RESEARCH ARTICLE

KHAI THÁC MÔ HÌNH NGÔN NGỮ LỚN ĐỂ CHUYỂN ĐỔI NGÔN NGỮ TỰ NHIÊN THÀNH TRUY VẤN CYPHER MỘT CÁCH HIỆU QUẢ

Đinh Minh Hòa, Trần Khải Thiện*

Khoa Công nghệ thông tin, Trường Đại học Ngoại ngữ -Tin học TP.HCM

hoadm@huflit.edu.vn, thientk@huflit.edu.vn

TÓM TẮT— Bài báo này nghiên cứu việc ứng dụng các mô hình ngôn ngữ lớn, cụ thể là GPT, trong tác vụ chuyển đổi ngôn ngữ tự nhiên thành truy vấn Cypher (Text-to-Cypher). Đây một thành phần quan trọng trong việc cải thiện hệ thống chatbot dựa trên cơ sở dữ liệu đồ thị. Chúng tôi phân tích các phương pháp nổi bật: zero-shot, few-shot và fine-tuning cùng với đề xuất một mô hình cải tiến của phương pháp few-shot. Sau cùng là đánh giá hiệu quả của chúng trong nhiệm vụ chuyển đổi đầu vào ngôn ngữ tự nhiên thành các truy vấn Cypher với độ chính xác và hiệu suất cao. Qua việc phân tích hiệu năng trong các kịch bản khác nhau, bài báo làm nổi bật sự đánh đổi giữa tính tổng quát, độ chính xác và yêu cầu tài nguyên. Kết quả nghiên cứu nhấn mạnh tầm quan trọng ngày càng tăng của các tác vụ Text-to-Cypher trong việc thúc đẩy công nghệ hội thoại do AI dẫn dắt.

Từ khóa— Mô hình ngôn ngữ lớn, ngôn ngữ truy vấn, đồ thị tri thức, cơ sở dữ liệu đồ thị, chatbot

I. GIỚI THIỆU

Chatbot đã trở thành công cụ không thể thiếu trong nhiều ngành công nghiệp, góp phần thay đổi cơ bản cách cung cấp dịch vụ và tiếp cận thông tin. Các ứng dụng của chatbot trải rộng trên nhiều lĩnh vực, đáp ứng những nhu cầu cấp thiết về hiệu quả, độ chính xác và khả năng tiếp cận. Trong lĩnh vực dịch vụ khách hàng [1], chatbot đóng vai trò then chốt khi cung cấp hỗ trợ tức thì, xử lý các câu hỏi thông thường và giải quyết vấn đề mà không cần sự can thiệp của con người. Điều này không chỉ rút ngắn thời gian phản hồi mà còn đảm bảo tính khả dụng liên tục 24/7, từ đó nâng cao sự hài lòng của người dùng. Tương tự, trong y tế [2], chatbot hỗ trợ bệnh nhân thông qua đánh giá ban đầu, đặt lịch hẹn, và nhắc nhở dùng thuốc, giảm tải cho nhân viên y tế đồng thời cải thiện sự gắn kết của bệnh nhân. Trong giáo dục [3], các tổ chức đang tận dụng chatbot để hỗ trợ quá trình học tập và quản lý hành chính. Từ việc trả lời câu hỏi của học sinh đến cung cấp các module học tập cá nhân hóa, chatbot thúc đẩy khả năng tiếp cận và tương tác trong giáo dục. Ngoài ra, trong lĩnh vực thương mại điện tử [4], chatbot hoạt động như các trợ lý mua sắm ảo, hướng dẫn khách hàng trong việc chọn sản phẩm, đưa ra các gợi ý và tối ưu hóa quy trình mua sắm. Việc sử dụng rộng rãi chatbot nhấn mạnh khả năng thích ứng của chúng với nhiều bối cảnh khác nhau, khiến chúng trở thành một phần không thể thiếu trong hệ sinh thái số hiện đại. Khi các doanh nghiệp và tổ chức nỗ lực đáp ứng kỳ vọng ngày càng cao về sự cá nhân hóa và hiệu quả trong cung cấp dịch vụ, chatbot sẽ tiếp tục đóng vai trò quan trọng trong thúc đẩy đổi mới và cải thiện trải nghiệm người dùng.

Trong bối cảnh các hệ thống chatbot hiện đại, việc tích hợp đồ thị tri thức [5] đã nổi lên như một phương pháp mang tính cách mạng nhằm nâng cao năng lực của chúng. Đồ thị tri thức cung cấp một cách biểu diễn thông tin có cấu trúc, cho phép chatbot diễn giải và suy luận [6] với dữ liệu một cách bối cảnh hóa và có ý nghĩa hơn. Bằng cách liên kết các thực thể, mối quan hệ và thuộc tính trong một mạng lưới giàu ngữ nghĩa, đồ thị tri thức giúp chatbot vượt qua các mô hình hỏi-đáp tĩnh, tạo điều kiện cho các tương tác động và theo ngữ cảnh. Khả năng tận dụng thông tin kết nối này đặc biệt quan trọng trong các kịch bản đòi hỏi chuyên môn cụ thể, chẳng hạn như y tế, giáo dục và dịch vụ pháp lý, nơi chatbot phải điều hướng qua các hệ thống dữ liệu phức tạp để cung cấp các phản hồi chính xác và hữu ích. Việc sử dụng cơ sở dữ liệu đồ thị để lưu trữ đồ thị tri thức càng nhấn mạnh vai trò quan trọng của chúng trong hệ thống chatbot. Các cơ sở dữ liệu đồ thị, chẳng hạn như Neo4j† hoặc ArangoDB‡, được thiết kế đặc biệt để quản lý và truy vấn dữ liệu kết nối quy mô lớn một cách hiệu quả. Khác với cơ sở dữ liệu quan hệ truyền thống, cơ sở dữ liệu đồ thị tận dụng cấu trúc đồ thị để lưu trữ và duyệt qua các mối quan hệ trực tiếp, từ đó cải thiện đáng kể tốc độ và độ chính xác của các truy vấn phức tạp. Điều này khiến chúng trở nên đặc biệt phù hợp với các ứng dụng chatbot, nơi mà việc truy hồi thông tin theo thời gian thực và khả năng mở rộng là rất cần thiết. Bằng cách sử dụng cơ sở dữ liệu đồ thị, các hệ thống chatbot có thể truy cập và khai thác dễ dàng các mối quan hệ phức tạp giữa các thực thể dữ liệu, hỗ trợ các phản hồi tinh vi và theo ngữ cảnh.

Để tận dụng tối đa sức mạnh của đồ thị tri thức, việc sinh ra các truy vấn Cypher để truy hồi dữ liệu đồ thị đã trở thành một thành phần quan trọng trong chức năng của chatbot. Cypher, một ngôn ngữ truy vấn khai báo dành cho cơ sở dữ liệu đồ thị, cho phép truy vấn dữ liệu có cấu trúc đồ thị một cách chính xác và linh hoạt. Việc sinh tự động các truy vấn Cypher giúp chatbot tương tác linh hoạt với đồ thị tri thức, chuyển hóa ý định người dùng

* Coressponding Author

† https://neo4j.com/

‡ https://arangodb.com/

thành các truy vấn cơ sở dữ liệu hiệu quả. Năng lực này không chỉ nâng cao khả năng phản hồi của chatbot mà còn đảm bảo tính mở rộng và khả năng thích ứng của hệ thống trong các lĩnh vực ứng dụng đa dạng. Khi nhu cầu về các hệ thống chatbot thông minh và theo ngữ cảnh tiếp tục tăng, sự phát triển các phương pháp sinh mã truy vấn Cypher mạnh mẽ sẽ đóng vai trò then chốt trong việc thúc đẩy lĩnh vực này và đáp ứng kỳ vọng ngày càng cao của người dùng.

Từ nhu cầu thực tiễn đó, trong bài báo này chúng tôi sẽ trình bày các phương pháp cơ bản khai thác Mô hình ngôn ngữ lớn (cụ thể là ChatGPT-4) để chuyển đổi ngôn ngữ tự nhiên thành truy vấn Cypher một cách hiệu quả. Bên cạnh đó, chúng tôi cũng đề xuất một phương pháp đơn giản nhưng hiệu quả nhằm nâng cao hiệu năng của kỹ thuật few-shot khi làm việc với ChatGPT-4. Cuối cùng, chúng tôi sẽ thực nghiệm tất cả các phương pháp này trên tập dữ liệu được chuẩn bị sẵn cho việc xây dựng Chatbot hỗ trợ thông tin tuyển sinh đại học. Kết quả nghiên cứu này sẽ cung cấp cho chúng ta cái nhìn sâu sắc hơn về ưu nhược điểm của từng phương pháp được đề cập.

Phần còn lại của bài báo được trình bày như sau: Mục II trình bày về các khái niệm và các công trình liên quan. Tiếp theo mục III là phương pháp được đề xuất bởi nhóm nghiên cứu. Mục IV trình bày quá trình thực nghiệm và các kết quả ghi nhận được. Cuối cùng là các mục V, VI, và VII lần lượt là phần kết luận, lời cảm ơn và các tài liệu tham khảo.

II. CÁC KHÁI NIỆM VÀ CÁC CÔNG TRÌNH LIÊN QUAN

A. MÔ HÌNH NGÔN NGỮ LỚN

Mô hình ngôn ngữ lớn (Large Language Model – LLM) là một bước đột phá quan trọng trong lĩnh vực trí tuệ nhân tạo, tận dụng các mạng nơ-ron tiên tiến để xử lý và tạo ra văn bản giống như con người. Những mô hình này được huấn luyện trên các tập dữ liệu khổng lồ, cho phép chúng học được sự phức tạp của ngôn ngữ, ngữ cảnh và ý nghĩa ở quy mô chưa từng có. LLM hoạt động dựa trên kiến trúc transformer [7], được Vaswani và cộng sự giới thiệu vào năm 2017, nổi bật nhờ khả năng hiểu và tạo ra các chuỗi văn bản thông qua các cơ chế như tự chú ý (self-attention) và mã hóa vị trí (positional encoding). Trong số các mô hình này, ChatGPT nổi bật như một triển khai chuyên biệt dành cho các bối cảnh hội thoại. Dựa trên nền tảng GPT (Generative Pre-trained Transformer), ChatGPT sử dụng phương pháp huấn luyện gồm hai giai đoạn: tiền huấn luyện và tinh chỉnh. Trong giai đoạn tiền huấn luyện, mô hình học các mẫu ngôn ngữ và tri thức từ các tập dữ liệu rộng lớn và đa dạng, bao gồm sách, bài báo và nội dung web. Giai đoạn tinh chỉnh, thường được hướng dẫn bởi phản hồi của con người, giúp điều chỉnh đầu ra của mô hình theo các mục tiêu cụ thể, đảm bảo tính liên quan, mạch lạc và tuân thủ các nguyên tắc đạo đức. ChatGPT có nhiều khả năng đa dạng, từ trả lời câu hỏi, soạn thảo nội dung, đến hỗ trợ giáo dục, dịch vụ khách hàng và hơn thế nữa. Mô hình này minh chứng cho cách LLM có thể ngữ cảnh hóa đầu vào, duy trì tính liên tục trong hội thoại và thích nghi với mục đích của người dùng. Tuy nhiên, vẫn còn những thách thức, chẳng hạn như giảm thiểu thiên kiến trong dữ liệu huấn luyện và đảm bảo độ tin cậy của thông tin được tạo ra. Sự phát triển của các LLM như ChatGPT nhấn mạnh tiềm năng chuyển đổi mạnh mẽ trong nhiều ngành công nghiệp, đồng thời nêu bật sự cần thiết của việc triển khai có trách nhiệm và liên tục cải tiến để tối đa hóa lợi ích xã hội mà chúng mang lại.

B. KỸ THUẬT TẠO GỢI Ý

Các mô hình ngôn ngữ lớn, tiêu biểu là ChatGPT, đã cách mạng hóa việc tạo gợi ý và phản hồi trong các hệ thống AI hội thoại. Những mô hình này, được huấn luyện trên lượng dữ liệu văn bản đa dạng khổng lồ, có khả năng tạo ra các gợi ý mạch lạc và phù hợp với ngữ cảnh. Quá trình tạo gợi ý với LLMs [8] dựa trên khả năng hiểu ngôn ngữ xác suất của chúng để dự đoán các phần hoàn chỉnh hoặc khuyến nghị phù hợp nhất dựa trên đầu vào của người dùng. Khả năng này được hỗ trợ bởi các kiến trúc tiên tiến như Transformer, cho phép mô hình nắm bắt các phụ thuộc ngữ cảnh sâu sắc trong văn bản. Trong bối cảnh ChatGPT, việc tạo gợi ý bắt đầu với kỹ thuật xây dựng prompt, nơi đầu vào ban đầu được thiết kế cẩn thận để tạo ra loại phản hồi mong muốn. Các prompt có thể được thiết kế bao gồm các hướng dẫn rõ ràng, ví dụ minh họa hoặc ngữ cảnh để định hướng đầu ra của mô hình đến các lĩnh vực hoặc nhu cầu cụ thể của người dùng. Chẳng hạn, cung cấp các prompt chi tiết về các tình huống y tế có thể giúp hướng dẫn mô hình tạo ra các khuyến nghị liên quan đến y tế với độ chính xác cao hơn. Ngoài ra, các kỹ thuật như zero-shot hoặc few-shot có thể nâng cao khả năng tạo gợi ý của mô hình trong các lĩnh vực mà nó thiếu dữ liệu huấn luyện sâu rộng.

C. TINH CHỈNH CÁC MÔ HÌNH NGÔN NGỮ LỚN

Tinh chỉnh các mô hình ngôn ngữ [8] là một bước quan trọng trong việc tùy chỉnh các hệ thống này cho các nhiệm vụ hoặc lĩnh vực cụ thể. Mặc dù các mô hình được huấn luyện sẵn như ChatGPT thể hiện khả năng tổng quát hóa vượt trội, hiệu suất của chúng có thể được cải thiện thông qua việc thích nghi cho các nhiệm vụ cụ thể. Quá trình tinh chỉnh cho phép điều chỉnh phản hồi của mô hình sao cho phù hợp với các yêu cầu người dùng, kiến thức đặc thù theo lĩnh vực, hoặc các ràng buộc vận hành, từ đó trở thành một phần không thể thiếu trong việc triển khai LLMs vào các ứng dụng thực tế. Tinh chỉnh bao gồm việc huấn luyện một mô hình được huấn

luyện sẵn trên các tập dữ liệu bổ sung được thiết kế phù hợp với ứng dụng mong muốn. Các tập dữ liệu này có thể bao gồm văn bản theo lĩnh vực, truy vấn do người dùng tạo ra, hoặc các ví dụ được chú thích, nhằm đảm bảo rằng mô hình phát triển sự hiểu biết sâu sắc hơn về các sắc thái cần thiết cho nhiệm vụ. Bằng cách sử dụng các kỹ thuật như tinh chỉnh có giám sát và học tăng cường, quá trình này cải thiện đáng kể độ chính xác, tính phù hợp và khả năng sử dụng của các đầu ra từ mô hình.

D. CƠ SỞ DỮ LIỆU ĐỒ THỊ VÀ NGÔN NGỮ TRUY VẤN CYPHER

Cơ sở dữ liệu đồ thị biểu diễn dữ liệu dưới dạng các nút (node), quan hệ (relationship), và thuộc tính (property), cho phép xử lý dữ liệu có tính liên kết cao một cách hiệu quả. Khác với cơ sở dữ liệu quan hệ truyền thống sử dụng bảng và hàng, cơ sở dữ liệu đồ thị mô hình hóa dữ liệu thành các thực thể (nút) được kết nối qua các cạnh (quan hệ). Cấu trúc này giúp truy vấn các mối quan hệ phức tạp như mạng xã hội, hệ thống gợi ý, và đồ thị tri thức trở nên hiệu quả.

Cypher [9] là ngôn ngữ truy vấn dạng khai báo, được thiết kế dành riêng cho cơ sở dữ liệu đồ thị, nổi bật là Neo4j. Cypher sử dụng cú pháp ASCII-art để biểu diễn các mẫu đồ thị, giúp các truy vấn trở nên trực quan. Hình 1 mô tả một ví dụ về mã Cypher.

```
MATCH (user:Person)-[:FRIEND]->(friend:Person)
WHERE user.name = 'Hòa Đinh'
RETURN friend.name;
```

Hình 1. *Mã Cypher tìm kiếm bạn của một người tên là "Hòa Đinh"*

Cách tiếp cận ngắn gọn và dễ đọc này đơn giản hóa việc làm việc với dữ liệu đồ thị, khiến Cypher trở thành lựa chọn phổ biến trong các ứng dụng dựa trên đồ thị.

E. CÁC CÔNG TRÌNH LIÊN QUAN

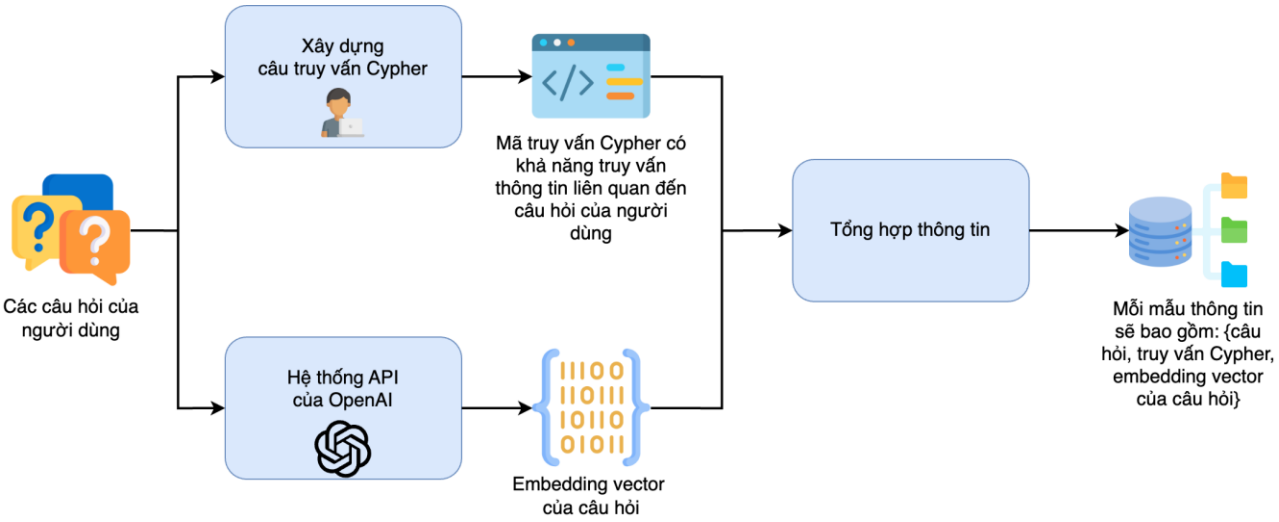
Tự động sinh truy vấn SQL, thường được gọi là Text-to-SQL hoặc Text-to-Cypher (trong ngữ cảnh cơ sở dữ liệu đồ thị), là một bài toán quan trọng ở giao điểm giữa xử lý ngôn ngữ tự nhiên (NLP) và hệ quản trị cơ sở dữ liệu. Nhiệm vụ này liên quan đến việc dịch các truy vấn ngôn ngữ tự nhiên thành các lệnh cơ sở dữ liệu có thể thực thi, giúp người dùng ít chuyên môn kỹ thuật truy cập và xử lý dữ liệu một cách dễ dàng. Nguồn gốc của vấn đề này có từ những năm 1980, khi các nhà nghiên cứu bắt đầu tìm hiểu các giao diện ngôn ngữ tự nhiên cho cơ sở dữ liệu [10] (NLIDBs). Tuy nhiên, các hệ thống ban đầu bị giới hạn bởi cách tiếp cận dựa trên luật [11] và năng lực ngôn ngữ hạn chế. Sự phát triển của học sâu và NLP từ giữa những năm 2010 đã thay đổi hoàn toàn lĩnh vực này, giới thiệu các mô hình mạnh mẽ hơn có khả năng nắm bắt mối quan hệ phức tạp giữa ngôn ngữ tự nhiên và dữ liệu có cấu trúc.

Những đột phá gần đây được thúc đẩy nhờ việc sử dụng các mô hình ngôn ngữ tiền huấn luyện như BERT [12], GPT và T5 [13], cùng với kiến trúc mạng thần kinh phù hợp cho dữ liệu cấu trúc, như mạng thần kinh đồ thị (GNNs) trong các bài toán Text-to-Cypher. Các tiêu chuẩn đánh giá như Spider [14] và các bộ dữ liệu dành riêng cho cơ sở dữ liệu đồ thị đã cung cấp khung đánh giá chuẩn hóa, thúc đẩy đổi mới. Các cách tiếp cận tiên tiến hiện nay tích hợp kỹ thuật zero-shot và few-shot [15] đã cải thiện đáng kể khả năng thích ứng của mô hình trên các lược đồ và ngôn ngữ cơ sở dữ liệu khác nhau. Dù đạt được nhiều thành tựu, bài toán này vẫn đối mặt với các thách thức như xử lý các truy vấn mơ hồ, đảm bảo khả năng tổng quát hóa giữa các miền, tối ưu hóa hiệu quả tính toán khi triển khai ở quy mô lớn và đặc biệt là sự phức tạp và thiếu thốn các tập dữ liệu huấn luyện trong tiếng Việt.

III. PHƯƠNG PHÁP ĐỀ XUẤT

Để cải thiện hiệu năng của mô-đun chuyển đổi ngôn ngữ tự nhiên thành truy vấn Cypher, chúng tôi đề xuất một phương pháp đơn giản nhưng hiệu quả, kết hợp giữa việc sử dụng vector nhúng (embedding vector) được tạo từ mô hình GPT và kỹ thuật few-shot. Quá trình chuyển đổi từ truy vấn ngôn ngữ tự nhiên của người dùng sang truy vấn Cypher được chia thành hai giai đoạn chính.

Giai đoạn 1 – Chuẩn bị dữ liệu. Giai đoạn này bao gồm ba bước cơ bản. Trước tiên, dữ liệu truy vấn từ hệ thống EduChat được lựa chọn kỹ lưỡng để đảm bảo mỗi chủ đề mà hệ thống hỗ trợ đều có ít nhất một đến hai câu hỏi người dùng. Sau đó, các chuyên gia của chúng tôi sẽ xây dựng câu truy vấn Cypher tương ứng với câu hỏi được chọn. Cuối cùng, ở bước thứ ba, các câu hỏi này được chuyển đổi thành vector nhúng thông qua API của OpenAI. Tổng quan quy trình được minh họa cụ thể trong hình 2.



Hình 2. *Quy trình chuẩn bị dữ liệu mẫu cho hệ thống*

Giai đoạn 2 – Chuyển đổi ngôn ngữ tự nhiên thành mã truy vấn Cypher. Khi người dùng gửi một truy vấn mới, hệ thống sẽ sử dụng API của OpenAI để tạo ra vector nhúng tương ứng với câu truy vấn đó. Sau đó, hệ thống áp dụng độ tương tự cosine (cosine similarity) để xác định các truy vấn tương đồng nhất. Những truy vấn này sẽ được sử dụng làm ví dụ trong kỹ thuật few-shot. Độ tương tự cosine được tính theo công thức (1).

$$Cosine\ Similaririty = \cos(\theta) = \frac{A \cdot B}{\| A \| \| B \|} \qquad (1)$$

Trong đó:

- A và B là 2 vector cần so sánh
- $A \cdot B$ là tích vô hướng (dot product) của 2 vector
- $\| A \|$ và $\| B \|$ lần lượt là độ dài (norm) của 2 vector A và B

IV. THỰC NGHIỆM

A. DỮ LIỆU THỰC NGHIỆM

Trong nghiên cứu này, chúng tôi sử dụng dữ liệu được chuẩn bị để xây dựng hệ thống EduChat – Chatbot hỗ trợ thông tin tuyển sinh đại học. Một số thông tin chi tiết về dữ liệu sẽ được trình bày bên dưới.

1. TẬP DỮ LIỆU EDUCHAT

Tập dữ liệu EduChat được xây dựng khi chúng tôi hiện thực ứng dụng EduChat [3]. Các bước xây dựng tập dữ liệu này như sau:

- *Bước 1*: Khảo sát người dùng về các câu hỏi liên quan đến chủ đề tuyển sinh.
- *Bước 2*: Thu thập, tổng hợp và tiền xử lý các thông tin liên quan đến chủ đề tuyển sinh cũng như các thông tin về Trường Đại học Ngoại ngữ - Tin học TP.HCM (HUFLIT). Nguồn dữ liệu bao gồm các thông báo và văn bản nội bộ của trường, các thông tin chính thức trên website HUFLIT[§], cũng như các tài liệu quảng bá tư vấn tuyển sinh và cẩm nang tuyển sinh năm 2024.
- *Bước 3*: Các chuyên gia sẽ tìm kiếm thông tin thu thập được ở bước 2 để kiểm tra xem chúng có thỏa mãn các câu hỏi đã thu thập ở bước 1 không. Với các câu hỏi không đủ thông tin trả lời, chúng tôi tiến hành tham vấn các chuyên gia tại trường để thu thập thêm thông tin.

Nội dung chính của tập dữ liệu bao gồm thông tin về trường, các khoa, ngành học, chuyên ngành, điểm chuẩn, hoạt động, và nhiều thông tin bổ trợ khác giúp sinh viên định hướng chọn ngành học phù hợp. Tập dữ liệu EduChat được xây dựng với hai mục tiêu quan trọng. Mục tiêu đầu tiên là cung cấp nền tảng thông tin mở rộng cho hệ thống EduChat, trong khi mục tiêu thứ hai là làm cơ sở để kiểm thử và đánh giá các mô-đun của hệ thống Chatbot trong quá trình phát triển. Phần lớn dữ liệu đã được tổ chức dưới dạng đồ thị tri thức và lưu trữ trong hệ quản trị cơ sở dữ liệu đồ thị Neo4j. Chi tiết về đồ thị tri thức này sẽ được trình bày cụ thể trong các phần bên dưới.

2. ĐỒ THỊ TRI THỨC

[§] https://huflit.edu.vn/

Như đã trình bày, dữ liệu sau khi thu thập và tiền xử lý sẽ được mô hình hóa thành đồ thị tri thức và lưu trữ trên Hệ quản trị cơ sở dữ liệu đồ thị Neo4j. Sơ đồ cơ sở dữ liệu (schema) đồ thị tri thức của hệ thống EduChat được minh họa chi tiết trong hình 3.

Hình 3. *Đồ thị tri thức của ứng dụng EduChat*

Đồ thị tri thức của chúng tôi bao gồm 16 loại nút, 32 loại quan hệ, dữ liệu liên quan đến 20 ngành học đang được đào tạo tại HUFLIT và một số thông tin liên quan được cập nhật vào thời điểm tuyển sinh năm 2024.

3. DỮ LIỆU TRUY VẤN

Các câu hỏi trong quá trình khảo sát và chạy thử nghiệm được chúng tôi tiến hành phân tích, đánh giá và xây dựng các câu truy vấn Cypher một cách thủ công. Các mã Cypher này được sử dụng huấn luyện cũng như đánh giá hiệu năng của các phương pháp mới trong quá trình nâng cấp. Bảng 1 mô tả một vài câu hỏi và mã truy vấn Cypher liên quan đến ngành Công nghệ thông tin.

Bảng 1. *Một số ví dụ minh họa về câu hỏi và mã truy vấn Cypher tương ứng*

STT	Câu hỏi	Truy vấn Cypher
1	Ngành Công nghệ thông tin là gì	MATCH (m: Major) WHERE lower(m.name) = 'công nghệ thông tin' RETURN m, COLLECT { MATCH (m)-[:HAS_SPECIALIZATION]->(s) RETURN s.name } as specialization, COLLECT { MATCH (m)-[:INTRODUCES]->(i) RETURN i.content } as introduction
2	HUFLIT có đào tạo ngành Công nghệ thông tin không	MATCH (m: Major) return m.name
3	Cơ hội việc làm ngành công nghệ thông tin	MATCH (m: Major)-[f: FITS]-(e: Employment) WHERE lower(m.name) = 'công nghệ thông tin' RETURN e.name
4	Thế mạnh đào tạo của ngành Công nghệ thông tin là gì	MATCH (m: Major)-[a: ADVANTAGES]-(i: Information) WHERE lower(m.name) = 'công nghệ thông tin' RETURN m.name, i.content
5	Những tố chất phù hợp với ngành Công nghệ thông tin	MATCH (m: Major)-[r: REQUIRES_QUALITY]->(q) WHERE lower(m.name) = 'công nghệ thông tin' RETURN m.name, r, q.quality
6	Tổ hợp xét tuyển ngành Công nghệ thông tin	MATCH (m: Major)-[a: ACCEPTS]-(ac: Admission_Combination) WHERE lower(m.name) = 'công nghệ thông tin' return m.name, a, ac.name

7	Ngành Công nghệ thông tin có những chuyên ngành nào	MATCH (m: Major)-[r:HAS_SPECIALIZATION]-(s) WHERE lower(m.name) = 'công nghệ thông tin' RETURN m.name, r, s.name
8	Điểm chuẩn ngành Công nghệ thông tin năm 2023 là bao nhiêu	MATCH (m: Major)-[r:REQUIRES_2023]->(am: Admission_Method) WHERE lower(m.name) = 'công nghệ thông tin' RETURN TYPE(r), r.value, am.name
9	Chương trình đào tạo ngành Công nghệ thông tin.	MATCH (m: Major) WHERE lower(m.name) = 'công nghệ thông tin' RETURN m, COLLECT {MATCH (m)-[c: CONTAINS]-(p: Training_Program) RETURN apoc.map.fromPairs(collect([p.name, c.value])) AS result}
10	Thời gian đào tạo ngành Công nghệ thông tin là bao lâu?	MATCH (m: Major) WHERE lower(m.name) = 'công nghệ thông tin' RETURN m.name, m.credits, m.study_duration

B. CÁC TIÊU CHÍ ĐÁNH GIÁ

Đánh giá các mô hình chuyển đổi ngôn ngữ tự nhiên thành truy vấn Cypher (text-to-cypher) đặt ra nhiều thách thức đáng kể. Thứ nhất, sự đa dạng trong cách diễn đạt ngôn ngữ tự nhiên đòi hỏi mô hình phải hiểu và xử lý ngữ cảnh chính xác. Thứ hai, truy vấn Cypher tuy đúng cú pháp nhưng có thể sai logic, gây khó khăn trong việc kiểm chứng kết quả. Thứ ba, các câu hỏi phức tạp liên quan đến quan hệ trong đồ thị đòi hỏi mô hình phải suy luận sâu. Đối mặt với những khó khăn đó và để làm rõ hiệu năng của mô hình chúng tôi lựa chọn bốn độ đo là BLEU [16], Valid SQL (VA), execution accuracy (EX) và Latency [17].

BLEU (Bilingual Evaluation Understudy) là một phương pháp tiêu chuẩn để đánh giá chất lượng văn bản do máy sinh ra, đặc biệt phù hợp với bài toán Text-to-Cypher, nơi cần chuyển đổi ngôn ngữ tự nhiên thành câu lệnh Cypher. BLEU đo lường định lượng mức độ tương đồng giữa đầu ra của mô hình và một tập hợp câu lệnh Cypher tham chiếu bằng cách đánh giá sự trùng lặp n-gram. Độ chính xác đã được điều chỉnh (modified precision) được sử dụng để hạn chế việc lặp lại quá mức, giới hạn số lần xuất hiện tối đa của một n-gram theo tần suất của nó trong các câu lệnh tham chiếu. Để xử lý vấn đề đầu ra quá ngắn, một hình phạt ngắn (brevity penalty - BP) được áp dụng, giảm điểm số nếu văn bản đầu ra ngắn hơn đáng kể so với tham chiếu. BLEU được tính toán theo công thức (2).

$$BLEU = BP \times \exp(\sum_{n=1}^{N} w_n \log p_n) \qquad (2)$$

Trong đó p_n là độ chính xác của $n-grams$ ở cấp độ n, w_n là trọng số gán cho mỗi cấp độ $n-grams$ (thường phân bố đều), và N là thứ tự $n-grams$ cao nhất được sử dụng. BP được tính bằng công thức (3).

$$BP = \begin{cases} 1; \ Nếu \ c > r \\ e^{(1-\frac{r}{c})}; \ Nếu \ c \leq r \end{cases} \qquad (2')$$

với c là độ dài của câu lệnh Cypher sinh ra và r là độ dài tham chiếu.

Bên cạnh đó, Valid SQL là một độ đo phổ biến trong các bài toán sinh câu lệnh SQL tự động, chẳng hạn như Text-to-Cypher hoặc các bài toán tương tự, để đánh giá tính hợp lệ cú pháp của câu lệnh SQL hoặc Cypher được sinh ra. VA tập trung vào việc kiểm tra xem các câu lệnh do mô hình sinh ra có được trình phân tích cú pháp (parser) chấp nhận hay không, tức là chúng có tuân thủ quy tắc cú pháp của ngôn ngữ truy vấn hay không. VA được tính bằng công thức (3).

$$VA = \frac{Số \ truy \ vấn \ hợp \ lệ}{Tổng \ số \ truy \ vấn} \times 100 \qquad (3)$$

Với một câu lệnh được coi là hợp lệ nếu nó không gây ra lỗi cú pháp khi gửi đến cơ sở dữ liệu hoặc hệ thống xử lý.

Trong khi VA chỉ đo tỉ lệ số truy vấn hợp lệ thì Execution Accuracy (EX) kiểm tra xem câu lệnh sinh ra có trả về kết quả đúng khi được thực thi trên cơ sở dữ liệu hay không. Một cách rõ ràng hơn thì EX tập trung kiểm tra tính đúng đắn ngữ nghĩa và khả năng phản ánh chính xác yêu cầu từ đầu vào. EX được tính bằng công thức (4).

$$EX = \frac{Số \ câu \ truy \ vấn \ trả \ về \ kết \ quả \ đúng}{Tổng \ số \ truy \ vấn} \times 100 \qquad (4)$$

Và cuối cùng, Latency phản ánh thời gian mà hệ thống cần để chuyển đổi một đầu vào ngôn ngữ tự nhiên thành câu lệnh truy vấn. Trong nghiên cứu này, chúng tôi sẽ tính trung bình cộng thời gian của tất cả truy vấn trong từng phương pháp để hỗ trợ làm rõ về ưu nhược điểm của từng phương pháp.

C. TIẾN HÀNH THỰC NGHIỆM

Về dữ liệu, chúng tôi rút trích 250 câu hỏi trong tập dữ liệu EduChat, bao gồm các thành phần:

- Câu hỏi của người dùng
- Mã Cypher để truy vấn các thông tin liên quan đến câu hỏi của người dùng (tương tự bảng 1)
- Các thông tin liên quan đến câu hỏi của người dùng

Dữ liệu này được chia thành hai phần:

- Phần thứ nhất gồm 200 câu hỏi[**] để đánh giá khả năng sinh mã của các phương pháp.
- Phần thứ hai gồm 50 câu hỏi mở rộng (các ngành chưa có tại HUFLIT, thông tin của các năm học trong tương lai) nhằm đánh giá khả năng mở rộng của các phương pháp.

Về mô hình, chúng tôi sử dụng các mô hình tiên tiến của OpenAI trong thời điểm hiện tại:

- Model ChatGPT-4o cho tác vụ sinh mã Cypher.
- Model text-embedding-3-small cho tác vụ tạo embedding vector.

Dữ liệu sẽ được thực nghiệm trên 4 phương pháp cơ bản:

- Zero-shot
- Few-shot
- Tinh chỉnh model ChatGPT-4o với dữ liệu liên quan ứng dụng EduChat
- Mô hình của chúng tôi đề xuất

Chúng tôi sẽ trình bày kết quả thực và giải thích chi tiết trong phần kế tiếp.

D. KẾT QUẢ THỰC NGHIỆM

Kết quả thực nghiệm được trình bày trong bảng 2, thông qua những số liệu này, chúng tôi đưa ra một số nhận xét sau về bài toán chuyển đổi ngôn ngữ tự nhiên thành mã truy vấn Cypher. Thứ nhất, các mô hình ngôn ngữ lớn đặc biệt là ChatGPT-4 trong thời điểm hiện nay đã đạt được nhiều thành tựu đáng kể, với phương pháp zero-shot, chỉ với một vài chỉ dẫn cụ thể, ChatGPT có thể sinh ra mã Cypher một cách đáng kinh ngạc với khoảng 78% mã hợp lệ và trên 50% cho kết quả chính xác với yêu cầu của người dùng. Tiếp theo, khi ChatGPT được cung cấp nhiều thông tin hơn trong bài toán cụ thể thông qua phương thức Few-shot hoặc tinh chỉnh (fine-tuning) thì hiệu năng của mô hình được cải thiện đáng kể (từ 8-10%) trên độ đo VA và EX. Cuối cùng, tận dụng khả năng tạo các vector nhúng của mô hình embedding, chúng tôi đề xuất phương pháp xây dựng thư viện các câu truy vấn mẫu, sử dụng độ đo tương tự cosine để tìm kiếm các mẫu gần với truy vấn mới của người dùng, các mẫu này được sử dụng để hướng dẫn ChatGPT trong quá trình sinh mẫu mới. Phương pháp của chúng tôi cho kết quả vượt trội so với các phương pháp tiếp cận truyền thống.

Mặc dù cho kết quả tương đối tốt về độ chính xác, nhưng phương pháp của chúng tôi cũng còn hạn chế về mặt thời gian thực thi. Trong phương pháp của chúng tôi, mô hình sẽ cần thêm 1 bước trong việc chuyển đổi câu truy vấn mới của người dùng thành vector nhúng, sử dụng thuật toán tính độ đo tương tự cosine để tìm kiếm các truy vấn tương đương để xây dựng các ví dụ trong kỹ thuật few-shot. Điều này dẫn đến việc phương pháp của chúng tôi tốn gấp 3 lần thời gian để chuyển đổi một yêu cầu của người dùng thành mã truy vấn Cypher.

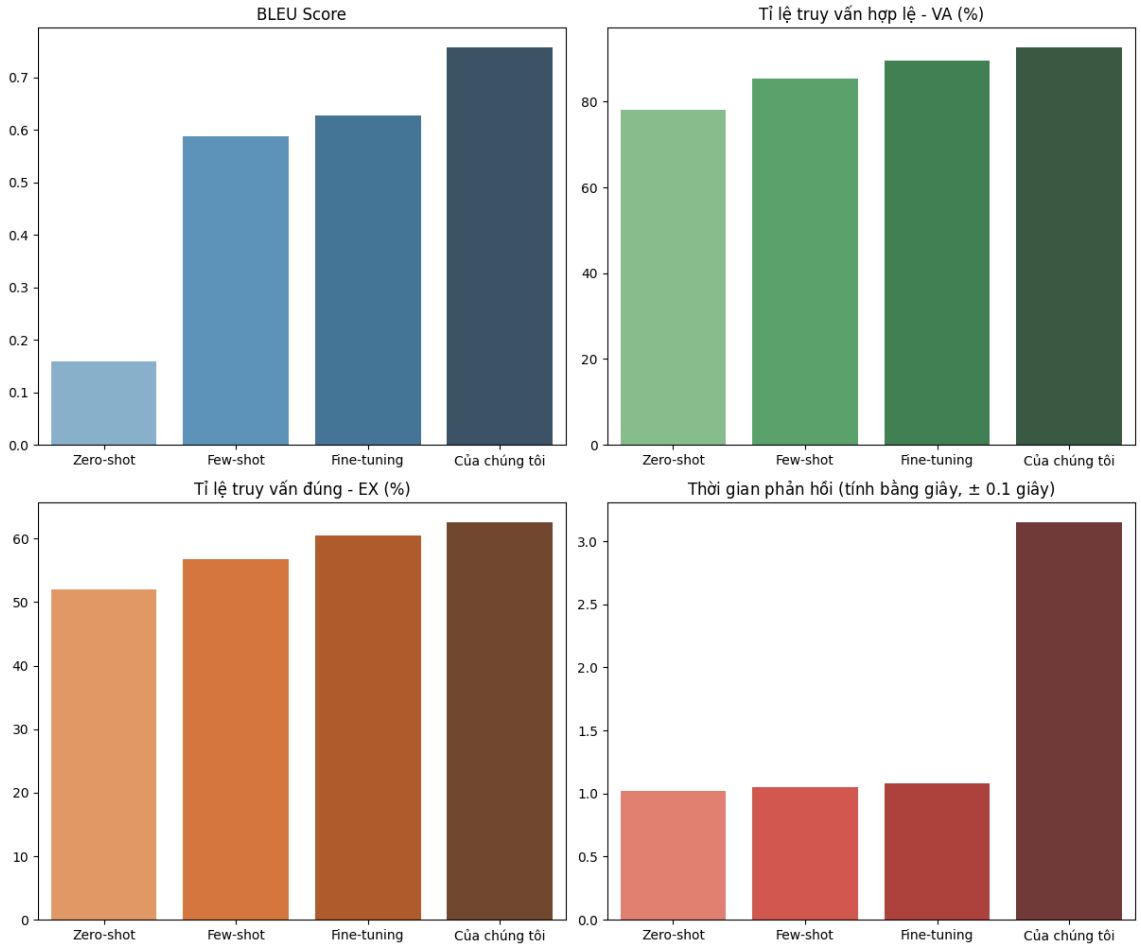
Bảng 2. *Kết quả thực nghiệm các phương pháp chuyển đổi ngôn ngữ tự nhiên thành mã Cypher*

Phương pháp	**BLEU**	**VA**	**EX**	**Thời gian phản hồi (tính bằng giây, ± 0,1 giây)**
Zero-shot	0,1588	78,12%	52,04%	1,02
Few-shot	0,5884	85,31%	56,81%	1,05
Fine-tuning	0,6268	89,52%	60,52%	1,08
Của chúng tôi đề xuất	0,7573	92,68%	62,56%	3,15

[**] https://tinyurl.com/Chatbot-Questions

Để làm rõ hơn về hiệu năng của các phương pháp, chúng tôi thực hiện trực quan hóa các số liệu trong hình 4, thông qua biểu đồ này có thể thấy, mô hình của chúng tôi đề xuất cho kết quả cao nhất trên 3 độ đo là BLEU, VA và EX, nhưng cũng có thể dễ dàng nhận thấy, để đạt được kết quả tốt hơn phương pháp của chúng tôi phải đánh đổi rất nhiều về thời gian thực hiện.

Bên cạnh đó, nhóm nghiên cứu cũng thử nghiệm trên một tập dữ liệu nhỏ để khảo sát khả năng mở rộng của mô hình trong tương lai (ví dụ hỏi về điểm chuẩn năm 2025). Nhờ khả năng linh hoạt của phương pháp few-shot và phương pháp chúng tôi đề xuất (tăng cường phương pháp few-shot), hệ thống có thể dễ dàng sinh mã Cypher linh hoạt theo thiết kế của hệ thống, trong khi zero-shot và fine-tuning gặp nhiều khó khăn hơn.



Hình 4. *Trực quan hóa kết quả thực nghiệm các phương pháp chuyển đổi ngôn ngữ tự nhiên thành mã Cypher*

V. KẾT LUẬN

Nghiên cứu của chúng tôi đã chứng minh tiềm năng chuyển đổi của các mô hình ngôn ngữ lớn như GPT trong việc chuyển đổi ngôn ngữ tự nhiên thành truy vấn Cypher. Thông qua phân tích so sánh các phương pháp zero-shot, few-shot, fine-tuning và một phương pháp mới kết hợp embedding vector cùng độ đo tương tự cosine, hiệu quả và độ chính xác đã được cải thiện đáng kể. Dù mô hình đề xuất vượt trội về độ chính xác truy vấn, chi phí tính toán, đặc biệt là độ trễ, vẫn là một hạn chế cần cân nhắc.

Trong tương lai, chúng tôi dự định sẽ tập trung tối ưu hóa phương pháp đề xuất để giảm độ trễ trong khi duy trì độ chính xác, cho phép ứng dụng thời gian thực trên các hệ thống quy mô lớn. Bên cạnh đó là tận dụng các tiến bộ trong học tăng cường có thể cải thiện khả năng xử lý các truy vấn phức tạp và mơ hồ của hệ thống.

VI. LỜI CẢM ƠN

Nghiên cứu được tài trở bởi Trường Đại học Ngoại ngữ - Tin học Thành phố Hồ Chí Minh trong khuôn khổ Đề tài mã số H2023-11.

VII. TÀI LIỆU THAM KHẢO

[1] E. W. T. Ngai, M. C. M. Lee, M. Luo, P. S. L. Chan, and T. Liang, "An intelligent knowledge-based chatbot for customer service," *Electron. Commer. Res. Appl.*, vol. 50, p. 101098, 2021, doi: https://doi.org/10.1016/j.elerap.2021.101098.

[2] K. Singhal *et al.*, "Towards Expert-Level Medical Question Answering with Large Language Models," May 16, 2023, *arXiv*: arXiv:2305.09617. Accessed: Sep. 11, 2024. [Online]. Available: http://arxiv.org/abs/2305.09617

[3] H. Dinh and T. K. Tran, "EduChat: An AI-Based Chatbot for University-Related Information Using a Hybrid Approach," *Appl. Sci.*, vol. 13, no. 22, p. 12446, Nov. 2023, doi: 10.3390/app132212446.

[4] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, "SuperAgent: A Customer Service Chatbot for E-commerce Websites," in *Proceedings of ACL 2017, System Demonstrations*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 97–102. doi: 10.18653/v1/P17-4017.

[5] S. Varitimiadis, K. Kotis, D. Pittou, and G. Konstantakis, "Graph-Based Conversational AI: Towards a Distributed and Collaborative Multi-Chatbot Approach for Museums," *Appl. Sci.*, vol. 11, no. 19, 2021, doi: 10.3390/app11199160.

[6] A. Saxena, A. Tripathi, and P. Talukdar, "Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 4498–4507. doi: 10.18653/v1/2020.acl-main.412.

[7] A. Vaswani *et al.*, "Attention is All you Need".

[8] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing," *ACM Comput Surv*, vol. 55, no. 9, Jan. 2023, doi: 10.1145/3560815.

[9] N. Francis *et al.*, "Cypher: An Evolving Query Language for Property Graphs," in *Proceedings of the 2018 International Conference on Management of Data*, in SIGMOD '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1433–1445. doi: 10.1145/3183713.3190657.

[10] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, "Natural Language Interfaces to Databases - An Introduction." 1995. [Online]. Available: https://arxiv.org/abs/cmp-lg/9503016

[11] K. Lin, B. Bogin, M. Neumann, J. Berant, and M. Gardner, "Grammar-based Neural Text-to-SQL Generation." 2019. [Online]. Available: https://arxiv.org/abs/1905.13326

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/V1/N19-1423.

[13] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J Mach Learn Res*, vol. 21, no. 1, Jan. 2020.

[14] T. Yu *et al.*, "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 3911–3921. doi: 10.18653/v1/D18-1425.

[15] M. Hornsteiner, M. Kreussel, C. Steindl, F. Ebner, P. Empl, and S. Schönig, "Real-Time Text-to-Cypher Query Generation with Large Language Models for Graph Databases," *Future Internet*, vol. 16, no. 12, p. 438, Nov. 2024, doi: 10.3390/fi16120438.

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Philadelphia, Pennsylvania: Association for Computational Linguistics, 2001, p. 311. doi: 10.3115/1073083.1073135.

[17] A. Liu, X. Hu, L. Wen, and P. S. Yu, "A comprehensive evaluation of ChatGPT's zero-shot Text-to-SQL capability," Mar. 12, 2023, *arXiv*: arXiv:2303.13547. doi: 10.48550/arXiv.2303.13547.

LEVERAGING LARGE LANGUAGE MODELS FOR EFFICIENT TRANSFORMATION OF NATURAL LANGUAGE INTO CYPHER QUERIES

Dinh Minh Hoa, Tran Khai Thien

ABSTRACT— This paper investigates the application of large language models, specifically GPT, for transforming natural language into Cypher queries (Text-to-Cypher). This is a critical component in enhancing chatbot systems based on graph databases. We examine prominent methods, including zero-shot, few-shot, and fine-tuning approaches, and propose an enhanced model for the few-shot method. Finally, we assess their effectiveness in converting natural language inputs into Cypher queries with high accuracy and efficiency. Through an analysis of performance across various scenarios, the paper highlights the trade-offs among generalization, accuracy, and resource requirements. The findings underscore the increasing significance of Text-to-Cypher tasks in advancing AI-driven conversational technologies.

Keywords — Large language model, cypher language, knowledge graph, graph database, chatbot.

TS. Trần Khải Thiện nhận học vị tiến sĩ ngành Khoa học máy tính tại trường Đại học Bách Khoa, ĐHQG-TP.HCM năm 2022. Hiện ông là giảng viên tại Khoa Công nghệ thông tin Trường Đại học Ngoại ngữ - Tin học Tp. Hồ Chí Minh (HUFLIT) và là Trưởng nhóm nghiên cứu IDPS của trường. TS. Thiện là bình duyệt viên và là tác giả của trên 20 công bố trong các tạp chí SCIE và các hội nghị quốc tế uy tín.

Hướng nghiên cứu chính của ông là Xử lý ngôn ngữ tự nhiên, Trí tuệ nhân tạo.

ThS. Đinh Minh Hòa là thạc sĩ ngành Công nghệ Thông tin tại Trường Đại học Ngoại ngữ - Tin học Tp. Hồ Chí Minh (HUFLIT) vào năm 2022. Hiện tại ông là giảng viên tại Khoa Công nghệ thông tin-HUFLIT và thành viên của nhóm nghiên cứu IDPS của trường.

Hướng nghiên cứu chính của ông là Xử lý ngôn ngữ tự nhiên, Trí tuệ nhân tạo.