Design Intelligent Educational Chatbot for Information Retrieval based on Integrated Knowledge Bases

Hien D. Nguyen, Tuan-Vi Tran, Xuan-Thien Pham, Anh T. Huynh, Vuong T. Pham, Diem Nguyen

***Abstract*— The intelligent chatbot has many applications in the real world, especially for supporting e-learning. An educational chatbot requires a complete knowledge base to help students to search knowledge content of courses. This paper proposes a model for integrating multiple knowledge domains, called the Integ-Rela model. Each knowledge domain is organized by an ontology Real-model, which is a knowledge model of relations. The Integ-Rela model manages the knowledge base of an intelligent chatbot that can intellectually retrieve information on multiple domains. Moreover, searching problems in a field have been studied and solved, and the issues for retrieval on numerous knowledge domains. The proposed method is applied to construct an intelligent chatbot for searching programming content in a university's Introduction to Programming and Object-oriented Programming courses. It can act as a tutor by providing the required knowledge of both approaches to students. The experimental results show that it is the potential to apply in practice.**

***Index Terms*— intelligent chatbot, knowledge base, ontology, knowledge engineering, e-learning**

I. Introduction

KNOWLEDGE representation in education significantly improves the performance of knowledge acquisition, synthesis, and inference [1, 2]. For building intelligent systems for science and technology education, engineering, and mathematics (STEM) education [3, 4], knowledge representation plays a vital role in organizing the knowledge bases of those intelligent systems [5].

In education, the intelligent searching bot has a knowledge base supporting conversations and finding appropriate knowledge for users [5, 6]. Its knowledge base has to adequately represent the contents of courses and give learners communication as an instructor [3, 7, 8].

Ontology is an effective method to improve the interface between data and search requests to bring results closer to user research requirements [9]. Furthermore, there are also ontological integration methods that have been applied in the knowledge fields related to medicine [10] and education [11, 12]. Semantic sensor network (SSN) ontology for dynamic IoT sensor data is also used through K-means clustering, new knowledge is extracted from input data, and semantic information is used to update the original ontology [13]. However, those methods are not suited for structuring a chatbot's knowledge base because they do not include any components to handle the user-chatbot communication.

Rela-model is an effective model for representing relational knowledge domains [14]. Its foundation includes the structures of concepts, relations between ideas, and inference rules of the domain. Rela-model is also helpful to apply for designing the knowledge bases of searching bots [15]. This study proposes a model for integrating multiple knowledge domains, called the Integ-Rela model. Ontology Real-model organizes each knowledge domain. The integration is worked based on the combination of concepts and linked rules between parts. The Integ-Rela model is used to manage the knowledge base of an intelligent chatbot that can retrieve the knowledge on multiple domains. Moreover, for making conversations on chatbot, its knowledge base is added to store general scripts of practical communication. Besides searching problems on a field, some difficulties for retrieval on multiple knowledge domains of a chatbot are designed. The built chatbot equipped with a complete knowledge base increases its searchability based on practical knowledge content on multiple domains. It can act as a tutor by providing users with the necessary information.

The proposed method is applied to construct an intelligent chatbot for searching programming contents in the Introduction to Programming (I2P) and Object-oriented Programming (OOP) courses in a university. This chatbot plays as a tutor to communicate with undergraduate students in Vietnamese. It helps the learner discover the content of lessons and definitions, apply them in practice, and get information related to search results [16]. This chatbot can be used for self-learning and enhance their skill of programming in those courses.

Manuscript received July 15, 2021; revised February 02, 2022. This research was funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number DSC2021-26-07.

Hien D. Nguyen is a senior lecturer at Faculty of Computer Science, University of Information Technology, Vietnam National University – Ho Chi Minh city (VNU-HCM), Vietnam (email: hiennd@uit.edu.vn).

Tuan-Vi Tran is an undergraduate student at Faculty of Computer Science, University of Information Technology, Vietnam National University – Ho Chi Minh city (VNU-HCM), Vietnam (email: 18520245@gm.uit.edu.vn).

Xuan-Thien Pham is an undergraduate student at Faculty of Computer Science, University of Information Technology, Vietnam National University – Ho Chi Minh city (VNU-HCM), Vietnam (email: 18520158@gm.edu.vn).

Anh T. Huynh is a lecturer at Faculty of Software Engineering, University of Information Technology, Vietnam National University – Ho Chi Minh city (VNU-HCM), Vietnam (email: anhht@uit.edu.vn).

Vuong T. Pham is a lecturer at Faculty of Information Technology, Sai Gon University, Ho Chi Minh city, Vietnam (email: vuong.pham@sgu.edu.vn).

Diem Nguyen is a lecturer at Faculty of Computer Science, University of Information Technology, Vietnam National University – Ho Chi Minh city (VNU-HCM), Vietnam (email: diemntn@uit.edu.vn).

The following section presents some related works. Section 3 describes two integrating models: the model integrating between ontology Rela-model and the structure of scripts, and the model integrating multiple knowledge domains which have Rela-model form. Section 4 presents some searching problems on a knowledge domain and multiple knowledge domains of an intelligent chatbot. Section 5 presents the experimental results of a question-answering chatbot supporting the learning of I2P and OOP courses in the Information Technology (IT) curriculum at universities. Section 6 compares the designed chatbot with others based on the criteria of an intelligent system in education. Section 7 summarizes the paper.

II. RELATED WORK

Chatbots have become more common in today's society with the development of artificial intelligence such as healthcare helpdesk systems to answer complex medical problems [17, 18], an Ontology-based dialogue management system for banking and financial dialogue systems [19], natural language processing, information retrieval, technical knowledge, ... The main goal of these chatbots is to become a trusted companion by providing the necessary information as requested by the user. However, these chatbots are still ineffective in supporting users to find academic knowledge so that they can self-study based on the semantics of the input queries of a previously asked conversation.

An online Question and Answer (QA) Healthcare Assistant system, called HHH, is a chatbot that applies an association model consisting of a knowledge graph and a model text analogy to answer complex medical questions [20]. The knowledge graph of HHH is built from medical data collected from the Internet. However, this bot is designed as a medical chatbot and is not pedagogical when assisting students.

The banking and finance sector holds excellent potential for context discrimination through a rich product set and specificity of proper nouns, entities, and named verbs. The study in [19] uses ontology as both a knowledge base and a basis for dialogue and composition managers. The knowledge base and the dialogue management component are linked in a certain sense. However, it does not have the knowledge base to solve the problem automatically.

In e-learning, a chatbot is a helpful tool to help with admissions advice [21, 22]. It gives some tips and guides learners to solve problems automatically. The tutoring of the chatbot simulates a teacher's tutoring method for students in reality [8, 23]. The searching chatbot helps users retrieve appropriate knowledge content, query on knowledge classifications, and recommend related knowledge for students [12, 24].

Educational apps have pedagogical criteria to provide helpful knowledge for courses. They require a knowledge base organized based on the course outline. Answering knowledge questions related to the content of the courses, the chatbot can naturally communicate with students like a teaching assistant or explain missing knowledge to students. It can use natural queries to retrieve knowledge from users that matches their requirements. The chatbot must be equipped with natural language processing to analyze the meaning of the entered queries in order to share with users easily [25, 26]. From that, it can extract the correct content matching the request from the user. Moreover, the chatbot can recommend more knowledge related to the content found by searching. This relevant knowledge can provide students with a deeper understanding of what to search for.

Besides, the knowledge includes many sub-domains, the method for organizing the knowledge base has the mechanism for knowledge integration. The integration has two kinds: integrating between components in a domain and integrating between sub-domains. This study presents a method for knowledge integration via a kernel as a relational knowledge model.

III. INTEGRATING KNOWLEDGE BASE FOR AN INTELLIGENT CHATBOT

An intelligent chatbot for intellectual searching needs an adequate knowledge base to organize the knowledge domain. In the real world, the knowledge is extracted from multiple knowledge domains. This section presents a method to integrate knowledge bases that forms a knowledge model of relations, the Rela-model [14]. The integrating model can build the knowledge base of an intelligent searching chatbot, including many knowledge domains.

A. Ontology Rela-model

Definition 3.1 [14]: A knowledge model of relations, called the Rela-model, consists of three components:

(C, R, Rules)

C is a set of concepts, and each concept is a class of objects; R is a set about relations between concepts in C, and Rules is a set of inference rules of the knowledge domain.

Each concept c $\in$ **C** is a class of objects. It has an instance set I_c. The structure of each objectis a tuple *(Attr, Inner, Proper, Key)*, where, *Attr* is a set of attributes of a concept, *Inner* is the set of inner-relations between attributes in *Attr*, *Proper* is the set of properties of the corresponding concept, and *Key* is a set of keywords related to other concepts.

<u>Example 3.1:</u> In the courses of Introduction to Programming and Object-oriented programming (OOP), there are some concepts as follows:

• ALGORITHM (in the course of Introduction to Programming) - the concept of algorithms

Attr := {Name, Chapter, Content}

– Name: The name of the Algorithm.
– Chapter: The chapter to which algorithm belongs to.
– Content: The definition and content of the algorithm.

Inner := {belong to}

– Name belong to Chapter

Proper := {}

Key := set of keywords of the algorithm.

• CLASS (in OOP course) - concept about Class.

Attr := {Name, Chapter, Content, Action, Usage}

– Name: The name of Class
– Chapter: The chapter which Class belongs to.
– Content: The definition and content of the Class.
– Action: The set of actions is able in Class.
– Usage: How to use Class.

Inner := {belong to, performance}

– Name belong to Chapter
– Usage is performance of Action

Proper := {} *Key* := set of keywords of the class.

R - set of relation

There are two kinds of relations between concepts in **C**:

$$\mathbf{R} = \mathbf{R}_{\mathbf{hirerachy}} \cup \mathbf{R}_{\mathbf{related\_to}}$$

In which:

• $\mathbf{R}_{\mathbf{hirerachy}}$ is a set of relations as "is-a" between concepts and they are able to be considered as inheritance relations.

c_i is_a c_j: It means c_i is a sub-concept of c_j and c_j.Attr $\subseteq$ c_i.Attr (c_i, $c_j \in \mathbf{C}$).

<u>Example 3.2:</u>

– "Multi-dimensional array" is_a "Array". It has all attributes in "Array".

– "Friend class" is_a "Class"

• $\mathbf{R}_{\mathbf{related\_to}}$ is a set of related knowledge with the main knowledge. When operating the proof layer, these rules assist in deducing the related knowledge.

c_i related_to c_j: It means c_i has the relationship with c_j (c_i, $c_j \in \mathbf{C}$)

<u>Example 3.3:</u>

– "Program" related_to "Programming language"

– "Statement" related_to {"DataTypes", "Key word", "Typedef"}. It means:

∗ "Statement" related_to "DataTypes"

∗ "Statement" related_to "Keyword"

∗ "Statement" related_to "Typedef"

Rules - set of rules

Definition 3.2 [14]: *Kinds of facts*

Let $\mathcal{K}$ be a knowledge domain as Rela-model. There are some facts as follows:

1. $o : c$ ($c \in \mathbf{C}$, o is an object): o is an object of the concept c.
2. Determine(o) ($c \in \mathbf{C}$, $o \in I_c$) : An object o is determined.
3. Determine($c.a$) ($c \in \mathbf{C}$, a is an attribute of a component of c).
4. $c_1\ \Theta\ c_2$ ($c_1, c_2 \in \mathbf{C}$, $\Theta \in \mathbf{R}$): the relation Θ between two concepts c_1 and c_2.

Each rule $r \in$ **Rules** is an inference rule, and it has the form: $u(r) \rightarrow v(r)$, where $u(r)$, $v(r)$ are sets of facts. Facts are concrete statements about "properties of relations","relations between concepts", "relations between attributes of concepts".

<u>Example 3.4:</u> $\forall\ c_1, c_2 \in \mathbf{C}$:

• $rule_1$: $\{c_1$ related_to $c_2\} \rightarrow \{c_2$ related_to $c_1\}$

• $rule_2$: $\{c_1$ is_a $c_2\} \rightarrow \{c_2.\text{Attr} \subseteq c_1.\text{Attr}\}$

B. Relations between two knowledge domains

In the intellectual content of courses, the learners have to study from basic level lessons to high-level lessons. For example, some knowledge in the OOP course needs to refer to primary content in the Introduction to Programming (I2P) course.

Let $\mathcal{K}_1 = (\mathbf{C_1, R_1, Rule_1})$ be the first knowledge domain, and $\mathcal{K}_2 = (\mathbf{C_2, R_2, Rule_2})$ be the second knowledge domain as Rela-model. Then, the relations between $\mathcal{K}_1$ and $\mathcal{K}_2$ can be presented as:

$$(c_{1i}, r_{1i}, rule_{1i})\ \Theta\ (c_{2i}, r_{2i}, rule_{2i})$$

In which Θ is the relations between two knowledge domains. It is: "related to", "link to".

$$(c_{1i}, r_{1i}, rule_{1i}) \in (\mathbf{C_1, R_1, Rule_1})$$
$$(c_{2i}, r_{2i}, rule_{2i}) \in (\mathbf{C_2, R_2, Rule_2})$$

<u>Example 3.4:</u> The declaring of some concepts in the OOP course related to the declaring others in the I2P system. For example, it displayed the idea "Class" in OOP related to reporting its attributes.

```
class <class name>
{
    <access modifiers> <variable name> <variable datatype>;
    <access modifiers> <function>;
}
```

The declaring of attributes in Class is the declaration of the variable name and variable datatype in the knowledge of I2P.

C. The structure of scripts

In an intelligent searching chatbot, there are some of most everyday conversations. They are general scripts to converse with users. The system will give some questions to make requirements of users and show some suitably explaining. The following are some scenarios for searching information by chatbot:

Scenario 1: Gather data and classify questions into corresponding categories.

Scenario 2: The results of the concept or attributes of the leading knowledge.

Scenario 3: Return the known of other domains.

Scenario 4: Return the practice test.

Scenario 5: Recommend related knowledge as a question.

Fig 1 is the process of intellectual searching of a chatbot.

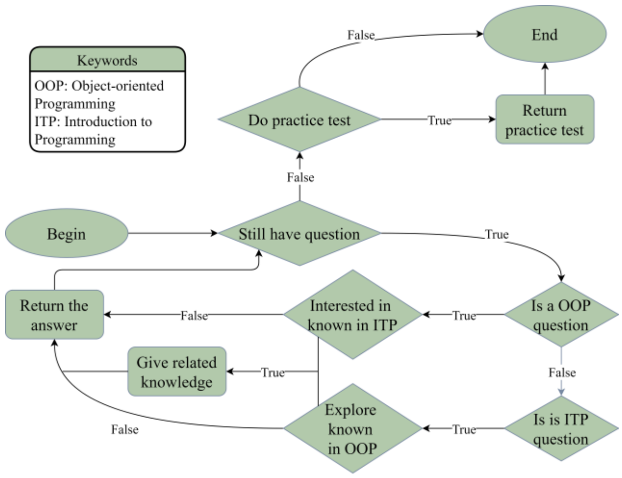


Fig 1. The process of searching

Definition 3.3 [24]: The structure of a script representing an answer-question process includes five elements:

(Name, Meaning, Question, Answer, InterRul)

In which:

- *Name*: The script's name.
- *Meaning*: The content of the script.
- *Question* = $[q_1,\ldots,q_m]$: list of questions for the corresponding script.
- *Answer* = $[a_1,\ldots,a_m]$: list of answers for the corresponding script, which each a_i is an answer for a question $q_i \in$ *Questions* ($1 \leq i \leq m$).

Questions in Ques and answers in Ans will be manually defined based on statements of the corresponding script.

- *InterRul*: It's a collection of deductive rules for selecting questions for the chatbot's script. Each rule $r \in$ *InnerRul* has the form: $u(r) \rightarrow v(r)$, where $u(r)$ is the set of facts and $v(r)$ is the result for choosing a question or an answer.

Table I presents a script for searching the definition of CLASS in the courses of OOP.

D. Integrating model of knowledge bases and scripts

The knowledge base of an intelligent searching chatbot has a component to support its conversation with users. This component contains general communication scripts. Moreover, for extracting the knowledge effectively, the knowledge base also includes multiple related knowledge domains. This section proposed the integrating model between ontology Rela-model and scripts representing a single knowledge domain for searching and the model for integrating various knowledge domains.

Definition 3.4: *The knowledge model for searching on a single domain*

The integrating model between ontology Rela-model and scripts, called Rela-Script model, has the following structure:

(C, R, Rules) + Scripts

In which, (**C, R, Rules**) is an ontology Rela-model as Definition 3.1, **Scripts**-component is a collection of scripts with the structure defined in Definition 3.2, each element in **Scripts** is a conversation frame for doing practical searching.

The Rela-Script model can be used to arrange the knowledge base of an intelligent chatbot for knowledge queries. The integration is based on the requirements listed below [6, 9, 11]:

- Build a relationship between scripts and concepts: Extract specific ideas of the knowledge domain based on the images and scripts' keyword components through the script's meaning.
- Adding more kinds of facts: Some facts related to the content of hands need to be connected with former sorts of points in the Rela model.
- Adding more rules from the scripts: When integrating the knowledge model, some rules emerged. They must be combined into the integrated model.

Table II presents the integration between ontology Rela-model and scripts.

In the real world, a knowledge domain $\mathcal{K}$ may contain some knowledge sub-domains $\mathcal{K}_i$ ($1 \le i \le n$). Each sub-domain has a distinct shape as well as particular interrelationships [27]. In this study, each sub-domain $\mathcal{K}_i$ is modeled by the Rela-Script model.

Definition 3.5: The structure of the model for multiple knowledge domains, called *Integ-Rela model*, is a tube:

(K, LINK)

In which:

- **K** = $\{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n\}$ is a collection of knowledge sub-domains, with each sub-domain, has the same form as the Rela-Script model.
- **LINK** is a collection of rules for connecting sub-domains $\mathcal{K}_i$ ($1 \le i \le n$).

The searching problems on the integrating model for multiple knowledge domains are proposed in Section IV.

IV. Problems for Searching on Integrating Knowledge Base of Chatbot

The intelligent chatbot for searching on academic knowledge can extract the knowledge being suitable with the query. It also gives some methods to apply them in practice. Besides, the system can recommend some other content related to the searching knowledge. There are three main practical requirements for a chatbot supporting the learning: Definition of a concept, applying the knowledge in practice, and the knowledge related to the searching knowledge.

A. The problems for searching on knowledge domain as Rela-model

When a query or a text is entered into the chatbot, it will be processed to retrieve the knowledge by extracting specific main keywords. Those keywords include two core components: the intent of the user and entities in a query.

The matching mechanism works by comparing the structure of the parts in the knowledge model, especially the components of concepts and relations. Inference rules aid in the deduction of other ties related to the query's content. Finally, the chatbot will display the retrieval results for the inputted query. The process for chatbot based on Rela-model is shown in Fig 2.

There are some significant problems for knowledge querying:

(1) Problem 1: **Classifying the intent of the user**. The problem will predict the kind of query entered into the system. Those terms are used to choose the appropriate conversation script of the chatbot and extract knowledge results from the knowledge base.

Intent classification could be done in various ways; Rasa NLU, for example, enables the use of a variety of models such as support vector machines (SVMs). In this study, bidirectional LSTM, a neural network architecture, is applied because of its ability to process a vector with both directions, forward and backward [28].

Solution: Fig 3 illustrates the intent classification system. There are three main stages: Tokenizer, Word Embedding, and training with Deep Bi-Directional LSTM.

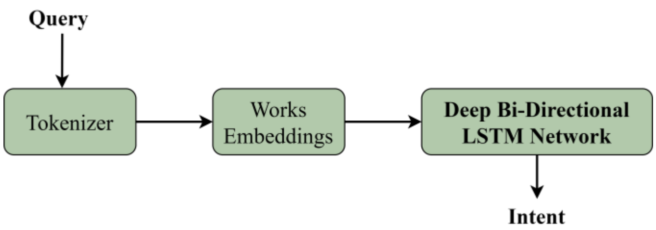


Fig 3. Intent classification architecture

PyVi toolkit [29] is used at the tokenizer stage. In the embedding stage, Word2VecVN [30] is used to choose appropriate pre-trained embedding. The Keras framework designed the deep learning model and added a softmax function into the output layer.

(2) Problem 2: **Recognizing and extracting entities in the query**. Named Entity Recognition (NER) is an fundamental challenge for NLU. In this task, the goal is to locate and to group named entities in text into any categories we choose, such as concepts (B-CON, I-CON), attributes (B-ATT, I-ATT), and properties (B-PRO, I-PRO). This approach helps identify concepts and make a comparison in more than two terms.

Solution: Fig 4 illustrates the processing of the NER task.

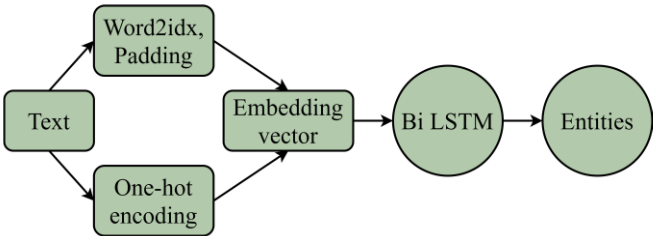


Fig 4. Named entity recognition processing

In the data processing stage, the text is transformed into a list of tuples with the following format: [(token 1, tag 1), ... (token *n*, tag *n*)]. We are then converting the data into a list of tuples. We need to ensure that all of the text is the same length to feed it to our Bi-LSTM model. After that, we do One-hot encoding for NER tags by using the Keras framework.

(3) <u>Problem 3:</u> **Matching the keywords to retrieve knowledge content.** Comparing the meaning of keywords and the material in the knowledge base is done by collecting keywords retrieved from the query. For the supplied question, this technique will identify the necessary contents and relations among concepts.

Solution: The dictionary is constructed in the first phase using the search system's knowledge base. It also comprises question terms that aid in the classification of different types of questions.

The collection of keywords from the utterance is matched to a lexicon to get a dictionary of key phrases. The search engine uses these keywords to obtain programming knowledge from ontology. Furthermore, depending on the relations between retrieval results and inference rules of the knowledge base, the system can search for related knowledge. Finally, all results will be returned as a document in the form of each type of intent.

The knowledge material is found, and a set of knowledge is returned based on the question's meaning. This search is carried out by determining whether the query's keywords and stored knowledge have same meanings. The method for solving the problem of knowledge searching has been translated in [15, 24].

(4) <u>Problem 4</u>**:** **Searching related knowledge for results of a query**. The system will retrieve the ability to answer and communicate with the user based on their question or entered text. Additionally, the system will identify some knowledge that is connected to the results of its own responses.

The system will retrieve the knowledge to answer and communicate with the user based on their query or entered text. Additionally, the system will identify some knowledge that connects to the results of its responses.

The system should be able to determine the question's target after querying it. Furthermore, based on the relationships between the query results and the knowledge base's inference rules, the system looks for relevant knowledge from the query results. Finally, all results will store as a document in the form of each type of query. The related knowledge in this study refers to a collection of knowledge related to the current search results in the ontology Rela-model that represents the knowledge domain.

B. Problem for searching on integrating knowledge bases

Let a knowledge domain $\mathcal{K}$ = ($\mathbb{K}$, $\mathbf{Link}$) integrate knowledge bases as Integ-Rela model including sub-domains $\mathcal{K}_i \in \mathbb{K}$ ($1 \le i \le n$). The searching problem on the Rela-model can decrypt some queries that only use the knowledge of a knowledge domain $\mathcal{K}_i$. However, it cannot solve the problems which require the combination of multiple knowledge domains for solving. In practice, the question will be analyzed and divided into fields. The knowledge domains will be determined and processed to extract content results through corresponding keywords sent to them.

Given a knowledge domain $\mathcal{K}$ = ($\mathbb{K}$, $\mathbf{Link}$) as Integ-Rela model, a query *q*. Using each knowledge domain connected to query q, this algorithm finds a set of knowledge content that matches the meaning of query *q*.

Input: Knowledge domain $\mathcal{K}$ as Integ-Rela model.
Query *q*.
Output: Set of knowledge content for the meaning of query *q*.
Algorithm 4.1:
Step 1:
Extract keywords in query *q*.
$W := keywords(q)$
Determine sub-domains in $\mathbb{K}$ which related to keywords in W.
$Domain := \{\mathbf{K} \in \mathbb{K} \mid \exists\, k \in W, k \text{ related to } \mathbf{K}\}$
Step 2: **Determine** other sub-domains in K connected to *Domain*.
Connect := {};
for *r* in $\mathbf{Link}$ **do**
if $\exists\, \mathbf{K} \in Domain$ and $\mathbf{K'} \in \mathbb{K} \setminus Domain$ and $r(\mathbf{K}, \mathbf{K'})$ **then**
$Connect := Connect \cup \{[\mathbf{K'}, r]\}$
od;
Step 3: Retrieve the searching results from multiple knowledge domains.
Search := {};
3.1 Extract knowledge results matching the query q.
for K in *Domain* **do**
$Key := \{k \in W \mid k \text{ related to } \mathbf{K}\}$
Search := Set of searching results on the knowledge domain **K** matching *Key* by using the method in Section IV.A
od;
3.2 Extract related knowledge from the connected sub-domains.
for *rela* in *Connect* **do** # $rela = [\mathbf{K'}, r]$
Key := Set of keywords related to *rela*.**K'** through the linked rule *rela*.*r*
Search := Set of searching results on the knowledge domain *rela*.**K'** matching *Key* by using the method in Section IV.A
od;
Step 4: Return results in *Search*.

V. Design an Intelligent Chatbot for Querying the Knowledge of Programming Courses

Courses of programming are the foundation of the IT curriculum at a university. They include Introduction to Programming (I2P), Data Structure and Algorithms, Object-Oriented Programming (OOP). This section presents the application of the Integ-Rela model to design an intelligent chatbot for querying the knowledge of I2P and OOP courses.

A. Organize the knowledge base of programming courses

The knowledge of I2P was collected from [31, 32], and OOP knowledge was obtained from [33, 34]. That knowledge covers concepts, relations, and rules of the courses. The collecting classifies in many ways, such as chapters, topics, issues, and exercises. The ontology Rela-model can fully represent each course's content, and the Integ-Rela model can connect the knowledge of both. Besides, the problems of knowledge querying knowledge are carried out using this knowledge base. The search engine fetches the results for the inputted query through the chatbot.

The primary capability of our chatbot is to search the knowledge of the course about Introduction to Programming and OOP for users' requests, such as inquiring about a concept, the technique for setting or practicing tasks, comparing two ideas, or providing relevant knowledge for querying results. Those courses cover the following topics as Table III:

TABLE III.
TOPICS OF COURSES

I2P course	OOP course
• Introduction to computer • Algorithms • Data types and operators of C++ • Control structures: condition and loop statements. • Function and procedure • Array and operators • Pointer • File	• Introduction to OOP • Class and Object • Constructor and Destructor • Friend function and friend class • Inheritance • Polymorphism • Operator overloading

B. Experimental results

The designed chatbot covers all of the essential knowledge programming and OOP principles. This chatbot plays as an instructor for coaching novices of programming learning. It can support the learner in searching the definition of a concept, applying it in practice, and the content related to the searching knowledge. Table IV shows the testing results of the chatbot for six different intents.

TABLE IV.
RESULTS OF TESTING FOR 06 KINDS OF QUERY

Intent	Quantity	Correct	Accuracy
Greeting	25	18	72.00%
Bye	22	17	77.27%
Concept	314	284	90.45%
Setting	282	188	66.67%
Comparison	295	245	83.05%
Related knowledge	150	128	85.33%
Out-of-scope	178	142	79.78%
Total	**1,243**	**1,034**	**79.22%**

The chatbot is designed to extract the requirements about determining a concept, its attributes, and properties. Thus, Table V presents the results of classification into three kinds of named entities recognition: concepts (CON), attributes (ATT), and properties (PROP).

Table VI shows the outcomes of the chatbot testing. Students input their questions to the chatbot. The system classifies queries into intents, concepts, comparisons, and settings; then, it will extract results for students. The process of the chatbot is also stored and checked by lecturers who are teaching those courses. Fig 6 (including Fig 6a and 6b) shows the accuracy of the chatbot for each course.

TABLE V.
RESULTS OF TESTING FOR NAMED ENTITIES RECOGNITION TASK

Entity	Quantity	Precision	Recall	F1
CON	525	75.44%	81.33%	77.30%
ATT	617	78.84%	77.31%	78.07%
PRO	608	80.14%	76.32%	78.18%

In general, the performance of the comparison queries is lower than both kinds of queries about concept and setting. Because the accuracy of the NER stage is not good in Vietnamese, those results also affect the accuracy of these queries containing two domain knowledge. Besides, the development of testing on combining questions, which need to use knowledge from both courses to receive correct results, get the accuracy being 71%. This thing shows that the proposed method works effectively for retrieval knowledge from multiple knowledge domains.

VI. DISCUSSIONS

Nowadays, many systems support learning, such as algebra in geometry education [35], Symbolab in mathematics [36]. Besides, some programs such as chatbots are built to tutor programming knowledge, such as ChiQuat-Tutor, to understand Computer Science [37] and Python-Bot for teaching programming in Python [22]. However, the intelligent chatbot for querying the knowledge of I2P and OOP courses in this study is constructed based on the integrating knowledge base, and it can communicate with learners through Vietnamese. The designed chatbot has full knowledge of both courses, which were organized similarly to their curriculums. Hence, the system can give suitable and useful content for the queries of students.

Wolfram Alpha is a knowledge search engine that covers a wide range of topics [38]. It can extract some programming expertise. Moreover, it helps learners better grasp your assignment by following the stages and tips that lead them from challenges to solutions.

FPT.AI Conversation is a platform that allows you to create and develop chatbots with user interfaces [39]. It can support many languages, such as Vietnamese, English, Malaysian, Indonesian, and Japanese. Users may create scripts for their work by creating pairs of questions and answers.

ProgBot is a chatbot for teaching JAVA, C++ and Python [40]. This bot can support the user to get the syntax of the program with its examples. Besides, it can give some quiz to test the knowledge of user.

Table VII shows the comparison between our system and other programs based on criteria of an intelligent system in education [3, 5, 41]:

- Sufficiently knowledge base: The system stores contents of the course altogether, and those contents are suitable with its syllabus.
- Ability to extract required knowledge: The system can retrieve information based on the inputted query, and the results meet users' requirements.
- Supporting the studying: The system support users to learn the knowledge of the course. It plays as a tutor to focus some actual contents of lessons.

VII. CONCLUSION AND FUTURE WORK

This paper proposed a model integrating multiple knowledge domains, called the Integ-Rela model. This integrating model can represent knowledge bases as the

Rela-Script model, which combines the knowledge model of relations and the structure of general scripts. The Integ-Rela model is equipped to build the knowledge base of an intelligent searching chatbot. Besides, the problems for knowledge searching on a domain are studied and solved. The difficulties for information retrieval from multiple knowledge domains are also designed. Those problems are applied to develop an intelligent chatbot for querying on integrated knowledge bases: searching definitions of concepts, explaining how to use the knowledge in practice, and recommending the releated knowledge.

The proposed method is applied to construct a conversational chatbot for learning Introduction to Programming and Object-oriented Programming courses in a university. This chatbot utilizes the approach of machine learning and natural language understanding such as LSTM network and named entity recognition for intents classification and information retrieval. It plays as a tutor to communicate with students in Vietnamese.

The experimental results show that it is helpful to find out the content of lessons, the meaning of definitions, how to apply them in practice and get information which are related to search results.

In the future, we will continue to enhance the knowledge base for designing the tutoring chatbot for solving problems in courses [42, 43, 44]. It can give some tips for students to find the step-by-step solution of an exercise [45]. This method also can apply to build a tutoring chatbot for problem-solving in mathematical courses [46, 47]. In addition, the natural language processing for Vietnamese will be improved to increase the analysis of queries' meaning more exactly through their grammar structure [48, 49] and retrieve knowledge from many sources more precisely. Moreover, the knowledge model will be studied more to integrate the knowledge of operators and functions [50, 51] for representing the knowledge base of tutoring chatbot in courses of STEM education [3, 52].

REFERENCES

[1] N. Noy, et al., "Final Report on the 2013 NSF Workshop on Research Challenges and Opportunities in Knowledge Representation". *National Science Foundation Workshop Report*, August 2013.

[2] C. Li, A. Li, Y. Wang, H. Tu, Y. S, "A Survey on Approaches and Applications of Knowledge Representation Learning", in *Proc. 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, Hongkong, China, 2020, pp. 312-319.

[3] H.D. Nguyen, N.V. Do, P.N. Tran, X.H. Pham, V.T. Pham, "Some criteria of the Knowledge Representation method for an Intelligent Problem Solver in STEM education", *Applied Computational Intelligence and Soft Computing*, Vol. 2020, Article ID 9834218, 2020.

[4] Y. Xie, M. Fang, K. Shauman, "STEM Education", *Annual Review of Sociology*, vol. 41, pp. 331 – 357, 2015.

[5] I. Hatzilygeroudis, J. Prentzas, "Chapter 9: Knowledge Representation in Intelligent Educational Systems", in *Web-Based Intelligent E-Learning Systems: Technologies and Applications*, Zongmin Ma (Eds.), IGI publisher, 2006.

[6] H. Nguyen, N. Do, V. Pham, "Chapter 10 - A method for designing Knowledge-based Systems and Application", in *Applications Computational Intelligengence Multi Disciplinary Research*, A. Elgnar et al. (Eds.), Academic Press, Elsevier, 2022.

[7] F. Colace, M. Santo, M. Lombardi, et al, "Chatbot for E-Learning: A Case of Study", *International Journal of Mechanical Engineering and Robotics Research*, Vol. 7, No. 5, 2018, pp. 528-533.

[8] H. Nguyen, D. Tran, H. Do, V. Pham, "Design an intelligent system to automatically tutor the method for solving problems", *International Journal of Integrated Engineering*, Vol. 12, No. 7, 2020, pp. 211 – 223.

[9] K. Munir, M. Anjum, "The use of ontologies for effective knowledge modelling and information retrieval", *Applied Computing and Informatics*, Vol. 14, No. 2, 2018, pp. 116 – 126.

[10] M. Dumontier, C.J. Baker, J. Baran, et al, The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery. *J. Biomed. Semant.*, Vol. 5, Article number 14, 2014.

[11] N.V. Do, H.D. Nguyen, T.T. Mai, "A Method of Ontology Integration for Designing Intelligent Problem Solvers", *Applied Sciences*, Vol. 9, No. 18, 3793, 2019.

[12] N.V. Do, H. Nguyen, L. Hoang, Some Techniques for Intelligent Searching on Ontology-based Knowledge domain in E-learning, in *Proc. 12th International Conference on Knowledge Engineering and Ontology Development (KEOD)*, Hungary, 2020, pp. 313 – 320.

[13] S. Lin, J. Li, C. Yu, "Dynamic Data Driven-based Automatic Clustering and Semantic Annotation for Internet of Things Sensor Data", *Sensors and Materials*, Vol. 31, No. 6, pp. 1789 – 1901, 2019.

[14] N. Do, H.D. 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, 2018, pp. 1047 – 1090.

[15] X.T. Pham, T.V. Tran, V. Nguyen-Le, V. Pham, H. Nguyen, "Build a search engine for the knowledge of the course about Introduction to Programming based on ontology Rela-model", in *Proc. 12th IEEE International Conference on Knowledge and Systems Engineering (KSE)*, Can Tho, Vietnam, 2020, pp. 207 – 212.

[16] P. Smutny, P. Schreiberova, "Chatbots for learning: A review of educational chatbots for the Facebook Messenger", *Computers & Education*, Vol. 151, 103862, 2020.

[17] A. Tjiptomongsoguno, A. Chen, H. Sanyoto, E. Irwansyah, B. Kanigoro, Medical Chatbot Techniques: A Review, in *Proc. Computational Methods in Systems and Software (CoMeSySo 2020)*, Vsetin, Czechia, 2020, pp. 346 – 356.

[18] A. Almurayah, "The Challenges of Using Abraic Chatbot in Saudi Universities," IAENG International Journal of Computer Science, vol. 48, no. 1, pp190 – 201, 2021.

[19] D. Altinok, An Ontology-Based Dialogue Management System for Banking and Finance Dialogue Systems, in *Proc. 7th International Conference on Language Resources and Evaluation (LREC)*, Paris, France, 2018.

[20] Q. Bao, L. Ni, J. Liu, "HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention", in *Proc. the Australasian Computer Science Week Multi-conference (ACSW 2020)*, Melbourne, Australia, 2020, pp. 1- 10.

[21] L. Zhou, J. Gao, D. Li, H. Shum, "The Design and Implementation of XiaoIce, an Empathetic Social Chatbot", *Computational Linguistics*, Vol. 46, No. 1, 2020, pp. 53 – 93.

[22] O. Chinedu, A. Ade-Ibijola, "*Python-Bot: A Chatbot for Teaching Python Programming*," Engineering Letters, vol. 29, no. 1, pp25 – 34, 2021.

[23] H.D. Nguyen, V.T. Pham, D.A. Tran, T. Le, "Intelligent tutoring chatbot for solving mathematical problems in High-school", in *Proc. 11th IEEE International Conference on Knowledge and Systems Engineering (KSE 2019)*, Da Nang, Vietnam, 2019, pp. 145-150.

[24] H. Nguyen, T. Tran, X. Pham, A. Huynh, N. Do, "Ontology-based Integration of Knowledge Base for Building an Intelligent Searching Chatbot", *Sensors and Materials*, Vol. 33, No. 9, pp. 3101 – 3123, 2021.

[25] L.K. Fryer, D. Coniam, R. Carpenter, D. Lăpușneanu, "Bots for language learning now: Current and future directions", *Language Learning & Technology*, Vol. 24, No. 2, 2020, pp. 8–22.

[26] H. Shum, X. He, D. Li, "From Eliza to XiaoIce: challenges and opportunities with social chatbots", *Frontiers Information Technology Electronic Engineering*, Vol. 19, 2018, pp. 10–26.

[27] H. Nguyen, N. Do, V. Pham, A. Selamat, E. Herrera-Viedma, "A method for knowledge representation to design Intelligent Problems Solver in mathematics based on Rela-Ops model", *IEEE Access*, Vol. 8, 2020, pp. 76991–77012.

[28] M. Basaldella, E. Antolli, G. Serra, C. Tasso, "Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction", in *Proc. Italian Research Conference on Digital Libraries (IRCDL 2018)*, Udine, Italy, 2018, pp. 180 – 187.

[29] Pyvi: https://pypi.org/project/pyvi

[30] Word2vecVN: https://github.com/sonvx/word2vecVN

[31] T. Tran, P. Nguyen, T. Dinh, T. Tran, *Introduction to Programming*, Publishes of Science and Technology, 2015. (in Vietnamese).

[32] J. Hubbard, *Theory and Problems of Fundamentals of Computing with C++*, McGraw-Hill, 1998.

[33] E. Balagurusamy, *Object Oriented Programming with C++ (8th ed.)*, McGraw-Hill, 2019.

[34] R. Johnsonbaugh, M. Kalin, *Object Oriented Programming in C++ (2nd ed.)*, Pearson, 1999.
[35] Z. Kovács , T. Recio , P. Richard , S. Vaerenbergh, M. Vélez, "Towards an ecosystem for computer-supported geometric reasoning", *International Journal of Mathematical Education in Science and Technology*, 2020. DOI: 10.1080/0020739X.2020.1837400
[36] Symbolab: https://www.symbolab.com/ (2021, September 20).
[37] O. Alzoubi, C. Brown, "ChiQuat-Tutor and its Architecture", in *Intelligent Support for Computer Science Education*, B. Eugenio, D. Fossati, N. Green (Eds.), CRC press, 2021.
[38] Wolfram Alpha: https://www.wolframalpha.com/ (2021, July 12).
[39] FPT.AI: https://fpt.ai/chatbot (2021, September 20).
[40] J. Sadhasivam, M. Ametwala, A. Mazumdar, "Implementation of Chatbot That Teach Programming Language", *Journal of Xi'an University of Architecture & Technology*, Vol. 2, No. (3), 2020, pp. 1095 – 1101.
[41] M. Kandlhofer, G. Steinbauer, S. Hirschmugl-Gaisch, P. Huber, "Artificial intelligence and computer science in education: From kindergarten to university", *Proceedings of 2016 IEEE Frontiers in Education Conference (FIE 2016)*, PA, USA, Oct. 2016.
[42] M. Phan, H. Nguyen, D. Tran, T. Le, N.P. Tran, "Design an Intelligent Problem Solver in Geometry based on Knowledge Model of Relations", *Engineering Letters*, Vol. 28, No. 4, 2020, pp. 1108 – 1117.
[43] X. Sánchez-Díaz, G. Ayala-Bastidas, P. Fonseca-Ortiz, L. Garrido, "*A Knowledge-Based Methodology for Building a Conversational Chatbot as an Intelligent Tutor*", in Proc. Mexican International Conference on Artificial Intelligence (MICAI 2018), Guadalajara, Mexico, 2018, pp. 165 – 175.
[44] L. Neagu, E. Rigaud, V. Guarnieri, et al, "Selfit–An Intelligent Tutoring System for Psychomotor Development", in Proc. International Conference on Intelligent Tutoring Systems (ITS 2021), Athens, Greece, 2021, pp. 291 – 295.
[45] H.D. Nguyen, L. Nguyen, N. Tran, V.T. Nguyen-Le, S. Vu, "Design an Intelligent Problem Solver in Mathematics based on Integrated-Knowledge Model", Proceedings of 13th IEEE International Conference on Knowledge and Systems Engineering (KSE 2021), Bangkok, Thailand, Nov. 2021.
[46] N. Do, H. Nguyen, "A Reasoning method on Knowledge Base of Computational Objects and Designing a System for automatically solving plane geometry problems", Proceedings of World Congress on Engineering and Computer Science 2011 (WCECS 2011), San Francisco, USA, October 2011, pp294-299.
[47] M. Eppe, C. Gumbsch, M. Kerzel, et al., "Intelligent problem-solving as integrated hierarchical reinforcement learning", *Natural Machine Intelligence*, Vol. 4, 2022, pp. 11–20.
[48] A. Aiti-Mlouk, L. Jiang, "KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data", *IEEE Access*, Vol. 8, 2020, pp. 149220 – 149230.
[49] H. Nguyen, T. Huynh, S.N. Hoang, V.T. Pham, I. Zelinka, "Language-oriented Sentiment Analysis based on the grammar structure and improved Self-attention network", in *Proc. of 15th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE 2020), Praha, Czech Public, 2020, pp. 339-346.
[50] N.V. Do, H. Nguyen, T. Mai, "Reasoning Method on Knowledge about Functions and Operators", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, no. 6, 2015, pp. 156 – 168.
[51] V.N. Do, H.D. Nguyen, "Reducing model of COKB about Operators Knlwedge and Solving problems about Operators", *New Trends in Computational Collective Intelligence*, Studies in Computational Intelligence, vol. 572, 2015.
[52] M. Honey, et al., *STEM education for the future – A vision report*, National Science Foundation, 2020.

Hien D. Nguyen received his B.S. and M.S. degrees from University of Sciences, VNU-HCM, Vietnam, in 2008 and 2011, respectively. He received his Ph.D degree from University of Information Technology, VNU-HCM, in 2020. He is a senior lecturer at the Faculty of Computer Science, University of Information Technology, VNU-HCM. His research interests include knowledge representation, automated reasoning, and knowledge engineering, especially intelligent systems in education, such as intelligent problem solvers. He received the Best Presentation Clip Award at AI-Socha (Ho Chi Minh City) in 2020 and Incentive Prizes of the Vietnam Fund for Scientific and Technological Creations (VIFOTEC) Award and the Technological Creation Awards of Binh Duong province, in 2016 and 2015, respectively.

Tuan-Vi Tran is an undergraduate student at the Faculty of Computer Science, University of Information Technology (UIT), VNU-HCM, Vietnam. His research interests are knowledge presentation and natural language processing, especially applying NLP to intelligent systems in education, such as constituency parsing and classification problems in Vietnamese. He received the Best Student Paper Award at the 12th International Conference on Knowledge and System Engineering (KSE 2020) and the Incentive Prize of the Technological Creation Awards of Binh Duong province in 2021.

Xuan-Thien Pham is an undergraduate student at the Faculty of Computer Science, University of Information Technology (UIT), VNU-HCM, Vietnam. His research interests are knowledge-based systems and automated reasoning, especially building intelligent systems in education. He received the Best Student Paper Award at the 12th International Conference on Knowledge and System Engineering (KSE 2020) and the Incentive Prize of the Technological Creation Awards of Binh Duong province in 2021.

Anh T. Huynh received his B.S. and M.Sc. degrees from University of Information Technology (UIT), VNU-HCM, Vietnam, in 2011 and 2014, respectively. From 2011 to 2014, he was an assistant lecturer at the Faculty of Software Engineering, UIT. Since 2014, he has been a lecturer at the Faculty of Software Engineering, UIT. His research interests are in ontology, knowledge bases, NLP, and machine learning. He received the Incentive Prize of the Technological Creation Awards of Binh Duong province in 2021.

Vuong T. Pham received his B.S. and M.Sc. from University of Sciences, VNU-HCM, Vietnam, in 2003 and 2008. From 2003 – 2006, he was a lecturer at University of Sciences, VNU-HCM, Vietnam. From 2006 – 2019, he was a lecturer at University of Information Technology, VNU-HCM, Vietnam. He also was the Vice Dean of Faculty of Software Engineering, University of Information Technology, VNU-HCM from 2012 – 2019. He is currently a lecturer at the Faculty of Information Technology, Sai Gon University, Vietnam. His research interests include artificial intelligence, software engineering, game development.

Diem Nguyen received her B.S. and M.S. degrees from University of Information Technology (UIT), VNU-HCM, Vietnam, in 2011 and 2015, respectively. From 2011 to 2015, she was an assistant lecturer at the Faculty of Computer Science, UIT. Since 2015, she has been a lecturer at the Faculty of Computer Science, UIT. Her research interests are in ontology, knowledge-based system, intelligent systems.

TABLE I.
THE SCRIPT FOR SEARCHING THE DEFINITION "CLASS"

Name	*Meaning*	*Question*	*Answer*	*InterRul*
Searching definition of a concept	Conversation about searching Class	q_1: What is a Class's definition?	a_1: It is a user-defined data type with its own set of data members and member methods that can be accessed and used by establishing a class instance. A C++ class is similar to an object's blueprint.	r_1: $\forall\ i, 1 \le i \le 7$ $\{q_i$.answer = "Yes"$\}$ $\rightarrow$ {**show** a_i} r_2: $\{q_2$.answer = "No"$\}$ $\rightarrow$ {**goto** q_3} r_3: $\{a_3$.answer has a choice$\}$ $\rightarrow$ {**goto** q_1 with that choice}
		q_2: Do you want to brush up on your skills by answering some practice questions?	a_2: Write a class with two private variables and one member function, which will return the rectangle area.	
		q_3: Do you need some other knowledge related to Class?	a_3: Show related knowledge with Class.	
		q_4: What is notice about variable when declare a class ?	a_4: When declaring a class, each variable must declare the variable name, variable datatype. Variable name must begin with a letter of the alphabet or an underscore (__). After the first initial letter, variable names can also contain letters and numbers.	
The question of amount		q_5: How many types of access modifiers in C++?	a_5: C++ uses the 03 modifiers called public, protected, and private.	
		q_6: How many various types of constructors?	a_6: There are three types of constructors: – Default Constructor, with no parameters. – Parametric Constructor, with Parameters. Create a new instance of a class and also pass arguments simultaneously. – Copy Constructor, which creates a new object as a copy of an existing object.	
The question of comparison		q_7: What exactly is the distinction between a class and an object?	a_7: A class is a template for objects. A class defines object properties, including a valid range of values and a default value. A class also describes object behavior. An object is a member or an "instance" of a class. An object has a state in which all of its properties have values that you either explicitly define or define by default settings.	

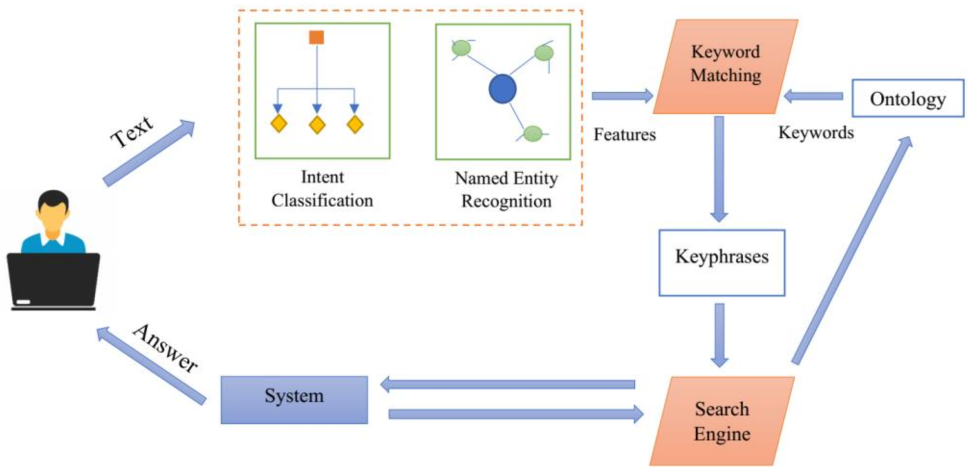
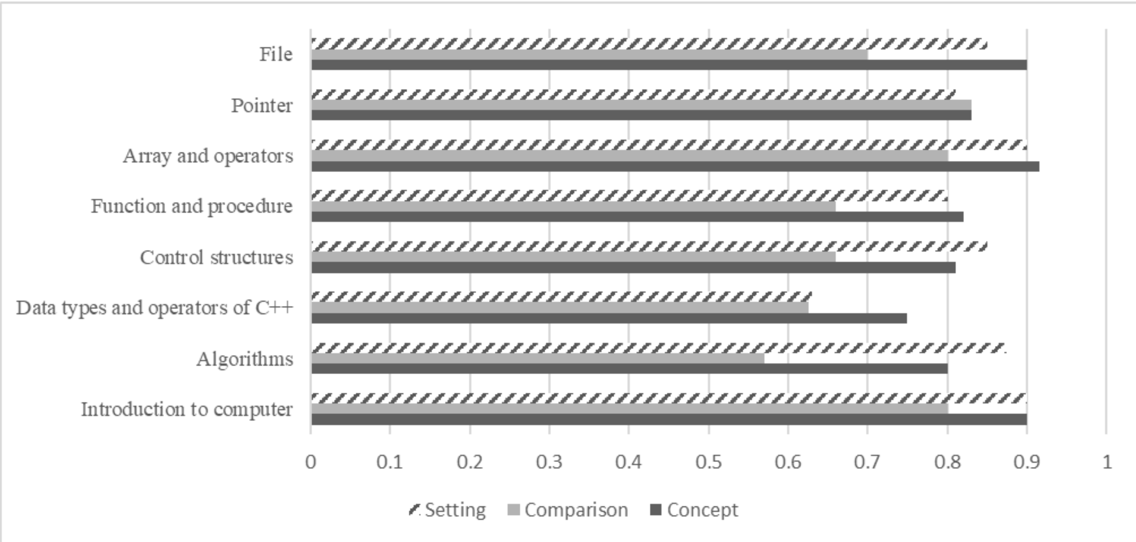


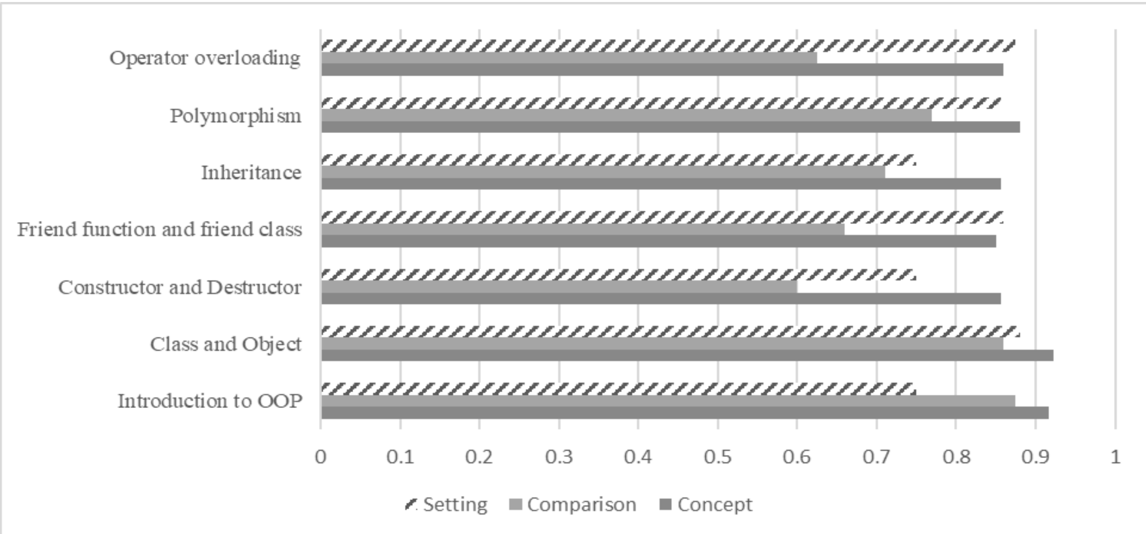
Fig 2. The searching process of chatbot based on Rela-model

TABLE II.
THE INTEGRATING BETWEEN ONTOLOGY RELA-MODEL AND SCRIPTS

Relations between a script and a concept	Adding Facts on Rela-Script model	Adding kinds of rules on Rela-Script model
Structure of a script s: s := (*Name*, *Meaning*, *Question*, *Answer*, *InterRul*) The script s is related to the concept c $\Leftrightarrow$ keywords($s.Meaning$) $\cap$ $c.Key \neq \varnothing$ where, keywords(*content*) are a set of keywords of the *content*.	f_1. Determine a pair of question and answer. Specification: (q_k, a_k) Condition: $q_k \in s.Ques$, $a_k \in s.Ans$, with $s \in Scripts$ f_2. Display an answer or a question Specification: Show(u) Condition: $u \in s.Ques \cup s.Ans$, $s \in Scripts$	r is a combining deductive rule. It has the form: $u(r) \rightarrow v(r)$ and $\neg u(r) \rightarrow s(r)$ where $u(r)$ is the set of facts and $v(r)$, $s(r)$ are results for choosing a question or an answer.



a. Results of knowledge querying on I2P course.



b. Results of knowledge querying on OOP course.

Fig 6. Results of testing on knowledge content of I2P and OOP courses.

TABLE VI.
RESULTS OF TESTING ON KNOWLEDGE CONTENT OF COURSES

Course	Content	Number of queries	Number of Correct results				Proportion (%)
			Concept	Comparison	Setting	Total	
Introduction to Programming	Introduction to computer	15	7	5	2	14	93%
	Algorithms	28	10	7	8	25	89%
	Data types and operators of C++	36	12	8	11	31	86%
	Control structures: condition and loop statements.	65	21	18	20	59	91%
	Function and procedure	76	23	15	25	63	83%
	Array and operators	70	24	16	22	62	89%
	Pointer	50	14	14	13	41	82%
	File	28	10	7	6	23	82%
	Total	**368**	**121**	**90**	**107**	**318**	**86%**
Object-oriented Programming	Introduction to OOP	27	12	8	4	24	89%
	Class and Object	66	26	15	18	59	89%
	Constructor and Destructor	50	14	10	12	36	72%
	Friend function and friend class	55	17	12	15	44	80%
	Inheritance	80	28	14	20	62	78%
	Polymorphism	62	18	13	21	52	84%
	Operator overloading	49	15	8	16	39	80%
	Total	**389**	**130**	**80**	**106**	**316**	**81%**
	Combine (*)	100	31	18	22	71	71%
	Total	**891**	**284**	**188**	**245**	**717**	**80%**

(*) Combine: Those are queries which need to use knowledge from both courses to receive correct results.

TABLE VII.
COMPARISON WITH OTHER PROGRAMS

<table>
<tr><th>Criteria</th><th>FPT.AI Conversation</th><th>Progbot</th><th>Wolfram Alpha</th><th>Our program</th></tr>
<tr><td rowspan="2">Sufficiently knowledge base</td><td rowspan="2">It just allows you to make a simple script by including a query and a response. Thus, it is only efficient for simple conversations.</td><td rowspan="2">• The knowledge of this bot includes some knowledge in JAVA, C++ and Python.
• The knowledge is organized as database. It is not represented following the curriculum of a course.</td><td colspan="2">The knowledge of programming is adequate for searching on basic knowledge.</td></tr>
<tr><td>• Its knowledge comes from a variety of sources. It's not structured like a course.</td><td>• Its knowledge base is structured according to the material of this course's curriculum.
• Its knowledge base is integrated by both components' knowledge base: Intro to programming and OOP.</td></tr>
<tr><td rowspan="2">Ability for extracting required knowledge</td><td rowspan="2">• It provides a built-in core NLU engine that makes us difficult to custom and fine-tune.
• It can only respond to individual queries and cannot link to relevant information.</td><td rowspan="2">• This bot can get the syntax of the program with its examples.
• The results are simple. Because its data is not organized as a completely course, it still lacks of some foundation knowledge in programming.</td><td colspan="2">• The system can retrieve information based on the meaning of the query entered.
• It also provides additional information about the primary search results.</td></tr>
<tr><td>• The system only processes a single query.
• Relationships between terms in the query and stored knowledge are used to retrieve related knowledge.</td><td>• Users interact with the system using a communication script. As a result, it can enable them in better comprehending the material they are looking for.
• The related knowledge is derived from the relationships between the search results and the query's meaning.</td></tr>
<tr><td>Supporting the studying</td><td>The system is designed for common users and does not include any educational features:
• It is difficult to organize and integrate knowledge.
• The capacity to respond is used to determine if a bot is known or unknown.</td><td>The system can give some quiz to test the knowledge of user. However, it has not yet been enough to support the learning:
• The data is not designed following the curriculum of a determine course.
• There is not the connection between the knowledge in different programming languages.</td><td>The system is designed for common users and does not include any educational features:
• The search results only provide answers to the question. It does not belong in any of the course's instructional units.
• The corresponding results are automatically extracted. Thus, some of them are not necessary for users</td><td>The system aims to service for studying:
• The knowledge base is structured according to the course's syllabus. As a consequence, the search results assist in better comprehending the knowledge's lesson.
• The bot's conversation serves as a tutor, explaining the significance of the knowledge material. As a result, consumers require the extracted knowledge.</td></tr>
</table>