural retrieval) với mô hình truy xuất dựa trên xác suất thống kê (probabilistic retrieval), nhóm nghiên cứu đã tạo ra một phương pháp lai (hybrid approach) vượt trội so với các phương pháp đơn lẻ. Kỹ thuật này không chỉ cải thiện hiệu quả truy xuất dữ liệu mà còn tăng tính linh hoạt và độ tin cậy của hệ thống.

Đặc biệt, nghiên cứu đã giải quyết thành công thách thức về tính khả thi của các mô hình học sâu phức tạp, điển hình như mô hình ColBERT. Thay vì dừng lại ở mức nghiên cứu lý thuyết, nhóm tác giả đã phát triển các giải pháp tối ưu hóa tài nguyên tính toán, giúp đưa các mô hình tiên tiến này vào ứng dụng thực tế một cách hiệu quả.

Một điểm nhấn quan trọng khác là sự giới thiệu phương pháp chọn số lượng mô hình tối ưu cho FASTopic - một phương pháp chủ đề mới được phát triển. Phương pháp này thể hiện tính sáng tạo và tiềm năng ứng dụng cao trong việc phân

tích và trích xuất tri thức từ các tập dữ liệu lớn và phức tạp.

Tóm lại, nghiên cứu không chỉ là một đóng góp học thuật quan trọng mà còn mở ra những triển vọng ứng dụng thực tiễn trong lĩnh vực xử lý ngôn ngữ và quản trị thông tin. Các kỹ thuật và phương pháp được giới thiệu trong nghiên cứu này hứa hẹn sẽ là nền tảng cho các hệ thống trí tuệ nhân tạo thế hệ mới, với khả năng xử lý thông tin thông minh, chính xác và hiệu quả hơn.

Mặc dù nghiên cứu đã đạt được những kết quả đáng khích lệ, tôi vẫn phải thừa nhận một số hạn chế quan trọng cần được xem xét và cải thiện trong các nghiên cứu tiếp theo:

Nghiên cứu hiện tại được thực hiện với bộ dữ liệu có quy mô tương đối hạn chế, tập trung chủ yếu trong một miền tri thức cụ thể. Điều này làm giảm khả năng khái quát hóa kết quả nghiên cứu và tính đại diện của mô hình. Kích thước mẫu nhỏ có thể dẫn đến việc không phản ánh đầy đủ tính phức tạp và đa dạng của không gian thông tin thực tế.

Mặc dù đã có những cải tiến đáng kể trong việc tối ưu hóa tài nguyên tính toán, hệ thống vẫn chưa đạt được hiệu suất tối ưu trong việc xử lý các truy vấn phức tạp và khôi lượng dữ liệu lớn. Các thử nghiệm cho thấy độ trễ xử lý và độ chính xác truy xuất thông tin vẫn còn một số hạn chế nhất định.

Quá trình đánh giá hiệu quả của hệ thống chủ yếu dựa trên các bộ dữ liệu và kịch bản thử nghiệm được xác định trước. Điều này có thể không phản ánh đầy đủ những tình huống phức tạp và động trong môi trường thực tế. Bởi vì dữ liệu nội bộ có thể có nhiều kiến thức chuyên sâu hơn và khó hiểu hơn.

2. Kiến nghị

2.1 Hướng phát triển nghiên cứu

Trong bối cảnh phát triển nhanh chóng của công nghệ trí tuệ nhân tạo, việc mở rộng và nâng cao chất lượng nghiên cứu về hệ thống truy xuất thông tin là một yêu cầu cấp thiết. Từ những kết quả và hạn chế của nghiên cứu hiện tại, tôi nhận thấy việc mở rộng phạm vi nghiên cứu sang các miền tri thức khác nhau là một trong những ưu tiên hàng đầu. Việc này không chỉ giúp tăng tính khái quát của mô

hình mà còn nâng cao khả năng thích ứng và linh hoạt của hệ thống. Tôi đề xuất xây dựng các bộ dữ liệu tham chiếu đa lĩnh vực, từ khoa học kỹ thuật đến khoa học xã hội, nhằm kiểm chứng và hoàn thiện hiệu quả của phương pháp nghiên cứu.

Một trong những ưu tiên hàng đầu là mở rộng phạm vi nghiên cứu sang các miền tri thức khác nhau. Việc này không chỉ giúp tăng tính khái quát của mô hình mà còn nâng cao khả năng thích ứng và linh hoạt của hệ thống. Tôi khuyến nghị xây dựng các bộ dữ liệu tham chiếu đa lĩnh vực, từ khoa học kỹ thuật đến khoa học xã hội, nhằm kiểm chứng và hoàn thiện hiệu quả của phương pháp nghiên cứu.

Sự phát triển không ngừng của các kỹ thuật học máy đòi hỏi chúng ta không ngừng cải tiến kiến trúc mô hình. Trong quá trình nghiên cứu, tôi đã tập trung vào việc cải tiến kiến trúc mô hình nhằm đáp ứng các thách thức về mặt kỹ thuật và ứng dụng. Cụ thể, tôi đã phát triển các phương pháp tối ưu hóa tài nguyên tính toán dựa trên kỹ thuật học tăng cường (transfer learning) và học liên tục (continual learning). Những cải tiến này cho phép mô hình thích ứng tốt hơn với các tác vụ mới mà không cần phải huấn luyện lại toàn bộ hệ thống, đồng thời duy trì được kiến thức đã học từ các tác vụ trước đó.

Về mặt hiệu năng, tôi đã tập trung nghiên cứu và phát triển các giải pháp toàn diện nhằm nâng cao khả năng xử lý của hệ thống. Điều này bao gồm việc tối ưu hóa các thuật toán truy vấn và tổng hợp thông tin, áp dụng các kỹ thuật cache thông minh, và phát triển các phương pháp song song hóa xử lý. Tôi cũng đã thiết kế và triển khai các thử nghiệm đánh giá hiệu năng một cách có hệ thống, từ đó đề xuất các giải pháp cải thiện độ trễ và tăng tốc độ xử lý mà vẫn đảm bảo chất lượng kết quả đầu ra.

Ngoài ra, tôi còn chú trọng việc phát triển các công cụ đánh giá và phân tích hiệu quả của hệ thống một cách toàn diện. Điều này bao gồm việc xây dựng các metrics đánh giá chất lượng kết quả, đo lường độ chính xác và độ phủ của thông tin, cũng như đánh giá khả năng mở rộng của hệ thống. Những công cụ này không chỉ giúp tôi có cái nhìn sâu sắc về hiệu quả của các phương pháp đề xuất mà còn cung cấp cơ sở cho việc so sánh và lựa chọn giải pháp phù hợp trong các bối cảnh ứng dụng khác nhau.

2.2 Kiến nghị ứng dụng thực tiễn

Từ kết quả nghiên cứu đạt được, tôi nhận thấy tiềm năng to lớn của hệ thống truy xuất thông tin thông minh trong việc chuyển đổi và nâng cao hiệu quả hoạt động của nhiều lĩnh vực. Các kiến nghị ứng dụng thực tiễn được đề xuất dựa trên nền tảng khoa học vững chắc, đồng thời hướng tới tính khả thi và giá trị thực tiễn cao.

Trước hết, trong lĩnh vực doanh nghiệp, tôi đề xuất phát triển các hệ thống chăm sóc khách hàng thông minh tích hợp công nghệ truy xuất thông tin tiên tiến. Hệ thống này không chỉ đơn thuần trả lời các câu hỏi thông thường mà còn có khả năng phân tích ngữ cảnh chuyên sâu, học hỏi từ tương tác trước đó, và đưa ra các giải pháp phù hợp với từng khách hàng. Đặc biệt, tôi chú trọng việc phát triển các thuật toán thông minh để tối ưu hóa trải nghiệm người dùng, đảm bảo tính nhất quán và chính xác trong việc cung cấp thông tin.

Về mặt quản trị tri thức doanh nghiệp, tôi đề xuất xây dựng nền tảng tích hợp các công nghệ tiên tiến như xử lý ngôn ngữ tự nhiên, học máy và trí tuệ nhân tạo. Nền tảng này sẽ giúp doanh nghiệp tự động hóa việc thu thập, phân loại và truy xuất tri thức nội bộ một cách hiệu quả. Tôi đặc biệt nhấn mạnh việc phát triển các công cụ phân tích dữ liệu phi cấu trúc, cho phép khai thác giá trị từ các nguồn thông tin đa dạng như email, tài liệu nội bộ và báo cáo kỹ thuật.

Trong lĩnh vực giáo dục, tôi đề xuất phát triển các hệ thống hỗ trợ học tập thông minh tích hợp công nghệ truy xuất thông tin. Những hệ thống này sẽ có khả năng phân tích mức độ hiểu biết của người học, đề xuất tài liệu phù hợp, và tự động điều chỉnh nội dung học tập dựa trên tiến độ và khả năng tiếp thu. Tôi cũng chú trọng việc phát triển các công cụ đánh giá và theo dõi quá trình học tập, giúp giảng viên và người học có cái nhìn toàn diện về hiệu quả đào tạo.

Để đảm bảo tính khoa học và hiệu quả trong việc triển khai các ứng dụng trên, tôi đề xuất một số cải tiến quan trọng về mặt phương pháp luận. Đầu tiên là việc xây dựng khung đánh giá toàn diện, bao gồm các metrics định lượng và định tính, cho phép đo lường chính xác hiệu quả của hệ thống trong môi trường thực tế.

Tiếp đến là việc phát triển các kỹ thuật tiền xử lý dữ liệu tiên tiến, đặc biệt chú trọng đến việc xử lý thông tin đa ngôn ngữ và đa phương thức.

Hướng phát triển tiếp theo của nghiên cứu sẽ tập trung vào việc tích hợp các thành tựu mới nhất từ nhiều lĩnh vực khác nhau. Tôi đặc biệt quan tâm đến việc kết hợp trí tuệ nhân tạo với khoa học nhận thức, nhằm nâng cao khả năng hiểu và xử lý ngôn ngữ tự nhiên của hệ thống. Song song đó, tôi cũng chú trọng việc phát triển các giải pháp bảo mật và bảo vệ quyền riêng tư, đảm bảo an toàn thông tin trong quá trình xử lý và lưu trữ dữ liệu.

Với những định hướng nghiên cứu và kiến nghị mang tính đột phá nêu trên, tôi tin tưởng sâu sắc rằng công nghệ truy xuất thông tin thông minh sẽ không ngừng phát triển và hoàn thiện, tạo nên những bước tiến vượt bậc trong việc quản lý và khai thác tri thức số. Thông qua việc kết hợp hài hòa giữa nghiên cứu học thuật chuyên sâu và ứng dụng thực tiễn đa dạng, những đóng góp này sẽ là nền tảng vững chắc cho sự phát triển của hệ thống truy xuất thông tin thế hệ mới, đáp ứng được những thách thức ngày càng phức tạp của kỷ nguyên số. Đặc biệt, trong bối cảnh chuyển đổi số toàn diện và sự bùng nổ của dữ liệu lớn, những cải tiến này không chỉ góp phần thúc đẩy sự phát triển của công nghệ số mà còn mở ra những chân trời mới trong việc ứng dụng trí tuệ nhân tạo vào thực tiễn, hướng tới một tương lai nơi tri thức được quản lý và khai thác một cách hiệu quả, thông minh và bền vững.

TÀI LIỆU THAM KHẢO

- [1] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," presented at the Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 2020.
- [2] T. N. S. Đoàn Huỳnh Công Sơn, "THIẾT KẾ ROBOT TRỢ LÝ GIẢNG DẠY GIAO TIẾP BẰNG GIỌNG NÓI," *Tạp Chí Khoa Học Giáo Dục Kỹ Thuật - Trường Đại Học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh*, 2020. [Online]. Available: <https://www.scribd.com/document/723886250/No61-P7-Doan-Huynh-Cong-Son>.
- [3] P. Brandtzaeg and A. Følstad, *Why People Use Chatbots*. 2017.
- [4] R. Miikkulainen and M. G. Dyer, "Natural language processing with modular PDP networks and distributed lexicon," *Cognitive Science*, vol. 15, no. 3, pp. 343-399, 1991, doi: 10.1207/s15516709cog1503_2.
- [5] Ş. Gökçearslan, C. Tosun, and Z. Erdemir, "Benefits, Challenges, and Methods of Artificial Intelligence (AI) Chatbots in Education: A Systematic Literature Review," vol. 7, pp. 19-39, 02/04 2024, doi: 10.46328/ijte.600.
- [6] T. T. Nguyen, A. D. Le, H. T. Hoang, and T. Nguyen, "NEU-chatbot: Chatbot for admission of National Economics University," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100036, 2021/01/01/ 2021, doi: <https://doi.org/10.1016/j.caeai.2021.100036>.
- [7] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, *Challenges in Chatbot Development: A Study of Stack Overflow Posts*. 2020.
- [8] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.
- [9] R. Baeza-Yates *et al.*, "Modern Information Retrieval," 07/17 1999.
- [10] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, pp. 333-389, 01/01 2009, doi: 10.1561/1500000019.

- [11] V. Karpukhin *et al.*, *Dense Passage Retrieval for Open-Domain Question Answering*. 2020.
- [12] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," presented at the Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 2020. [Online]. Available: <https://doi.org/10.1145/3397271.3401075>.
- [13] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, "Evaluation of Retrieval-Augmented Generation: A Survey," *ArXiv*, vol. abs/2405.07437, 2024.
- [14] B. Eyal, M. Mahabi, O. Haroche, A. Bachar, and M. Elhadad, *Semantic Decomposition of Question and SQL for Text-to-SQL Parsing*. 2023, pp. 13629-13645.
- [15] F. Ma *et al.*, *Task Navigator: Decomposing Complex Tasks for Multimodal Large Language Models*. 2024, pp. 2248-2257.
- [16] H. Zhang, J. Cai, J. Xu, and J. Wang, "Complex Question Decomposition for Semantic Parsing," in *Annual Meeting of the Association for Computational Linguistics*, 2019.
- [17] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735-80, 12/01 1997, doi: 10.1162/neco.1997.9.8.1735.
- [18] A. Vaswani *et al.*, "Attention is all you need," presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017.
- [19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019.
- [20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," presented at the Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Montreal, Canada, 2014.

- [21] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1-140:67, 2019.
- [22] U. Chauhan and A. Shah, "Topic Modeling Using Latent Dirichlet allocation: A Survey," *ACM Computing Surveys*, vol. 54, pp. 1-35, 09/30 2022, doi: 10.1145/3462478.
- [23] X. Wu, T. Nguyen, D. Zhang, W. Wang, and A. Luu, *FASTopic: A Fast, Adaptive, Stable, and Transferable Topic Modeling Paradigm*. 2024.
- [24] E. Perez *et al.*, "Unsupervised Question Decomposition for Question Answering," 2020. [Online]. Available: <https://doi.org/10.18653/v1/2020.emnlp-main.713>.
- [25] K. Shridhar, A. Stolfo, and M. Sachan, "Distilling Reasoning Capabilities into Smaller Language Models," in *Annual Meeting of the Association for Computational Linguistics*, 2022.
- [26] T. Nguyen *et al.*, "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset," 11/28 2016, doi: 10.48550/arXiv.1611.09268.
- [27] G. Izacard *et al.*, "Atlas: few-shot learning with retrieval augmented language models," *J. Mach. Learn. Res.*, vol. 24, no. 1, p. Article 251, 2024.
- [28] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers," presented at the Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 2020.
- [29] G. Team *et al.*, *Gemma 2: Improving Open Language Models at a Practical Size*. 2024.
- [30] Z. J. a. F. F. X. a. L. G. a. Z. S. a. Q. L. a. J. D.-Y. a. Y. Y. a. J. C. a. G. Neubig, "Active Retrieval Augmented Generation," 2023. [Online]. Available: <https://aclanthology.org/2023.emnlp-main.495>.