DOI: https://doi.org/10.59294/HIUJS.KHQG.2024.002

PHƯƠNG PHÁP TÍCH HỢP MÔ HÌNH TRI THỨC VÀ BIỂU DIỄN TÀI LIỆU CHO HỆ TRUY VẤN KIẾN THỨC

Đỗ Văn Nhơn[1]*, Mai Trung Thành[1] và Huỳnh Thị Thanh Thương[2]

[1]Trường Đại học Quốc tế Hồng Bàng
[2]Đại học Công nghệ thông tin, Đại học Quốc gia TP.HCM

TÓM TẮT

Việc nghiên cứu phát triển phương pháp và kỹ thuật để thiết kế các ứng dụng truy vấn kiến thức trong giáo dục điện tử là nhu cầu cấp thiết trong thực tế. Trước đây đã có nhiều công trình nghiên cứu liên quan đến biểu diễn tri thức và một số kỹ thuật cho truy vấn tri thức nhưng còn hạn chế cần nghiên cứu cải thiện và cải tiến. Hạn chế chủ yếu là việc sử dụng mô hình biểu diễn tri thức cho giải vấn đề thông minh nên chưa phù hợp với nhu cầu truy vấn tri thức, và cũng chưa chỉ ra sự trích dẫn tài liệu chuyên môn. Bài báo này trình bày một nghiên cứu mới về việc xây dựng mô hình tri thức với sự tích hợp mô hình biểu diễn nội dung tài liệu phục vụ thiết kế hệ thống hỗ trợ truy vấn kiến thức kèm theo trích dẫn từ tài liệu chuyên môn có chứa tri thức đó. Phương pháp và kỹ thuật đề xuất trong bài báo cũng được dùng trong thiết kế một ứng dụng thử nghiệm cụ thể; kết thử nghiệm và đánh giá cho thấy hệ thống khả thi và hiệu quả cho việc tra cứu kiến thức trong dạy và học.

Từ khóa: *biểu diễn tri thức, biểu diễn tài liệu, tích hợp tri thức, hệ cơ sở tri thức, phần mềm giáo dục thông minh*

AN INTEGRATED METHOD OF KNOWLEDGE REPRESENTATION AND SEMANTIC DOCUMENT REPRESENTATION FOR DESIGNING KNOWLEDGE QUERYING SYSTEMS

Do Van Nhon, Mai Trung Thanh and Huynh Thi Thanh Thuong

ABSTRACT

Researching and development of methods and techniques for designing knowledge querying systems in e-learning is essential in practice. There have been many related works in knowledge representation and some techniques for knowledge querying, but there are still limitations that need further improvement and adaptation. Previous studies still have limitations because they were used for designing intelligent problem-solving systems. Therefore, they are not suitable for handling knowledge querying requirements and related document citation methods. The paper will present an integration method of knowledge representation and semantic document representation for designing knowledge querying systems that have the ability to respond to users with results and citations of corresponding documents. The proposed methods and techniques in this paper are also used in designing a specific experimental application; experimental results and evaluations show that the integration method is useful and effective for designing knowledge querying systems in teaching and learning.

Keywords: k*nowledge representation, document representation, knowledge intergration, knowledge base systems, interlligent educational software*

* Tác giả liên hệ: ThS. Mai Trung Thành, Email: thanhmt@hiu.vn
(Ngày nhận bài: 04/03/2024; Ngày nhận bản sửa: 19/4/2024; Ngày duyệt đăng: 04/05/2024)

1. TỔNG QUAN NGHIÊN CỨU

Thiết kế các lớp ứng dụng hỗ trợ học tập thông minh là rất cần thiết và ý nghĩa trong giáo dục [1]. Đặc biệt là các lớp ứng dụng hỗ trợ được các nhóm chức năng cho phép người dùng có thể tra cứu hay truy vấn kiến thức, truy tìm nội dung tài liệu [2, 3]. Người dùng có thể đưa vào (input) hệ thống các *câu truy vấn* mà người dùng mong muốn, hệ thống phải trả về các kết quả phù hợp với mong muốn của người dùng. Bên cạnh đó, ngoài việc đảm bảo kết quả trả về phù hợp theo yêu cầu người dùng, các kết quả cần đảm bảo độ tin cậy. Nghĩa là, mỗi yếu tố tri thức hay nội dung được trả về từ hệ thống cũng phải kèm theo thông tin chúng được trích dẫn từ các tài liệu nào, vị trí nào trong những tài liệu tương ứng.

Để thiết kế được loại hệ thống vừa có khả năng cho phép tìm kiếm truy vấn kiến thức và trả về kết quả có trích dẫn rõ nguồn tài liệu, đòi hỏi hỏi cần có những giải pháp trong việc kết hợp hay tích hợp các tri thức và các các nội dung của tài liệu. Việc đưa ra giải pháp tích hợp tri thức và các nội dung của tài liệu sẽ làm cơ sở khoa học, giúp cho các nhà phát triển có thể thiết kế được các lớp ứng dụng có khả năng hỗ trợ chức năng truy vấn kiến thức và đảm bộ tin cậy.

Hiện nay, đã có nhiều giải pháp trong thiết kế các lớp ứng dụng hỗ trợ truy vấn hay tìm kiếm nội dung. Ta có thể điểm qua một số kết quả nổi bật hiện nay như sau:

Trong công trình [4, 5], nhóm tác giả đã đưa ra một số phương pháp biểu diễn tri thức theo phương pháp tích hợp, từ đó làm cơ sở để hướng đến thiết kế hệ thống hỗ trợ truy vấn tri thức trong học tập. Hệ thống có ưu điểm đó là cho phép người dùng có thể truy vấn tri thức theo phân loại kiến thức, đồng thời có khả năng xử lý được các truy vấn dưới dạng ngôn ngữ tự nhiên. Các kết quả nghiên cứu đã được vận dụng thử nghiệm trên một số miền tri thức như Nhập môn lập trình, Toán cấp trung học phổ thông với kết quả trả về có độ chính xác cao, phù hợp với nội dung của yêu cầu được nhập vào từ người dùng. Tuy nhiên, các kết quả trả về của hệ thống chỉ cho phép trả về các kết quả là nội dung hiển thị cho người dùng, chưa quan tâm đến trích dẫn của kết quả trả về từ hệ thống.

Nhóm ứng dụng hỗ trợ truy vấn tri thức [3-8], đây là các nhóm giải pháp biểu diễn tri thức hướng đến thiết kế các lớp ứng dụng có khả năng hỗ trợ truy vấn tri thức dựa trên quy ước câu truy vấn. Các kết quả nghiên cứu cũng đã đưa ra được phương pháp biểu diễn tri thức và xét các lớp bài toán trong việc truy vấn tri thức. Ưu điểm của các ứng dụng này đó là cho phép truy vấn tri thức theo phân loại tri thức, và có khả năng hỗ trợ truy vấn đa yêu cầu [7]. Bên cạnh đó, các nhóm hệ thống này có khả năng xử lý các câu truy vấn và trả về các kết quả phù hợp với yêu cầu được diễn đạt qua câu truy vấn. Kết quả cũng được triển khai thử nghiệm trên một số miền tri thức như Nhập môn lập trình [7], Toán cấp trung học phổ thông [6, 8], Lý thuyết đồ thị [3]. Dù hỗ trợ tốt trong việc cung cấp dạng quy ước trong việc truy vấn tri thức theo sự phân loại, tuy nhiên các kết quả trả về từ hệ thống lại chưa quan tâm đến trích dẫn rõ tài liệu của kết quả trả về.

ChatGPT (**C**hat **G**enerative **P**re-training **T**ransformer) là một ứng dụng của công nghệ AI, người dùng có thể diễn đạt mong muốn bằng cách nhập vào hệ thống các câu ngắn (sentence short) hoặc là các cụm từ (phrase) chúng được gọi là các prompt [9]. Từ các yêu cầu được diễn đạt bằng ngôn ngữ tự nhiên prompt, dựa trên mô hình ngôn ngữ lớn LLM (**L**ager **L**anguage **M**odel) [11], ChatGPT sẽ cho ra được các câu trả lời phù hợp với nội dung của prompt. Nội dung trả về phù hợp với "*ngữ nghĩa*" của các prompt, được trình bày bằng ngôn ngữ tự nhiên. Điểm mạnh của công cụ này đó là có thể "*hiểu*" được ngữ nghĩa của ngôn ngữ tự nhiên và khả năng hỗ trợ đa dạng yêu cầu từ người dùng có thể kể đến một số chức năng phổ biến như: tìm kiếm, hỏi – đáp, dịch ngôn ngữ, tạo/sửa các mã lệnh lập trình, tạo/sinh nội dung văn bản và đặc biệt có thể hỗ trợ đa ngôn ngữ. Tuy nhiên, kết quả trả về từ công cụ này chưa đảm bảo được độ tin cậy, người dùng cần xác thực lại các thông tin trả về từ công cụ này.

Với một cách tiếp cận mới [2, 11], nhóm tác giải đã đưa ra được giải pháp trong việc biểu diễn nội dung tài liệu CK-ONTO (***C**lassed **K**eyphrase* based ***Onto**logy*) [2], từ cơ sở đó cho phép biểu diễn được nội dung tài liệu một cách ***sâu*** hơn về mặt ngữ nghĩa. Giải pháp đã được vận dụng vào triển khai các lớp ứng dụng tìm kiếm theo nội dung ngữ nghĩa trên một số miền tri thức, kết quả trả về của các hệ thống này là các tài liệu có liên quan đến nội dung tìm kiếm. Dù có nhiều điểm mạnh, tuy nhiên chúng chỉ trả về danh

mục các tài liệu đến câu truy vấn, mà chưa thể trích xuất được chi tiết về vị trí bên trong mỗi tài liệu có liên quan, điều này gây ra rất nhiều khó khăn cho người dùng trong việc đọc và trải nghiệm các kết quả.

Các kết quả nghiên cứu ở trên nhìn chung đều có những điểm mạnh, đó là có khả năng hỗ trợ được truy vấn tri thức hay tìm kiếm theo nội dung ngữ nghĩa của tài liệu. Bên cạnh đó, các điểm yếu của những nhóm hệ thống này đó là hệ thống có kết quả trả về chưa có sự đảm bảo độ tin cậy, khi chưa thể chỉ rõ các nguồn trích dẫn như trong [3-8]. Một điểm mạnh của [2, 11] dù đã đưa ra được các tài liệu liên quan, tuy nhiên chúng lại chưa chỉ dẫn được chi tiết vị trí của nội dung quan tâm hay liên quan có đề cập trong từng tài liệu.

Với những nhu cầu từ thực tế cùng các điểm yếu của một số công trình đã nghiên cứu, bài báo sẽ trình bày một tiếp cận mới trong việc tích hợp một mô hình biểu diễn tri thức tựa COKB (Computational Objects Knowledge Base) và phương pháp biểu diễn nội dung tài liệu văn bản có giới hạn là các tài liệu dạng Ebook. Trên cơ sở tích hợp, bài báo sẽ xem xét lớp bài toán truy vấn tri thức, có các kết quả là tri thức được trả về từ hệ thống cần đảm bảo được sự tin cậy, nghĩa là kết quả trả về không chỉ gồm các tri thức, mà còn chỉ rõ được vị trí có đề cập đến tri thức trong các tài liệu. Với lớp bài toán được đề xuất, các phương pháp, kỹ thuật và thuật giải cũng sẽ được trình bày trong bài báo. Bên cạnh đó, một kết quả vận dụng giải pháp trong thiết kế hệ thống hỗ trợ truy vấn tri thức trên miền tri thức về lập trình trên máy tính cũng sẽ được đề cập chi tiết trong bài báo.

2. PHƯƠNG PHÁP TÍCH HỢP MÔ HÌNH BIỂU DIỄN TRI THỨC VÀ MÔ HÌNH BIỂU DIỄN TÀI LIỆU EBOOK

Định nghĩa 2.1 Mô hình biểu diễn tri thức lập trình trên máy tính là mô hình tựa COKB (Computational Objects Knowledge Base), gồm bộ 6 thành phần như sau: **(C, R, Funcs, Rules, Problems, Methods)**. Trong đó:

- C là một tập hợp hệ thống khái niệm trong miền tri thức, trong đó mỗi khái niệm $c \in$ C là một lớp các đối tượng. Mỗi lớp đối tượng được phân cấp dựa trên cấu trúc hay cách xác định của đối tượng [7, 12]. Dựa trên mô hình COKB hiệu chỉnh, ta có một đối tượng *o* có cấu trúc gồm các thành phần (*Attrs, Facts, InerRules, Contents*). *Attrs* là tập các thuộc tính của khái niệm *c*; *Facts* là tập các tính chất nội tại của khái niệm *c*; *InnerRules* là tập các luật nội tại có dạng luật dẫn r: $h \rightarrow g$, với *h* là tập sự kiện giải thiết và *g* là tập các sự kiện kết luận; *Contents* là một tập các yếu tố/đặc trưng/thông tin mô tả ngữ nghĩa của đối tượng, chúng được diễn đạt bằng ngôn ngữ tự nhiên, *Contents* = $\{f_1, f_2, \ldots, f_n\}$, mỗi $f \in$ *Contents* có cấu trúc là một bộ *key-value*.

- R là thành phần tri thức về quan hệ hai ngôi giữa các khái niệm. Để hỗ trợ cho các vấn đề truy vấn và hiển thị thông tin, mỗi quan hệ r cấu trúc gồm 2 thành phần: phần diễn đạt thông tin ngữ nghĩa của quan hệ dưới dạng ngôn ngữ tự nhiên; phần cấu trúc của quan hệ vẫn được giữ lại theo cấu trúc của quan hệ trong mô hình COKB, chi tiết cấu trúc được trình bày trong tài liệu [12].

- Funcs là tập các tri thức hàm có trong miền tri thức. Mỗi tri thức hàm có cấu trúc gồm bộ (*name, return_datatype, syntax, parameters, def, meaning*). Trong đó: *name* là tên của hàm; *return_datatype* là kiểu trả của hàm; *syntax* là cú pháp của hàm; *parameters* là danh sách các tham số đầu vào của hàm; *def* là định nghĩa của hàm dưới dạng thủ tục; *meaning* là thành phần diễn đạt thông tin ngữ nghĩa của hàm dưới dạng ngôn ngữ tự nhiên.

- Rules là tập các tri thức dạng luật trong miền tri thức. Mỗi luật suy diễn gồm có 2 phần: phần diễn đạt thông tin ngữ nghĩa của luật được diễn đạt dưới dạng ngôn ngữ tự nhiên; phần cấu trúc chính của luật, cấu chính luật r có dạng luật dẫn có dạng r: $h \rightarrow g$, vơi *h* là tập các sự kiện giả thiết, *g* là tập các sự kiện kết luật. Các sự kiện và thuật giải hợp nhất sự kiện được trình bày trong [7, tr.2].

- Problems là tập các dạng bài tập/bài toán trong miền tri thức, mỗi dạng bài toán *p* trong miền tri thức gồm có các thành phần (*name, input, output, examples, statement*). Trong đó, *name* là tên của dạng bài toán *p*; *input* là thành phần diễn đạt các dữ kiện/thông tin đầu vào của dạng bài toán *p*; *output* là thành

phần diễn đạt các dữ kiện/thông tin yêu cầu đầu ra của bài toán *p*; *examples* là tập các ví dụ cụ thể cho dạng bài toán *p*; *statement* là thành phần phát biểu bài toán dưới dạng ngôn ngữ tự nhiên.

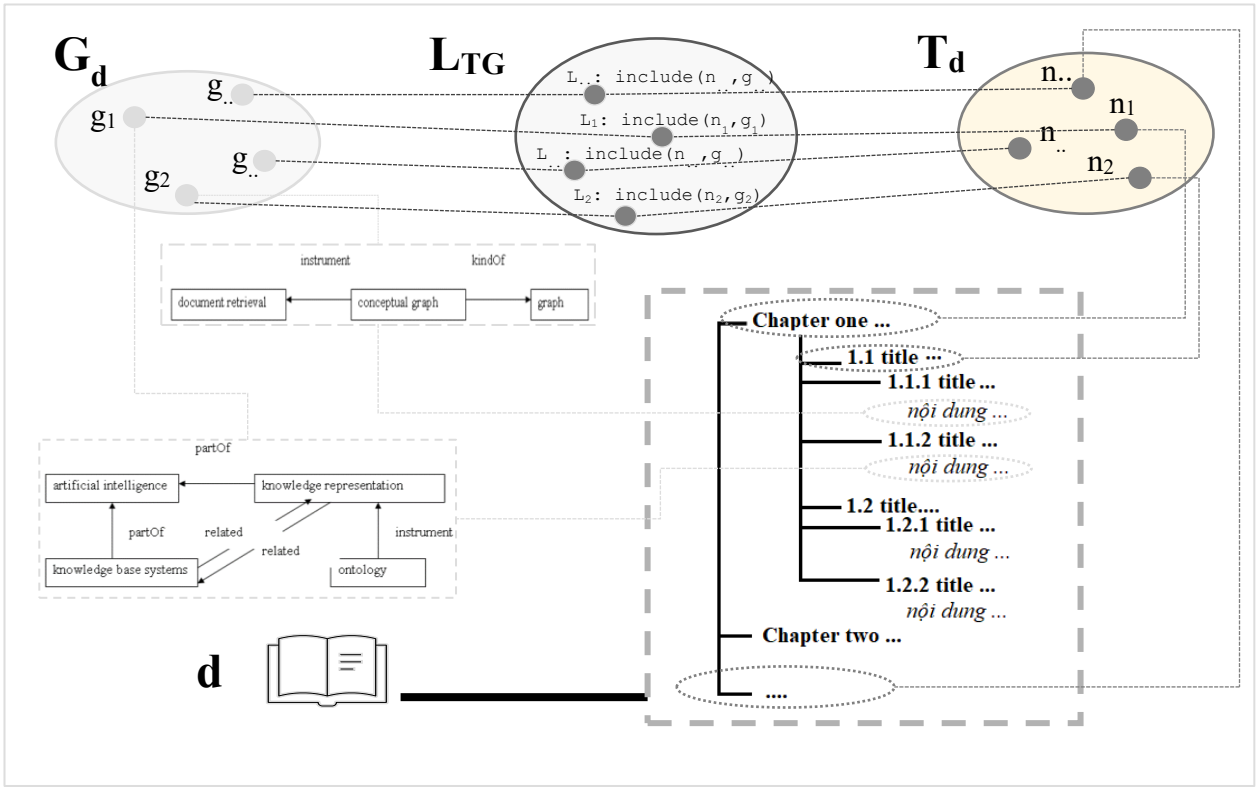
- Methods là tập các phương pháp giải cho các dạng bài toán trong **Problems**, mỗi phương pháp giải *m* trong miền tri thức có các thành phần (*name, p, algorithm, meaning, examples*). Trong đó *name* là tên của phương pháp giải *m*; *p* là dạng bài tập được giải bởi phương pháp giải *m*; *algorithm* là thuật giải/phương pháp giải được trình bày dưới dạng thủ tục bằng mã giả hoặc mã code bởi ngôn ngữ lập trình; *meaning* là thành phần diễn đạt ngữ nghĩa của phương pháp giải được trình bày dưới dạng ngôn ngữ tự nhiên; *examples* là tập các ví dụ mẫu về phương pháp *m* để giải các dạng bài tập mẫu trong *p*.

Định nghĩa 2.2 Mô hình biểu diễn nội dung cho các tài liệu Ebook gồm bộ 3 thành phần: **(K, R_{KK}, D)**. Trong đó:

- K là tập các keyphrase (theo [11]) có trong miền tri thức, trong đó tên gọi của một yếu tố tri thức trong miền tri thức cũng là một keyphrase trong **K**. Ví dụ: trong phạm vi tri thức về lập trình trên máy tính ta có một số khái niệm như: Mã nguồn, mã máy, chương trình, trình biên dịch, dữ liệu, kiểu dữ liệu, biến được xem là các keyphrases.

- R_{KK} là tập các quan hệ hai ngôi giữa các keyphrases, gồm các quan hệ như: quan hệ đồng nghĩa, quan hệ viết tắt và quan hệ gần nghĩa giữa các keyphrases.

- D là tập các tài liệu là các Ebook, **D** = $\{d_1, d_2, ..., d_n\}$. Mỗi tài liệu $d_i \in$ **D** có cấu trúc gồm ($\mathbf{T_d}$, $\mathbf{G_d}$, $\mathbf{L_{TG}}$). $\mathbf{T_d}$ là thành phần diễn đạt bố cục chương mục của Ebook d_i. Cấu trúc cây chương mục có thể được biểu diễn bởi cấu trúc cây gồm tập các nút (node) N và tập quan hệ R_{NN}, để chỉ quan hệ "CHA" – "CON" giữa các nút; $\mathbf{G_d}$ = $\{g_1, g_2, g_3, ..., g_n\}$, mỗi $g_i \in \mathbf{G_d}$ là một đồ thị keyphrase được trình bày dựa trên cơ sở [11]. Mỗi đồ thị keyphrase $g_i = (V_K, E_K)$, với V_K là tập các keyphrase $k \in$ **K**. E_K là tập các quan hệ hai ngôi giữa hai keyphrases, mỗi quan hệ $r_e \in E_K$ được diễn đạt bằng một bộ gồm tên quan hệ r_e và k_1, k_2 là hai *Từ khóa* có quan hệ r_e. L_{TG} là một tập các liên kết, mỗi liên kết sẽ chỉ mối quan hệ "*include*" giữa một nút trong $\mathbf{T_d}$ và một đồ thị $g_i \in \mathbf{G_d}$.



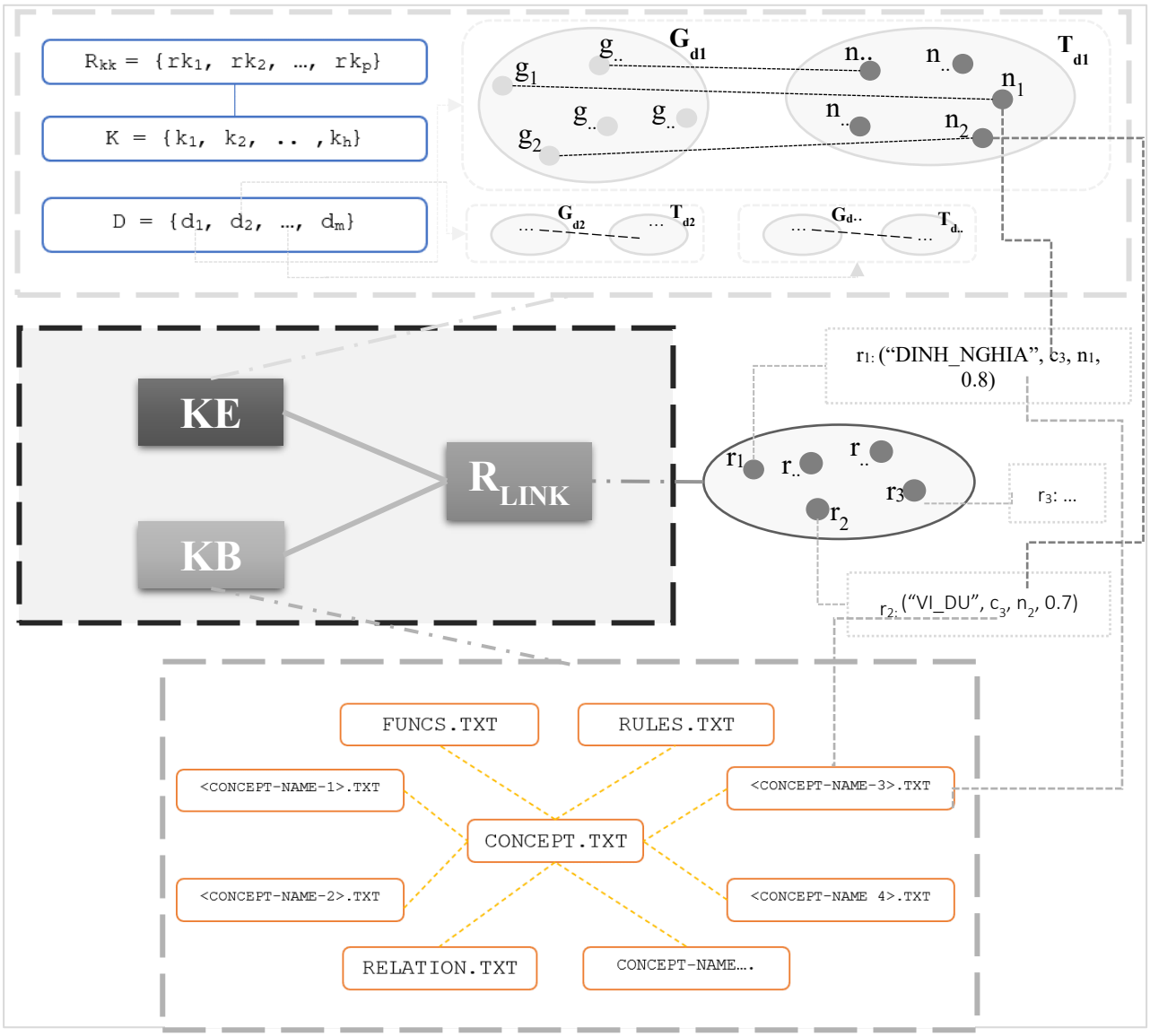
Hình 1. Hình minh họa cấu trúc của một tài liệu d

Định nghĩa 2.3 Mô hình biểu diễn tri thức tích hợp *cơ sở tri thức* và *cơ sở nội dung tài liệu Ebook* là một bộ gồm 3 thành phần: **(KB, KE, RLINK)**

- KB (Knowledge Base) là thành phần biểu diễn cho cơ sở tri thức, **KB** là một mô hình có cấu trúc tựa COKB theo **định nghĩa 2.1**.

- KE (Knowledge base from Ebook) là thành phần biểu diễn nội dung cho tài liệu là các Ebook, **KE** là một mô hình có cấu trúc tựa CK-ONTO theo định nghĩa 2.2.

- RLINK (Linking Relation) là tập hợp các quan hệ hai ngôi giữa các yếu tố tri thức trong **KB** và **KE**, mỗi quan hệ hai ngôi r_{link} là để chỉ một *mối liên kết* giữa yếu tố tri thức trong **KB** và **KE**. Mỗi quan hệ liên kết được xác định qua tên liên kết, có cấu trúc gồm 4 thành phần (*linkname, e, n, w*). Trong đó *linkname* là tên của liên kết; *e* là một yếu tố tri thức trong **KB** ($e \in \mathbf{C} \cup \mathbf{R} \cup \mathbf{Funcs} \cup \mathbf{Rules} \cup \mathbf{Problems} \cup \mathbf{Methods}$); *n* là một nút trong cây $\mathbf{T_d}$ của tài liệu *d* ($d \in \mathbf{D}$); *w* là trọng số để chỉ mức độ *đề cập* của tri thức *e* trong nút *n*, $w \in [0, 1]$. *w* càng gần 0 thì mức độ đề cập của *e* trong nút *n* càng thấp, ngược lại *w* càng gần 1 thì mức độ đề cập của *e* trong node *n* càng cao, trong số *w* được gán bởi chuyên gia trong lĩnh vực. Ta gọi mức độ liên quan của tri thức *e* với tài liệu *d* là w_{de}, ta có thể xác định w_{de} bằng công thức: $w_{de} = max\{w_1, w_2, \ldots, w_v\}$ (*), w_i là giá trị mức độ đề cập đến tri thức *e* của nút n_i ($n_i \in \mathbf{N_d}$, $\mathbf{N_d}$ là tập nút trong cây của tài liệu *d*).



Hình 2. Kiến trúc tổ chức mô hình tri thức tích hợp

3. BÀI TOÁN TRUY VẤN TRI THỨC VÀ THUẬT GIẢI

3.1. Bài toán truy vấn tri thức

Truy vấn tri thức là một trong những nhu cầu thiết yếu của con người trong quá trình học tập kiến thức. Một trong những mong muốn cơ bản đó là truy tìm các tri thức theo sự phân loại tri thức như các khái niệm, các tri thức có dạng là các quy tắc, các dạng bài tập cùng phương pháp giải. Đặc biệt là nhu cầu truy vấn phức tạp, có sự kết hợp hay liên kết các tri thức thông qua các mối quan hệ giữa chúng. Một quy ước đơn giản và phù hợp sẽ cho phép người dùng có thể thuận tiện trong việc giao tiếp với hệ thống. Một số quy ước hỗ trợ truy vấn từ [7] đã được đề xuất, cho phép người dùng thuận tiện và dễ dàng đặc tả các yêu cầu truy vấn các tri thức theo sự phân loại, cùng các truy vấn phức tạp có kết hợp các mối quan hệ giữa các tri thức.

Với các yêu cầu đặc tả theo quy ước (câu truy vấn), hệ thống cần trả về các kết quả cho người dùng. Các kết quả trả về là những nội dung có chứa các tri thức được truy tìm trong cơ sở tri thức, những nội dung khi được trả về cho người dùng sẽ kèm theo đó là các trích dẫn từ các tài liệu là các Ebook. Các tài liệu này cũng sẽ được sắp xếp theo thứ tự về *mức độ liên quan* giữa các tri thức trong nội dung và tài liệu.

Định nghĩa 3.1: Cho miền tri thức về lập trình trên máy tính **K** và tập tài liệu Ebook **D** có chứa các nội dung đề cập đến tri thức trong **K**. Tri thức **K** và tài liệu **D** được cấu trúc hóa theo *mô hình tri thức tích hợp* được trình bày theo **định nghĩa 2.3**. Từ câu truy vấn *q* hãy thực xử lý và trả về kết quả *results* phù hợp với các yêu cầu được đặc tả trong câu truy vấn *q*. Nội dung *results* là tập các tri thức *e* trong **K** cùng các trích dẫn của mỗi tri thức *e* có đề cập chi tiết bên trong các tài liệu D_e ($D_e \subseteq$ **D**). Ta có thể mô hình hóa bài toán truy vấn tri thức **P** có dạng sau đây: ***query q*** $\rightarrow$ ***results***.

query q là một dạng quy ước truy vấn có cấu trúc đặc tả được trình bày theo [7]; ***results*** là nội dung cần trả về theo các yêu cầu được diễn đạt theo câu truy vấn *q*, ***results*** = $\{rs_1, rs_2, ..., rs_u\}$, với mỗi *rs* $\in$ ***results*** gồm hai thành phần: *e* là yếu tố tri thức trong cơ sở tri thức **K** và tập tài liệu D_e có đề cập đến tri thức *e*, $D_e = [d_{e1}, d_{e2}, ..., d_{em}]$. Mỗi d_e trong D_e có cấu trúc là một bộ gồm (d, N_e, w_{de}), với *d* là một tài liệu Ebook, N_e là tập các nút có trong cây T_d của tài liệu *d* mà có đề cập đến tri thức *e*, w_{de} là trọng số chỉ mức độ đề cập của tài liệu *d* với tri thức *e*, w_{de} được tính theo công thức (*).

3.2. Thuật giải

Dựa trên sự kế thừa thuật giải [7, tr.3-5], trong việc trả về các tri thức theo yêu cầu truy vấn từ q. Thuật giải 3.1 dưới đây sẽ trình bày quá trình có thể tìm được các trích dẫn cho từng tri thức, dựa trên kiến trúc tri thức tích hợp. Bài toán **P** có thể được tìm thấy bởi thuật giải sau:

Thuật giải 3.1

Base: cơ sở tri thức tích hợp gồm tri thức **K**, tập tài liệu Ebook **D** (có cấut trúc được mô hình hóa theo định nghĩa 3.1)

Input: query *q*; với *q* có cú pháp đặc tả theo [7, tr.2-3].

Output: *results; với results* = $\{rs_1, rs_2, ..., rs_u\}$.

Giai đoạn 1: thực hiện quá trình xử lý câu truy vấn q và truy vấn theo các yêu cầu được đặc tả trong q trên cơ sở tri thức KB dựa trên cơ sở thuật toán 3.1 được trình bày trong tài liệu [7, tr.3-5].

Giai đoạn 2: thực hiện gắn trích dẫn tài liệu vào các tri thức được trả về ở giai đoạn 1 *ResultsFromKB*. Quá trình gắn các trích dẫn vào tri thức được thực hiện qua các bước xử lý sau:

```
results:={}:
for r in ResultsFromKB do
        rs := []: docs:={}
        for d in D do
        de:=[]: Nd := {}: wd := 0:
```

```
                for n in Node(d) do
                        for rela in RLINK do
                                if rela [3] = e and rela [2] = n and rela[3] >0 then
                                        Nd:= Nd ∪ {n}:
                                        if rela[3] > wd then wd := rela[3] end if:
                                end if;
                        end do:
                end do:
                if nops(Nd) > 0 then de:=[d, Nd, wd]: end if:
        docs:= docs ∪ {de}:
        end do:
        if nops(docs) > 0 then
                rs := [e, rd]: results:= results ∪ {rs}:
        end if:
end do:
```

Ví dụ: trong phạm vi kiến thức lập trình, người dùng mong muốn truy vấn về định nghĩa array (mảng) một chiều. Người dùng có thể đặc tả yêu cầu trên theo quy ước ([7, tr.3]) như sau:
Trong đó: SEARCH: là *Từ khóa* chỉ yêu cầu tìm kiếm; MANG_MOT_CHIEU: là yếu tố tri thức cần

```
SEARCH:      MANG_MOT_CHIEU.DINH_NGHIA
```

tìm. Thuật giải 1 sẽ xử lý và trả về kết quả truy vấn kèm theo các trích dẫn theo bảng dưới đây:

Bảng 1. Bảng minh họa kết quả của chương trình theo thuật giải 1

Results = {rs1}					
rs_i	**e**	**D_e**			
		Ebook d	**Node n**	**w**	**contents of node n**
rs_1	MANG_MOT_CHIEU. DINH_NGHIA	Nhập môn lập trình TG: Trần Đan Thư	Chương 7 Dữ liệu Mảng và Chuỗi ký tự II. Mảng một chiều	1.0	Mảng một chiều đơn giản là dãy của nhiều phần tử giống nhau. Một cuộn phim chụp ảnh có 36 kiểu chính là mảng một chiều gồm 36 phần tử, là những tấm phim có cùng kích thước. Thao tác trên mảng một chiều (cũng như mảng nhiều chiều), với ngôn ngữ C/C++, là rất linh động. Tuy nhiên, ở đây chúng ta chỉ khảo sát ác định tính cơ bản của kiểu dữ liệu này và người đọc dễ dàng nhận thấy nét tương đồng trong một số ngôn ngữ lập trình cấp cao như C#, Java, Pascal.
			Chương 7 Dữ liệu Mảng và Chuỗi ký tự II.1. Tạo mảng một chiều	0.6	Để tạo ra một mảng tĩnh, chúng ta sẽ sử dụng câu lệnh với ba thông tin cần thiết sau: Kiểu của mỗi phần tử được lưu trữ trong mảng. Tên của biến (mảng) Số lượng các phần tử (hay kích thước) của mảng.
		Kỹ thuật lập trình	Chương 5 Cấu trúc dữ	1.0	Mảng là cấu trúc dữ liệu cho phép quản lý một danh sách hữu hạn các dữ liệu

Results = {rs_1}					
rs_i	e	D_e			
		Ebook d	**Node n**	**w**	**contents of node n**
		cơ sở với C/C++ TG: Dương Thăng Long	liệu mảng và con trỏ 5.1 Cấu trúc dữ liệu mảng		cùng kiểu và có thứ tự. Mảng được cấp cấp hầu hết trong các ngôn ngữ lập trình.
		Kỹ thuật lập trình C TG: Phạm Văn Ất	Chương 2 Hằng, Biến, Mảng Bài 5 Khái niệm về mảng, cách khai báo	0.6	Mảng có thể hiểu là một tập hợp nhiều phần tử có cùng một kiểu giá trị và có chung một tên. Mỗi phần tử mảng có vai trò như một biến và chứa được một giá trị
		Kỹ thuật lập trình C TG: Phạm Văn Ất	Chương 2 Hằng, Biến, Mảng Bài 9 Biến, mảng tự động	0.2	Các biến (mảng) khai báo bên trong thân của một hàm kể cả hàm main gọi là biến (mảng) tự động hay cục bộ.

4. CÀI ĐẶT THỬ NGHIỆM VÀ SO SÁNH - ĐÁNH GIÁ

Kết quả nghiên cứu của bài báo được vận dụng vào thử nghiệm trên miền tri thức về lập trình trên máy tính, các tài liệu cũng được thu thập và được biểu diễn là các giáo trình của các tác giả là chuyên gia trong lĩnh vực, đang được sử dụng giảng dạy tại các trường đại học trong nước.

4.1. Thiết kế cơ sở tri thức tích hợp

Thành phần KB được thiết kế và tổ chức lưu trữ theo định nghĩa 2.1, 2.3, gồm các thành phần tri thức như: tập các tri thức khái niệm; tập các tri thức quan hệ giữa các khái niệm; tập các tri thức dạng hàm; tập các tri thức dạng luật; tập các dạng bài tập; tập các phương pháp giải cho các bài tập.

Thành phần KE được thiết kế và tổ chức lưu trữ theo định nghĩa 2.2, 2.3, gồm các bộ tài liệu Ebook liên quan đến phạm vi lập trình trên máy tính gồm [13-16].

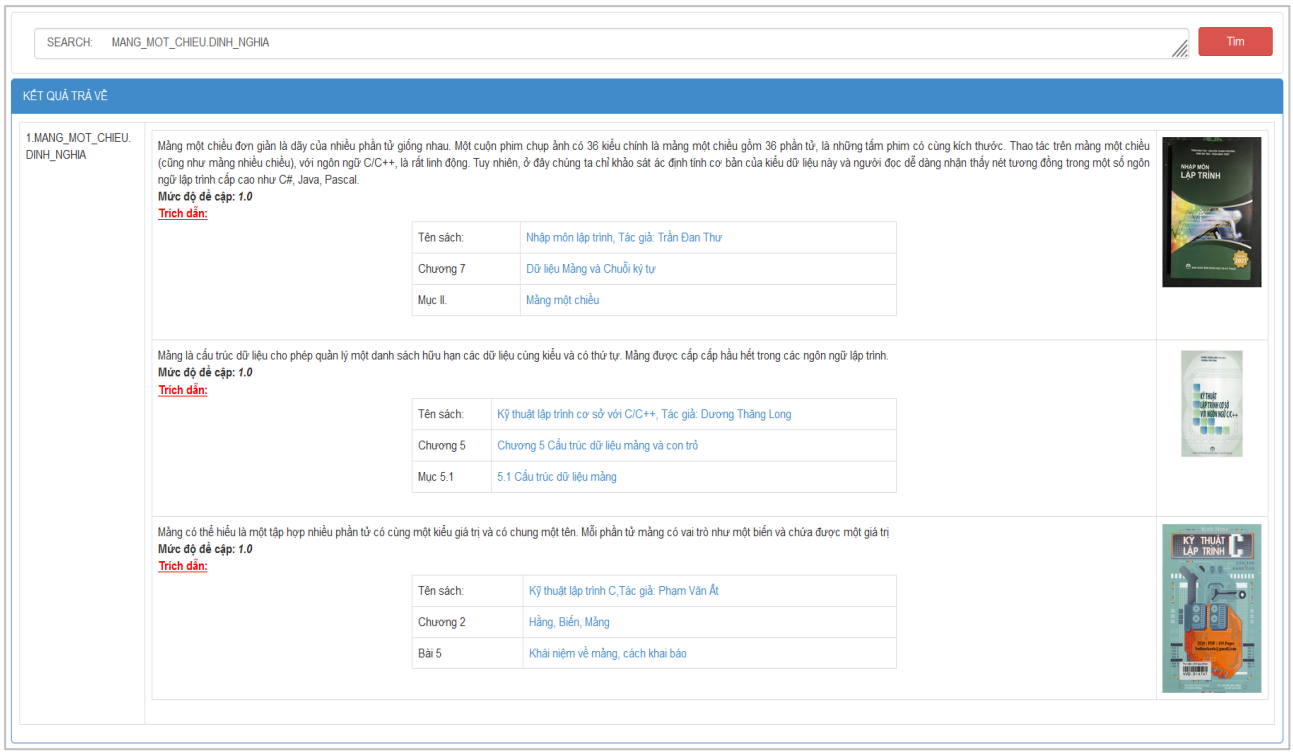
Thành phần liên kết R_{LINK} được thiết kế và tổ chức lưu trữ theo định nghĩa 2.3, dựa trên các ý kiến đóng góp của các chuyên gia là các giảng viên đang theo dạy các học phần về lập trình trên máy tính như môn Nhập môn lập trình, Cấu trúc dữ liệu và Giải thuật, Lập trình hướng đối tượng.

4.2. Thiết kế các module

Các module chính của hệ thống gồm module chính như: module xử lý câu truy vấn; module truy tìm tri thức từ câu truy vấn; module xử lý các kết quả trả về. Trong đó module xử lý câu truy vấn và động cơ truy tìm tri thức sẽ được viết bằng mã nguồn lập trình Maple trên cơ sở tri thức tích hợp. Module thực hiện xử lý và hiển thị kết quả chính sẽ được thiết kế bằng mã nguồn lập trình C# và công nghệ ASP.NET MCV.

4.3 Kết quả thử nghiệm

Đề tài cũng đã vận dụng và cài đặt thành công chương trình dưới dạng web application, bên dưới đây là một phần giao diện của hệ thống.



Hình 3. Giao diện kết quả của hệ thống truy vấn tri thức với yêu cầu "SEARCH: MANG_MOT_CHIEU.DINH_NGHIA"

Đề tài đã thử nghiệm trên một số dạng câu truy vấn đơn giản về tri thức khái niệm, tri thức dạng hàm, tri thức về dạng bài tập và phương pháp giải. Kết quả thử nghiệm được tổng hợp theo bảng tóm tắt sau đây:

Bảng 2. Tóm tắt minh họa kết quả thử nghiệm của chương trình

STT	Các yêu cầu	Số lượng
1	Tri vấn tri thức loại khái niệm	35
2	Truy vấn tri thức là tri thức hàm	20
3	Truy vấn tri thức là tri thức dạng bài tập	15
4	Truy vấn tri thức là tri thức dạng phương pháp giải	15
Tổng cộng		**85**

4.4. So sánh và đánh giá

Trong phần này đề tài cũng đã thực hiện so sánh thử nghiệm với một số hệ thống có cùng nhóm chức năng hỗ trợ truy vấn tri thức, có thể phân thành các nhóm sau đây:

Nhóm các hệ thống [3-8] cho phép truy vấn kiến thức trên một số miền tri thức như các kiến thức về lập trình, các kiến thức về toán cấp trung học phổ thông. Nhóm hệ thống hỗ trợ truy vấn với các yêu cầu đa dạng như truy tìm tri thức theo sự phân loại, thực hiện so sánh các tri thức khái niệm. Hệ thống này cho phép giao tiếp tốt với người dùng bằng ngôn ngữ tự nhiên tiếng Việt. Kết quả xử lý và trả về phù hợp và có độ chính xác cao.

Nhóm hệ thống AI - ChatGPT, đây là nhóm ứng dụng được đánh giá có hỗ trợ chức năng tìm kiếm theo nội dung ngữ nghĩa của yêu cầu được nhập vào từ người dùng, trên nhiều lĩnh vực. Hệ thống hỗ trợ giao tiếp tốt bằng đa ngôn ngữ gồm cả tiếng Anh lẫn tiếng Việt. Nhóm ứng dụng cũng được người sử dụng đánh giá cao về khả năng hiểu và trả lời của hệ thống.

Nhóm ứng dụng [2, 11] cho phép tìm kiếm các tài liệu có liên quan đến câu truy vấn được nhập vào. Nhóm ứng dụng có ưu điểm là cho phép biểu diễn đa dạng tài liệu gồm các bài báo, ebook, và các kho tài nguyên khác.

So với một số nhóm ứng dụng trên, hệ thống được trình bày có những điểm hạn chế như chỉ hỗ trợ người sử dụng truy vấn tri thức dựa trên ngôn ngữ quy ước, chưa hỗ trợ được xử lý các yêu cầu được diễn đạt bằng ngôn ngữ tự nhiên. Tuy nhiên, hệ thống cũng có những ưu điểm khi xử lý được các yêu cầu có độ chính xác tốt hơn so với nhóm hệ thống [3-8], bên cạnh đó, các kết quả trả về của hệ thống cũng có độ tin cậy cao khi có thể trích dẫn các nguồn tài liệu một cách chi tiết hơn so với [2, 11] và ChatGPT, tham khảo *Hình 3. Giao diện kết quả của hệ thống truy vấn.*

5. KẾT LUẬN

Từ sự kế thừa và hiệu chỉnh một số kết quả đã có, bài báo đã trình bày một giải pháp trong việc tích hợp mô hình cơ sở tri thức tựa COKB và một phương pháp biểu diễn nội dung tài liệu là các Ebook. Từ cơ sở mô hình tích hợp, nghiên cứu cũng đã đưa ra một số lớp bài toán, đặc biệt là lớp bài toán truy vấn tri thức với yêu cầu trả về các tri thức được diễn đạt theo quy ước câu truy vấn kèm theo các trích dẫn tài liệu. Đồng thời đề xuất các thuật giải để giải quyết lớp bài toán truy vấn tri thức cùng với các trích dẫn của tri thức trong các tài liệu Ebook.

Bài báo cũng đã thực hiện việc cài đặt thành công một ứng dụng, có hỗ trợ chức năng truy vấn tri thức trên miền tri thức về lập trình trên máy tính. Hệ thống cho phép người dùng nhập vào các yêu cầu được diễn đạt qua các quy ước đơn giản. Hệ thống cũng đã xử lý và trả về được các các kết quả là các tri thức phù hợp với yêu cầu theo câu truy vấn, bên cạnh đó các tri thức được trả về cho người dùng cũng được chỉ rõ trích dẫn từ các tài liệu liên quan.

Dù đã có những điểm mới so với một số kết nghiên cứu trước đây, tuy nhiên phương pháp chỉ mới là kết quả khởi đầu, chưa khai thác hết các tiềm năng của hướng tiếp cận biểu diễn nội dung bằng các đồ thị keyphrases. Việc ước lượng bởi chuyên gia về mức độ liên qua của tri thức và các nút trong cây chương mục của mỗi tài liệu còn mang tính thủ công, điều này gây hạn chế đến khả năng phát triển rộng của hệ thống. Trong nghiên cứu tiếp theo, đề tài sẽ tập trung nghiên cứu hoàn thiện các kỹ thuật còn giới hạn, đồng thời xem xét bài toán rút trích các keyphrase tự động để tăng cường tính khả thi trong quá trình triển khai nhân rộng vào các phạm vi tri thức hay lĩnh vực khác.

LỜI CẢM ƠN

Nghiên cứu này được Trường Đại học Quốc tế Hồng Bàng cấp kinh phí thực hiện dưới mã số đề tài GVTC17.30.

TÀI LIỆU THAM KHẢO

[1] N. V. Do, "Các hệ thống trí tuệ nhân tạo ứng dụng trong giáo dục," *Hong Bang International University Journal of Science,* 2023.

[2] ThanhThuong T. Huynh, et al, "A Method for Designing Domain-Specific Document Retrieval Systems using Semantic Indexing," *International Journal of Advanced Computer Science and Applications,* vol. 10, no. 10, 2019.

[3] Nhon V. Do, et al, "Intelligent Educational Software in Discrete Mathematics and Graph Theory," in *International Conference on Intelligent Software Methodologies, Tools, and Techniques*, Spain, 2018.

[4] Hien D. Nguyen, et al, "Design an Ontology-based model for Intelligent Querying system in Mathematics Education," *Journal of Interdisciplinary Mathematics,* vol. 26, no. 3, pp. 449-473, 2023.

[5] Xuan-Thien Pham, et al, "Build a search engine for the knowledge of the course about Introduction to Programming based on ontology Rela-model," in *International Conference on Knowledge and Systems Engineering*, Can Tho, 2022.

[6] T. T. Mai, et al, "A Knowledge-Based Model for Designing the Knowledge Querying System in Education," in *International Conference on Research, Innovation and Vision for the Future* , Ho Chi Minh, 2022.

[7] Nhon D. Van, et al, " knowledge representation model for designing the knowledge querying system in Programming Language C/C++," in *RIVF International Conference on Computing and Communication Technologies (RIVF)*, HaNoi, 2023.

[8] Hien D. Nguyen, et al, "Some Techniques for Intelligent Searching on Ontology-based Knowledge Domain in e-Learning," in *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2020.

[9] S. Ekin, "Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips,," Texas A&M University, 2023.

[10] Wayne Xin Zhao, et al, "A Survey of Large Language Models," in arXiv:2303.18223, 2023.

[11] ThanhThuong T. Huynh, et al, "A semantic document retrieval system with semantic search technique based on knowledge base and graph representation," in *in Proceedings of The 17th International Conference on New Trends in Intelligent Software Methodologies, Tools, and Techniques, IOS Press*, 2018.

[12] Nhon Do, et al, "Perfect COKB Model and Reasoning Methods for the design of Intelligent Problem Solvers," in *International Conference on Intelligent Software Methodologies, Tools, and Techniques*, Japan, 2017.

[13] Trần Đan Thư, et al, Nhập môn lập trình, NXB: Khoa học kỹ thuật, 2018.

[14] Mai Tiến Dũng, et al, Nhập môn lập trình, NXB: ĐH QG, 2021.

[15] Phạm Văn Ất, et al, Kỹ thuật lập trình C, NXB: BK Hà Nội, 2021.

[16] Dương Thăng Long, Kỹ thuật lập trình cơ sở, NXB: Khoa học kỹ thuật, 2015.