Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 26 (2023), No. 3, pp. 449–473
DOI : 10.47974/JIM-1674

Design an ontology-based model for intelligent querying system in mathematics education

Hien D. Nguyen [‡]
University of Information Technology
Ho Chi Minh city
Vietnam

and

Vietnam National University
Ho Chi Minh city
Vietnam

Hoa Huynh [†]
Industrial University of Ho Chi Minh city (IUH)
Ho Chi Minh city
Vietnam

Thanh T. Mai [§]
Hong Bang International University
Ho Chi Minh city
Vietnam

Dung A. Tran [*]
Diem Nguyen [@]
University of Information Technology
Ho Chi Minh City
Vietnam

and

This research is funded by University of Information Technology – Vietnam National University HoChiMinh City, under grant number D1-2022-38.

[‡] *E-mail:* `hiennd@uit.edu.vn`
[†] *E-mail:* `hqhoa99@gmail.com`
[§] *E-mail:* `thanhmt@hiu.vn`
[*] *E-mail:* `dungta@uit.edu.vn(Corresponding author)`
[@] *E-mail:* `diemntn@uit.edu.vn`

Vietnam National University
Ho Chi Minh city
Vietnam

Van-Thanh Nguyen-Le [#]
HCMC University of Technology and Education
Ho Chi Minh city
Vietnam

Abstract

Distance learning can be facilitated through e-learning, and students can enhance their understanding of course materials through the use of an intelligent querying system that enables self-learning. In this paper, an ontology-based method for organizing the knowledge of mathematics in high-school is presented. Ontology Rela-Ops model, which is a knowledge model for representing the knowledge of relations and operators, is combined with the knowledge component of functions. Based on the combining model, a structure of knowledge graph is proposed to represent relations of intellectual contents. An intelligent querying system is designed for the domain of high-school mathematics using these structures. This searching system can query contents of the course and extracts other knowledge related to querying results. The experimental results show that the proposed method is useful and effective to apply in the practice.

Subject Classification: *00A05 (Mathematics in general), 03B70 (Logic in computer science).*

Keywords: *Ontology, Knowledge graph, Intelligent searching system, High-school mathematics, Education, Knowledge engineering.*

1. Introduction

In the fourth industry, e-learning is a modern way for studying and is increasing its usage daily [1, 2]. For the effectiveness of this kind of learning, it needs an intelligent system to support learners for searching the knowledge of courses more conveniently [3]. Based on standards for an intelligent educational system, the requirements of an intelligent search system in education are determined [4, 5]:

- To represent the knowledge of courses, the system has a comprehensive knowledge base.
- The system has the ability to search and query data from educational resources. It can interpret the meaning of a query and retrieve from its knowledge base results that are appropriate for that interpretation.

[#] *E-mail:* **thanhnlv@hcmute.edu.vn**

- The system can foresee the user's requirements and make recommendations based on current search results

Ontologies make it easier to gain access to knowledge and transmit information between people and heterogeneous systems [6, 7]. The ability to diverse distributed application systems is one of features of an ontology [8, 9]; hence, an intelligent search engine's knowledge base can be represented by using ontology [10], particularly e-learning search engines. In [11], an ontology including concepts, relations between concepts, operators, rules, problems and their solving methods, is proposed. The application of this ontology leads to the design of an intelligent search system for high-school mathematics knowledge.

The study of mathematics is crucial for high school students [12, 13]. Students are in great need of assistance in order to master this material, particularly when they study for their final tests. This paper presents a method for organizing high-school mathematics knowledge using an ontology-based approach. This approach involves integrating the Rela-Ops model, which represents relations and operators, with the knowledge component of functions. Based on the combining model, a structure of knowledge graph is proposed to represent relations of intellectual contents. Those structures are used to design a knowledge-based query engine for high school maths. The built system can organize this knowledge domain completely. The system utilizes the knowledge base to retrieve the course query contents and extract additional information related to the query results, students need to know those details in order to help their learning.

The paper is organized as follows. Some related works for designing of intelligent systems in education are presented in the next section. Section 3 introduces an ontology for knowledge of relations and operators, Rela-Ops model. This section also proposes a Rela-Ops model for integrating ontologies and knowledge of functions [14], *Onto-Funcs model*. It presents a structure of the knowledge graph to clearly describe relations between components of Onto-Funcs model. Section 4 of the document details the issues related to intelligent searching within the Onto-Funcs ontology, along with their corresponding solutions. There are two problems for searching: Classifying the input query and extracting main keyphrases of the query, and Matching the keyphrases with the content of the knowledge. The architecture of the search system, which is based on the knowledge model of functions, is presented in Section 5. The experimental results are shown in this section. Finally, we draw conclusions and consider future work in Section 6.

2. Related work

Intelligent Tutoring systems (ITSs) are learning systems incorporated in order to create more individualized educational systems [15]. Intelligent searching system is a part of an ITS. This system can support the querying on the knowledge of courses for students. There are many artificial intelligent techniques applied in intelligent searching systems are applied in ITSs in education, such as: ontology, rule-based reasoning, data mining, and Bayesian network [16].

The model of intelligent tutoring system for learner assessment was studies in [17, 18]. Authors in [17] was developed based on a Bayesian network. In an online learning environment, the Bayesian network was used to determine the student's current level of knowledge so that the model might develop and provide new knowledge to enhance learner performance. However, this model did not have a component to store the content of learning courses.

EduplaX[1] is a platform for students to study and self-testing their knowledge in high-shool mathematics. This system can assess the level of understanding of students. It also can track users' progress through their tests and diagnose the content that needs to be enhanced based on the test results. Nevertheless, the function for searching on this knowledge domain only shows contents in each lesson, it has not yet worked on an inputted query.

Ontology COKB (Computational Objects Knowledge Base) was employed to represent the knowledge domains of graph theory [19] and discrete mathematics [20]. However, this ontology is too complex for designing an intelligent searching system, it is suitable for intelligent problem solver systems. Rela-model is an intellectual model of relations as ontology. This model and its extension was used to organize university courses about algorithms [21, 22]. Nonetheless, this model lacks of the knowledge of functions. Hence, Rela-model is not enough to design the knowledge base for querying on the knowledge domain about mathematics with contents of functions.

3. Ontology for Knowledge of Functions and Knowledge graph

3.1 *Ontology Rela-Ops model*

Definition 3.1 [23]: The knowledge model of relations and operators, called *Rela-Ops model*, is an ontology including 04 components:

$$(\mathbf{C, R, Ops, Rules})$$

[1] EduplaX: https://www.eduplax.com/

where, **C** is the set of concepts which are classes of objects, each concept has a structure to represent concepts of the domain course; **R** is the set of relations between concepts in **C**; **Ops** is the set of operators; and **Rules** is the set of inference rules. The structure of each component are presented as follows.

3.1.1 **C** – set of concepts

The structure of each concept $c \in$ **C** is:
(Symbol, Attrs, Inner-Rela, Inner-Rul, Contents, Chapter, Key)

In which,

- *Symbol*: a text or keyphrase to recognize an idea in the field of knowledge.
- *Attrs*: a set of a concept *c*'s attributes.
- *Inner-Rela*: The set of relations among attributes in *c.Attrs*.
- *Inner-Rul*: set of inner rules of concept *c*. These rules are used to deduce other attributes from determined attributes in concept *c*.
- *Synonym*: it is a set of synonyms with concept *c*.
- *Contents*: Information of the concept *c*. They include some text paragraphs. Each paragraph is specified as [*name_par*, *content_par]* where *name_par* is the paragraph's ID and *content_par* is its content.
- *Chapter*: the information about chapter, section in the curriculum of the course.
- *Key*: set of keywords related to the corresponding concept c.

Table 1 is an example of description the concept about LINEAR_ EQUATION in the course of Linear Algebra.

Table 1
The concept about LINEAR_EQUATION

Component	Content
Symbol	$a.x + b = 0$
Attrs	$\{a: \mathbb{R}, b: \mathbb{R}; x\text{: variable}\}$ where $\mathbb{R}$ is a set of real nmbers.
Inner-Rela	{ }
Inner-Rul	r_1: $\{b \neq 0\} \rightarrow \{x = -b/a\}$ r_2: $\{b = 0, a \neq 0\} \rightarrow \{x\text{: NULL}\}$ (there is not solution for the equation) r_3: $\{b = 0, a = 0\} \rightarrow \{x \in \mathbb{R}\}$

Contd...

Synonym	Linear equation, first-order equation, equation of line
Content	*Content* := [[$name_1$, $content_1$], [$name_2$, $content_2$]] • $name_1$:= "*Definition*" $content_1$:= "A linear equation is an algebraic equation where each term has an exponent of 1 and when this equation is graphed, it always results in a straight line." • $name_2$:= "*Form*" $content_2$:= "The standard form or the general form of linear equations in one variable is written as, $a*x + b = 0$; where a and b are real numbers, and x is a single variable."
Chapter	Grade 10, Chapter 3, Unit 2.
Key	algebraic equation, straight line, linear equations system, quadric equation.

3.1.2 **R** – set of relations

Let $r \in \mathbf{R}$, $r \subseteq \mathbf{C} \times \mathbf{C}$. This implies that r is a binary relation between the concepts in **C**, which can be categorized into two types of relations in R - hierarchical relations and non-hierarchical relations..

$$\mathbf{R} = \mathbf{R}_{\mathbf{hicher}} \cup \mathbf{R}_{\mathbf{others}}$$

In which,

- $\mathbf{R}_{\mathbf{hicher}}$ = {is-a, include} consists of two relations: "is-a" and "include", where:

$$\forall c_i, c_j \in \mathrm{C},\ c_i \text{ is-}a\ c_j \Leftrightarrow \begin{cases} c_j.Attrs \subseteq c_i.Attrs \\ c_j.\text{Inner-Rela} \subseteq c_i.\text{Inner-Rela} \\ c_j.\text{Inner-Rul} \subseteq c_i.\text{Inner-Rul} \end{cases}$$

$$\forall c_s, c_t \in \mathrm{C},\quad c_s \text{ include } c_t \Leftrightarrow c_t \in c_s.Attrs$$

- $\mathbf{R}_{\mathbf{others}}$ consists of all other binary relations in the knowledge domain.

3.1.3 **Ops** – set of operators

Beside operators between real numbers in R, set **Ops** also consists of operators on **C**. There are two kinds of operators in **Ops**: unary and binary operators.

$$\mathrm{Ops} = \mathrm{Ops}_1 \cup \mathrm{Ops}_2$$

In which,

$\mathbf{Ops_1} \subset \{\Theta: I_c \to I_{c'} \mid c, c' \in \mathbf{C}\}$ is the set of unary operators.

$\mathbf{Ops_2} \subset \{\otimes: I_c \times I_{c'} \to I_{c''} \mid c, c', c'' \in \mathbf{C}\}$ is the set of binary operators.

The specification of an operator:

operator-def ::= OPERATOR <name>
ARGUMENT: arg-def+
RETURN: return-def;
PROPERTY: prop-type;
END
arg-def ::= name, <name>: type
return-def ::= <name> : type
prop-type ::= associative | commutative | identity

Definition 3.2: Definition of expression

$$< express >::= obj \mid \otimes < express > \mid < express > \Theta < express >$$

where, *obj*: an object in **C** Θ: binary operator $\otimes$: unary operator

If Θ is associative then: (p, q, r are expressions)

$$p \,\Theta\, q \,\Theta\, r = (p \,\Theta\, q) \,\Theta\, r = p \,\Theta\, (q \,\Theta\, r)$$

3.1.4 **Rules –** set of inference rules

There are three kinds of rules in **Rules**-set: inference rules, equivalent rules and equation rules. The structures of a rule r in **Rules** is described in Table 2.

$$\text{Rules} = \text{Rule}_{\text{infer}} \cup \text{Rule}_{\text{equi}} \cup \text{Rule}_{\text{equa}}$$

Table 2

Kind of rules

$\mathbf{Rule_{infer}}$	$\mathbf{Rule_{equi}}$	$\mathbf{Rule_{equa}}$
rul is an inference rule: $p(rul) \to q(rul)$ where $p(rul)$, $q(rul)$ are sets of facts.	The structure of an *equivalent rule*: $h(rul), p(rul) \leftrightarrow q(rul)$ in which, $h(rul)$, $p(rul)$ and $q(rul)$ are sets of facts, they meet requirements: $h(rul), p(rul) \to q(rul)$, and $h(rul), q(rul) \to p(rul)$ are true.	*rul* is an *equation rule*: $l = r$ where l, r are expressions of objects. Denote: $left(rul) = l$, $right(rul) = r$

The facts of Rela-Ops model are classified into 5 kinds which are shown in Table 3.

Table 3

Kind of facts

Kind	Meaning	Specification	Condition
1	Kind of an object	$x{:}c$	x is an object, $c \in \mathbf{C}$
2	Determine an object	o	$o \in \mathrm{I}_c$, $c \in \mathbf{C}$
3	Determine an object through constant	$obj = <const>$	$obj \in \mathrm{I}_{c1}$, $c_1 \in \mathbf{C}$
4	Determine a relation between two objects	$x\ \Xi\ y$	$\Xi \in \mathbf{R}$, Ξ is a relation. $x \in \mathrm{I}_c$, $y \in \mathrm{I}_{c'}$, $c, c' \in \mathbf{C}$
5	Equation between two expresses	$<exp_1> = <exp_2>$	$<exp_1>$, $< exp_2>$: expression

3.2 *Knowledge model of functions*

In the knowledge domain about maths in high-school, the knowledge of functions is an intellectual foundation. Thus, the contents of functions must be fully represented when building an intelligent search system for this domain. In this section, a model focusing on the integration of the Rela-Ops ontology model and functional knowledge is studied, called ontology for knowledge of functions, *Onto-Funcs model*. Rela-Ops model and functional knowledge integration is being worked on:

- Creating the component's structure: This structure needs to be appropriate such that the new specification of functions is seamless and consistent to concepts of Rela-Ops model. It also has specification language using to describe its structure.
- Adding more kinds of rules about functions.
- Solving the searching problem on the functions component and combining with ontology Rela-Ops model.

3.2.1 Specification of a function

In the practice, each function is determined by one of the followed methods: 1/ Determined by a formula, or 2/ Determined by a sequence of commands. The functions need to be specified to represent the functions intellectual in the domain.

Definition 3.3: The following is a function specification:

Function-def := FUNCTION <name>:
    ARGUMENT : arg-definition+
    RETURN: return-definition;
    [facts];
    [variable];
    [statements];
    [properties];
END;
statement := assign-state | if-state | for-state
assign-state := <name>:= expression
if-state := IF logic-expression THEN statements+ END;
for-state := FOR name IN [range] DO statement+ END;
properties := continous | differentiable | monotonic | commutative

Eg. 3.1:

- Determined by a sequence of commands.
  The function PointProject(P : POINT, d : LINE) returns a projection point of a point P on a line d (where POINT and LINE are concepts in **C**).

  FUNCTION PointProject(P : POINT, d : LINE)
      ARGUMENT : *P, d*
      RETURN: *H* : POINT
      Constraint: *P* :: POINT, *d* :: LINE, *H* :: POINT
      Facts: NOT(*P* belongs *d*)
      Statement: Let *d'*: LINE with (*P* belongs *d'*) and ($d' \perp d$);
    $H = d'$ intersect *d*;
  END;

- Determined by a formula. For example, the function:

$$f(x) = \frac{3x^2 - 5x + 4}{2x + 1}$$

3.2.2 Ontology Onto-Funcs model for knowledge of functions

A collection of functions makes up the knowledge component of functions, called ***Funcs**-set*. The structure of this set is described as followed:

Definition 3.4: Let $F = \{g \mid g$ is a function specified as Definition 3.3$\}$ and $\alpha: F \to \mathbb{N}$: determining the function's number of parameters.

$$\mathbf{Funcs} \subseteq \{g \in F \mid k = \alpha(g),\ g: \mathrm{I}_{c1} \times \mathrm{I}_{c2} \ldots \times \mathrm{I}_{ck} \to \mathrm{I}_c,\ c_1, c_2, \ldots, c_k, c \in \mathbf{C}\}$$

Each function in the **Funcs**-set has its properties tested as well.

Definition 3.5: The knowledge model of functions is an integration of Rela-Ops ontology model with a set of functions, referred *Onto-Funcs model*. There are five components to this integrating model:

(C, R, Ops, Rules) ⊕ Funcs

where, **(C, R, Ops, Rules)** is the knowledge model of relations and operators as ontology Rela-Ops model, and Funcs is a set of functions specified as Definition 3.4.

The symbol "⊕" means the integrating of ontology Rela-Ops model as **(C, R, Ops, Rules)** and the knowledge of functions as Funcs-set. Besides the connection on concepts, the integration is worked on additional facts of functions and an additional kind of rules.

Table 4 describes details of the integration between Rela-Ops model and Funcs set.

Table 4

The integration between Rela-Ops model and Funcs-set.

<table>
<tr><th>Foundation</th><th colspan="2">Detail of integration</th></tr>
<tr><td rowspan="2">Additional Facts</td><td>Kind 6: Identify a function.
Specification:
$g(y_1, y_2, \ldots, y_m)$;
Condition: $g \in$ Funcs, $m = \alpha(g)$,
g: $I_{c1} \times I_{c2} \times \ldots \times I_{cm} \rightarrow I_c$,
$y_i \in I_{ci}$, $c_i \in \mathbf{C}$ $(1 \le i \le m)$.</td><td>Kind 7: Determine an object through a function.
Specification:
$o = g(y_1, y_2, \ldots, y_m)$;
Condition: $g \in$ Funcs, $m = \alpha(g)$,
g: $I_{c1} \times I_{c2} \times \ldots \times I_{cm} \rightarrow I_c$,
$y_j \in I_{cj}$, $c_j \in \mathbf{C}$ $(1 \le j \le m)$,
$o \in I_c$, $c \in \mathbf{C}$.</td></tr>
<tr><td>Kind 8. Determine a function through an expression.
Specification:
$g(y_1, y_2, \ldots, y_m)$ = <exp>.
Condition: $g \in$ Funcs, $m = \alpha(g)$,
g: $I_{c1} \times I_{c2} \times \ldots \times I_{cm} \rightarrow I_c$,
$y_i \in I_{ci}$, $c_i \in \mathbf{C}$ $(1 \le i \le m)$, $c \in \mathbf{C}$,
<exp>: expression.</td><td>Kind 9. Determine a relation between functions.
Specification:
$g(y_1, y_2, \ldots, y_m)\ \Phi\ h(x_1, x_2, \ldots, x_n)$.
Condition: $\Phi \in \mathbf{R}$,
$g \in$ Funcs, $h \in$ Funcs,
$m = \alpha(g)$, $n = \alpha(h)$; $c, c' \in \mathbf{C}$;
g: $I_{c1} \times I_{c2} \times \ldots \times I_{cm} \rightarrow I_c$,
$y_i \in I_{ci}$, $c_i \in \mathbf{C}$ $(1 \le i \le m)$
h: $I_{b1} \times I_{b2} \ldots \times I_{bn} \rightarrow I_{c'}$,
$x_j \in I_{bj}$, $b_j \in \mathbf{C}$ $(1 \le j \le n)$.</td></tr>
<tr><td>Additional kinds of Rules</td><td colspan="2">The equation rule r has the following form:
l = r,
where l, r are functions or expressions.
Denote: $left(r) = l$ $right(r) = r$</td></tr>
</table>

3.3 *Knowledge graph and ontology for knowledge of functions*

The Onto-Funcs model is an ontology that contains ideas, relationships between ideas, operators, rules, and knowledge-related functions. This model is fully and effectively to organize the knowledge base of high-school mathematics. For doing the query on this knowledge domain, the relations between intellectual components need to be impressed to retrieve their content more precisely. This section proposes a structure of the knowledge graph to clearly describe relations between components of Onto-Funcs model.

3.3.1 Knowledge graph

Knowledge graph is a model used graph-structured to organize the knowledge base. This model is a directed graph, consisting of two main components, nodes and edges. Each node is an entity, and an edge connects two nodes, which represents the relationship between those two nodes.

Definition 3.6: Given the knowledge domain $\mathcal{K}$ = **(C, R, Ops, Rules)** $\oplus$ **Funcs** as ontology Onto-Funcs model. The knowledge graph for this domain is a tube:

$$(N, E)$$

where, **N** is a set of nodes expressing knowledge of concepts or functions. Each node has the structure similarly to the structures of a concept in Section 3.1.1 or a function as Def. 3.3.

E is a set of edges of this graph representing relations between nodes in **N**. Each edge is a tube (*Attrs, Properties, Key, Start, End*), wherein, *Attrs*

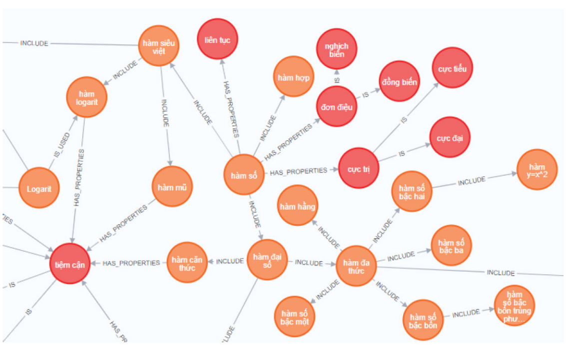

Figure 1

A part of the knowledge graph of the high-school mathematics domain.

is a collection of a relation's attributes, *Properties* is a group of the relevant relation's properties, *Key* is a set of keyphrases related to the corresponding relation, *Start* is a start node of the corresponding relation, and *End* is an end node of the corresponding relation.

Figure 1 shows a part of the knowledge graph of the domain of knowledge related to high-school mathematics. This graph can perform relationships of most of concepts in detail. It is helpful to support for knowledge querying and recommending related knowledge on this domain.

3.3.2 Database for supporting to represent ontology for knowledge of function

Although the knowledge graph can represent concepts of ontology Onto-Funcs completely, it cannot perform the knowledge of operators and rules in this model. Besides, the knowledge graph also has not yet been effective to represent binary relations between two items that share the same notion. Thus, a database is used to store that knowledge and combined with a knowledge graph to support each other. The structure of the database consists relationship tables as follows: **(Knowledge, Rules, Relations)**

In which:

- **Knowledge** table: includes operators and rules which cannot be represented by the graph, such as the definition of "Derivative", "Maximum Value".
- **Rules** table: includes rules with associated relations, functions, and operators.

  Eg. 3.2: The rule of three perpendicular lines includes:

  - Related relations: "perpendicular (⊥)", "belong to"
  - Related function: Projection
  - Related concepts: LINE, PLANE
- **Relations** table: consist of binary relations between two objects.

  Eg. 3.3: "Parallel (//)" between two lines (or two segments).

  - Concept: Line (or Segment)
  - Relation: Parallel (//)

4. Problems and Algorithms

In addition to the usual searching based on lessons and keywords, intelligent searching can search on every aspect of the knowledge domain, as well as each type of it, such as searching for concepts, rules, exercises, or solving methods, and it can also extract related knowledge from the current results. This section presents some intelligent searching for problems within the knowledge domain such as Onto-Funcs ontology, along with their respective solutions.

When a user inputs a query, the system collects data and searches for relevant material that matches the query. Additionally, the search engine suggests other related content in addition to the current results. There are two problems for searching: 1/ Classifying the input query and extracting main keyphrases of the query, and 2/ Matching the keyphrases with the content of the knowledge.

4.1 *Classifying the input query and Extracting main keyphrases*

The query is inputted into the system as Vietnamese. It will be extracted main keyphrases and those keyphrases are used to retrieve in the knowledge base. At first, the query is classified the intent of the user. There are several methods for classifying intent. Bidirectional LSTM, a neural network design, is used in this study because of its capacity to process vectors with both forward and backward directions [24]. There are three main stages for the intent classification system: Word Embedding, Tokenizer, and training with Deep Bi-Directional LSTM (Figure 2).

The tokenizer stage makes use of the PyVi toolkit[2]. Word2VecVN[3] is used to select the best pre-trained embedding during the embedding stage. The deep learning model was created using the Keras framework, which also adds a softmax function to output layer.

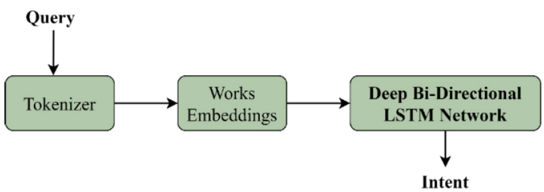

Figure 2

Intent classification architecture

[2] PyVi Toolkit: https://pypi.org/project/pyvi/
[3] Word2vecVN: https://github.com/sonvx/word2vecVN

After that, keyphrases are extracted from a deep learning model, using PhoBert model to tokenize the query into keyphrases. PhoBert is a pre-trained language model for Vietnamese. is a pre-trained model with a large Vietnamese dataset, high accuracy. This model is used as a black box, just pass the data through this box, the keywords will be combined into keyphrases. At the data processing stage, the text is transformed into a list of tuples in the following format: [(token$_1$, tag$_1$),... (token$_n$, tag$_n$)]. The information is then transformed into a collection of tuples. To feed the text into our Bi-LSTM model, we must make sure that it is of the same length. Then, using the Keras framework, One-hot encoding is used for Named Entity Recognition tags. From there, the entered query's primary keyphrases are extracted.

4.2 *Searching on the knowledge base combining of knowledge graph*

From extracted keyphrases of the query, the system will calculate the similarity between those keyphrases and keyphrases in the knowledge base. From then, the system leverages relations on the knowledge graph to determine concepts and functions for extracting the query's results based on data from the knowledge base.

Solution: In the first step, the dictionary is created using the search engine's knowledge base. Also, it has terminology for questions that help categorize various question types. To create a dictionary of key phrases, The set of keywords extracted from the utterance is compared with a lexicon. These keywords are used by the search engine to retrieve ontology-based programming knowledge. The system can also look for relevant knowledge based on the connections between the retrieval results and the inference rules of the knowledge base. Ultimately, a document containing each type of intent will be returned for all outcomes.

Algorithm 4.1: Determine the knowledge content that is pertinent to a given input query.

Input: Knowledge domain K as Onto-Funcs model,

The knowledge graph G = **(N, E)** as Definition 3.6.

The database D of the knowledge K includes relational tables as section 3.3.2

Query q.

Output: *Results* : Knowledge content for query q.

Algorithm 4.1:

```
Phrases ← Extract the keyphrases from query sentence q.
DOps ← KnowledgeTable(D.Knowledge) ;// list of operators
RConcepts ← [ ];
RFuncs ← [ ];
ROps ← [ ];
Results ← [ ];

for k in Phases do
    for node in G.N do
     if (type(node) is concept) then
            if (node.Symbol = k) then
              Update node into RConcepts;
        else
              ParsC ← [ ];
              for pa in node.Contents do
                 if (pa.name_par = k) then
                        Update pa into ParsC;
               end do;
               if ParsC  ≠  [ ] then
                   Update ParsC into RConcepts;
               end if;
        end if;
     else  // node is a function
            if (FunctionName(node) = k) then
          Update node into RFuncs;
        end if;
     end if;      //check type of node
    end do;

    for opr in DOps do
         if (Name(opr) = k) then
```

Update *opr* into *ROps*;

end do;

end do;

Results ← [*RConcepts, RFuncs, ROps*];

return *Results*;

The knowledge source is located, and a set of knowledge is retrieved by analyzing the meaning of the query. This process involves scrutinizing the resemblances between the keywords in the query and the knowledge that is stored. The approach for addressing the knowledge searching problem has been translated in [21, 25].

The system provides additional information about the results it has found for a given query based on those findings. Algorithm 4.2 describes the solution to extract knowledge which are related to results of querying.

Algorithm 4.2: Determine the knowledge about a query's outcomes.

Input: Knowledge domain K as Onto-Funcs model,

The knowledge graph G = **(N, E)** as Definition 3.6.

The database D of the knowledge K includes relational tables as section 3.3.2

Query q.

Output: *Related* : a set of knowledge pertaining to results of query q.

Algorithm 4.2:

Results ← [*RConcepts, RFuncs, ROps*]; // list of results of query q which was determined by Algorithm 4.1, where:

- *RConcepts*: Set of results as concepts.
- *RFuncs*: Set of results as functions.
- *ROps*: Set of results as operators.

RelatedConcepts ← [ ]; // related knowledge as concepts

RelatedFuncs ← [ ]; // related knowledge as functions

RelatedOps ← [ ]; // related knowledge as operators

Related ← [ ];

```
for n in Results do
  for edge in G.E do
    if (edge.Start = n) then
      if (type(edge.End) is concept) then
        Update edge.End into RelatedConcepts;
      else if (type(edge.End) is function) then
        Update edge.End into RelatedFuncs;
      else
        Update edge.End into RelatedOps;
      end if;
    end if;  // check edge.Start
  end do;
  for rel in D.Relations do
      rel is a binary relation, so it has the form rel(a, b), where a, b are
arguments of relation rel.
if (rel.a  = n) then
      if (type(rel.b) is concept) then
        Update rel.b into RelatedConcepts;
      else if (type(rel.b) is function) then
        Update rel.b into RelatedFuncs;
      else
        Update rel.b into RelatedOps;
  end if; // checking type of rel.b
else if (rel.b  = n) then
      if (type(rel.a) is concept) then
        Update rel.a into RelatedConcepts;
      else if (type(rel.a) is function) then
        Update rel.a into RelatedFuncs;
      else
        Update rel.a into RelatedOps;
  end if; // checking type of rel.a
end if;
  end do;
end do; // searching related knowledge in Results

Related ← [RelatedConcepts, RelatedFuncs, RelatedOps];
return Related;
```

Also, the system will locate some knowledge that is linked to the outcomes of its own answers using the knowledge graph it has created. Moreover, the system also searches for relevant knowledge from the query results, taking into account the relationships between the results and the inference rules in the knowledge base stored in the database. Eventually, the user will be presented with all the results, which will be saved in the form of a document containing the query.

5. Testing and Experimental results

5.1 *The architecture of the searching system based on Onto-Funcs model*

Incorporating the Onto-Funcs ontology presented in Section 3 and the searching algorithms outlined in the same section, the intelligent searching system has been constructed, which incorporates the knowledge of high-school mathematics in Vietnam. The knowledge base of the system is derived from the data gathered in this knowledge domain [26]. This technology can assist students in their academic pursuits by offering necessary course materials in response to user searches. Using the structure of an intelligent system based on its knowledge base [27], Figure 3 presents the architecture of this system which is built based on its knowledge base including the knowledge graph and database which are organized as Onto-Funcs model.

The system extracts keyphrases from user-inputted queries and represents its meaning through relations between those keyphrases. After

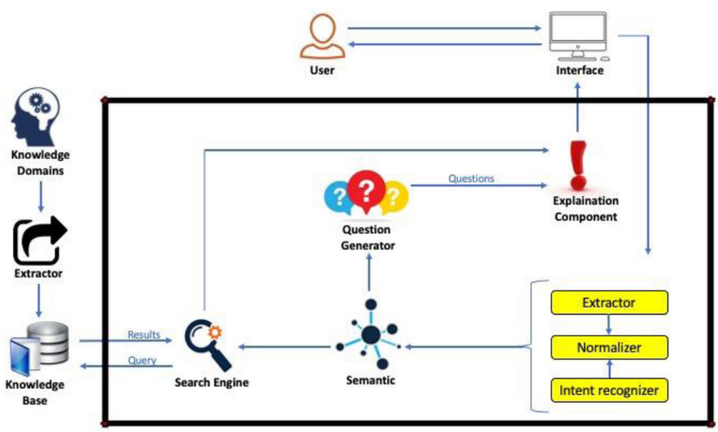

Figure 3

The architecture of the searching system based on Onto-Funcs model

that, the search engine processes to retrieve results from the knowledge base. In this process, if it lacks information for searching, the search engine can generate a basic question for the user to get more information. The determining of searching results is worked based on Algorithms 4.1 and 4.2. After receiving results, the system can show them by readable language through the explanation component.

5.2 *Testing*

Eg. 5.1: Query "Hàm số là gì ?" (It means "What is Function?")

After the processing this query, we have a set of keyphrases S_1 := {"Hàm số", "là gì"} (it means {"Function", "What is"})

- Compare the set S_1 with the dictionary containing question words. The word "là gì" ("what is") is a question word, therefore, this is a "What is" query, the system will return the content of the concept.
- The system uses the remaining keyphrases of set S_1 to find the content on the knowledge base.
- Then, based on relations of the knowledge graph, the system will return knowledge related to "Function", such as "Graph of a function", "Quadratic function".

The output of the system:

- Definition of Function: If a quantity y depends on a variable quantity x for which for a value of x, there is always one and only one corresponding value of y then y is called a function of x.
- Lesson: Grade 10, Chapter 2, Unit 1
- Related knowledge: Graph of a function, Quadratic function.

Eg. 5.2: Query "So sánh hàm số đồng biến và hàm số nghịch biến ?" (It means "Compare the increasing function and the decreasing function?")

- After the processing this query, we have a set of keyphrases S_2 = {"Hàm số đồng biến", "Hàm số nghịch biến", "so sánh"} (it means {"increasing function", "decreasing function", "Compare"})
- Compare the set S_2 with the dictionary containing question words. The word "so sánh" ("compare") is the question word, therefore, this is a "Compare" query, the system will return the contents of concepts and their attributes and properties.

- The system uses the remaining keyphrases of set S_2 to find the content on the knowledge base.
- Then, based on relations of the knowledge graph, the system will return the knowledge related to "increasing" and "decreasing", such as "Monotonic", "the extremum of a function".

The output of the system:

- Compare:
  - Increasing function: Suppose K is an interval, a segment, or a half interval, and $y = f(x)$ is a determined function on K. The function $y = f(x)$ is said to be an increasing function on K iff: $\forall x_1, x_2 \in K, x_1 < x_2$ then $f(x_1) < f(x_2)$
  - Decreasing function: Suppose K is an interval, a segment, or a half interval, and $y = f(x)$ is a determined function on K. The function $y = f(x)$ is said to be a decreasing function on K iff: $\forall x_1, x_2 \in K, x_1 < x_2$ then $f(x_1) > f(x_2)$
- Lesson: Grade 12, Chapter 1, Unit 1
- Related knowledge: Monotonic, the extremum of a function.

5.3 *Experimental results*

High-school students in Ho Chi Minh city (Vietnam) tested the search system for high-school maths queries. They input queries for searching the knowledge of this course. There are two kinds of queries: "*What is*" and "*Compare*" queries. The extracted results have been verified twice:

- If the results are appropriate for the students, they will evaluate the system.
- Those results are also saved. High-school mathematical teachers will reassess them.
- The results will be accurate if they are verified by students and teachers.

There are 70 queries that have been inputted by students. The test results are displayed in Table 5.

Table 5

Results for testing of queries

Kind	Quantity	Correct	Rate
"What is" query	40	29	72.5%
"Compare" query	30	20	66.67%
Total	**70**	**49**	**70%**

In the kind of "What is" query, there are some keyphrases of queries do not exist in the dictionary of the system, so they are not recognized. Hence, the system did not return correct results for those queries. In "Compare" queries, with the same reasons, there are queries cannot do not have appropriate results of comparison. This weak-point can be improved by enriching the dictionary and using pre-train model to recognize synonym words in the dictionary.

Moreover, the method of this search system can process complex relations in the queries including inheritance relation, relationships which are far apart. The advantage of the built system is using the knowledge graph combining a structured database to speed up the search both in terms of content and relations. There are various opportunities to combine the suggested approach with other more advanced Vietnamese processing methods, including employing grammatical rules [28] and adding more data to the knowledge graph [29, 30, 31], it will increase the accuracy of searching.

6. Conclusions and Future work

This study proposes an ontology-based model for representing mathematical knowledge in high-school is presented, called Onto-Funcs model. This method is an integration between the knowledge of functions and the understanding of relations and operators. Based on the integrated model, a structure of knowledge graph is proposed to represent relations of intellectual contents. This knowledge graph also combines databases to organize the knowledge domain more clearly. Those structures are used to design an intelligent querying system in high-school mathematics through solving problems for querying on this knowledge domain: Classifying the input query and Extracting main keyphrases, Searching the content of the knowledge based on the combination of knowledge graph and database. The system can retrieve contents of queries and extracts other knowledge relevant to querying results which require for students to support their learning.

In the future, the intelligent searching system will be combined with a system for evaluating the intellectual level of students through multiple choice testing [18, 32]. The system for combining will be an entirely intelligent tutoring system for students to review in high-school mathematics. Moreover, it can develop as an answering-question system to become a personalized recommendation system of learning [33, 34].

References

[1] N.W. Gleason, Higher education in the era of the fourth industrial revolution, Springer Nature, (2018).

[2] A. Meena, A. Vaish, S. Meena, et. al., A mathematical model for analysing the significance of Information Technology resources in school education during pandemic, *Journal of Interdisciplinary Mathematics*, 25(3), pp. 741-752, (2022).

[3] M. Littman, et. al., Gathering Strength, Gathering Storms: The One Hundred Year Study on Artificial Intelligence (AI100) 2021 Study Panel Report, Stanford University, Stanford, CA, Sept. 2021. http://ai100.stanford.edu/2021-report

[4] H. Nguyen, N. Do, N. Tran, X.H. Pham, V. 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).

[5] I. Hatzilygeroudis, J. Prentzas, Knowledge Representation in Intelligent Educational Systems, In: Z. Ma (Ed.), Web-Based Intelligent E-Learning Systems: Technologies and Applications, IGI Global, (2006).

[6] M. Tapia-Leon, A. Rivera, J. Chicaiza, S. Luján-Mora, Application of ontologies in higher education: A systematic mapping study, *Proceedings of 2018 IEEE Global Engineering Education Conference (EDUCON 2018)*, pp. 1344-1353, (2018).

[7] N. Noy, D. McGuinness (Eds.), Final Report on the 2013 NSF Workshop on Research Challenges and Opportunities in Knowledge Representation, NSF Workshop Report, (2013).

[8] K. Giri, Role of ontology in semantic web. *DESIDOC Journal of Library Information Technology*, 31(2), 116-120, (2011).

[9] A.C. Khadir, H. Aliane, A. Guessoum, Ontology learning: Grand tour and challenges, *Computer Science Review*, 39, 100339, (2021).

[10] C. Grosan, A. Abraham, Intelligent Systems: A modern approach, Springer, Berlin, Heidelberg, (2011).

[11] N.V. Do, H.D. Nguyen, L.N. Hoang, Some Techniques for Intelligent Searching on Ontology-based Knowledge Domain in e-Learning, *Proceedings of 12th International Conference on Knowledge Engineering and Ontology Development (KEOD 2020)*, pp. 313-320, Budapest, Hungary, (2020).

[12] A. Espen, Why you should choose math in high school, *Ubiquity*, 7(11), (2006). https://ubiquity.acm.org/article.cfm?id=1125361

[13] P.G.W. Adnyana, I.M. Suarsana, I.G.P Suharta, Multi-Representation Discourse Model and Math Problem Solving Skills of High School Students, *Journal of Learning Improvement and Lesson Study*, 1(1), 40-48, (2021).

[14] S.M. Stefanov, On some properties of linear functionals, *Journal of Interdisciplinary Mathematics*, 25(8), pp. 2491-2502, (2022).

[15] D.H. Nguyen, A.D. Tran, P.H. Do, et. al., Design an intelligent system to automatically tutor the method for solving problems, *International Journal of Integrated Engineering*, 12(7), pp. 211-223, (2020).

[16] E. Mousavinasab, N. Zarifsanaiey, R. Niakan, et. al., Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods, *Interactive Learning Environments*, 29(1), pp. 142-163, (2021).

[17] R.B. Kaliwal, S.L. Deshpande, Study on Intelligent Tutoring System for Learner Assessment Modeling Based on Bayesian Network. *Proceedings of 5th International Joint Conference on Advances in Computational Intelligence* (IJCACI 2021), pp. 389-397, Dhaka, Bangladesh, Oct. 2021, (2021).

[18] L.W. Somathilake, A mathematical model for control strategies of COVID-19, *Journal of Interdisciplinary Mathematics*, 25(8), pp. 2365-2386, (2022).

[19] N. Do, H. Nguyen, T. Mai, Intelligent Educational Software in Discrete Mathematics and Graph Theory, *Proceedings of 17th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET 2018),* pp. 925 - 938, Granada, Spain, (2018).

[20] H.D., Nguyen, N. Do, Intelligent Problems Solver in Education for Discrete Mathematics, *Proceedings of 16th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET 2017)*, pp. 21-34, Kitakyushu, Japan, (2017).

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

[22] T. Le, et. al., Knowledge representation method for designing an Intelligent Tutoring System in Learning of courses about Algorithms. *Proceedings of 25th Asia-Pacific Conference on Communications (APCC 2019)*, pp. 310-315, Ho Chi Minh city, Vietnam, (2019).

[23] H. Nguyen, et. al., A method for knowledge representation to design Intelligent Problems Solver in mathematics based on Rela-Ops model. *IEEE Access* 8, 76991-77012, (2020).

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

[25] H.D. Nguyen, V. Tran, T. Pham, A.T. Huynh, N. Do, Ontology-based Integration of Knowledge Base for Building an Intelligent Searching Chatbot, Sensors and Materials, Vol. 33, No. 9, pp. 3101-3121, (2021).

[26] Vietnam Ministry of Education and Training, Textbooks and workbooks of mathematics in high school, Publisher of Education, (2021).

[27] H. Nguyen, N. Do, T. Pham, A method for designing Knowledge-based Systems and Application. In: A. Elgnar et. al. (Eds.). Applications Computational Intelligence Multi-Disciplinary Research, Academic Press, Elsevier, (2022).

[28] H.D. 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*, Proceedings of 15th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2020), pp. 339-346, Prague, Czech Public, May 2020.

[29] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, *Expert Systems with Applications*, 141, 112948, (2020).

[30] X. Wu' Y. Li, J. Wang, Q. Qian, Y. Guo, UBAR: User Behavior-Aware Recommendation with knowledge graph, *Knowledge-Based Systems*, Vol. 254, 109661, (2022).

[31] M. Gionfriddo, A remarkable connection between k-permutations and colourings of graphs, *Journal of Interdisciplinary Mathematics*, 14(1), pp. 95-104, (2011).

[32] T. Mai, et. al. An Intelligent Support System for the Knowledge evaluation in high-school mathematics by Multiple choices testing. *Proceedings of 5th NAFOSTED Conference on Information and Computer Science* (NICS 2018), pp. 282-287, Ho Chi Minh city, Vietnam, (2018).

[33] H. Chen, Personalized recommendation system of e-commerce based on big data analysis. *Journal of Interdisciplinary Mathematics*, 21(5), pp.1243-1247, (2018).

[34] Y. Zhou, C. Huang, Q. Hu, et. al., Personalized learning full-path recommendation model based on LSTM neural networks, *Information Sciences*, 444, pp.135-152, (2018).