

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: XÂY DỰNG HỆ THỐNG TRA CỨU KIẾN THỨC
MÔN CẤU TRÚC DỮ LIỆU DỰA TRÊN ONTOLOGY QUERY-ONTO

Giảng viên hướng dẫn: TRẦN PHONG NHÃ

Sinh viên thực hiện: NGUYỄN TRUNG KIÊN

Mã sinh viên: 5851071038

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 58

Tp. Hồ Chí Minh, năm 2021

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: XÂY DỰNG HỆ THỐNG TRA CỨU KIẾN THỨC
MÔN CẤU TRÚC DỮ LIỆU DỰA TRÊN ONTOLOGY QUERY-ONTO

Giảng viên hướng dẫn: TRẦN PHONG NHÃ

Sinh viên thực hiện: NGUYỄN TRUNG KIÊN

Mã sinh viên: 5851071038

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 58

Tp. Hồ Chí Minh, năm 2021

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP
BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-----***-----

Mã sinh viên:.....5851071038 **Họ tên SV:**.....Nguyễn Trung Kiên

Khóa:58 **Lớp:**Công nghệ thông tin

1. Tên đề tài:

Xây dựng hệ thống tra cứu kiến thức môn cấu trúc dữ liệu dựa trên Ontology Query-
Onto

2. Mục đích, yêu cầu:

Thiết kế và xây dựng thành công hệ thống tra cứu kiến thức môn cấu trúc dữ liệu. Yêu cầu hiểu được mô hình Ontology Query-Onto, áp dụng được mô hình đó.

3. Nội dung và phạm vi đề tài:

Sử dụng mô hình Ontology query-onto kết hợp với trường kiến thức cấu trúc dữ liệu để xây dựng nên hệ thống. Trường kiến thức để xây dựng hệ thống tra cứu là kiến thức cấu trúc dữ liệu trong bộ môn cấu trúc dữ liệu và giải thuật.

4. Công nghệ, công cụ và ngôn ngữ lập trình:

Phương pháp biểu diễn tri thức Ontology nói chung và Query-Onto nói riêng, .NET MVC, C#, Html, Css, Visual studio, Dialogflow.

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng:

Xây dựng thành công trang web hỗ trợ việc tra cứu kiến thức môn cấu trúc dữ liệu có tích hợp chatbot. Giúp cho sinh viên nói chung và sinh viên công nghệ thông tin nói riêng có thể tra cứu kiến thức một cách nhanh chóng và đơn giản. Ngoài ra, khi hiểu rõ được mô hình này, sẽ dễ dàng mở rộng ra với những trường kiến thức khác.

6. Giáo viên và cán bộ hướng dẫn

Họ tên:Trần Phong Nhã

Đơn vị công tác: Trường ĐH Giao thông vận tải phân hiệu tại Thành phố Hồ Chí Minh

Điện thoại:.....0906761014.....Email:..... tpnha@utc2.edu.vn.....

Ngày tháng 07 năm 2021

Đã giao nhiệm vụ TKTN

Trưởng BM Công nghệ Thông tin

Giáo viên hướng dẫn

ThS. Trần Phong Nhã

Đã nhận nhiệm vụ TKTN

Sinh viên: Nguyễn Trung Kiên

Ký tên:

Điện thoại: 0975522206

Email: 5851071038@utc2.edu.vn

LỜI CẢM ƠN

Trong thời gian thực hiện đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Em xin gửi lời cảm ơn chân thành đến Ths. Trần Phong Nhã, giảng viên Bộ môn Công nghệ thông tin – trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh đã hướng dẫn, chỉ bảo em trong suốt quá trình làm khóa luận.

Em cũng xin chân thành cảm ơn các thầy cô giảng viên trong trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh nói chung, các thầy cô trong Bộ môn Công nghệ thông tin nói riêng đã dạy dỗ em kiến thức từ các môn cơ bản cho đến các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập. Để ngày hôm nay, em đã có thể hoàn thành đồ án với đề tài:

“Xây dựng hệ thống tra cứu kiến thức môn Cấu trúc dữ liệu dựa trên
Ontology Query-Onto”

Cuối cùng, em xin chân thành cảm ơn gia đình, bạn bè luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình thực hiện đồ án tốt nghiệp

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm

Giảng viên hướng dẫn

Trần Phong Nhã

MỤC LỤC

CHƯƠNG 1: MỞ ĐẦU	1
CHƯƠNG 2: THIẾT KẾ HỆ THỐNG TRA CỨU KIẾN THỨC MÔN HỌC	4
2.1. Kiến trúc hệ thống tra cứu kiến thức môn học	4
2.1.1. Tổ chức cơ sở tri thức cho môn cấu trúc dữ liệu	4
2.1.2. Hệ thống tra cứu kiến thức các môn học	4
2.2. Cơ sở tri thức biểu diễn hệ thống tra cứu kiến thức môn học	6
2.2.1. Ontology tri thức quan hệ Rela-model	6
2.2.2. Ontology biểu diễn tri thức cho hệ thống tra cứu kiến thức môn học	7
2.2.3. Cấu trúc thành phần tri thức (Exer, proof)	9
2.3. Các vấn đề tra cứu dựa trên ontology biểu diễn kiến thức môn học	10
2.3.1. Vấn đề tra cứu nội dung kiến thức môn học	10
CHƯƠNG 3: XÂY DỰNG MÔ HÌNH CƠ SỞ DỮ LIỆU	16
3.1. Tổ chức cơ sở tri thức chatbot	16
3.1.1. Mô hình cơ sở dữ liệu	16
3.1.2. Mô tả các bảng dữ liệu	16
3.1.3. Một số ràng buộc về dữ liệu	18
3.2. Thiết kế chatbot	20
3.2.1. Cách chatbot được huấn luyện	20
3.2.2. Giới thiệu Dialogflow	22
3.2.3. Một số khái niệm cơ bản trong Dialogflow	22
3.2.4. Xây dựng chatbot cho trang web	24
CHƯƠNG 4: KẾT QUẢ THỬ NGHIỆM	31
4.1. Kết quả tra cứu theo nội dung kiến thức	31
4.2. Kết quả tra cứu theo sự phân loại kiến thức	31
4.3. Kết quả tra cứu theo sự phân loại kiến thức của chatbot	33
4.4. Kết quả khuyến nghị các kiến thức liên quan	34
CHƯƠNG 5 : TỔNG KẾT	35

DANH MỤC HÌNH ẢNH

Hình 1. Kiến trúc hệ thống tra cứu kiến thức môn học.	4
Hình 2. Kết quả tìm kiếm nội dung kiến thức cho nội dung “danh sách liên kết”	5
Hình 3. Khuyến nghị kiến thức liên quan với nội dung đang được tra cứu	6
Hình 4. Lưu đồ thuật toán cho thuật giải 4.1	12
Hình 5. Lưu đồ thuật toán cho thuật giải 4.2	15
Hình 6. Mô hình cơ sở dữ liệu cho hệ thống	16
Hình 7. Các từ khóa được phân theo các entities.....	26
Hình 8. Các từ khóa được hệ thống xác định.....	26
Hình 9. Giao diện làm việc để thêm Entities tương ứng với các từ khóa.....	27
Hình 10. Ví dụ tạo Entity cho từ khóa đại diện cho CTDL.....	28
Hình 11. Quản lý hoặc chỉnh sửa các entity đã tạo ra	28
Hình 13. Training phrases.....	29
Hình 12. Action and parameters	29
Hình 14. Responses.....	30
Hình 15. Kết quả khi tra cứu theo nội dung kiến thức với từ khóa "dslk"	31
Hình 16. Phân các loại kiến thức để tra cứu	31
Hình 17. Kết quả tra cứu thành phần hoạt động cho từ khóa mảng	32
Hình 18. Kết quả khi thực hiện tìm kiếm khái niệm "mảng động"	33
Hình 19. Khuyến nghị kiến thức liên quan cho "Cây nhị phân"	34

DANH MỤC BẢNG BIỂU

Bảng 1. Cơ sở tri thức cấu trúc dữ liệu	8
Bảng 2. Thuật giải 4.1	11
Bảng 3. Thuật giải 4.2	14
Bảng 4. Mô tả bảng Lesson.....	16
Bảng 5. Mô tả bảng Construct	16
Bảng 6. Mô tả bảng Concept.....	17
Bảng 7. Mô tả bảng Operator	17
Bảng 8. Mô tả bảng Relationship	17
Bảng 9. Mô tả bảng Exercise	18

CHƯƠNG 1: MỞ ĐẦU

1.1. Giới thiệu

Trong cuộc cách mạng công nghiệp lần thứ tư, e-learning (hay học tập thông minh) là việc học thông qua các phương tiện điện tử và ứng dụng công nghệ thông tin trong giáo dục. Dễ thấy trong những năm gần đây, việc học và dạy học trực tuyến không còn là vấn đề xa lạ. Không những thế, để đạt được hiệu quả của việc học tập thông minh, hệ thống hỗ trợ tra cứu, tìm kiếm kiến thức là công cụ cần thiết để người học có thể truy xuất các nội dung môn học cần thiết cho quá trình học, giúp người học có thể tự ôn tập lại các kiến thức đã được học cũng như tìm hiểu về kiến thức sắp được học. Hệ thống này được xây dựng dựa trên nền tảng cơ sở tri thức được tổ chức để biểu diễn nội dung môn học một cách đầy đủ. Bên cạnh đó, hệ thống tra cứu kiến thức môn học cần phải đáp ứng được các yêu cầu việc tra cứu hỗ trợ học tập. Tra cứu nội dung theo các chung và bài học của môn học, tra cứu các khái niệm, tính chất, các dạng bài tập thông dụng cũng như phương pháp giải tương ứng. Để giúp người học có thể nắm vững kiến thức đang tìm hiểu, hệ thống có thể tự động khuyến nghị các kiến thức liên quan đến nội dung đang được tra cứu.

Ontology là một phương pháp hữu ích để tổ chức cơ sở tri thức cho các hệ thống giáo dục thông minh. Relational-model là ontology biểu diễn tri thức quan hệ và đã được ứng dụng trong việc biểu diễn các miền tri thức trong giáo dục và y khoa [3, 4]. Ontology COKB (Computational Object Knowledge Base) biểu diễn tri thức về các đối tượng tính toán đã được sử dụng để biểu diễn cho các miền tri thức về Toán rời rạc và Lý thuyết đồ thị [4]. Tuy nhiên, các kết quả này chỉ hiệu quả cho việc thiết kế các hệ thống hỗ trợ giải bài tập tự động của môn học. Ontology Search-Onto được nghiên cứu và ứng dụng để xây dựng hệ thống tìm kiếm kiến thức trong môn Toán bậc Trung học phổ thông [3]. Tuy nhiên, mô hình Search-Onto mang tính đặc thù chỉ áp dụng được cho kiến thức Toán bậc trung học phổ thông và không áp dụng cho môn học khác.

Trong nghiên cứu này, một mô hình ontology biểu diễn cho miền kiến thức của môn học được nghiên cứu gọi là Query-Onto. Cấu trúc Query-Onto được cải tiến dựa trên cấu trúc của ontology tri thức quan hệ Rela-model [5]. Sự cải tiến này được nghiên cứu ở hai khía cạnh:

- Cải tiến cấu trúc thành phần khái niệm trong mô hình để hỗ trợ việc truy xuất nội dung kiến thức trong quá trình tra cứu.
- Tích hợp thành phần tri thức biểu diễn các dạng bài tập và phương pháp giải. Bên cạnh đó, một số kỹ thuật tra cứu kiến thức dựa trên ontology Query-onto cũng được nghiên cứu: tra cứu nội dung kiến thức, tra cứu theo phân loại kiến thức và khuyến nghị các kiến thức liên quan. Ontology Query -Onto và các kỹ thuật tra cứu kiến thức được ứng dụng để xây dựng hệ thống tra cứu kiến thức các môn học Cấu trúc dữ liệu. Hệ thống này có thể đáp ứng các yêu cầu tra cứu kiến thức của hệ thống hỗ trợ học tập.

1.2. Mục tiêu nghiên cứu

Trong nghiên cứu này, một mô hình ontology biểu diễn cho miền kiến thức của môn học được nghiên cứu gọi là Query-Onto. Cấu trúc Query-Onto được cải tiến dựa trên cấu trúc của ontology tri thức quan hệ Rela-model [5]. Sự cải tiến này được nghiên cứu ở hai khía cạnh:

- Cải tiến cấu trúc thành phần khái niệm trong mô hình để hỗ trợ việc truy xuất nội dung kiến thức trong quá trình tra cứu.
- Tích hợp thành phần tri thức biểu diễn các dạng bài tập và phương pháp giải. Bên cạnh đó, một số kỹ thuật tra cứu kiến thức dựa trên ontology Query-onto cũng được nghiên cứu: tra cứu nội dung kiến thức, tra cứu theo phân loại kiến thức và khuyến nghị các kiến thức liên quan.

1.3. Phạm vi

Kiến thức được nghiên cứu để áp dụng hệ thống được trình bày trong nghiên cứu này là kiến thức môn cấu trúc dữ liệu và giải thuật, được giới hạn lại là kiến thức về cấu trúc dữ liệu. Các hoạt động bao gồm tra cứu nội dung kiến thức, tra cứu theo phân loại kiến thức và khuyến nghị các kiến thức liên quan được áp dụng

theo mô hình trên. Ontology Query -Onto và các kỹ thuật tra cứu kiến thức được ứng dụng để xây dựng hệ thống tra cứu kiến thức các môn học Cấu trúc dữ liệu. Hệ thống này có thể đáp ứng các yêu cầu tra cứu kiến thức của hệ thống hỗ trợ học tập.

1.4. Cấu trúc báo cáo thực tập tốt nghiệp

1.4.1. Chương 1: Mở đầu

1.4.2. Chương 2: Thiết kế hệ thống tra cứu kiến thức môn học

1.4.3. Chương 3: Xây dựng mô hình cơ sở dữ liệu

1.4.4. Chương 4: Kết quả thử nghiệm

1.4.5. Chương 5: Tổng kết

CHƯƠNG 2: THIẾT KẾ HỆ THỐNG TRA CỨU KIẾN THỨC MÔN HỌC

2.1. Kiến trúc hệ thống tra cứu kiến thức môn học

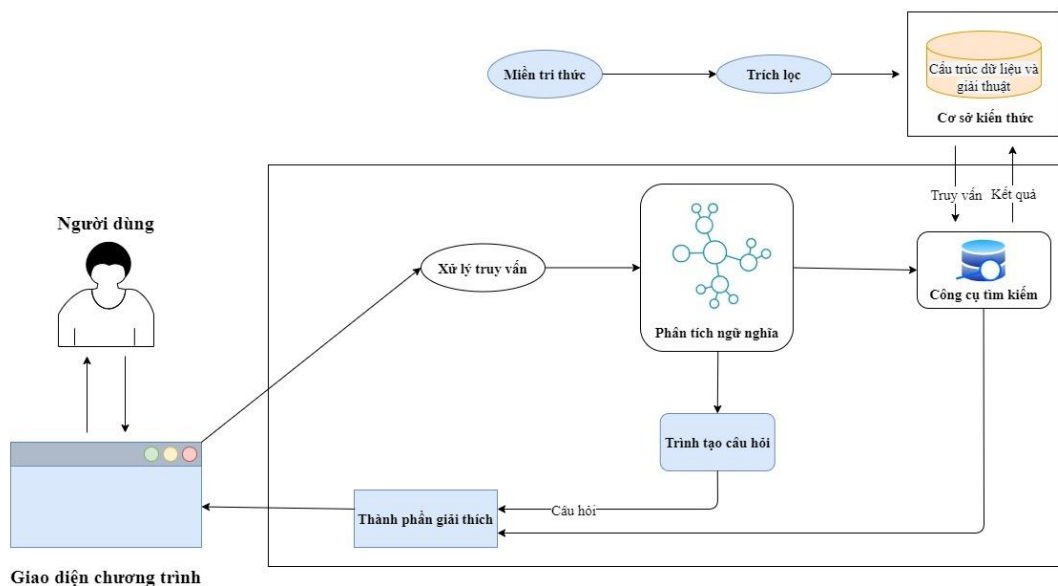
2.1.1. Tổ chức cơ sở tri thức cho môn cấu trúc dữ liệu

Ứng dụng ontology Query-Onto để biểu diễn kiến thức môn cấu trúc dữ liệu. Từ đó, dựa trên kỹ thuật giải quyết các vấn đề về tra cứu kiến thức trong chương 3, hệ thống tra cứu nội dung kiến thức cho các môn học cũng được xây dựng.

Kiến thức môn học cấu trúc dữ liệu và giải thuật được thu thập từ [6, 7, 8]. Miền tri thức này được biểu diễn bằng mô hình Query-Onto.

2.1.2. Hệ thống tra cứu kiến thức các môn học

Hệ thống tra cứu kiến thức môn học được xây dựng trên nền tảng web có thể hỗ trợ việc tìm kiếm, tra cứu nội dung môn học Cấu trúc dữ liệu. Cơ sở tri thức của hệ thống được tổ chức gồm cơ sở tri thức cho nội dung môn học Cấu trúc dữ liệu theo ontology Query-Onto như chương 4 mục 4.1. Động cơ thực hiện việc tra cứu kiến thức của hệ thống được thiết kế dựa trên các thuật giải giải quyết các vấn đề tra cứu kiến thức đã được trình bày trong chương 3. Kiến trúc của hệ thống tra cứu kiến thức như Hình 1.



Hình 1. Kiến trúc hệ thống tra cứu kiến thức môn học.

Chương trình có thể giúp tra cứu nội dung các môn học đó một cách nhanh chóng. Chương trình hỗ trợ việc tra cứu nội dung theo từng bài học, cũng như tra cứu theo phân loại của kiến thức. Bên cạnh đó, hệ thống được xây dựng có thể khuyến nghị các kiến thức liên quan với nội dung đang được tra cứu. Hình 2 là giao diện của chương trình khi tra cứu theo từ khóa. Sau khi truy xuất kết quả nội dung tra cứu, hệ thống sẽ khuyến nghị các kiến thức liên quan đến nội dung đang được người dùng tra cứu (Hình 3). Trong kiến trúc này, việc xây dựng, thiết kế cơ sở tri thức biểu diễn được hệ thống tra cứu kiến thức môn học và động cơ để tìm kiếm, tra cứu kiến thức đóng vai trò hết sức quan trọng.

LEARNICORN

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

Nhập tên kiến thức

Tìm kiếm

Tìm kiếm theo các thành phần

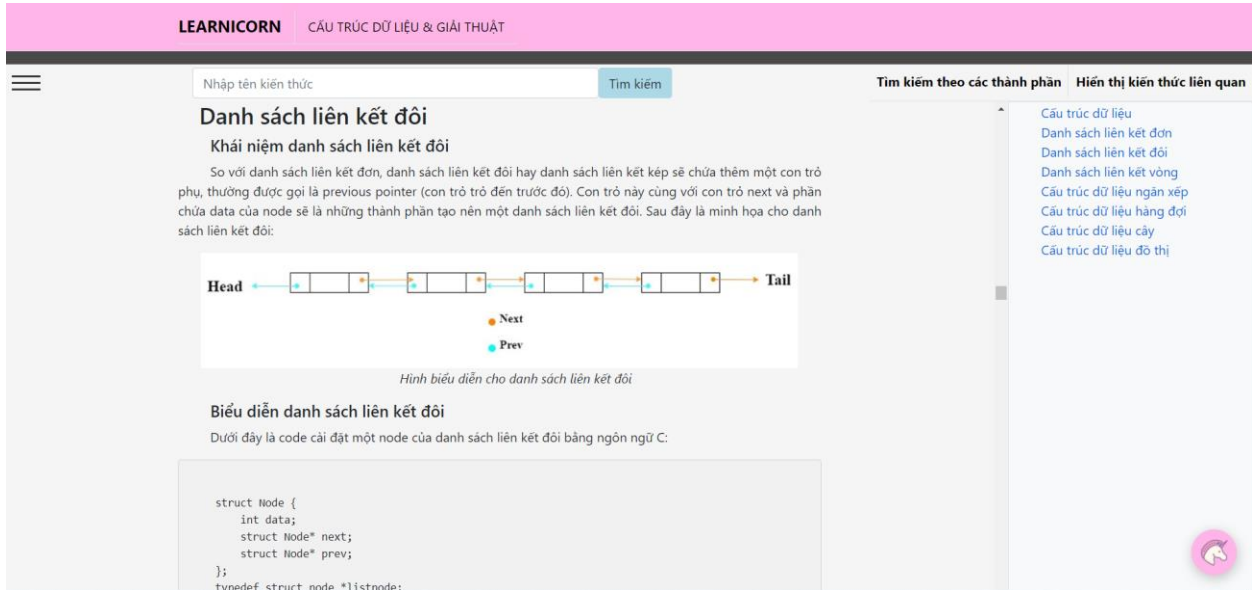
Hiển thị kiến thức liên quan

NỘI DUNG KIẾN THỨC CHO TỪ KHÓA "danh sách liên kết"

Bài học	Khái niệm	Khởi tạo	Toán tử	Bài tập
Danh sách liên kết đơn	Khái niệm cấu trúc dữ liệu danh sách liên kết đơn	Biểu diễn danh sách liên kết đơn	Hoạt động trên danh sách liên kết đơn	Bài tập về danh sách liên kết đơn
Danh sách liên kết đôi	Khái niệm cấu trúc dữ liệu danh sách liên kết đôi	Biểu diễn danh sách liên kết đôi	Hoạt động trên danh sách liên kết đôi	Bài tập về danh sách liên kết đôi
Danh sách liên kết vòng	Khái niệm cấu trúc dữ liệu danh sách liên kết vòng	Biểu diễn danh sách liên kết vòng	Hoạt động trên danh sách liên kết vòng	Bài tập về danh sách liên kết vòng

Trở về trang chính

Hình 2. Kết quả tìm kiếm nội dung kiến thức cho nội dung “danh sách liên kết”



2.2. Cơ sở tri thức biểu diễn hệ thống tra cứu kiến thức môn học

2.2.1. Ontology tri thức quan hệ Rela-model

Ontology là phương pháp hữu hiệu để biểu diễn các tri thức quan hệ, đặc biệt là sự liên kết giữa các thành phần kiến thức trong các môn học. Ontology về tri thức quan hệ, còn gọi là Rela-model là mô hình được biểu diễn tri thức quan hệ gồm có 03 thành phần sau:

$$K = (C, R, \text{Rules})$$

Trong đó:

- **C** là tập các khái niệm của miền tri thức. Mỗi khái niệm là một lớp các đối tượng.
- **R** là tập các quan hệ giữa các khái niệm trong tri thức, mỗi quan hệ này là một quan hệ hai ngôi giữa hai khái niệm trong tập **C**.
- **RULES** là một tập các luật của miền tri thức. Các luật trong mô hình này ở dạng luật dẫn và biểu diễn các suy luận từ các mối quan hệ giữa các khái niệm trong tri thức.

Cấu trúc các thành phần trong mô hình Rela-model đã được trình bày như trên. Tuy nhiên, cấu trúc đối tượng của khái niệm trong nghiên cứu này có sự thay đổi như sau: Mỗi đối tượng của một khái niệm có cấu trúc gồm 03 phần:

$$(Attrs, Ops, RulObj, Content, Key),$$

Trong đó:

- *Attrs* là tập hợp các thuộc tính của đối tượng.
- *Ops* là tập hợp các toán tử trên đối tượng.
- *RulObj* là tập hợp các luật dẫn trên một đối tượng.
- *Content* là nội dung của khái niệm.
- *Key* là tập hợp các từ khóa để xác định khái niệm.

2.2.2. **Ontology biểu diễn tri thức cho hệ thống tra cứu kiến thức môn học**

Trong quá trình học tập, bên cạnh việc tra cứu các kiến thức môn học, người dùng cũng cần phải tra cứu các dạng bài tập cũng như các phương pháp giải tương ứng. Vì vậy, đối với cơ sở tri thức của hệ thống tra cứu kiến thức, tri thức cần phải có thành phần biểu diễn về các kiến thức dạng bài tập và phương pháp giải [4]. Các thành phần này cần phải có sự liên kết với nội dung kiến thức của môn học để giúp hệ thống có thể tra cứu các kiến thức một cách hiệu quả hơn. Ontology biểu diễn tri thức cho việc tra cứu kiến thức môn học là sự kết hợp giữa ontology tri thức quan hệ Rela-model, tri thức về các bài tập thông dụng và các phương pháp giải.

Ontology biểu diễn miền kiến thức cho việc tra cứu, gọi là *Query-Onto*, là sự kết hợp giữa ontology tri thức quan hệ và thành phần tri thức về các dạng bài tập. Mô hình Query-Onto như sau:

(C, R, Rules) + (Exer, proof)

Trong đó:

- (C, R, Rules) là ontology biểu diễn tri thức quan hệ Real-model, với các khái niệm trong **C** được cải tiến về cấu trúc thành phần để biểu diễn chính xác hơn ý nghĩa của khái niệm trong miền tri thức. Bảng 1 trình bày một số kiến thức của môn học được biểu diễn bởi 03 thành phần (C, R, Rules)

Bảng 1. Cơ sở tri thức cấu trúc dữ liệu

CẤP	C	R	RULES
$C_{(0)}$	<p>- Các khái niệm cơ bản:</p> <ul style="list-style-type: none"> + KIỂU DỮ LIỆU: Mô tả kiểu của các dữ liệu. + DỮ LIỆU: Các giá trị hoặc tập hợp các giá trị. Tập này có tập thể hiện $I_{\text{Dữ liệu}}$ + TÊN BIẾN: Là tên gọi để gọi các cấu trúc dữ liệu được đặt cho. + ĐỘ LỚN: Số lượng các phần tử dữ liệu. + CHỈ MỤC: Vị trí được đánh số của các phần tử dữ liệu cơ bản trong một cơ sở dữ liệu. + CON TRỎ: CON TRỎ ĐƯỢC TRỎ TỚI CHỈ MỤC ĐỂ LẤY DỮ LIỆU Ở ĐÓ. 	<p>$R = \{\text{is-a, related, created}\}$</p> <ul style="list-style-type: none"> + $c_i \text{ is-a } c_j$: khái niệm c_i là khái niệm c_j, với $c_i, c_j \in C_{(1)}$ * Quan hệ “is-a” có tính chất bắc cầu. <p>Ví dụ: MẢNG ĐÔNG is-a MẢNG</p> <ul style="list-style-type: none"> + $c_i \text{ related } c_j$: khái niệm c_i liên quan đến khái niệm c_j, với $c_i, c_j \in C_{(1)}$ * Quan hệ “related” có tính chất đối xứng. <p>Ví dụ: HÀNG ĐỘI liên quan đến DANH SÁCH LIÊN KẾT</p> <ul style="list-style-type: none"> + $c_i \text{ created } c_j$: khái niệm c_i được tạo bởi khái niệm c_j, với $c_i, c_j \in C_{(1)}$ <p>Ví dụ: NGĂN XẾP được tạo bởi MẢNG</p>	<p>Luật r_1:</p> $\forall c_i, c_j \in C_{(1)}: \{c_i \text{ is-a } c_j\} \rightarrow \{c_j.Attrs \subseteq c_i.Attrs, c_j.Ops \subseteq c_i.Ops, c_j.RulObj \subseteq c_i.RulObj\}$ <p>Luật r_2:</p> $\forall c_i, c_j \in C_{(1)}: \{c_i \text{ created } c_j\} \rightarrow \{c_i \text{ related } c_j\}$ <p>Luật r_3:</p> $\forall c_i, c_j \in C_{(1)}: \{c_i \text{ related } c_j\} \rightarrow \{c_i.Key \text{ related } c_j.Key\}$ <p>Luật r_4:</p> $\forall c_i, c_j, c_k \in C_{(1)}: \{c_i \text{ created } c_j, c_j \text{ is-a } c_k\} \rightarrow \{c_i \text{ created } c_k\}$
$C_{(1)}$	<p>Mỗi khái niệm thuộc $C_{(1)}$ là một lớp các đối tượng. Cấu trúc mỗi khái niệm là một bộ gồm 5 phần:</p> <p>(<i>Attrs, Ops, RulObj, Content, Key</i>)</p> <p>$C_{(1)} = \{\text{MẢNG và MẢNG ĐÔNG, DANH SÁCH LIÊN KẾT và các loại danh sách, NGĂN XẾP, HÀNG ĐỘI, CÂY và các loại cây}\}$</p> <p>Ví dụ: Khái niệm DANH SÁCH LIÊN KẾT $\in C_{(1)}$ có cấu trúc:</p> <p>$Attrs = \{type, head, tail, element\}$</p> <p><i>type</i>: Kiểu dữ liệu.</p> <p><i>head</i>: con trỏ đầu của danh sách</p> <p><i>tail</i>: con trỏ đuôi của danh sách</p>		

<p><i>element</i>: Dãy các giá trị của danh sách</p> <p><i>Ops</i> = {chèn, xóa, duyệt, Lấy giá trị phần tử, tìm kiếm phần tử, cập nhật}: là các toán tử trên khái niệm danh sách</p> <p><i>RulesObj</i> = { }</p> <p><i>Content</i>: Nội dung về lý thuyết của danh sách liên kết đã được thu thập.</p> <p><i>Key</i> = {"danh sách", "liên kết"}</p>		
--	--	--

- EXER là tập hợp các dạng bài tập thông dụng trong kiến thức môn học, các dạng bài này là như bài tập cơ bản và đặc trưng của môn học.
- Proof là tập hợp các phương pháp giải tương ứng cho các dạng bài tập trong tập hợp EXER.

2.2.3. Cấu trúc thành phần tri thức (Exer, proof)

Exer là tập hợp các dạng bài tập thông dụng trong môn học. Cấu trúc của mỗi phần tử trong tập này là một bộ gồm 6 thành phần:

(*Name, Content, Intellectual, Hypothesis, Goal, Example*)

Trong đó:

- *Name*: Thành phần để xác định dạng bài tập. Thành phần này gồm các từ khóa (keyword) hay mã (code) của dạng bài tập.
- *Content*: Thành phần này mô tả nội dung của các bài tập dưới dạng ngôn ngữ tự nhiên.
- *Intellectual*: Thành phần này biểu diễn sự liên kết của dạng bài tập với các thành phần tri thức của môn học được biểu diễn theo Rela-model.
- *Hypothesis*: Thành phần đặc tả các giả thiết của bài tập cũng như các đối tượng cần thiết trong bài tập.
- *Goal*: Thành phần đặc tả các mục tiêu của bài tập.
- *Example*: Danh sách một số bài tập ví dụ cho dạng bài tập tương ứng.

proof là tập hợp các phương pháp giải cho các dạng bài tập tương ứng. Mỗi phương pháp sẽ được sử dụng để giải quyết một dạng bài tập tương ứng. Cấu trúc một phương pháp giải gồm các thành phần sau:

(Name, Idea, Exercise, Solution, Example)

Trong đó:

- *Name*: Thành phần để xác định phương pháp giải. Thành phần này có thể gồm các từ khóa (keyphrase) hay mã (code) của dạng bài tập.
- *Idea*: Thành phần này mô tả ý tưởng của phương pháp giải.
- *Solution*: Thành phần đặc tả chi tiết nội dung các bước giải trong phương pháp.
- *Example*: Danh sách các bài giải ví dụ cho các bài tập tương ứng cũng trong thành phần *Example* của Exer .

2.3. Các vấn đề tra cứu dựa trên ontology biểu diễn kiến thức môn học

Hệ thống tra cứu kiến thức thông minh cần phải đáp ứng được các yêu cầu về tra cứu các thành phần kiến thức như các khái niệm, các quan hệ trong miền kiến thức và khuyến nghị các kiến thức liên quan với nội dung đang được tìm kiếm. Bên cạnh đó, hệ thống tra cứu kiến thức thông minh hỗ trợ học tập còn phải hỗ trợ được việc tra cứu kiến thức theo các nội dung bài học, các tính chất, định lý, cũng như các dạng bài tập thông dụng và phương pháp giải để giải các bài tập đó.

2.3.1. Vấn đề tra cứu nội dung kiến thức môn học

Trong vấn đề này, việc tra cứu nội dung kiến thức môn học gồm có hai loại:

- Tra cứu nội dung theo chương và bài học
- Tra cứu nội dung theo nhóm kiến thức thông qua các từ khóa đã được xác định.

Việc xác định thông tin truy xuất phù hợp với nội dung kiến thức được thực hiện dựa trên cấu trúc kiến thức môn học được tổ chức theo ontology Query-Onto.

Cho K là một miền tri thức có dạng ontology Query-Onto hoặc K là một thành phần của ontology Search-Onto, khi đó ta kí hiệu:

- $Com(K)$ là tập hợp các thành phần của K .
- $keyword(o)$ là tập hợp các từ khóa của đối tượng o .

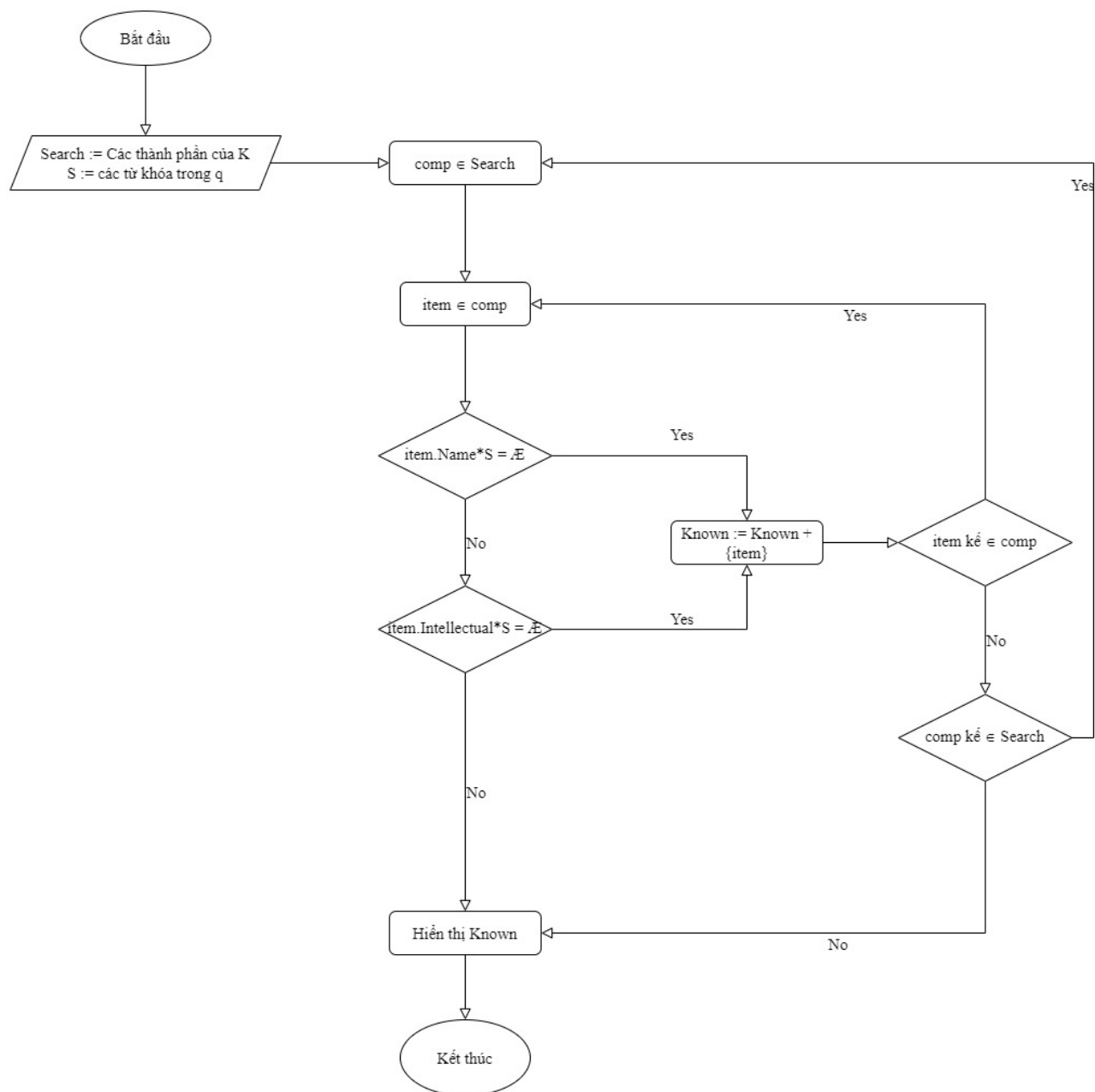
Cho miền tri thức K có dạng ontology Query-Onto, và nội dung tra cứu q .

Thuật giải xác định tập hợp các kiến thức phù hợp với nội dung tra cứu q .

Bảng 2. Thuật giải 4.1

Thuật giải 4.1: <i>Xác định nội dung kiến thức</i>
Input: Miền tri thức K có dạng ontology Query-Onto. Nội dung tra cứu q . Output: Tập hợp các kiến thức trong K cho tra cứu q .
Bước 1: Khởi tạo $S := keyword(q)$. //các từ khóa có trong q . $Known := \{\}$ // tập hợp các kết quả trả về. $Search := Com(K)$ Bước 2: Xác định thành phần khái niệm For $comp$ in $Search$ do For $item$ in $comp$ do If ($item.Name * S \neq \emptyset$) then $Known := Known + \{item\}$ Else if ($item.Intellectual * S \neq \emptyset$) then $Known := Known + \{item\}$ Bước 3: Phân lớp các kiến thức trong $Knowledge$ thành từng loại theo các thành phần của ontology Query-Onto. Bước 4: Hiển thị các kết quả là các nội dung kiến thức đã được phân lớp For $item$ in $Known$ do Show $item.Content$;

Trong thuật giải trên, tại bước 3, phép toán $*$ và $+$ lần lượt là các phép toán giao và hợp giữa hai tập hợp theo nghĩa “so khớp từ”.



Hình 4. Lưu đồ thuật toán cho thuật giải 4.1

2.3.2. Vấn đề tra cứu theo sự phân loại kiến thức

Miền kiến thức của một môn học gồm có nhiều thành phần tri thức như các khái niệm và các thuộc tính của chúng, và các quan hệ giữa các khái niệm, bên cạnh đó, còn có các định lý, tính chất, các luật, và các dạng bài tập thường gặp của môn học đó cùng với phương pháp giải. Vì vậy, việc tra cứu theo sự phân loại kiến thức giúp người học có thể xác định các thông tin cần thiết mà họ cần. Quá trình tra cứu này được thực hiện dựa trên việc xác định loại kiến thức được tìm kiếm trong yêu cầu tra cứu. Thuật giải cho việc tra cứu này gồm có hai giai đoạn chính như sau:

Giai đoạn 1: Xác định loại kiến thức cần tra cứu.

Việc xác định được giải quyết bằng cách sử dụng kho ngữ liệu của miền tri thức được tổ chức lưu trữ dưới dạng các từ khóa trong tri thức. Thông qua kho ngữ liệu đó, hệ thống có thể phân loại kiến thức mà người dùng muốn tra cứu. Trước tiên, hệ thống sẽ chia một câu truy vấn thành các cụm thành phần của nó, gồm: NP (Cụm danh từ), VP (Cụm động từ) và QP (Cụm câu hỏi). Tuy nhiên, trong nghiên cứu hiện tại, chúng tôi chỉ xét đến câu truy vấn gồm có NP và QP. Khi đó, NP sẽ chứa thông tin nội dung muốn tra cứu và QP giúp xác định loại kiến thức mà người dùng muốn tra cứu.

Ví dụ:

- Câu truy vấn: “*Danh sách liên kết là gì ?*”: hệ thống sẽ xác định NP = “*Danh sách liên kết*” và QP = “*là gì*”, và câu này sẽ được phân loại vào việc tra cứu khái niệm. “*Bài tập về Danh sách liên kết*”.
- Câu truy vấn: “*Bài tập về Danh sách liên kết*”: hệ thống sẽ xác định NP = “*Danh sách liên kết*” và QP = “*bài tập*”, và câu này sẽ được phân loại vào việc tra cứu dạng bài tập.

Để việc xác định được hiệu quả, bên cạnh dựa trên cấu trúc của ontology Query-Onto biểu diễn miền tri thức K, các heuristic cũng được sử dụng để xác định các từ khóa. Sau khi xác định được các từ khóa trong NP và QP, hệ thống sẽ xác định loại tương ứng cho nội dung kiến thức cần tra cứu.

Giai đoạn 2: So khớp kiến thức cần tra cứu và nội dung trong thành phần tri thức.

Từ loại kiến thức đã được xác định ở Giai đoạn 1, hệ thống sẽ truy xuất kiến thức từ các thành phần tương ứng ontology Query-Onto. Việc truy xuất này được thực hiện dựa trên việc áp dụng Thuật giải 4.1 để tìm kiếm nội dung kiến thức phù hợp trong các thành phần tương ứng và phù hợp với thông tin được trích xuất từ câu truy vấn.

2.3.3. Vấn đề khuyến nghị các kiến thức liên quan

Kiến thức liên quan là tập hợp các kiến thức có sự liên kết đến kiến thức đang được tra cứu. Sự liên kết giữa các kiến thức được đặc tả trong ontology Query-Onto biểu diễn miền tri thức K. Khuyến nghị kiến thức liên quan giúp người dùng có thể hiểu thêm nội dung tra cứu của họ, nắm bắt rõ hơn nội dung kiến thức của môn học.

Cho K là miền tri thức dạng ontology Query-Onto, từ khóa s chỉ nội dung kiến thức được tra cứu. Thuật giải sau sẽ xác định tập hợp các kiến thức liên quan đến nội dung tra cứu s.

Bảng 3. Thuật giải 4.2

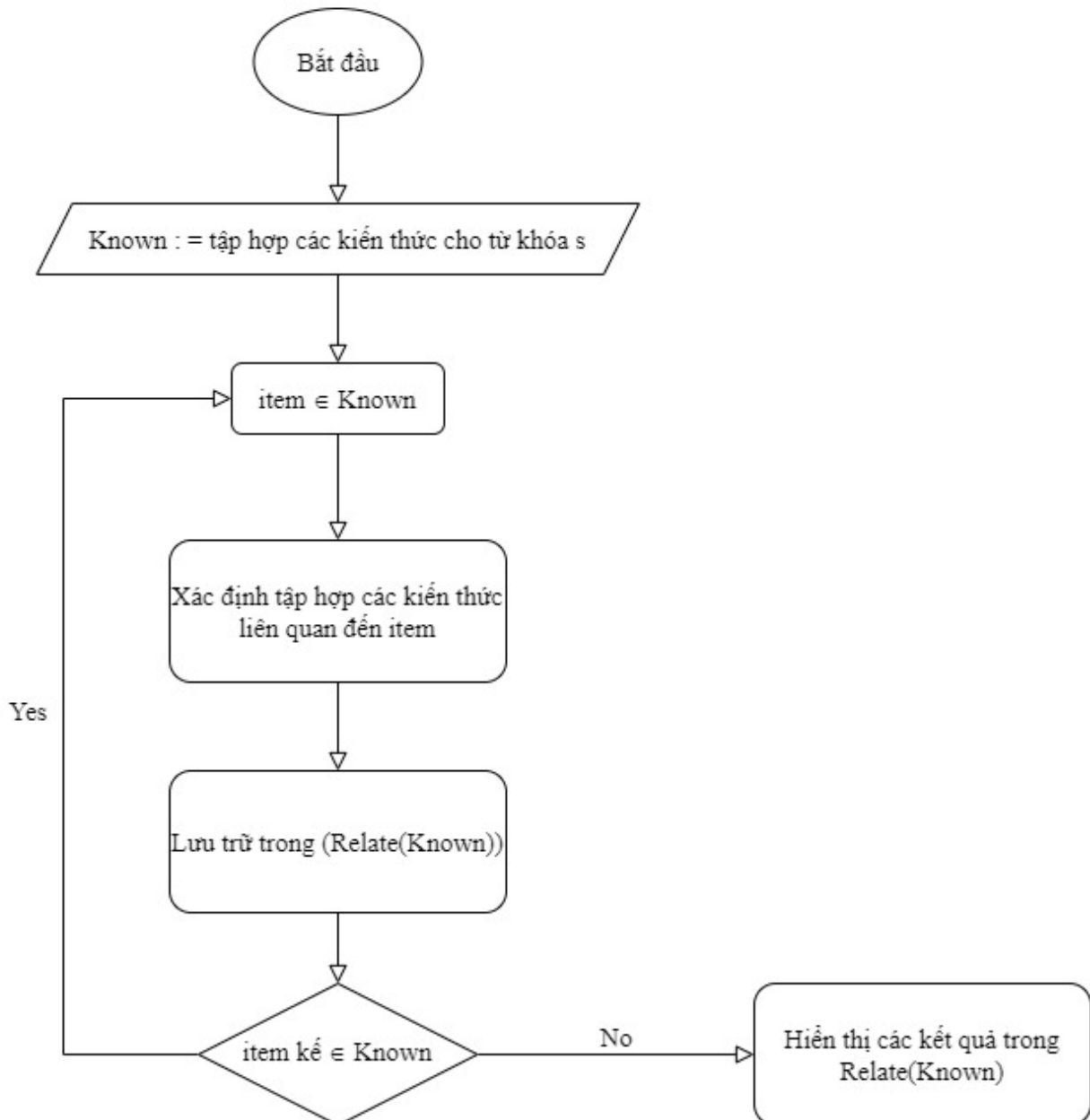
Thuật giải 4.2: <i>Xác định kiến thức liên quan</i>
Input: Miền tri thức K, một từ khóa s. Output: Tập hợp các kiến thức liên quan đến tập các kiến thức của từ khóa s.
Bước 1: <i>Known</i> := tập các kiến thức tìm được cho từ khóa s.// Áp dụng thuật giải 4.1 để xác định Bước 2: Foreach <i>item</i> in <i>Known</i> do Xác định tập hợp các kiến thức liên quan đến <i>item</i> $Related(item) := \{item' \mid \exists r \in K.R, r(item, item')\};$ với $r(x, y)$ là quan hệ giữa kiến x và kiến thức y. Bước 3: Xác định tập hợp các kiến thức liên quan đến <i>Knowledge</i> $Relate(Known) := \bigcup_{item \in Known} Related(item)$

Phân lớp các kiến thức trong $Relate(Known)$ thành từng loại theo các thành phần của ontology Query-Onto.

Bước 4: Hiển thị các kết quả trong $Relate(Known)$ đã được phân lớp.

For $item$ **in** $Relate(Known)$ **do**

Show $item.Content$;



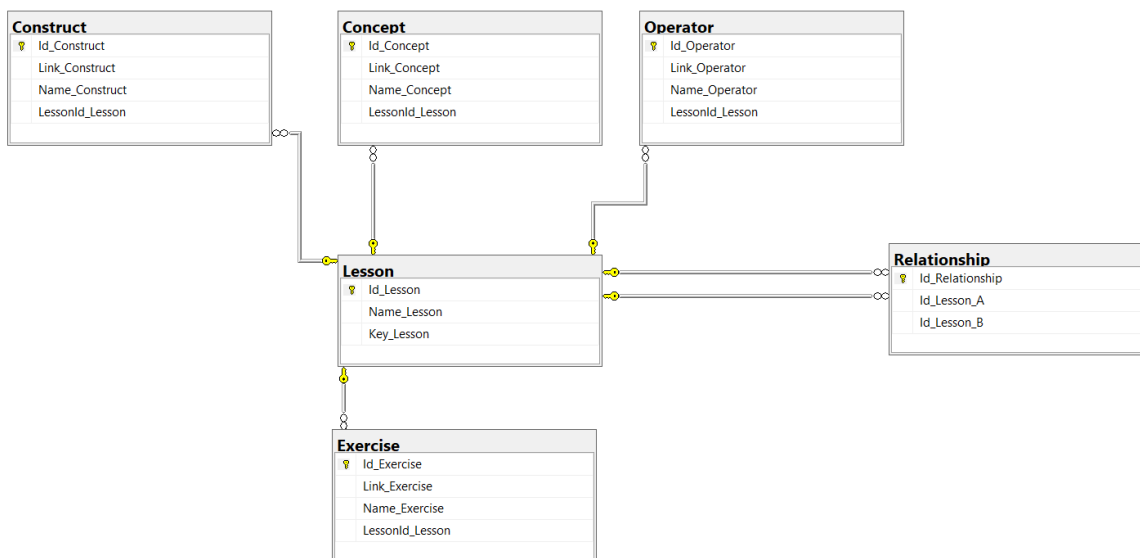
Hình 5. Lưu đồ thuật toán cho thuật giải 4.2

CHƯƠNG 3: XÂY DỰNG MÔ HÌNH CƠ SỞ DỮ LIỆU

Trong chương này, chúng tôi sẽ trình bày về việc xây dựng hệ thống tra cứu kiến thức môn học để tìm kiếm, truy vấn kiến thức môn học trên nền tảng chatbot bằng Dialogflow

3.1. Tổ chức cơ sở tri thức chatbot

3.1.1. Mô hình cơ sở dữ liệu



Hình 6. Mô hình cơ sở dữ liệu cho hệ thống

3.1.2. Mô tả các bảng dữ liệu

Bảng 4. Mô tả bảng Lesson

Tên cột	Kiểu dữ liệu	Ràng buộc	Diễn giải
Id_Lesson	Text	Khóa chính, khác null	Mã khái niệm, cũng là đường dẫn đến html nội dung
Name_Lesson	Text		Tên khái niệm
Key_Leson	Text		Từ khóa tìm kiếm

Bảng 5. Mô tả bảng Construct

Tên cột	Kiểu dữ liệu	Ràng buộc	Diễn giải
Id_Construct	Text	Khóa chính, khác null	Mã khởi tạo
Link_Construct	Text		Đường dẫn đến html nội dung

Name_Construct	Text		Nội dung khởi tạo tương ứng với khái niệm
LessonId_Lesson	Text	Khóa ngoại (tham chiếu đến bảng Lesson)	Mã khái niệm

Bảng 6. Mô tả bảng Concept

Tên cột	Kiểu dữ liệu	Ràng buộc	Diễn giải
Id_Concept	Text	Khóa chính, khác null	Mã định nghĩa
Link_Concept	Text		Đường dẫn đến html nội dung
Name_Concept	Text		Nội dung định nghĩa tương ứng với khái niệm
LessonId_Lesson	Text	Khóa ngoại (tham chiếu đến bảng Lesson)	Mã khái niệm

Bảng 7. Mô tả bảng Operator

Tên cột	Kiểu dữ liệu	Ràng buộc	Diễn giải
Id_Operator	Text	Khóa chính, khác null	Mã hoạt động/ toán tử
Link_Operator	Text		Đường dẫn đến html nội dung
Name_Operator	Text		Nội dung hoạt động tương ứng với khái niệm
LessonId_Lesson	Text	Khóa ngoại (tham chiếu đến bảng Lesson)	Mã khái niệm

Bảng 8. Mô tả bảng Relationship

Tên cột	Kiểu dữ liệu	Ràng buộc	Diễn giải
Id_Relationship	Text	Khóa chính, khác null	Mã mối quan hệ
Id_Lesson_A	Text	Khóa ngoại (tham chiếu đến bảng Lesson)	Mã khái niệm

Id_Lesson_B	Text	Khóa ngoại (tham chiếu đến bảng Lesson)	Mã khái niệm
-------------	------	---	--------------

Bảng 9. Mô tả bảng Exercise

Tên cột	Kiểu dữ liệu	Ràng buộc	Diễn giải
Id_Exercise	Text	Khóa chính, khác null	Mã dạng bài tập
Link_Exercise	Text		Đường dẫn đến html nội dung
Name_Exercise	Text		Nội dung các dạng bài tập tương ứng với khái niệm
LessonId_Lesson	Text	Khóa ngoại (tham chiếu đến bảng Lesson)	Mã khái niệm

3.1.3. Một số ràng buộc về dữ liệu

❖ Ràng buộc khóa nội

- **R1:** Mỗi khái niệm có 1 mã khái niệm duy nhất, không trùng với các mã khái niệm khác.
 - **Biểu diễn:** $\forall \text{les1, les2} \in \text{Lesson}$, sao cho $\text{Lesson.les1} \neq \text{Lesson.les2}$.
- **R2:** Mỗi khởi tạo có 1 mã khởi tạo duy nhất, không trùng với các mã khởi tạo khác.
 - **Biểu diễn:** $\forall \text{cons1, cons2} \in \text{Construct}$, sao cho $\text{Construct.cons1} \neq \text{Construct.cons2}$.
- **R3:** Mỗi định nghĩa có 1 mã định nghĩa duy nhất, không trùng với các mã định nghĩa khác.
 - **Biểu diễn:** $\forall \text{conc1, conc2} \in \text{Concept}$, sao cho $\text{Concept.conc1} \neq \text{Concept.conc2}$.
- **R4:** Mỗi hoạt động có 1 mã hoạt động duy nhất, không trùng với các mã hoạt động khác.
 - **Biểu diễn:** $\forall \text{oper1, oper2} \in \text{Operator}$, sao cho $\text{Operator.oper1} \neq \text{Operator.oper2}$.

- **R5:** Mỗi mối quan hệ có 1 mã mối quan hệ duy nhất, không trùng với các mã mối quan hệ khác.
 - **Biểu diễn:** $\forall \text{ rela1, rela2} \in \text{Relationship, sao cho Relationship.rela1} \neq \text{Relationship.rela2.}$
- **R6:** Mỗi dạng bài tập có 1 mã dạng bài tập duy nhất, không trùng với các mã dạng bài tập khác.
 - **Biểu diễn:** $\forall \text{ exer1, exer2} \in \text{Exercise, sao cho Exercise.exer1} \neq \text{Exercise.exer2.}$
- ❖ **Ràng buộc khóa ngoại**
- **R7:** Mỗi bộ khởi tạo phải có một mã khái niệm ứng với mã khái niệm trong bảng Lesson.
 - **Biểu diễn:** $\text{Construct[LessonId_Lesson]} \subseteq \text{Lesson[Id_Lesson]}$
- **R8:** Mỗi bộ định nghĩa phải có một mã khái niệm ứng với mã khái niệm trong bảng Lesson.
 - **Biểu diễn:** $\text{Concept[LessonId_Lesson]} \subseteq \text{Lesson[Id_Lesson]}$
- **R9:** Mỗi bộ hoạt động phải có một mã khái niệm ứng với mã khái niệm trong bảng Lesson.
 - **Biểu diễn:** $\text{Operator[LessonId_Lesson]} \subseteq \text{Lesson[Id_Lesson]}$
- **R10:** Mỗi bộ mối quan hệ phải có một mã khái niệm A ứng với mã khái niệm trong bảng Lesson. Và một khái niệm B tương ứng với mã khái niệm trong bảng Lesson sao cho $\text{lesson A} \neq \text{Lesson B}$
 - **Biểu diễn:** $\text{Relationship[Id_Lesson_A]} \subseteq \text{Lesson[Id_Lesson]},$
 $\text{Relationship[Id_Lesson_B]} \subseteq \text{Lesson[Id_Lesson]}, \text{ Id_Lesson_A} \neq \text{Id_Lesson_B.}$
- **R11:** Mỗi bộ dạng bài tập phải có một mã khái niệm ứng với mã khái niệm trong bảng Lesson.
 - **Biểu diễn:** $\text{Exercise[LessonId_Lesson]} \subseteq \text{Lesson[Id_Lesson]}.$

3.2. Thiết kế chatbot

3.2.1. Cách chatbot được huấn luyện

Một số mô hình học máy huấn luyện chatbot thường gồm hai chức năng chính: nhận dạng Intents và nhận dạng Entities. Nhận dạng Intent giúp cho hệ thống hiểu được ý định của người dùng ở mỗi thông điệp đầu vào. Nhận dạng Entities giúp hệ thống lấy được các thuộc tính (hay thực thể Entities) của Intents tương ứng. Để thực hiện hai nhiệm vụ này, người ta sẽ sử dụng 2 mô hình học máy làm 2 chức năng khác nhau đó là phân loại Intents và phân loại Entities.

a. Phân loại Intent

Để làm nhiệm vụ này, cần một mô hình phân loại chủ đề văn bản. Dữ liệu là tập hợp các văn bản đã được gán sẵn nhãn chủ đề. Mô hình phân loại văn bản phổ biến sử dụng Bag of words làm đặc trưng phân loại. Ý tưởng chính là đếm tần suất xuất hiện của các từ xuất hiện trong văn bản, dựa trên từ điển là tập hợp tất cả các từ xuất hiện trong toàn văn bản

Ví dụ, ta có 2 câu sau:

- Kiên thích xem phim ma, My cũng thích xem phim ma.
- Kiên còn thích xem cả phim hành động.

Với từ vựng của 2 thông điệp trên, ta thành lập được từ điển sau:

[“Kiên”, “thích”, “xem”, “phim”, “ma”, “My”, “cũng”, “còn”, “cả”, “hành”, “động”]

Bằng cách này ta chuyển được dòng thông điệp trên thành kiểu số chỉ số lần hiển thị của từ vựng để máy tính xử lý

- [1, 2, 2, 2, 2, 1, 1, 0, 0, 0, 0]
- [1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1]

Hai thông điệp trên giờ đã được chuyển hóa thành 2 vector. Bản chất của xử lý ngôn ngữ tự nhiên là dùng những phép biến đổi để biến dạng chữ thành số, sao cho giữ được những đặc trưng của văn bản

Ví dụ: Phân loại Intent #Greeting và Intent #Complaint

Người ta dùng 2 tập dữ liệu là những câu mà người dùng hay dùng để chào hỏi và tập những câu mà người dùng hay dùng để phàn nàn:

Chào hỏi: #Greeting

Ví dụ những câu như: “Chào!”, “Chào bạn”, “Hello”, “Hi”,...

Phàn nàn: #Complaint

Ví dụ những câu như: “Dịch vụ tệ”, “Sản phẩm kém chất lượng”, “Sản phẩm xấu”, “Màu sắc nhạt nhòa không sinh động”,...

Sau khi huấn luyện hệ thống với tập dữ liệu thứ nhất, hệ thống có khả năng phân loại chính xác các thông điệp nhận vào mới có nội dung giống các thông điệp trong tập huấn luyện này vào Intent #Greeting. Nếu người dùng nhập vào những thông điệp không nằm trong tập huấn luyện nhưng có xu hướng có những từ vựng như: “Chào”, “ơi”, “hey”,... thì hệ thống cũng có xác suất cao phân loại những thông điệp này vào Intent #Greeting.

Đối với tập dữ liệu thứ hai cũng tương tự, nếu người dùng gõ một trong những câu đã được huấn luyện, hệ thống sẽ phân loại chính xác vào Intent #Complaint. Đối với những thông điệp chưa được huấn luyện, nhưng có mức độ tương tự cao, như có các từ vựng mô tả sự phàn nàn đã được huấn luyện trong tập huấn luyện như: “Kém”, “thiếu”, “xấu”, ... sẽ có xu hướng cao được phân loại vào Intent #Complaint.

b. Phân loại Entities

Trong xử lý ngôn ngữ tự nhiên, để nhận ra một thực thể (Tên người, tên địa danh, ngày tháng, con số, ...) trong một văn bản, người ta sử dụng một kỹ thuật học máy có tên gọi Name Entity Recognition (NER) hay còn gọi là Entity Extraction (Nhận dạng thực thể định danh hay trích xuất thực thể)

Ví dụ:

“Tôi muốn đi tới Hà Nội vào lúc 19 giờ ngày 31/03/2021”

Để hệ thống nhận ra được Hà Nội là **@Location**, 19 giờ là **@Time** và 31/03/2021 là **@Date**, người ta thường dùng một mô hình học máy để huấn luyện và gán nhãn cho mỗi token của văn bản.

Dựa theo ví dụ trên, bằng bộ tách từ, thông điệp được chia thành các token. Có thể gán nhãn cho từng token trong thông điệp trên như sau:

“Tôi/**O_W** muốn/**O_W** đi/**O_W** tới/**O_W** Hà Nội/**@Location** vào/**O_W** lúc/**O_W** 19 giờ /**@Time** ngày/**O_W** 31/03/2021/**@Date**

Bằng cách gán nhãn **@Location** cho Hà Nội, **@Time** cho 19 giờ, **@Date** cho 31/03/2021, và tất cả các từ còn lại nhãn **O_W**, khi đưa vào tập dữ liệu huấn luyện đủ lớn, hệ thống có thể nhận diện được chính xác những gì đã huấn luyện. Đối với những thông điệp mới, nếu chưa những token viết hoa ký tự đầu và nằm ngay sau token “từ...”, “đến...”, “tới...”

Sẽ được dự đoán là **@Location**. Những token có dạng DD/MM/YYYY có xu hướng được phân loại là **@Date** và các token có dạng HH:MM hoặc chứa “AM”, “PM”, “Giờ”,... sẽ được phân loại là **@Time**. Tất cả các token còn lại không có những đặc tính như trên sẽ có xu hướng được phân loại là **O_W**.

3.2.2. Giới thiệu Dialogflow

Dialogflow (trước đây được gọi là API.AI) là một nền tảng hiểu ngôn ngữ tự nhiên được sử dụng để thiết kế và tích hợp giao diện người dùng trò chuyện vào các ứng dụng di động, ứng dụng web, thiết bị, bot, hệ thống phản hồi giọng nói tương tác và các mục đích sử dụng liên quan.

3.2.3. Một số khái niệm cơ bản trong Dialogflow

a. Agents (Tác nhân)

Một Agent là một tác nhân ảo xử lý các cuộc hội thoại với người dùng cuối của bạn. Nó là một mô-đun hiểu ngôn ngữ tự nhiên, hiểu được các sắc thái của ngôn ngữ con người. Dialogflow dịch văn bản hoặc âm thanh của người dùng cuối trong cuộc hội thoại sang dữ liệu có cấu trúc mà ứng dụng và dịch vụ của bạn có thể hiểu được. Bạn thiết kế và xây dựng một tác nhân Dialogflow để xử lý các loại hội thoại cần thiết cho hệ thống của bạn.

Ví dụ : Với mỗi nội dung cuộc trò chuyện với bot khác nhau bạn có thể tạo các agent khác nhau để xử lý. Như bot về Thời tiết, Đồ ăn, Địa điểm,...

b. Intents (Ý định)

Một Intent phân loại một ý định của người dùng cuối cho một lượt trò chuyện. Đối với mỗi agent, bạn xác định nhiều intent, trong đó các việc kết hợp các intent có thể xử lý một cuộc hội thoại hoàn chỉnh.

Khi người dùng cuối viết hoặc nói điều gì đó, được gọi là biểu thức người dùng cuối , Dialogflow sẽ phân loại ý định của người dùng phù hợp với một intent cũng được gọi là phân loại ý định .

Ví dụ: Nếu người dùng cuối nói “*Mãng là gì ?*”, Hộp thoại sẽ khớp biểu thức của người dùng cuối đó với mục đích nêu lên khái niệm của *Mãng*. Bạn cũng có thể xác định ý định trích xuất thông tin hữu ích từ biểu thức của người dùng cuối.

Một Intents cơ bản có chứa:

- **Trainning Phases**

Đây là những cụm từ ví dụ cho những gì người dùng cuối có thể nói. Khi một biểu thức người dùng cuối giống với một trong những cụm từ này, Dialogflow sẽ khớp với ý định. Bạn không phải xác định mọi ví dụ có thể, bởi vì học máy tích hợp của Dialogflow mở rộng trong danh sách của bạn với các cụm từ tương tự khác.

- **Action**

Ta có thể xác định một hành động cho từng ý định. Khi một ý định được khớp, Dialogflow cung cấp hành động cho hệ thống của bạn và bạn có thể sử dụng hành động để kích hoạt một số hành động được xác định trong hệ thống của mình.

- **Tham số**

Khi một ý định được khớp trong thời gian chạy, Dialogflow cung cấp các giá trị được trích xuất từ biểu thức của người dùng cuối dưới dạng tham số . Mỗi tham số có một loại, được gọi là loại thực thể , chỉ ra chính xác cách trích xuất dữ liệu. Không giống như đầu vào của người dùng cuối, các tham số là dữ liệu có cấu trúc có thể dễ dàng được sử dụng để thực hiện một số logic hoặc tạo phản hồi.

- **Phản hồi**

Ta phải xác định văn bản, lời nói hoặc phản hồi trực quan để trả về người dùng cuối. Chúng có thể cung cấp cho người dùng cuối câu trả lời, hỏi người dùng cuối để biết thêm thông tin hoặc chấm dứt cuộc trò chuyện.

c. Entities (Thực thể)

Mỗi tham số của intent được xếp vào mỗi loại riêng, được gọi là kiểu thực thể, quy định chính xác cách trích xuất dữ liệu từ biểu thức người dùng cuối.

Ví dụ: có các thực thể hệ thống để khớp ngày, giờ, màu sắc, địa chỉ email, v.v. Bạn cũng có thể tạo các thực thể tùy chỉnh của riêng mình để khớp dữ liệu tùy chỉnh. Ví dụ: bạn có thể xác định một thực thể rau có thể phù hợp với các loại rau có sẵn để mua trong cửa hàng tạp hóa.

d. Contexts (Kịch bản)

Kịch bản hội thoại tương tự như kịch bản ngôn ngữ tự nhiên. Nếu một người nói “màu cam”, ta cần kịch bản để hiểu họ đang đề cập đến điều gì. Tương tự, đối với Dialogflow để xử lý một biểu thức người dùng cuối như vậy, nó cần được cung cấp với ngữ cảnh để khớp chính xác với một intent.

Sử dụng kịch bản có thể kiểm soát luồng của cuộc trò chuyện. Có thể định cấu hình kịch bản cho một mục đích bằng cách đặt kịch bản đầu vào và đầu ra, được xác định bởi tên chuỗi. Khi một ý định được khớp, mọi kịch bản đầu ra được cấu hình cho ý định đó sẽ hoạt động. Trong khi bất kỳ kịch bản nào đang hoạt động, Dialogflow có nhiều khả năng khớp với các ý định được cấu hình với các kịch bản đầu vào tương ứng với các kịch bản hiện đang hoạt động.

e. Follow-up intents (Ý định kéo theo)

Ta có thể sử dụng các ý định tiếp theo để tự động đặt kịch bản cho các cặp ý định. Mục đích tiếp theo chỉ được khớp khi ý định cha được khớp trong lượt trò chuyện trước đó. Ta cũng có thể tạo nhiều cấp độ ý định lồng nhau.

3.2.4. Xây dựng chatbot cho trang web

a. Xác định các kịch bản

Sau khi xây dựng được hệ thống hỗ trợ tra cứu kiến thức, chúng ta có thể xác định được các kịch bản mà người dùng có thể hỏi cho chatbot. Dựa trên vấn đề tra cứu theo sự phân loại kiến thức mà đã được nhắc đến ở chương 3 mục 3.2, những câu hỏi người dùng đưa ra sẽ được phân loại thành các kịch bản để chatbot trả lời.

Ví dụ:

- Câu truy vấn: “Danh sách liên kết là gì”, hệ thống xác định được các từ khóa từ bộ từ khóa có sẵn sau đó trả về kết quả như mong muốn cho người dùng.

Việc xác định bộ từ khóa sẽ được đề cập đến ở phần tiếp theo

b. Xác định các từ khóa (tokens)

Từ các hệ thống hỗ trợ tra cứu kiến thức được xây dựng ở trên, ta có thể xác định được các từ khóa để xây dựng nên các kịch bản. Bộ từ khóa bao gồm:

- Các cụm danh từ là các tên của các bài tập cần tra cứu.
- Các cụm câu hỏi như: là gì, như thế nào,...

Cụm câu hỏi giúp nhận biết loại kiến thức mà người dùng muốn tìm kiếm, kết hợp với cụm động từ, hệ thống sẽ xác định được phân loại kiến thức của một bài tập nhất định. Ví dụ:

- Người dùng nhập câu truy vấn: “Mình muốn tìm hiểu về bài tập về mảng”. Hệ thống sẽ xác định được cụm câu hỏi “*bài tập*” và cụm danh từ “*mảng*” để xác định phân loại vào việc tra cứu dạng bài tập của mảng, kèm thêm khuyến nghị các kiến thức liên quan, bài tập của mảng động cũng được đưa ra.

Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	e_baitap	@E_BaiTap	\$e_baitap	<input type="checkbox"/>
<input type="checkbox"/>	e_cautrucdulieu_mang	@E_CauTrucDuLieu_Mang	\$e_cautrucdulieu_mang	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Hình 7. Các từ khóa được phân theo các entities

Training phrases

Search training phrases

Add user expression

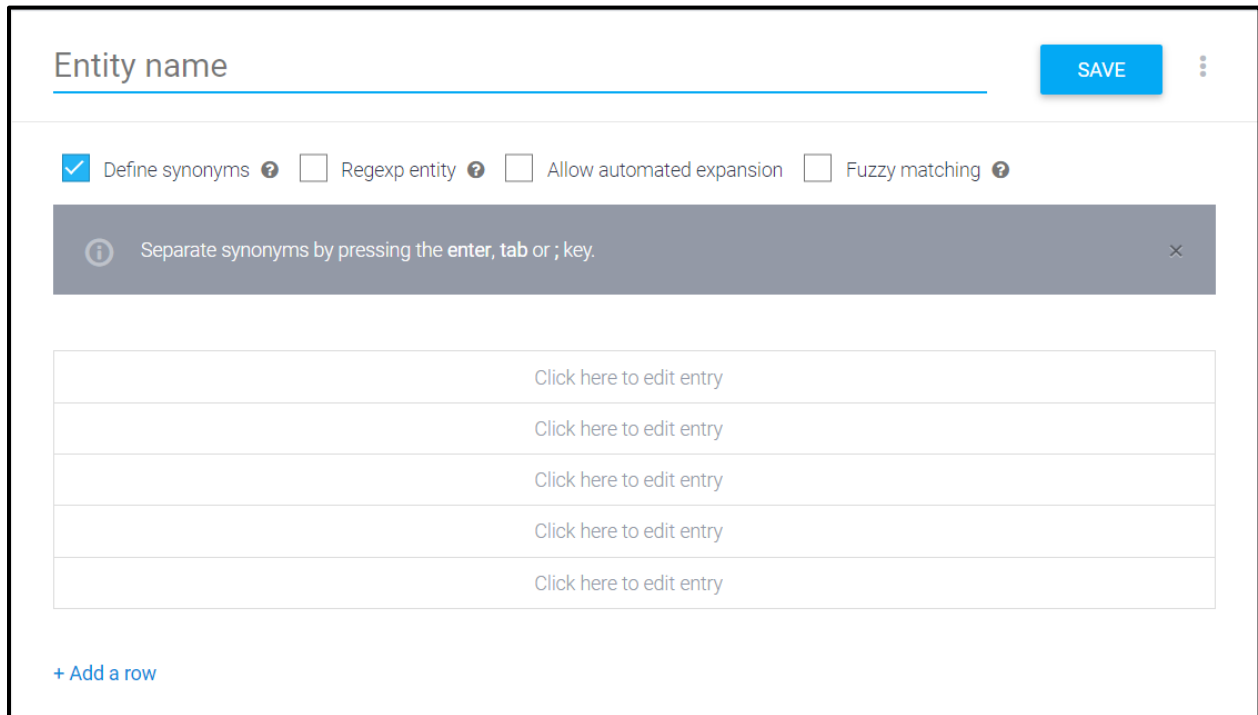
mảng có bài tập nào

Bài tập về mảng

Hình 8. Các từ khóa được hệ thống xác định

c. Tạo các entities từ các từ khóa

Đây là một trong 2 bước quan trọng trong việc xây dựng chatbot. Từ các từ khóa mà ta đã xác định, chúng ta tiến hành xây dựng các Entities tương ứng:



The screenshot displays the 'Create entity' interface. At the top, there is a text input field labeled 'Entity name' and a blue 'SAVE' button. Below this, there are four checkboxes: 'Define synonyms' (checked), 'Regex entity', 'Allow automated expansion', and 'Fuzzy matching'. A grey informational box states: 'Separate synonyms by pressing the enter, tab or ; key.' Below this is a table with five rows, each containing a 'Click here to edit entry' link. At the bottom left, there is a '+ Add a row' button.

Hình 9. Giao diện làm việc để thêm Entities tương ứng với các từ khóa

Đầu tiên, ta chọn “Create entity” hoặc nhấn vào dấu “+” để mở giao diện thêm Entity mới. Tiếp theo, ta đặt tên entity ví dụ mình xác định từ khóa cho entity cơ sở dữ liệu. Sau đó ta liệt kê các từ đặt để xác định entity và từ đồng nghĩa của chúng bằng cách nhập giá trị vào “Enter Referece Value” và “Enter Synonym”. Chúng ta cũng có thể thêm các giá trị khác nhau bằng cách nhấn vào “Add a row”.

E_CTDL

SAVE

☒ Define synonyms ⓘ
 ☐ Regexp entity ⓘ
 ☐ Allow automated expansion
 ☐ Fuzzy matching ⓘ

cấu trúc dữ liệu	cấu trúc dữ liệu, ctdl
Data Structure	Data Structure
Enter reference value	Enter synonym

+ Add a row

Hình 10. Ví dụ tạo Entity cho từ khóa đại diện cho CTDL

Chúng ta có thể chỉnh sửa hoặc quản lý lại các Entity đã tạo ra bằng các về giao diện quản lý Entity

Entities

CREATE ENTITY

Custom System

Search entities

1 OF 4

@_E_Bellman-Ford
@_E_BoNho
@_E_Cay_AVL
@_E_Cay_DuyetCay_HauThuTu
<input type="checkbox"/> @_E_Cay_DuyetCay_TienThuTu
@_E_Cay_DuyetCay_TrungThuTu
@_E_Cay_Heap
@_E_Cay_Khung
@_E_Cay_Khung_Min
@_E_Cay_NhiPhan
@_E_CSDL_DSLK_Doi
@_E_CSDL_DSLK_Don
@_E_CSDL_DSLK_Vong
@_E_CTDL
@_E_CTDL_Cay
@_E_CTDL_DanhSachLienKet
@_E_CTDL_DoThi
@_E_CTDL_HangDoi
@_E_CTDL_Mang
@_E_CTDL_Mang_Dong

1 OF 4

Hình 11. Quản lý hoặc chỉnh sửa các entity đã tạo ra

d. Tạo các intents từ các kịch bản

Sau khi xây dựng được các entity, bạn có thể dễ dàng xây dựng các kịch bản (intent) từ kịch bản đã xây dựng. Thực ra khi 1 intent được xác định nội dung câu nói. Nó sẽ được kích hoạt và Dialogflow sẽ tự động bóc tách các từ khoá đó lưu vào các parameters. Từ khoá trong parameter này sẽ được thay thế bởi từ khoá bạn đã khai báo trong entity.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	e_dacdiem	@E_DacDiem	\$e_dacdiem	<input type="checkbox"/>
<input type="checkbox"/>	e_giaithuat	@E_GiaiThuat	\$e_giaithuat	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Hình 13. Action and parameters

Training phrases

Search training phrases

” Add user expression

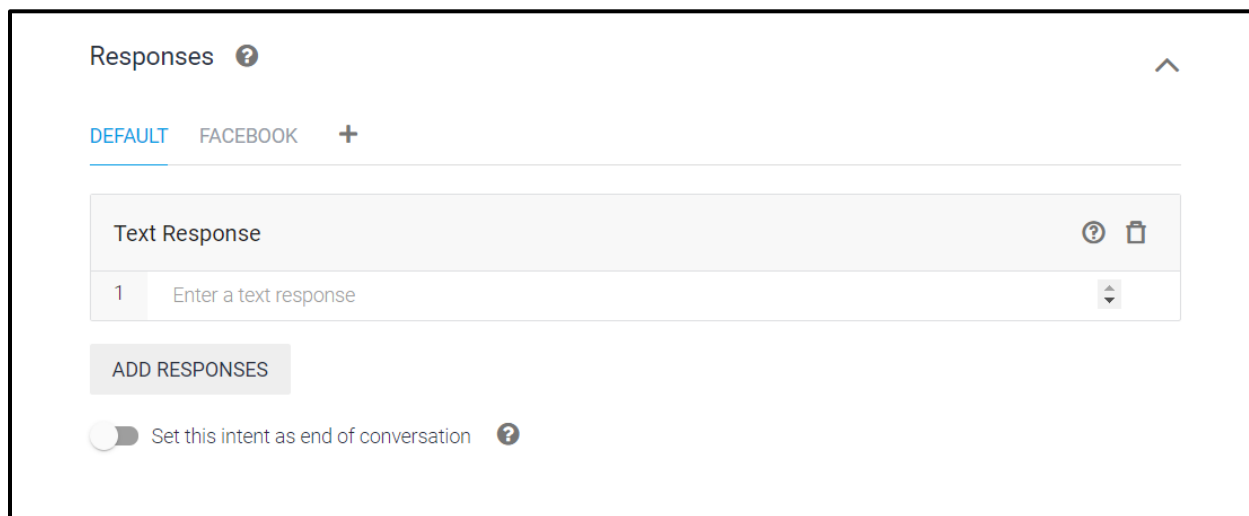
” đặc điểm thuật toán

” Đặc điểm giải thuật

Hình 12. Training phrases

Action và parameters rất quan trọng trong việc giúp máy hiểu chính xác câu nói và là tham số để chúng ta lập trình. Các thông số này sẽ được gán tự động khi người dùng giao tiếp đúng với một tình huống của một intent cụ thể.

Responses giúp bot trả lời bạn, đây là trường để bạn nhập câu trả lời. Dialogflow cho phép nhập một hoặc nhiều câu trả lời để bạn không cảm thấy nhàm chán khi giao tiếp với bot.




Hình 14. Responses

Có nhiều cách response khác nhau, đôi lúc không chỉ dùng mỗi text đơn giản, bạn có thể sử dụng trả lời bằng hình ảnh, nút nhấn hay link khi khách yêu cầu. Nếu có nhiều hơn 1 response trong 1 intent thì hệ thống sẽ sử dụng ngẫu nhiên một trong các response đó. Vì vậy khi tạo intent bạn nên thêm vài response để cuộc trò chuyện thêm phong phú. Điều đáng chú ý nhất là response của intent phải trả lời đúng yêu cầu người dùng và nó có thể là câu gợi ý để cuộc nói chuyện hay hơn.


CHƯƠNG 4: KẾT QUẢ THỬ NGHIỆM

4.1. Kết quả tra cứu theo nội dung kiến thức

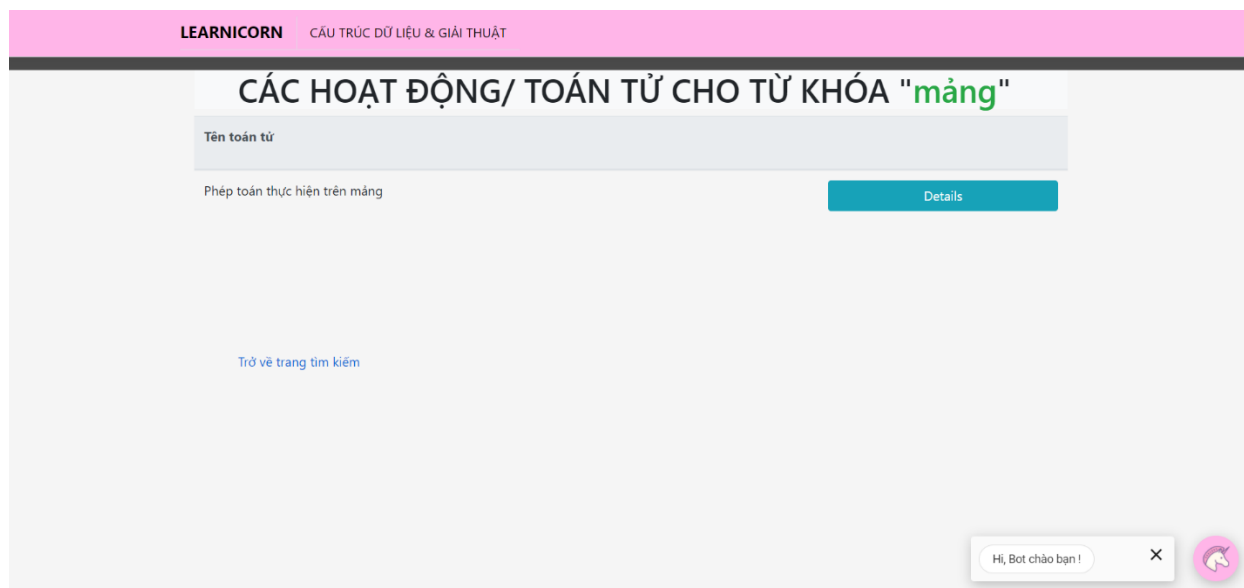
LEARNICORN CẤU TRÚC DỮ LIỆU & GIẢI THUẬT				
⋮	<input type="text" value="Nhập tên kiến thức"/>	<input type="button" value="Tìm kiếm"/>	Tìm kiếm theo các thành phần	Hiển thị kiến thức liên quan
NỘI DUNG KIẾN THỨC CHO TỪ KHÓA "dslk"				
Bài học	Khái niệm	Khởi tạo	Toán tử	Bài tập
Danh sách liên kết đơn	Khái niệm cấu trúc dữ liệu danh sách liên kết đơn	Biểu diễn danh sách liên kết đơn	Hoạt động trên danh sách liên kết đơn	Bài tập về danh sách liên kết đơn
Danh sách liên kết đôi	Khái niệm cấu trúc dữ liệu danh sách liên kết đôi	Biểu diễn danh sách liên kết đôi	Hoạt động trên danh sách liên kết đôi	Bài tập về danh sách liên kết đôi
Danh sách liên kết vòng	Khái niệm cấu trúc dữ liệu danh sách liên kết vòng	Biểu diễn danh sách liên kết vòng	Hoạt động trên danh sách liên kết vòng	Bài tập về danh sách liên kết vòng
Trở về trang chính				
Hi, Bot chào bạn ! <input type="button" value="X"/> 				

Hình 15. Kết quả khi tra cứu theo nội dung kiến thức với từ khóa "dslk"

4.2. Kết quả tra cứu theo sự phân loại kiến thức

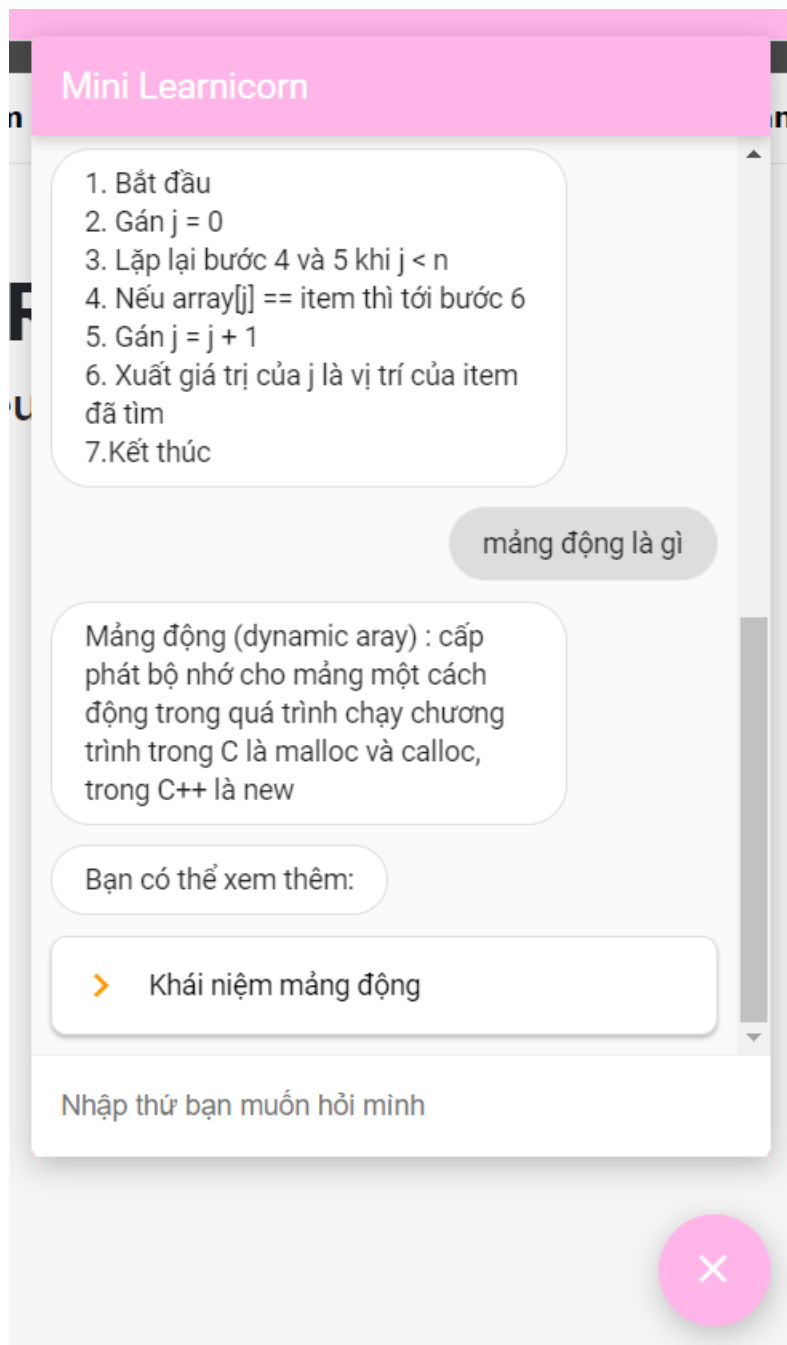
LEARNICORN CẤU TRÚC DỮ LIỆU & GIẢI THUẬT	
⋮	<input type="text" value="Nhập tên kiến thức"/> <input type="button" value="Tìm kiếm"/>
Tìm kiếm theo các thành phần Hiển thị kiến thức liên quan	
Tra cứu theo từng trường khác nhau	
Tìm kiếm theo tên kiến thức	
<input type="text" value="Nhập tên kiến thức"/>	<input type="button" value="Tìm kiếm"/>
Tìm kiếm khái niệm của:	
<input type="text" value="Nhập tên kiến thức"/>	<input type="button" value="Tìm kiếm"/>
Tìm kiếm cách biểu diễn / khởi tạo của:	
<input type="text" value="Nhập tên kiến thức"/>	<input type="button" value="Tìm kiếm"/>
Tìm kiếm hoạt động/ toán tử của:	
<input type="text" value="Nhập tên kiến thức"/> <input type="button" value="Tìm kiếm"/>	
Hi, Bot chào bạn ! <input type="button" value="X"/> 	

Hình 16. Phân các loại kiến thức để tra cứu



Hình 17. Kết quả tra cứu thành phần hoạt động cho từ khóa mảng

4.3. Kết quả tra cứu theo sự phân loại kiến thức của chatbot



Hình 18. Kết quả khi thực hiện tìm kiếm khái niệm "mảng động"

4.4. Kết quả khuyến nghị các kiến thức liên quan

LEARNICORN

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

Nhập tên kiến thức

Tìm kiếm

CÂY NHỊ PHÂN

Khái niệm

Cây nhị phân là một loại cây đặc biệt mà mỗi nút của nó chỉ có nhiều nhất là 2 nút con. Khi đó, 2 cây con của mỗi nút được gọi là cây con trái và cây con phải

```
graph TD; 1((1)) --- 2((2)); 1 --- 3((3)); 2 --- 4((4)); 2 --- 5((5)); 4 --- 7((7)); 3 --- 6((6))
```

Cây nhị phân là cây có cấu trúc đơn giản và có nhiều ứng dụng trong tin học. Một số dạng cây nhị phân đặc biệt và được ứng dụng nhiều nhất là:

- **Cây nhị phân đầy đủ:** là cây nhị phân mà mỗi nút không phải là đều có đúng 2 nút con và các nút lá phải có cùng độ sâu.

Tim kiếm theo các thành phần

Hiển thị kiến thức liên quan

Cây Heap

Cấu trúc dữ liệu cây

Cấu trúc dữ liệu

Cấu trúc dữ liệu danh sách liên kết

Cấu trúc dữ liệu mảng

Hì, Bot chào bạn !

Hình 19. Khuyến nghị kiến thức liên quan cho "Cây nhị phân"

CHƯƠNG 5 : TỔNG KẾT

Trong bài báo cáo này, em đã trình bày cấu trúc ontology Query-Onto để tổ chức cơ sở tri thức một môn học cho hệ thống tra cứu kiến thức. Ontology này là sự cải tiến của mô hình tri thức quan hệ Rela-model [3, 4] ở hai khía cạnh: Cải tiến cấu trúc thành phần khái niệm trong mô hình để hỗ trợ việc truy xuất nội dung kiến thức trong quá trình tra cứu, và tích hợp thành phần tri thức biểu diễn các dạng bài tập và phương pháp giải. Sự cải tiến này giúp cho việc tra cứu kiến thức đã đáp ứng được yêu cầu của hệ thống hỗ trợ học tập. Dựa trên cấu trúc mô hình Query-Onto, các chức năng tra cứu đã được nghiên cứu và thiết kế các kỹ thuật để giải quyết các vấn đề về tra cứu nội dung kiến thức, tra cứu theo sự phân loại của miền tri thức và khuyến nghị các kiến thức liên quan.

Tuy nhiên việc áp dụng và xây dựng hệ thống còn có những thiếu sót nhất định. Nhưng vẫn có thể đáp ứng được cơ bản những yêu cầu đã đặt ra cho hệ thống.

TÀI LIỆU THAM KHẢO

- [1] <https://dialogflowmaster.com/> (truy cập ngày 20 tháng 5 năm 2021)
- [2] <https://blog.vietnamlab.vn/gioi-thieu-ve-dialogflow/> (truy cập ngày 20 tháng 5 năm 2021)
- [3] Hatzilygeroudis, J. Prentzas, “Knowledge Representation in Intelligent Educational Systems”, In Web-Based Intelligent E-Learning Systems: Technologies and Applications, pp. 175–192, IGI publisher, 2006.
- [4] N. Do, H. Nguyen, T. Mai, “Intelligent Educational Software in Discrete Mathematics and Graph Theory”, In Proceedings of 17th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET 2018), pp. 925 - 938, 2018.
- [5] N. Do, H. Nguyen, A. Selamat, “Knowledge-Based model of Expert Systems using Rela-model”, International Journal of Software Engineering and Knowledge Engineering (IJSEKE), vol. 28, no. 8, pp. 1047 – 1090, 2018.
- [6] Nguyễn Đức Nghĩa, Cấu trúc dữ liệu và Thuật toán, NXB Bách Khoa Hà Nội, 2021.
- [7] Đỗ Văn Nhơn, Trịnh Quốc Sơn, Giáo trình Cấu trúc dữ liệu và Giải thuật, NXB ĐHQG-HCM, 2014.
- [8] N. Karumanchi, *Data Structures and Algorithms Made Easy (5th ed.)*, CareerMonk Publications, 2016.