

ỨNG DỤNG ĐỒ THỊ TRI THỨC CHO TÁI XẾP HẠNG CÁC ĐOẠN VĂN BẢN TRONG TRUY XUẤT TĂNG CƯỜNG TẠO SINH

APPLICATION OF KNOWLEDGE GRAPHS TO RERANKING TEXT CHUNKS IN RETRIEVAL-AUGMENTED GENERATION

Trần Quốc Hưng

Khoa Điện tử và Kỹ thuật máy tính, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

Đến Tòa soạn ngày 25/12/2025, chấp nhận đăng ngày 06/3/2026

Tóm tắt: Các hệ thống truy xuất tăng cường tạo sinh (RAG) hiện đại thường dựa vào sự tương đồng ngữ nghĩa để truy xuất ngữ cảnh dùng làm đầu vào cho các mô hình ngôn ngữ lớn. Tuy nhiên, phương pháp này trả về các đoạn văn bản thường thiếu liên quan về mặt dữ kiện, dẫn đến việc các mô hình ngôn ngữ lớn sinh ra các câu trả lời thiếu căn cứ. Để giải quyết vấn đề này, chúng tôi đề xuất một phương pháp tái xếp hạng sau truy xuất bằng cách tận dụng tri thức có cấu trúc được trích xuất tự động từ chính kho văn bản của miền dữ liệu. Phương pháp của chúng tôi đánh giá lại các tài liệu được truy xuất ban đầu bằng cách tính toán một điểm số tương quan. Bằng cách ưu tiên các tài liệu thể hiện sự liên kết thông tin chặt chẽ với câu hỏi, chúng tôi cung cấp cho mô hình một ngữ cảnh đáng tin cậy hơn. Thử nghiệm trên bộ dữ liệu rag-mini-bioasq cho thấy phương pháp của chúng tôi đã cải thiện chỉ số truy xuất so với hệ thống RAG cơ sở.

Từ khóa: RAG, đồ thị tri thức, tái xếp hạng.

Abstract: Modern retrieval-augmented generation (RAG) systems typically rely on semantic similarity to retrieve context for large language models (LLMs). However, this approach often returns passages that lack factual relevance, leading LLMs to produce unsupported answers. To address this issue, we propose a post-retrieval reranking method that leverages structured knowledge automatically extracted from the domain's own corpus. Our method reassesses the initially retrieved documents by computing a correlation-based relevance score. By prioritizing documents that exhibit tighter informational linkage to the query, we provide the model with more reliable context. Experiments on the rag-mini-bioasq dataset show that our approach improves retrieval metrics over baseline RAG systems.

Keywords: RAG, Knowledge Graph, Rerank.

1. GIỚI THIỆU

Trong những năm gần đây, các mô hình Ngôn ngữ lớn (LLM) như là Llama, GPT-4, DeepSeek đã tạo ra một cuộc cách mạng trong lĩnh vực Xử lý ngôn ngữ tự nhiên [1, 2, 3]. Các mô hình này thể hiện khả năng tạo sinh văn bản, tóm tắt, dịch thuật và trả lời câu hỏi một cách ấn tượng. Tuy nhiên, sức mạnh của chúng chủ yếu đến từ các tri thức được tham số

hóa và lưu trữ hoàn toàn trong hàng tỷ trọng số của mạng nơ-ron sau quá trình huấn luyện trên một kho dữ liệu văn bản khổng lồ [4].

Kiến thức tham số hóa này vốn có những hạn chế cố hữu. Thứ nhất, khi một mô hình đã được huấn luyện, kiến thức của nó bị cố định và không thể tự động cập nhật với các sự kiện mới nhất hoặc thông tin thời gian thực. Thứ hai, các LLM không thể truy cập vào các

nguồn kiến thức riêng tư, độc quyền hoặc chuyên ngành mà không được đưa vào trong bộ dữ liệu huấn luyện ban đầu. Những hạn chế này dẫn đến một vấn đề nghiêm trọng được gọi là ảo giác, hiện tượng LLM tạo ra các câu trả lời nghe có vẻ hợp lý và tự tin nhưng lại hoàn toàn sai lệch về mặt thực tế [5]. Đây là một vấn đề quan trọng làm ảnh hưởng đến độ tin cậy của các LLM, đặc biệt trong các lĩnh vực yêu cầu độ chính xác cao như y tế, tài chính và pháp lý.

Để giải quyết những thách thức này, kiến trúc Truy xuất - Tăng cường - Tạo sinh (RAG) đã nổi lên như một giải pháp hàng đầu. RAG thay đổi mô hình hoạt động của LLM bằng cách giới thiệu một thành phần kho kiến thức bên ngoài có thể truy cập được [6]. Thay vì chỉ dựa vào kiến thức nội tại, hệ thống RAG trước tiên thực hiện một bước truy xuất: khi nhận được một truy vấn từ người dùng, nó sẽ tìm kiếm trong một cơ sở kiến thức để lấy ra những thông tin liên quan nhất. Sau đó, những thông tin được truy xuất này sẽ được đưa vào ngữ cảnh của câu lệnh cùng với truy vấn ban đầu. Bằng cách này, LLM được cung cấp ngữ cảnh thực tế để tạo ra một câu trả lời dựa vào các nguồn đáng tin cậy, qua đó giảm thiểu đáng kể hiện tượng ảo giác. Phương pháp này cũng cho phép người dùng tùy chỉnh LLM với dữ liệu độc quyền của họ mà không cần quá trình tái huấn luyện tốn kém.

Kỹ thuật RAG đã giúp cải thiện chất lượng của LLM trong nhiều tác vụ [7]. Tuy nhiên, RAG cũng gặp rất nhiều thách thức như nhiễu trong quá trình truy xuất, sự mơ hồ về ngữ nghĩa hay hiện tượng Lost in the middle [8] khi ta tăng cường câu lệnh với quá nhiều tri thức. Do đó, việc thiết kế một phương pháp cải thiện sau bước truy xuất cho hệ thống RAG là một điều thiết yếu.

Để vượt qua những hạn chế của RAG, ta cần

xây dựng một quá trình truy xuất có khả năng tái xếp hạng các văn bản được truy xuất ban đầu. Tái xếp hạng là một kỹ thuật nền tảng trong các hệ thống truy xuất thông tin, được thiết kế như một quy trình xử lý hai giai đoạn, đánh đổi giữa độ chính xác và độ trễ. Giai đoạn truy xuất ban đầu, sử dụng một phương pháp nhanh để lọc ra một tập hợp lớn các ứng viên tiềm năng, tối ưu hóa cho độ phủ. Giai đoạn thứ hai, sắp xếp lại, áp dụng một mô hình phức tạp và tốn kém hơn về mặt tính toán trên tập hợp ứng viên nhỏ để tính toán lại điểm liên quan một cách chính xác hơn. Kiến trúc này cho phép hệ thống đạt được sự cân bằng tối ưu giữa chất lượng kết quả và hiệu suất hoạt động. Các phương pháp sắp xếp lại hiện đại thường dựa trên các mô hình học sâu như bộ mã hóa chéo [9, 10, 11] để đánh giá sự tương tác ngữ nghĩa giữa câu truy vấn và từng tài liệu ứng viên.

Một hướng tiếp cận khác cho các hệ thống tái xếp hạng là áp dụng Đồ thị tri thức (KG). KG là một phương pháp biểu diễn dữ liệu dưới dạng cấu trúc được sử dụng rất nhiều trong lĩnh vực trí tuệ nhân tạo [12]. KG có cấu trúc dưới dạng đồ thị bao gồm các nút biểu diễn các thực thể. Các nút sẽ được nối với nhau bởi các đường, các đường này biểu diễn mối quan hệ có thể có tính chất ngữ nghĩa, hoặc tính chất logic giữa các thực thể này. Một cách tổng quát, các kết nối thực thể, quan hệ và thực thể được gọi là các triples. Việc KG tổ chức các mối quan hệ triple này một cách có cấu trúc, trực quan giúp cho máy tính có thể hiểu và tổng quát được các mối quan hệ của các thực thể trong thực tế [13]. Việc tích hợp KG vào RAG giúp cung cấp một cấu trúc dữ liệu tường minh, liên quan hơn.

Trong bài báo này đề xuất một phương pháp tái xếp hạng sử dụng KG. Phương pháp này sẽ đánh giá mức độ tương quan giữa câu truy

vấn và các đoạn văn bản ứng viên dựa trên khoảng cách giữa các thực thể. Chúng tôi cũng sẽ đánh giá phương pháp đề xuất trên bộ rag-mini-bioasq so với RAG gốc trên các độ đo MRR và nDCG.

2. ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Các hệ thống tái xếp hạng

Các hệ thống RAG hiện nay thường theo quy trình truy xuất 2 bước là truy xuất ban đầu và tái xếp hạng. Trong đó các phương pháp tái xếp hạng bao gồm:

- Phương pháp dựa trên từ vựng: là phương pháp truy xuất thông tin truyền thống, tập trung vào việc khớp chính xác các từ khóa giữa truy vấn của người dùng và các tài liệu trong kho dữ liệu. Phương pháp này hoạt động dựa trên các nguyên tắc thống kê, phân tích tần suất xuất hiện và độ hiếm của các từ để tính toán điểm liên quan. Đại diện cho phương pháp là thuật toán BM25 [14, 15]. Ưu điểm của phương pháp này là tốc độ nhanh, độ chính xác cao đối với các truy vấn yêu cầu khớp từ khóa cụ thể. Tuy nhiên, chúng thiếu khả năng hiểu ngữ nghĩa và rất dễ bị ảnh hưởng bởi nhiễu hoặc tài liệu đánh lạc hướng [16].
- Cross-encoder [17, 18] là kiến trúc tái xếp hạng phổ biến. Cross-encoder nhận cả truy vấn và một tài liệu làm một cặp đầu vào duy nhất, và đưa chúng qua toàn bộ mô hình Transformer cùng một lúc. Quá trình xử lý chung này cho phép mô hình thực hiện cơ chế "attention" giữa các token của truy vấn và các token của tài liệu, từ đó tạo ra một điểm số tương quan duy nhất, phản ánh mức độ phù hợp của tài liệu đó với truy vấn. Bằng cách phân tích câu truy vấn và tài liệu trong cùng một ngữ cảnh, cross-encoder có thể nắm bắt được các mối quan hệ phức tạp và sự phụ

thuộc theo ngữ cảnh giữa chúng. Tuy nhiên, chi phí tính toán của cross-encoder là rất lớn.

- Mô hình Late-interaction [19, 20] là một kiến trúc lai giữa Cross-encoder và Dual-encoder. Mô hình này sẽ mã hóa câu truy vấn và văn bản riêng biệt. Tuy nhiên, thay vì nén toàn bộ văn bản thành một vector duy nhất, nó tạo ra một ma trận các vector embedding, mỗi vector tương ứng với một token trong văn bản. Cơ chế tương tác muộn sẽ được thực hiện ở giai đoạn truy vấn, trong đó mô hình sẽ tìm kiếm vector token tương đồng nhất trong ma trận tài liệu. Mô hình này có khả năng nắm bắt mức độ tương quan ngữ nghĩa tốt đồng thời tốc độ tính toán nhanh trong quá trình truy vấn.

- Mô hình tái xếp hạng dựa trên LLM [21, 22] tận dụng khả năng mạnh mẽ của các mô hình LLM để thực hiện việc chấm điểm tái xếp hạng. Phương pháp tiếp cận này dựa trên việc tinh chỉnh các mô hình Pre-trained. Việc tái xếp hạng được thực hiện khi LLM được cung cấp câu lệnh chứa truy vấn và danh sách các tài liệu ứng viên, sau đó mô hình được yêu cầu sắp xếp lại danh sách này. Phương pháp có độ linh hoạt cao tuy nhiên lại rất tốn kém chi phí và thời gian tính toán. Ngoài ra việc phương pháp này cũng thiếu ổn định phụ thuộc vào khả năng tuân thủ chỉ dẫn của LLM. Một hạn chế nữa chính là giới hạn của cửa sổ ngữ cảnh.

2.2. Ứng dụng KG vào tái xếp hạng

Một hướng khác trong bài toán tái xếp hạng đó là ứng dụng KG. KG là một cấu trúc dữ liệu mạnh mẽ, mô hình hóa thông tin dưới dạng một mạng lưới các thực thể (nodes) và các mối quan hệ (edges) giữa chúng. Thay vì xem các tài liệu như những đơn vị thông tin độc lập, KG cho phép hệ thống RAG vượt qua ngữ cảnh phân mảnh, thực hiện suy luận đa bước và cung cấp các câu trả lời có tính giải thích cao hơn.

Một cách tiếp cận phổ biến hiện nay là ứng dụng mạng nơ-ron đồ thị (GNN) cho việc tái xếp hạng [23, 24]. GNN là một lớp các mô hình học sâu được thiết kế để hoạt động trên dữ liệu có cấu trúc đồ thị. Chúng học các biểu diễn vector cho các nút bằng cách tổng hợp thông tin từ các nút lân cận. Các biểu diễn này nắm bắt cả đặc điểm của nút và cấu trúc cục bộ xung quanh nó. Điều này giúp cho mô hình nắm được các mối liên hệ giữa các thực thể, từ đó có thể tái xếp hạng hiệu quả hơn. GNN có khả năng tự động học các đặc trưng cấu trúc phức tạp, thường mang lại độ chính xác cao nhất trong các bài toán tái xếp hạng. Tuy nhiên việc xây dựng một GNN hiệu quả đòi hỏi một lượng lớn dữ liệu huấn luyện. Ngoài ra việc ứng dụng GNN vào RAG cũng rất tốn kém tài nguyên tính toán; bản thân GNN có tính chất "hộp đen" và khó giải thích.

2.3. Phương pháp đề xuất

Trong phần này, bài báo sẽ giới thiệu đề xuất phương pháp cải thiện truy xuất cho RAG bằng cách ứng dụng dữ liệu có cấu trúc của KG. So với phương pháp GNN, phương pháp đề xuất đơn giản hơn trong việc triển khai cũng như có khả năng giải thích. Nguyên lý trung tâm của phương pháp này là mức độ liên quan ngữ nghĩa giữa câu truy vấn và các đoạn văn bản có thể đo lường bằng khoảng cách quan hệ giữa các thực thể liên quan trên KG.

Để kiểm chứng giả thuyết này, bài báo đề xuất xây dựng một quy trình truy xuất hai giai đoạn được tăng cường bởi KG:

- **Xây dựng KG toàn cục:** Đầu tiên, một KG toàn cục được xây dựng từ toàn bộ kho tài liệu. Giống như cơ sở dữ liệu vector, quá trình này chỉ cần thực hiện một lần và có thể được cập nhật định kỳ. Chúng tôi sử dụng Gemini 2.0 Flash trên nền tảng của Neo4j cho tác vụ này.

- **Giai đoạn truy xuất ban đầu:** Khi có truy vấn của người dùng, một hệ thống tìm kiếm vector tiêu chuẩn sẽ được sử dụng để truy xuất một tập hợp lớn gồm K đoạn văn bản ứng viên. Giai đoạn này được tối ưu hóa để tối đa hóa độ phủ, đảm bảo rằng tất cả các đoạn văn bản có khả năng liên quan đều được đưa vào xem xét.

- **Tái xếp hạng dựa trên KG:** Đây là bước cốt lõi của phương pháp đề xuất. Hệ thống trích xuất các thực thể từ truy vấn của người dùng và từ mỗi đoạn văn bản trong số K ứng viên. Đối với câu truy vấn Q và đoạn văn bản ứng viên C, một điểm $\text{Score}(Q, C)$ được tính toán.

Quá trình tính điểm $\text{Score}(Q, C)$ như sau:

- Đầu tiên hệ thống trích xuất tập hợp các thực thể từ câu truy vấn $E_Q = \{e_{q1}, e_{q2}, \dots, e_{qn}\}$, và thực thể từ từng đoạn văn bản ứng viên $E_C = \{e_{c1}, e_{c2}, \dots, e_{cn}\}$.

- Tiếp theo, hệ thống sẽ tính điểm tương đồng ngữ nghĩa dựa trên khoảng cách ngắn nhất trên KG. Đối với mỗi cặp thực thể (e_q, e_c) , ta có công thức:

$$\text{sim}(e_q, e_c) = \frac{1}{1 + \text{ShortestPathLength}(e_q, e_c)}$$

Trong đó $\text{ShortestPathLength}$ được tính là khoảng cách ngắn nhất giữa e_q và e_c trên KG.

- Điểm $\text{Score}(Q, C)$ cuối cùng sẽ được tính bằng việc lấy tổng $\text{sim}(e_q, e_c)$ của toàn bộ các cặp (e_q, e_c) có thể. Sau đó điểm được chuẩn hóa bằng cách chia cho số lượng các cặp đang xét.

$$\text{Score}(Q, C) = \frac{\sum_{e_q \in E_Q} \sum_{e_c \in E_C} \text{sim}(e_q, e_c)}{|E_Q| |E_C|}$$

- Cuối cùng hệ thống sẽ tái xếp hạng các văn bản ứng viên dựa trên điểm $\text{Score}(Q, C)$ theo

thứ tự các văn bản hạng cao sẽ có điểm số cao, và ngược lại các văn bản hạng thấp sẽ có điểm số thấp.

3. KẾT QUẢ VÀ THẢO LUẬN

Trong phần này, chúng tôi sẽ thực hiện đánh giá phương pháp thông qua hai trường hợp là hệ thống RAG gốc và hệ thống RAG được tăng cường bởi KG. Để đánh giá khả năng truy xuất của hệ thống, chúng tôi sử dụng tham số MRR và nDCG.

3.1. Bộ dữ liệu

Bộ dữ liệu sử dụng là rag-mini-bioasq. Bộ dữ liệu này bao gồm 2 phần chính:

- Thành phần text-corpus bao gồm các đoạn văn bản có giá id.
- Bộ dữ liệu kiểm thử dưới dạng QA, trong đó bao gồm câu hỏi, câu trả lời và một danh sách các id của các đoạn văn bản trong text_corpus tương ứng có nội dung liên quan đến câu hỏi.

3.2. Kết quả thực nghiệm

Để đánh giá độ hiệu quả của phương pháp, chúng tôi tiến hành thí nghiệm với hai trường hợp là hệ thống RAG gốc và hệ thống RAG được tăng cường bởi KG trên bộ dữ liệu rag-mini-bioasq. Các tham số đánh sử dụng để đánh giá là MRR và nDCG. Kết quả thí nghiệm với K=5, K=10 lần lượt được thể hiện trong Bảng 1, Bảng 2.

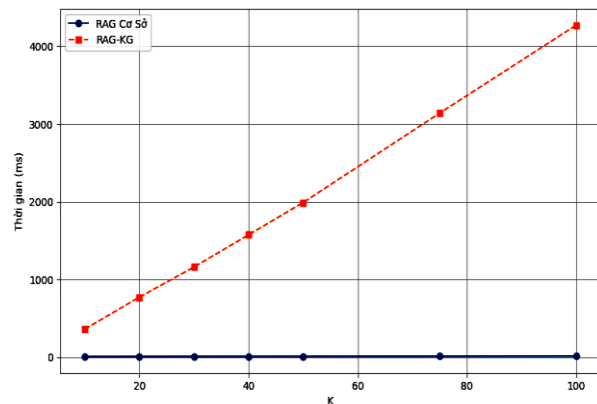
Bảng 1. Đánh giá trên top 5 văn bản

	MRR@5	nDCG@5
RAG	21.94	22.33
RAG-KG	24.15	23.98

Bảng 2. Đánh giá trên top 10 văn bản

	MRR@10	nDCG@10
RAG	22.1	22.21
RAG-KG	26	25.22

Hình 1 thể hiện chi phí tính toán trong quá trình truy xuất của hệ thống RAG gốc và RAG được tăng cường bởi KG.



Hình 1. Chi phí tính toán

3.3. Thảo luận

Kết quả từ Bảng 1 và Bảng 2 cho thấy phương pháp của chúng tôi cho kết quả tốt hơn so với RAG cơ sở trên cả hai chỉ số MRR và nDCG tại cả hai mức K=5 và K=10. Điều này chứng tỏ phương pháp tái xếp hạng dựa trên sự kết nối tri thức đã thành công trong việc đẩy các tài liệu liên quan hơn lên các vị trí cao hơn trong danh sách kết quả.

Sự khác biệt về hiệu suất trở nên đặc biệt rõ rệt khi xem xét top 10 tài liệu. Điều này cho thấy phương pháp đề xuất có hiệu quả trong việc tinh lọc một danh sách các ứng cử viên ban đầu lớn hơn, xác định và ưu tiên các tài liệu có giá trị cốt lõi mà phương pháp truy xuất dựa trên vector thuần túy có thể đã xếp hạng thấp hơn.

Hình 1 cho thấy thời gian xử lý của hệ thống RAG được tăng cường bởi KG có mối quan hệ tuyến tính rõ rệt với giá trị của K. Thời gian tăng từ khoảng 400 ms tại K=10 lên đến hơn 4000 ms (4 giây) tại K=100. Nguyên nhân là do hệ thống phải xử lý tuần tự từng tài liệu trong số K tài liệu được truy xuất ban đầu: trích xuất triplet bằng mô hình LLM và sau đó tính toán điểm số dựa trên KG.

4. KẾT LUẬN

Trong nghiên cứu này, chúng tôi đã đề xuất và đánh giá một phương pháp tái xếp hạng ứng dụng KG nhằm cải thiện chất lượng truy xuất RAG. Kết quả thực nghiệm đã chứng minh rằng phương pháp đề xuất có cải thiện chất lượng truy xuất của RAG. Điều này khẳng định rằng việc tích hợp một lớp xác thực dựa

trên tri thức có cấu trúc giúp hệ thống ưu tiên các tài liệu có tính liên quan về mặt dữ kiện một cách hiệu quả hơn, thay vì chỉ dựa vào sự tương đồng ngữ nghĩa bề mặt. So với các phương pháp tái xếp hạng phức tạp hiện nay thì phương pháp đề xuất là một giải pháp ít tốn kém, có hiệu quả và giải thích được.

TÀI LIỆU THAM KHẢO

- [1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, & Guillaume Lample, “*LLaMA: Open and Efficient Foundation Language Models*”, (2023).
- [2] OpenAI, “*GPT-4 Technical Report*”, (2024).
- [3] DeepSeek-AI, “*DeepSeek-V3 Technical Report*”, (2025).
- [4] Chaitanya Sharma, “*Retrieval-Augmented Generation: A Comprehensive Survey of Architectures, Enhancements, and Robustness Frontiers*”, (2025).
- [5] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, & Haofen Wang, “*Retrieval-Augmented Generation for Large Language Models: A Survey*”, (2024).
- [6] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, & Qing Li, “*A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models*”, (2024).
- [7] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, & Suranga Nanayakkara, “*Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering*”, (2022).
- [8] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, & Percy Liang, “*Lost in the Middle: How Language Models Use Long Contexts*”, (2023).
- [9] Rodrigo Nogueira & Kyunghyun Cho, “*Passage Re-ranking with BERT*”, (2020).
- [10] Rodrigo Nogueira, Zhiying Jiang, & Jimmy Lin, “*Document Ranking with a Pretrained Sequence-to-Sequence Model*”, (2020).
- [11] MacAvaney, S., Yates, A., Cohan, A., & Goharian, N., “*CEDR: Contextualized Embeddings for Document Ranking*”, In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM (2019).
- [12] Xuhui Jiang, Chengjin Xu, Yinghan Shen, Xun Sun, Luminyuan Tang, Saizhuo Wang, Zhongwu Chen, Yuanzhuo Wang, & Jian Guo, “*On the Evolution of Knowledge Graphs: A Survey and Perspective*”, (2025).
- [13] Hogan, A., Blomqvist, E., Cochez, M., D’amato, C., Melo, G., Gutierrez, C., Kirrane, S., Gayo, J., Navigli, R., Neumaier, S., Ngomo, A.C., Polleres, A., Rashid, S., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A., “*Knowledge Graphs*”, ACM Computing Surveys, 54(4) (2021).
- [14] Soares, L., Gillick, D., Cole, J., & Kwiatkowski, T., “*NAIL: Lexical Retrieval Indices with Efficient Non-Autoregressive Decoders*”, In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2023).

- [15] Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, & Suzan Verberne, “*Injecting the BM25 Score as Text Improves BERT-Based Re-rankers*”, (2023).
- [16] Lovisa Hagström, Ercong Nie, Ruben Halifa, Helmut Schmid, Richard Johansson, & Alexander Junge, “*Language Model Re-rankers are Fooled by Lexical Similarities*”, (2025).
- [17] Shashank Verma, Fengyi Jiang, & Xiangning Xue, “*Beyond Retrieval: Ensembling Cross-Encoders and GPT Rerankers with LLMs for Biomedical QA*”, (2025).
- [18] Petrov, A.V., MacAvaney, S., Macdonald, C., “*Shallow Cross-Encoders for Low-Latency Retrieval*”, In: Goharian, N., et al. Advances in Information Retrieval. ECIR 2024. Lecture Notes in Computer Science, vol 14610. Springer, Cham (2024).
- [19] Ziwei Ji, Himanshu Jain, Andreas Veit, Sashank J. Reddi, Sadeep Jayasumana, Ankit Singh Rawat, Aditya Krishna Menon, Felix Yu, & Sanjiv Kumar, “*Efficient Document Ranking with Learnable Late Interactions*”, (2024)
- [20] Khattab, O., & Zaharia, M., “*ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*”, In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery (2020).
- [21] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, L. Yan, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang, and M. Bendersky, “*Large language models are effective text rankers with pairwise ranking prompting*”, In Findings of the Association for Computational Linguistics: NAACL 2024 (2024).
- [22] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, & Zhaochun Ren, “*Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents*”, (2024).
- [23] Andrea Giuseppe Di Francesco, Christian Giannetti, Nicola Tonellotto, & Fabrizio Silvestri, “*Graph Neural Re-Ranking via Corpus Graph*”, (2024).
- [24] Jialin Dong, Bahare Fatemi, Bryan Perozzi, Lin F. Yang, & Anton Tsitsulin, “*Don't Forget to Connect! Improving RAG with Graph-based Reranking*”, (2024).

Thông tin liên hệ: **Trần Quốc Hưng**

Điện thoại: 0909873655 - Email: tqhung.dt@uneti.edu.vn

Khoa Điện tử và Kỹ thuật máy tính, Trường Đại học Kinh tế - Kỹ thuật
Công nghiệp