# Ontology-Based Topic Model for Document Retrieval Systems in Information Technology

Thanh Dien Nguyen [1,*], Van Nhon Do [2,*], and Hoang Tung Tran [3]

[1] Department of Computing Fundamental, FPT University, Ho Chi Minh City, Vietnam
[2] Department of Information Technology, Hong Bang International University, Ho Chi Minh City, Vietnam
[3] Department of Information and Communication Technology, Vietnam France University, Ha Noi, Vietnam
Email: diennt4@fe.edu.vn (T.D.N.); nhondv@hiu.vn (V.N.D.); tran-hoang.tung@usth.edu.vn (H.T.T.)
*Corresponding author

***Abstract*—Most current academic document retrieval systems for topic-based search rely on simple keyword matching or statistical topic modeling. In these methods, topics are formed either from sets of frequent keywords or from statistical clusters. While these approaches work in some contexts, they cannot fully capture the rich semantic meaning of topics as understood by human experts. This often leads to search results that fail to match the intended meaning of the topic, causing gaps between what users need and what the system returns. This study aims to overcome these limitations by developing a topic-based retrieval system that represents topics in a more semantically rich and human-aligned way. The system is designed to help Information Technology (IT) students search for topic-relevant materials—specifically English-language ebooks and research papers—from a curated faculty repository. We introduce C-ONTO, a structured knowledge model that includes topic names, descriptions, learning objectives, real-world application scenarios, and concept graphs describing internal semantic relationships. Documents are also modeled as concept graphs, enabling accurate semantic similarity calculations. An intelligent query analysis module interprets user intent and maps it to the system's semantic structure. Testing with 425 real student queries in four IT domains shows that the system achieves 81.18% accuracy, outperforming keyword-based and statistical methods in precision, recall, coverage, and F1-Score. By aligning computational topic modeling with human understanding, the proposed system improves accuracy, semantic consistency, and educational relevance in academic document retrieval for information technology and related fields.**

***Keywords*—ontology, knowledge representation, topic modeling, semantic document retrieval, concept graph**

## I. Introduction

In the current landscape of information retrieval, most systems still rely heavily on keyword-based search or document popularity. While these approaches offer basic functionality, they often fall short in specialized domains such as Information Technology (IT), where topic-based search capabilities remain underdeveloped. In many cases, topics are represented merely as keyphrases, lacking the semantic richness and structured content necessary for effective knowledge discovery. This limitation frequently leads to search results that fail to align with the nuanced semantic scope of a given topic, creating a significant gap between user intent and system output. Consequently, students, particularly those engaged in fields like IT, face considerable challenges in retrieving academic materials that truly resonate with their specific learning objectives and conceptual understanding.

Previous research in topic-based document retrieval, particularly within academic contexts, has largely relied on simplistic keyword matching or statistical topic modeling techniques. While these methods offer a foundational approach, they inherently struggle to capture the rich semantic meaning of topics as understood by human experts. This often results in a disconnect between the intended meaning of a user's query and the retrieved documents, leading to suboptimal search outcomes. For instance, models proposed [1, 2] represented topics as mere groups of keywords or with limited components (Name, Description, Contents), failing to encapsulate the intricate semantic relationships crucial for comprehensive knowledge representation. Similarly, the CK_ONTO model [3], while a step forward, still required significant improvement in its ability to represent document semantics effectively.

To improve the effectiveness of intelligent information retrieval systems, researchers need to select appropriate knowledge representation models and methods. Among these, ontology is considered one of the most important models, along with supporting tools such as Protégé and GATE. Ontology serves as a knowledge representation model for a specific domain, allowing reasoning about objects within that domain and the relationships among them. It provides a unified vocabulary that includes concepts, attributes, inference rules, axioms, and definitions, forming a foundation for organizing and exploiting knowledge more effectively.

In educational contexts, students frequently seek documents aligned with specific areas of interest. This necessitates intelligent retrieval systems capable of interpreting user queries semantically and delivering relevant content accordingly.

Manuscript received June 26, 2025; revised August 1, 2025; accepted August 29, 2025; published November 25, 2025.

Building such systems requires not only an effective knowledge modeling framework but also a querying mechanism that supports comprehensive knowledge specifications including concepts, definitions, properties, rules, and their interrelations. Furthermore, to meet the needs of educational environments, these systems should be capable of interpreting near-natural language queries and aligning them with the semantic structure of course content. The intended users of the proposed system include students and educators in the field of Information Technology, as well as domain professionals from other areas such as mathematics and road traffic law.

In response to these critical limitations, this paper proposes a novel semantic-based thematic document retrieval system specifically designed to address the unique challenges within Information Technology education. Our work distinguishes itself from existing research through several key contributions:

(1) A unified knowledge-based retrieval framework (C-ONTO): We propose the C-ONTO model, which integrates a domain ontology, a topic model, and a concept model to represent knowledge in a structured and semantically consistent manner.
(2) Structured keyphrase extraction: A method for extracting and organizing keyphrases is introduced, ensuring that topic–document matching captures both semantic meaning and domain-specific context.
(3) Document representation and semantic similarity computation: We develop techniques to represent documents and topics within the ontology structure, enabling accurate calculation of semantic similarity between queries and knowledge units.
(4) Experimental application and evaluation: We implement the proposed model in a topic-based document retrieval system and conduct extensive experiments. Results show that our system achieves higher accuracy, coverage, recall, and F1-score compared to existing keyword-based and ontology-based approaches, thus providing both precise and comprehensive retrieval.
(5) Evaluation in multiple domains: In addition to the Information Technology domain, the model is also tested in other domains (e.g., Mathematics, Road Traffic) and demonstrates strong performance with minimal configuration changes, confirming its robustness and scalability.

By aligning computational topic modeling with human understanding and addressing the pervasive semantic gaps, our proposed system not only improves accuracy and semantic consistency but also enhances the educational relevance of academic document retrieval for information technology and related fields. This work represents a significant step forward in building intelligent retrieval systems that truly support deep learning and research within specialized academic disciplines.

This paper is clearly structured into several sections to present its content systematically. Section I presents the Introduction of this study. Section II reviews relevant literature, providing the theoretical foundation for the research. Section III introduces the proposed knowledge-based model, referred to as the C-ONTO model, and describes how problems and algorithms are applied to design the architecture of an ontology-based knowledge querying system. Section IV presents the experimental results along with detailed analysis and discussion. Section V presents the limitations and future. Finally, Section VI summarizes the study, highlights the advantages and limitations of the improved model, and proposes potential directions for future work.

## II. Literature Review

Ontology as theoretical foundation—Ontology has been widely recognized as a formal and explicit specification of a shared conceptualization of a domain [4]. It provides a structured framework for representing knowledge by defining concepts, their properties, and the relationships among them. In the context of information retrieval, ontology plays a crucial role in enabling semantic search and reasoning, as it allows systems to move beyond keyword matching toward understanding the meaning of terms and their interconnections. Ontology-based approaches enhance retrieval accuracy by capturing domain-specific semantics, supporting interoperability, and facilitating knowledge reuse. In our work, the ontology serves as the foundation for modeling topics in the IT domain, ensuring that document retrieval is aligned with both the conceptual structure of the domain and the intended user needs.

Building an ontology typically involves several key steps: (i) identifying the scope and purpose of the ontology, (ii) collecting domain knowledge from experts, documents, and datasets, (iii) defining core concepts, attributes, and relationships, and (iv) formalizing these components into a structured representation, often using description logics or semantic web standards such as OWL (Web Ontology Language). In practice, ontology development requires iterative refinement, ensuring both conceptual clarity and practical usability.

To facilitate this process, various tools and editors have been developed. Among the most widely used is Protégé, an open-source ontology editor that supports ontology modeling, visualization, and reasoning with OWL and RDF(S). Other tools such as TopBraid Composer, OntoStudio, and NeOn Toolkit also provide advanced capabilities for managing large ontologies, integrating heterogeneous data sources, and supporting collaborative ontology engineering. These tools not only accelerate ontology construction but also help ensure consistency and interoperability, which are essential for applications in semantic search and document retrieval.

In document retrieval, ontologies enable semantic interpretation of queries and documents, going beyond keyword matching or statistical correlations. By encoding explicit domain knowledge, they allow systems to retrieve information that aligns more closely with expert understanding, improving accuracy and relevance. Moreover, ontologies can be adapted to different fields by modifying their domain concepts, ensuring semantic consistency across applications.

This section critically reviews prior work along three major strands that underpin topic-based academic document retrieval: (i) keyword/statistical topic modeling, (ii) ontology-based knowledge representation for retrieval, and (iii) education-oriented retrieval in IT. We analyze representative approaches, highlight their methodological and practical limitations, and position our contribution relative to these gaps.

### A. Keyword and Statistical Topic Modeling Approaches

In the theory of document topic identification and building a topic-based information search language, the concept of "topic" is one of the most important and basic concepts. In the literature, there have been many different concepts about the concept of document topic. Paying attention to ideology leads to drawing out "idea topic" and attaching importance to sticking to content with "real topic".

According to Vietnamese dictionary, topic is defined as follows: "topic is the main issue that is thoroughly understood in the content of a literary and artistic work according to a certain ideological trend". Realizing that, in the above interpretation, the topic is always associated with a literary work and the ideology is always emphasized. That definition does not cover the general concept of the topic. In fact, not only literary works have themes, but any document, no matter how it is published and presented, has a topic. Meanwhile, a report to the Cambridge dictionary, a topic is an issue that is discussed, written about or researched and the Oxford dictionary, a topic is an issue presented in a text, essay or conversation. However, the concept of the research topic is not a problem that cannot be inherited and developed.

Early and mainstream systems for topic-based retrieval typically rely on keyword matching or statistical topic modeling (e.g., frequency-based keyphrase extraction, probabilistic clustering) [5–9]. These methods are attractive for their simplicity and scalability, and they can discover latent co-occurrence structures in large corpora without manual curation. However, from a semantic and pedagogical standpoint, they present three critical limitations:

(1) Semantic shallowness: Topics are inferred as bags of co-occurring terms or clusters of frequent keyphrases. Such constructs do not encode conceptual relations, learning goals, or application context, yielding topics that may be statistically coherent but conceptually incomplete or ambiguous for learners.
(2) Misalignment with human understanding: Because the modeling units are word-level statistics rather than structured concepts, the resulting topics often deviate from the way domain experts define and teach topics, especially in IT where prerequisite chains and conceptual hierarchies matter.
(3) Poor educational relevance: The lack of explicit semantics prevents mapping user intent (expressed in near-natural language) to the deeper knowledge structure needed to recommend materials that match learning objectives and real-world use cases.

These limitations manifest in retrieval errors such as topical drift, low precision for conceptually nuanced queries, and inadequate support for curriculum-aligned search.

### B. Ontology-Based Knowledge Representation for Retrieval

Ontologies promise structured semantics through explicit concepts, properties, and relations, supporting integration and reasoning. Several lines of work have explored this direction:

- Concept-light topic formulations: Nhon *et al.* [1] defined a topic as a group of keywords, while Dien *et al.* [2] proposed Topic = (Name, Description, Contents) for AI. Although these formulations introduce structure, they still reduce content to keyword sets, omitting concept relations, learning outcomes, and application context. As such, they are insufficient to support semantic matching between queries and documents in education scenarios.
- Domain ontologies with simple concept layers: CK_ONTO [3, 10], Rela-KG (combining Rela-Ops and a two-layer knowledge graph) [11], an OWL ontology for climate data [12], an IU domain ontology defined as $O := \langle C, R\_sub, P, A \rangle$ [13], and healthcare-oriented ontologies [14] demonstrate feasibility across domains. However, a common shortcoming is the simplicity of the embedded concept model: concepts are modeled as classes/entities or keyphrase-based surrogates without rich intra-topic semantics. This undermines their ability to capture preconditions, dependencies, and usage scenarios critical for accurate, intent-aware retrieval. Moreover, when ontologies are constructed primarily from keyphrases [15], their semantic expressiveness is inherently limited.

In summary, while ontology-based approaches move beyond raw statistics, many remain conceptually shallow for the purposes of topic-based retrieval in IT education. They lack multi-faceted topic structures (e.g., learning objectives, real-world scenarios, concept graphs) and algorithms that leverage these structures for semantic similarity computation and ranking.

### C. Education-Oriented Retrieval in IT

Across the above strands, a persistent gap is the lack of systems explicitly designed for IT education. Surveys and reports indicate that IT students struggle to retrieve materials aligned with specific learning objectives and conceptual understanding [15–17]. Existing academic search tools often prioritize keyword matching and popularity signals, which neglect deeper semantics and prerequisite relations. Ontology-based models developed for other domains (e.g., healthcare, climate) are not readily transferable to IT curricula due to differences in concept decomposition, dependency structures, and pedagogical requirements. Consequently, learners encounter results that are topically adjacent but not thematically coherent or pedagogically suitable.

### D. Positioning and Distinct Contributions of This Work

Our work directly addresses these limitations through an ontology-driven topic model tailored to IT education:

(1) Rich, human-aligned topic representation (C-ONTO): Each topic includes Name, Description, Learning Objectives, Real-World Application Scenarios, and an internal Concept Graph. This goes substantially beyond prior keyword-based or class-level concept models, capturing the semantics necessary for intent interpretation and pedagogical alignment.

(2) Document-as-concept-graph representation: Documents are encoded as concept graphs, enabling semantic similarity computation that respects conceptual structure rather than surface term overlap.

(3) Intelligent query analysis and ranking: We introduce dedicated algorithms to map user intent to the semantic topic structure and to rank documents accordingly, overcoming the statistical and ontology-only limitations observed in prior systems.

(4) Empirical validation in IT domains: On 425 real student queries spanning four IT domains, the system achieves 81.18% accuracy and outperforms keyword/statistical baselines across precision, recall, coverage, and F1, demonstrating practical gains rooted in richer semantics.

Although many previous studies have proposed concepts, definitions, knowledge-based models, and topic modeling techniques across various domains, these developments have not been adequately explored in the context of IT education. Several reports and surveys [16, 17] indicate that a significant portion of IT students struggle with retrieving academic materials that align with their specific learning objectives and conceptual understanding. Existing academic search systems are often based on simplistic keyword-matching techniques, which overlook the deeper semantic relationships between topics and user queries. Moreover, current ontology-based models remain fragmented or are tailored to other fields such as healthcare or biology, offering limited relevance to IT-specific curricula. This domain-specific gap highlights the pressing need for a structured, semantic, and topic-oriented retrieval system designed explicitly for IT education. By addressing this gap, the proposed study aims to support learners in accessing thematically coherent, pedagogically relevant documents that better align with their study goals.

Collectively, these advances close the gap between statistical topic surrogates and shallow ontology layers on one side, and the pedagogically grounded, semantically rich topic modeling required for IT education on the other.

### E. Summary Comparison with Related Works

TABLE I. COMPARISON OF RELATED WORKS AND OUR APPROACH

| Ref. | Approach/Technique | Limitations | Our Approach |
|---|---|---|---|
| [1] | Ontology-based retrieval for general domain | Ontology structure not optimized for IT domain; lacks topic–concept integration | Proposes C-ONTO with explicit topic–concept relationships tailored to IT |
| [2] | Keyword-based academic search | Fails to capture semantic intent; results often irrelevant | Integrates ontology-based semantic similarity for more relevant retrieval |
| [3] | LDA topic modeling | Topics lack domain-specific coherence; limited interpretability | Uses ontology-based topic model to enhance semantic clarity |
| [5] | Knowledge base for subject-specific retrieval | Static model; not easily extendable to other fields | Modular design allows extension to multiple domains |
| [6] | Keyword expansion using ontology terms | Limited by ontology coverage; manual updates needed | Uses automated extraction from C-ONTO for richer vocabulary |
| [7] | Semantic search for academic papers | Semantic scope narrow; no topic-level modeling | Combines ontology topic modeling with semantic similarity computation |
| [8] | IT course ontology for learning resources | Focused only on course mapping; lacks retrieval optimization | Integrates ontology into a full retrieval architecture |
| [10] | Concept-based retrieval using hierarchical ontology | Manual ontology building is time-consuming; incomplete knowledge coverage | Automated keyphrase extraction and structured ontology expansion |
| [11] | Rela-KG model combining extended ontology and KG | Simple concept model; not aligned with current research | Enhances concept modeling with structured keyphrases in C-ONTO |
| [12] | Multi-ontology integration | Complex alignment process; high manual effort | Uses consistent C-ONTO structure for easier integration |
| [13] | Ontology-driven e-learning platform | Retrieval component not optimized for thematic queries | Employs topic-based semantic similarity for better thematic search |
| [14] | Semantic indexing for academic content | Indexing schema limited to keywords | Uses concept–topic–keyphrase indexing for richer semantic access |
| [15] | Knowledge graph search with manual tagging | Labor-intensive; prone to omissions | Automates tagging via keyphrase extraction from documents |
| [18] | Domain-specific knowledge graph retrieval | Limited scalability; lacks multi-domain adaptability | Designed for minimal-configuration adaptation to other domains |
| [19] | Hybrid keyword and ontology search | Still relies heavily on keywords; semantic layer shallow | Employs structured keyphrases and deeper semantic similarity computation |
| [20] | Statistical topic model for e-learning | Ignores semantic relationships; topics are statistical groupings | Explicit semantic linking between concepts and topics |
| [21] | Semantic enrichment for academic retrieval using domain ontologies | Limited to narrow domains; lacks cross-domain adaptability | Our method supports domain adaptation with minimal changes |
| [22] | Ontology-based topic extraction for specialized corpora | Lacks integration of keyphrase structure for semantic clarity | Incorporates structured keyphrases in C-ONTO to improve semantic representation |

Table I summarizes the key approaches from related studies, outlining their main techniques, identified limitations, and how the proposed method addresses these shortcomings. This comparative view not only highlights the gaps in existing research but also positions our ontology-based topic model as a more robust and adaptable solution for semantic document retrieval in the Information Technology domain.

By structuring the literature along methodological lines and critically assessing their suitability for IT education, we motivate the need for an ontology-driven, concept-graph-based topic model. C-ONTO operationalizes this need by uniting rich topic semantics, graph-based document modeling, and intent-aware algorithms, yielding demonstrable improvements in retrieval quality and educational relevance.

## III. Materials and Methods

### A. Specification for The Requirements of Topic-Based Search

The architecture of an intelligent search engine built upon a knowledge base. This knowledge base is derived from a domain-specific body of knowledge collected from real-world sources. The process of the knowledge querying system based on the ontology-base in information technology is as follows:

- Firstly, the user will input a query as topic to the system which analyzes the semantic of query and classifies it to a searching problem suitably. If the system cannot understand the query, it will generate some questions to determine the exact searching content.
- Secondly, the search engine is worked to do some intelligent searching techniques based on an organized knowledge base.
- Finally, the system returns a set of documents including ebooks and papers that are relevant to the topic and ranked in descending order of relevance for the user.

### B. The Architecture of Knowledge Retrieving System

The architecture of the knowledge retrieval system is built based on knowledge base model (C-ONTO) as Fig. 1.

- Semantic database: A model created to organize and control collections of documents on computer systems. It enables functions such as accessing, processing, and searching for documents based not only on keywords but also on their meaning. This model brings together several components: sets of documents in the field of information technology, storage files linked to each document, a directory system with naming rules, and structures managed through the relational database approach. Ontology is also incorporated to link these components and define their relationships.
- Semantic search engine: The system applies a specialized matching algorithm to compare the representations of queries and documents, producing a ranked list of documents based on their relevance. Through its interface, the search engine allows users to interact and further refine the search results.
- User interface: This component enables users to enter query sentences into the system and obtain results generated by the system. The retrieved results typically consist of documents, such as e-books or research papers, which are ranked according to their relevance to the user's intended content.
- Query analyzer: This module extracts keyphrases from the user's query, interprets its semantics, and categorizes it into an appropriate search problem. The analysis generates a semantic graph, which is stored in the keyphrase graph component and then passed to the search engine for processing.

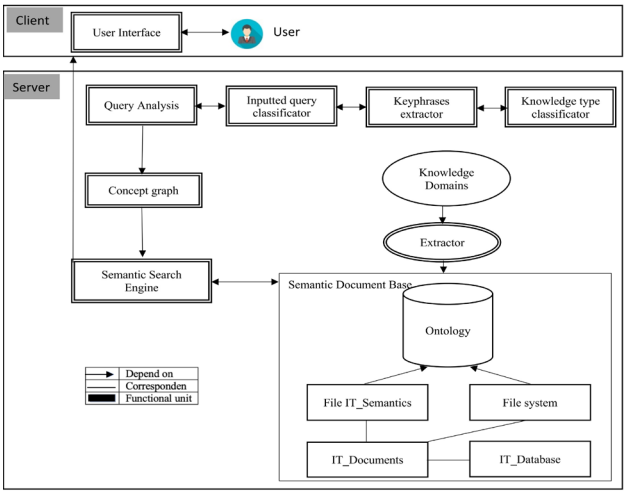


Fig. 1. The architecture of the knowledge retrieve system based on the ontology-base.

### C. Knowledge Based Model

Definition: A knowledge base model (C-ONTO) provides an explicit semantic base to support solving important tasks for document retrieval applications including ebooks and papers by topic, represented by the following 8 components as Fig. 2:

$$(K, C, H, R, \text{Ops}, \text{Rules}, \text{Problems}, \text{Methods})$$

where:

$K$ represents a set of keyphrases within a specific knowledge domain. In the field of information technology, these keyphrases can be categorized into three structural types: single keyphrases ($K_1$), combined keyphrases ($K_2$), and modified keyphrases ($K_3$).

(1) *Single keyphrase set ($K_1$)*: Is a set of single keyphrases, each single keyphrase is a noun or a phrase. fixed cannot be analyzed further in a technical context. Structure of a single keyphrase: $K_1 = \{w / w \in N\}$, where

- $w$ is a word.
- $N$ is a set of nouns or indivisible noun phrases in the technical context.

For example: $K_1$ = {"Algorithm", "Data", "Operating System"}.

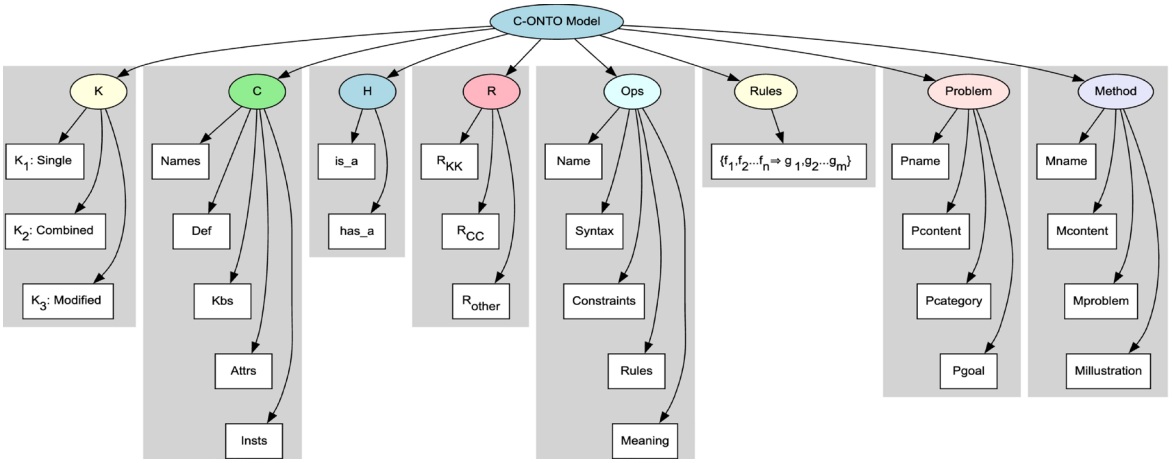


Fig. 2. Main components in the knowledge base model (C-ONTO).

(2) *Combined keyphrase set ($K_2$)*: Is a combination of multiple single keyphrases or between single keyphrases. Structure for compound keyphrases: $K_2 = \{(w_1, w_2, ..., w_n) \mid w_i \in T \text{ and } (w_1, w_2, ..., w_n)\}$ where

- $w_i$ are the words in the combined keyphrase.
- $T$ is the set of all word classes such as nouns, adjectives, verbs, gerunds, etc.
- The words $w_1$, $w_2$, ..., $w_n$ must have semantic relations to each other, such as in the following structures such as [Adj + N], [N + N], [N + Prep + N], [N + C + N], [V + N], [V-ing + N], [N + Prep + Adj], [Adj + N + N], [V + Prep + N].

For example: $K_2$ = {"Distributed Systems", "Open Source", "Big Data"}.

(3) *The set of modified keyphrases ($K_3$)*: Set of modified keyphrases, in which a keyphrase —either single or combined—is enriched with additional meaning by an adjective or an adverb. The general structure for modified keyphrases is: $K_3 = \{(K, \text{adj}) \mid K \in (K_1 \cup K_2) \text{ and adj} \in \text{Adjective}\}$ or $K_3 = \{(K, r, \text{adj}) \mid K \in (K_1 \cup K_2), r \in \text{Adverb and adj} \in \text{Adjective}\}$, where

- $K$ is a keyphrase belonging to the set $K_1$ or $K_2$.
- Adj is an adjective.
- r is an adverb that complements an adjective.

For example: $K_3$ = {( "Big Data" , "important"), ("Intelligent system", "advanced")}

Thus, the set of keyphrases includes: $K = K_1 \cup K_2 \cup K_3$.

(4) *Labeling function for classifying keyphrase*: The semantics of a keyphrase depends not only on its content but also on the classification level in the system. To ensure systematic and scientific nature, the classification labeling of keyphrases is performed based on the ACM Computing Classification System (ACM CCS)—a standard classification system in the field of information technology as Table II.

$C$ is the set of key concepts in a given knowledge domain in the public sector information technology consists of concepts. A concept is a fundamental unit of knowledge that represents a set or class of entities or "objects" within a specific knowledge domain. The component $C$ is divided into the following types:

(1) $C_0 = \{c \mid \textit{domain (c)} \subset \mathbb{R} \textit{ or Attrs(c)} = \emptyset \textit{ or c is a foundation concept of domain}\}$, dom($c$) is the value domain of $c$. For example, the concept *ARITHMETIC OPERATOR* is a foundation concept in programming PHP. Its instance set Insts includes the operators {+, −, *, /, %}.

(2) $C_k = \{c \mid (\exists x \in \text{Attrs}(c) \cup \text{Insts}(c), c' \in C_{k-1}, \forall y \in \text{Attrs}(c') \cup \text{Insts}(c'), x \in c', y \notin \text{Attrs}(c) \cup \text{Insts}(c))\}, (k \geq 1)$. For example, the concept *FOR LOOP* has Insts = {for($i$ = 0; $i$ < $n$; $i$++), for each $x$ in list, for $i$ in range($n$)}, and the concept *WHILE LOOP* has Insts = {while ($x$ > 0), while not end_of_file(), while(True)}. Both *FOR LOOP* and *WHILE LOOP* belong to $C_1$. The concept *LOOP STATEMENT* belongs to $C_2$ as Table III.

Each $c$ ($c \in C$) has the following structure: (*Names, Definition, K_base, C_Attrs, C_Insts*), where:

(1) *Names* $\neq \emptyset$ *:* The concept's name is used to represent the concept. It is represented as a string. *Example*: Names = "Algorithm".

(2) *Definition:* An element that describes the definition of a concept. Each element $d \in$ Def that follows this structure: *(d_name, d_text)*, where:

- *d_name*: The text string used to name the definition.
- *d_text*: A text string describing the content of the definition.

TABLE II. LABELING KEYPHRASES ACCORDING TO ACM STANDARD

| Keyphrase | Label |
|---|---|
| "Knowledge representation" | Computing methodologies → Artificial intelligence → Knowledge representation and reasoning |
| "Machine Learning" | Computing methodologies → Machine learning |
| "Neural Network" | Computing methodologies → Neural networks |
| "Cybersecurity" | Security and privacy → Security services |
| "Cloud Computing" | Computer systems organization → Distributed architectures |
| "Blockchain" | Security and privacy → Cryptography |

For example: The concept "Algorithm" has the following definition. ("Informal Definition", "An algorithm is a step-by-step method for solving a problem or accomplishing a task").

We have collected 15,241 concepts in the field of information technology, which are presented in Table IV.

TABLE III. CLASSIFICATION OF SOME CONCEPTS IN THE FIELD OF INFORMATION TECHNOLOGY

| Level ($C_k$) | Concept | Example | Describe |
|---|---|---|---|
| $C_0$ | Variable | {} | Variables in general, represent memory areas that store data in programming. |
| $C_1$ | Integer Variable | {int, short int, long int, unsigned int} | Integer type variable. |
| - | Floating-Point Variable | {float, double, long double} | Real number type variable. |
| - | Character Variable | {char, wchar_t} | Character variable. |
| $C_2$ | Array Variable | {int[], float[], char[], string[]} | Array variables contain multiple elements. |
| - | Pointer Variable | {int*, char*, void*} | Pointer variable points to memory address. |
| - | Reference Variable | {int&, float&} | Reference variable. |
| $C_3$ | Class Variable | {object, instance} | A variable belongs to a class in object oriented programming. |
| - | Generic Variable | {template<T>, generic<T>} | Global variable (used in generic programming). |

TABLE IV. CONCEPTS IN THE FIELD OF INFORMATION TECHNOLOGY

| Name | Type | External Linked Content | Internal Content of the Concept | Definition | References |
|---|---|---|---|---|---|
| Knowledge Representation | Composite | Knowledge, AI, Semantic Web | Problems in Knowledge Representation, Methods of Knowledge Representation, Applications of Knowledge Representation, Reasoning Techniques in Knowledge Representation | Methods for Describing and Representing Knowledge, helping computers understand and process information like humans. | Book: Artificial Intelligence: A Modern Approach–Russell and Norvig [23] |
| Computational Network | Composite | Semantic Network, Cloud Computing, Machine Learning, Distributed Network | Problems of Computational Networks, Problem-Solving Methods for Computational Networks, Applications of Computational Networks | Each computational network is a semantic network containing variables and relationships that can be configured for computation. We consider a computational network as a set of variables along with a set of relationships (such as formulas) for computing between the variables. In specific applications, the values of these variables are often associated with specific concepts about objects, and the relationships reflect a knowledge-based understanding of these objects. | Book: Knowledge Representation and Reasoning by Nhon [24] |
| INTEGER | Basic | Mathematics, Computer Science, Number Theory | Properties of Integers, Integer Operations (Addition, Subtraction, Multiplication, Division, Modulus), Integer Representation (Binary, Decimal, Hexadecimal), Applications in Computing and Cryptography. | A whole number that can be positive, negative, or zero, without any fractional or decimal component. | Book: "Discrete Mathematics and Its Applications"–Rosen [25] |
| Boolean | Basic | Logic, Computer Science, Programming | Boolean Values (True, False), Boolean Operations (AND, OR, NOT, XOR), Boolean Algebra, Boolean Expressions, Truth Tables | A fundamental data type representing two possible values: True (1) or False (0). Used in logic, computation, and decision-making. | Book: Discrete Mathematics and Its Applications–Rosen [26] |
| SQL | Composite | relational algebra, MySQL, PostgreSQL, Oracle, SQL Server. | SQL structure, data management in a relational database management system. | SQL (Structured Query Language) is a structured query language used to manage and manipulate data in a database. | Book: A First Course in Database Systems, Ullman and Widom, Third Edition [27] |
| Database | Composite | Data, Information, Database Management System, Data Integrity and Security, Data Storage and Querying | Database Concept, Organizational Structure | A collection of information that exists over a long period of time. A collection of related data. managed by a DBMS. | Book: A First Course in Database Systems, Ullman and Widom, Second Edition [28] |
| Data Model | Composite | Database, Database Management System, Data Structures | Structure of the data, Operations on the datal, Constraints on the data | A set of concepts used to describe data, organize data, and define relationships between them. | Book: A First Course in Database Systems, Ullman and Widom, First Edition [29] |

(3) *K_base* ⊆ *K*: A concept can be described by a set of content elements that capture its essential meaning. These elements represent the core knowledge defining the concept within the knowledge system and are usually expressed as a set of textual items. For example, the concept *Database System* may be characterized by the following content elements: Kbs = {Definition of database systems, Characteristics of database systems, Types of database systems, Evaluation methods of database systems, Applications of database systems}. Among these, the formal definition of the concept serves as the primary basis for identifying its fundamental keyphrases.

(4) *C_Attrs:* Attributes in Table V are fundamental components that define a concept. All instances of a given concept share the same set of attributes, but each instance may hold different values for them. Formally, each attribute a Attrs is represented as a triple (a_Name, a_Type, a_Range). Here, Name ∈ $K_1$ denotes the keyword identifying the attribute; Type specifies the attribute's data type, which may be a primitive computer type such as string, integer, float, or boolean; and Range defines the possible values. In some cases, the value of an attribute can be an instance of another concept, meaning the attribute's range is a set of concepts from which such instances can be derived.

(5) *C_Insts:* Instances form the set of elements that belong to a concept, representing its extended components. Each instance follows the structure defined by the concept and is expressed as a pair (Name, Values), where Name ∈ $K \backslash K_1$ denotes the keyword phrase identifying the instance, and Values represent the corresponding set of attribute values. The sets of instances and attributes are assumed to be disjoint. In cases where a concept has no attributes (empty Attrs) but does include instances (non-empty Insts), each instance in Insts is described by its name along with an empty value set as Table VI.

TABLE V. SOME PROPERTIES OF THE CONCEPT "OPERATING SYSTEM"

| Attribute name | Type | Range | Sample Value |
|---|---|---|---|
| isMultiUser | Boolean | {true, false} | true, false |
| isOpenSource | Boolean | {true, false} | true, false |
| systemType | Instance | {Real-time, Distributed} | Real-time, Distributed |
| kernelType | Instance | {Monolithic, Hybrid} | Monolithic, Hybrid |
| supportedDevices | Instance | {Device Type} | Printer, Disk, Network card |
| fileSystemType | Instance | {File System} | NTFS, ext4, FAT32 |
| applications | Instance | {Application Domain} | Mobile devices, Servers |

TABLE VI. THE INSTANCES OF THE CONCEPT "OPERATING SYSTEM"

| Name | Attribute | Values |
|---|---|---|
| Windows 10 | isMultiUser | True |
| - | isOpenSource | False |
| - | systemType | Time-sharing |
| - | kernelType | Hybrid |
| - | supportedDevices | Printer, disk, GPU |
| - | fileSystemType | NTFS |

*H* refers to the set of relations of type *IS_A* and *HAS_A*, defined both over the set of concepts *C* and over the instances of each concept ($c \in C$). For example, the concept *Computer* has a *HAS_A* relation with *CPU.*

*R* is binary relations in the knowledge domain. *R* has 3 types:

(1) $Relation_{KK}$*:* This component defines binary relations among keyphrases within a knowledge domain. C-ONTO not only models concepts and their links but also functions as a lexical model by grouping keyphrases with similar meanings and assigning semantic relation labels. A binary relation *r* on keyphrases *K* is a subset of $K \times K$, where $(x, y) \in r$ means that keyphrase *x* is related to keyphrase *y*, written as *x r y*. These relations, which can be conceptual, semantic, or lexical, play an important role in semantic retrieval:

- *Semantic equivalence relations:* Equivalence relations connect keyphrases with the same or similar meanings, allowing them to be used interchangeably. Two main types exist. The first is the *abbreviation relation* (*rabbr*), which links an acronym to its full form, such as DBMS and Database Management System. Since different terms can share the same acronym (e.g., Data Mining and Direct Messaging both abbreviated as DM), this relation is not symmetric or transitive. The second is the *synonymy relation* (*rsyn*), which ties together keyphrases that can fully substitute for each other, such as Knowledge Graph and Semantic Network. This relation is both symmetric and transitive, making it useful for grouping semantically equivalent terms. In practice, synonym sets are organized using a hub-and-spoke model, where one central keyphrase—typically the most widely adopted in the literature—acts as the hub connecting to its synonyms.
- *Relations of syntactical:* These relations connect compound keyphrases with their constituent parts. In *dvanda compounds*, a simple *formed-by* relation ($r_{formby}$) links the compound to each component. For *endocentric compounds*, the structure is more specific: one part serves as the head and the other as a modifier. Accordingly, two relations are defined—*headed-by* ($r_{headby}$) and *modified-by* ($r_{modby}$)—to associate the compound with its head and modifier.
- *Concept-driven semantic relations:* In information retrieval, many tasks rely on processing terms and their relationships, even without analyzing deeper conceptual structures. To support such tasks, our model extends $R_{KK}$ with a set of relations adapted from $R_{CC}$: hierarchical ones ($r_{hyp}$, $r_{part}$, $r_{sub}$) and non-hierarchical ones ($r_{hypsib}$, $r_{partsib}$, $r_{subsib}$, $r_{range}$, $r_{related}$).

Each keyphrase-to-keyphrase relation can be explicitly defined or derived from elements of $R_{CC}$ as Fig. 3. Because a keyphrase may denote a concept, an attribute, or an instance, specific mapping rules are required to align keyphrase relations with concept-level relations.

(2) *Relation$_{CC}$:* The set of binary relations between concepts in the Information Technology domain constitutes an essential component of its knowledge structure: A binary relation r on a set of concepts $C$ can be described as a subset of the Cartesian product $C \times C$. Formally, this means that $r \subseteq C \times C$, where each element of $r$ is an ordered pair ($c_1$, $c_2$) with $c_1$, $c_2 \in C$. The role of such a relation is to indicate connectivity between concepts: concept $c_1$ is related to concept $c_2$ precisely when the pair ($c_1$, $c_2$) is contained in $r$. The expression ($c_1$, $c_2$) $\in r$ is interpreted as "concept $c_1$ is related to concept $c_2$ through relation $r$". This relation is commonly denoted in infix form as $c_1\ r\ c_2$.

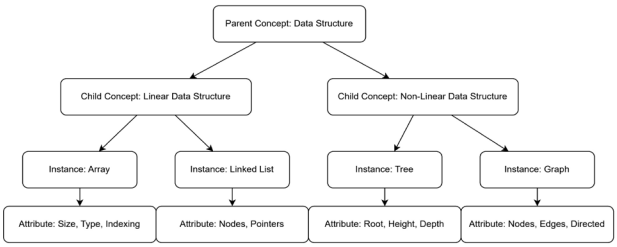


Fig. 3. Structure of concept data structure in information technology domain.

*1) Relations of hierarchical among concepts:* The primary forms of these relations are:

*Relations of hierarchical*: One of the most fundamental types of relations is Hyponymy (also known as the "is-a" or "kind of" relation). This relation connects a specific concept to a more general one. For example, the concept BINARY TREE is a more specific case of the broader concept DATA STRUCTURE. We denote this relation as $r_{HYP} \in Relation_{CC}$. An important characteristic of hyponymy is that it provides insights into both the attributes and instances of concepts. Formally, for two concepts $c_1, c_2 \in C$, the relation $c_1\ r_{HYP}\ c_2$ holds if and only if the following conditions are satisfied.

- For any concepts $c_1, c_2 \in C$, every instance of $c_1$ is also an instance of $c_2$.
- Every attribute that belongs to $c_2$ is also inherited by $c_1$.

A class may contain several subclasses or be part of other broader classes. A subclass is defined as a class that inherits certain properties from its corresponding superclass. Such inheritance relationships naturally establish a hierarchical structure among classes.

*Relation of meronymy* ($r_{PART}$), also referred to as the *part-of*, *part–whole*, or *has-a* relation, represents another important type of hierarchical relation between concepts. For instance, a KEYBOARD can be regarded as a part of a COMPUTER.

*Relation of sub-topic* ($r_{SUB}$), specifies that one concept can be considered a sub-area of another, for example, MACHINE LEARNING as a sub-area of DATA SCIENCE, or NETWORK SECURITY as a sub-area of CYBERSECURITY. While these so-called "topical" concepts are not always easy to capture structurally, identifying their hierarchical relations is essential in various information retrieval tasks.

*2) Relations of non-hierarchical:* From the previously defined hierarchical relations, we can also derive three sibling relations, denoted as $r_{HYPSIB}$, $r_{PARTSIB}$, and $r_{SUBSIB}$. Two concepts are treated as siblings when they share the same immediate parent within the hierarchy. Moreover, there exists the domain–range relation, $r_{RANGE}$, which connects one concept to another that falls within the value range of its attributes. More precisely, for $c_1, c_2 \in C$, if an attribute a of $c_1$ is typed as an "instance" and $c_2$ belongs to the range of a, then the relation can be expressed as $c_2\ r_{RANGE}\ c_1$. For example: PROGRAMMING LANGUAGE is within the range of the attribute "implemented in" of SOFTWARE, so (PROGRAMMING LANGUAGE, SOFTWARE) $\in r_{RANGE}$.

TABLE VII. ILLUSTRATIVE PROPERTIES OF RELATIONS FOR THE CONCEPT "DATABASE"

| Relation Type | Typical Properties |
|---|---|
| Relation of hierarchical | Often transitive (sub-databases inherit from general databases), reflexive, antisymmetric |
| Relation of domain-range | Usually antisymmetric because the attribute range has a one-way association. |
| Relations of sibling | Can be symmetric (two databases at the same level are siblings), reflexive, and sometimes transitive if they belong to the same hierarchy. |

Depending on the specific knowledge domain in the Information Techonlogy, concepts can also be connected through a wide range of non-hierarchical semantic relations. Unlike hierarchical ones, these relations do not establish parent–child structures and are often less strictly defined. Examples of such connections include relations like Expansion, Cause, Influence, Instrument, Creation, Ownership, Source, Goal, Location, Time, Manner, Support, Beneficiary, Attribute, Agent, Context, and many others. Similar to binary relations in general, these non-hierarchical links can exhibit logical properties such as symmetry, transitivity, or reflexivity. A summary of common relation properties in the Relation$_{CC}$ framework is provided in Table VII.

*3) Relation$_{Other}$:* The set of binary relations can also be defined among other components of a knowledge-based model. For instance, a relation may link a problem with a method through the statement "a problem can be addressed by a method", or connect a method with a rule through the statement "a method makes use of a rule".

*Ops:* The operation model on the concept has the following structure: *(name, syntax, constraints, rules, meaning)* as Table VIII, where:

- name: Set of operation names. (*Set[String]*)
- syntax: Represents the syntax. (*String*)
- constraints: Constraint checking function. (*Function[I]* → *Boolean*)
- rules: Function to execute the operation. (*Function[I]* → *O*)
- meaning: Mapping the operation name with the description. (*String*)

*Rules:* A collection of deductive rules can be established over facts associated with keyphrases and concepts. Each

rule is generally expressed in the form: $r$: $\{t_1, t_2, ..., t_n\} \Rightarrow \{k_1, k_2, ..., k_m\}$ where $\{t_1, t_2, ..., t_n\}$ represent the premise facts, and $\{k_1, k_2, ..., k_m\}$ denote the conclusion (or goal) facts that follow from the rule.

TABLE VIII. SOME PROPERTIES OF THE CONCEPT "ALGORITHM"

| name | syntax | constraints | rules | meaning |
|---|---|---|---|---|
| Append | list.append(item) | List must exist | Add item to end of list | Add element to list |
| Insert | list.insert(index, item) | index must be valid | Insert item into index position | Insert element into list |
| Remove | list.remove(item) | item must exist | Remove the first element with the value item | Remove element from list |
| Pop | list.pop(index) | index must be valid (default is -1) | Get and remove element at index | Remove and return elements |
| Sort | list.sort() | Elements must have the same data type | Sort the list in ascending order | Sort list |
| Reverse | list.reverse() | Do not have | Reverse the list | Reverse the order of elements |
| Extend | list.extend(iterable) | iterable must be an iterable list or set | Add elements from iterable to list | Expand the list |
| Index | list.index(item) | item must exist in list | Returns the first index of the item | Find element location |
| Count | list.count(item) | Do not have | Count the number of times an item appears | Count duplicate elements |

Facts are specific assertions that describe aspects such as the properties of relations, the connections between keyphrases, or the links among concepts. The following notes outline the main types of facts considered in this context:

(1) Facts about relation properties are expressed in the form [<relation symbol> is <property>]. For instance, [$r_{syn}$ is symmetric] indicates that the synonym relation between keyphrases possesses the property of symmetry.

(2) Facts about relations between keyphrases are expressed in the form [<first keyphrase> <relation symbol> <second keyphrase>]. For example, ["Relational database" $r_{hyp}$ "Database system"] indicates that the keyphrase *Relational database* is a hyponym of the keyphrase *Database system.*

(3) Facts about relations between concepts are represented in the form [<first concept> <relation symbol> <second concept>]. For instance, ["DATA MINING" $r_{SUB}$ "DATA SCIENCE"] expresses that the concept *Data Mining* is a sub-topic of the concept *Data Science.*

Some examples of rule include can be described as follows: $\forall k_1, k_2, k_3 \in K$, $\forall r \in S_{R_{KK}}$ where $S_{R_{KK}}$ is a set of symbols (or names) of the relations in $Relation_{KK}$.

- $r_1$: if [$r$ is symmetric] and $[k_1 r k_2]$ then $[k_2 r k_1]$.
- $r_2$: if [$r$ is transitive] and $[k_1 r k_2]$ and $[k_2 r k_3]$ then $[k_1 r k_3]$.
- $r_3$: if $[k_1\, r_{syn}\, k_2]$ and $[k_2 r k_3]$ then $[k_1 r k_3]$.

*Problems:* Problems is a set specific problems, general problems, scientific issues, specific applications are mentioned in the knowledge domain. Each $p \in$ Problems structure consists of 4 components as follows:

$$(P_{name}, P_{content}, P_{category}, P_{goal})$$

Each component is described in Table IX and an example in Table X.

TABLE IX. PROBLEM COMPONENTS

| Element | Type | Meaning |
|---|---|---|
| $P_{name}$ | $P_{name} \in$ String | Use a string to describe the topic name |
| $P_{content}$ | $P_{content}$: P → Text | Describe the problem as HTML text |
| $P_{category}$ | $P_{category} \in$ {"specific", "general", "scientific problem", "application"} | Categorize the problem, possibly into classes such as "specific", "general", "scientific problem", or "application" |
| $P_{goal}$ | $P_{goal}$: P → String | Problem objectives (solution-oriented, results achieved) |

TABLE X. EXAMPLE OF THE PROBLEM "DATABASE DESIGN"

| Element | Content |
|---|---|
| $P_{name}$ | Database Design |
| $P_{content}$ | Designing a database involves defining its schema, structure, and relationships between entities to meet specific application requirements. The process typically includes identifying entities, attributes, and relationships, creating an Entity-Relationship (ER) diagram, normalizing data to avoid redundancy, and implementing the design using a database management system (DBMS). Challenges include ensuring scalability, efficiency, and data integrity. |
| $P_{category}$ | specific |
| $P_{goal}$ | To create a well-structured database schema that ensures efficient data storage, retrieval, and management, supporting the functional requirements of an application while maintaining data integrity and consistency. |

*Methods:* A set of methods to solve problems. Each m ∈ methods has a structure consisting of 4 components:

$$(M_{name}, M_{content}, M_{problem}, M_{illustration})$$

Each component is described in Table XI and an example in Table XII.

TABLE XI. METHOD COMPONENTS

| Element | Type | Meaning |
|---|---|---|
| $M_{name}$ | $M_{name} \in$ String | A string that identifies the method, using only the most common keyphrase even if multiple names exist. |
| $M_{content}$ | $M_{content} \in$ Text | Is the HTML text that describes the contents of the method. |
| $M_{problem}$ | $M_{Problem} \in$ Text | Is the problem that this method is designed to solve. |
| $M_{illustration}$ | $M_{Illustration} \in$ Text | Is a collection of figures illustrating some practical issues for the method. |

TABLE XII. EXAMPLE OF "RELATIONAL DATABASE NORMALIZATION" METHOD

| Element | Content |
|---|---|
| $M_{name}$ | Relational Database Normalization |
| $M_{content}$ | A ddressing data redundancy and ensuring data integrity in relational database design. |
| $M_{Problem}$ | Relational database normalization is a systematic approach to organizing data in a database to reduce redundancy and dependency. The process involves dividing a database into smaller tables and defining relationships between them. It typically follows a series of normal forms (1NF, 2NF, 3NF, BCNF, etc.), each with specific requirements. For example, 1NF eliminates duplicate columns, 2NF removes partial dependencies, and 3NF ensures that no transitive dependencies exist. |
| $M_{illustrator}$ | In a student database, separating personal details (eg, name, address) from academic records (eg, courses, grades) ensures data integrity and avoids duplication. |

### D. Topic Model

*Definition:* A topic within a field of study includes many related concepts. A topic often covers various important concepts, problems, and different aspects of a research area. The topic model in the field of Information Technology has the following structure:

(*Topics, R, Rules, Problems, Methods*), where

(1) *Topics:* Set of topics. Each $t \in$ Topic has a structure including:

(*Name, Content, Goals, Level, Domain*)

Each component is described in Table XIII.

TABLE XIII. STRUCTURAL ELEMENT OF A TOPIC REPRESENTATION

| Element | Type | Meaning |
|---|---|---|
| Name | Name ⊆ String | The name of the topic, represented as a string of characters. |
| Content_Base | $G = (V, E)$ be a concept graph representing the topic. Where:<br>• $V$ is the set of nodes, where each node $v \in V$ corresponds to a key content element and has the same structure as a concept in a knowledge base model.<br>• $E$ is the set of edges, where each edge $e = (v_i, v_j) \in E$ represents a semantic relationship between two content elements $v_i$ and $v_j$. | The list of feature content of the topic is described using a concept graph. |
| Goals | Goals ⊆ String | The goal of the topic, represented as a string of characters. |
| Level | Level ⊆ String | Use a string to represent the level, which should correspond to one of the following levels: basic, intermediate, or advanced. |
| Domain | Domain ⊆ String | Use a string to represent the field, which should correspond to one of the following areas: Artificial Intelligence, IoT, Cybersecurity, etc. |

*Example*: Topic "Knowledge Representation" is represented as follows.

- Name: "Knowledge representation".
- Intro: "Knowledge Representation (KR) in AI is the process of structuring knowledge in a formal way so machines can understand, reason, and solve problems. It enables intelligent tasks like reasoning, planning, and natural language understanding".
- Content_Base: {Knowledge representation methods, Knowledge models, Knowledge, representation tools}. This components are represented as a concept graph as Fig. 4.
- Goals: "Building intelligent systems that can "understand", reason about, and act upon the world".
- Level: "advanced".
- Domain: "Artificial Intelligence".

(2) *R:* Includes 2 types of relationships $R_1$ and $R_2$ as follows:

- $R_1$: $R_1$ is the set of relationships between topics. $R_1 = \{r_{sub}, r_{related}\}$, where $r_{sub} \subseteq$ *Topics*: Hierarchical relationship between topics. Example: $r_{sub}$("Deep Learning", "Machine Learning"): Deep Learning is a sub branch of Machine Learning. $r_{related} \subseteq$ *Topics*: Relationships between topics. Example: rrelated ("Python", "Machine Learning") Python and Machine Learning are related.
- $R_2$: Is the relationship between the components in the topic model, example: Machine Learning topics can suffer from overfitting problems.

(3) *Rules*: A collection of deductive rules can be established over facts associated with keyphrases and concepts. Each rule is generally expressed in the form: $r$: $\{t_1, t_2, ..., t_n\} \Rightarrow \{k_1, k_2, ..., k_m\}$ where $\{t_1, t_2, ..., t_n\}$ represent the premise facts, and $\{k_1, k_2, ..., k_m\}$ denote the conclusion (or goal) facts that follow from the rule.

- The occurrence of a relationship between topics is expressed in the form: [<first topic >< relation symbol >< second topic >] Example: ['Knowledge representation methods' $r_{SUB}$ 'Knowledge representation'] means that the topic 'Knowledge representation methods' is a subtopic of the topic 'Knowledge representation'.
- The relationship law between topics has the form: $\forall tp_1, tp_2, tp_3 \in$ Topics, $[tp_1\ R\ tp_2] \wedge [tp_2\ R\ tp_3] \Rightarrow [tp_1\ R\ tp_3]$ where, R is a relation between topics, Example: $r_{SUB}$ (sub-topic). Example: [Deep Learning $r_{SUB}$ Machine Learning] ∧ [Machine Learning $r_{SUB}$ AI] ⇒ [Deep Learning $r_{SUB}$ AI].

(4) *Problems and Methods:* This model is discussed in detail in Section III.D.

We selected the C-ONTO model because it provides a structured and semantically rich representation of domain knowledge, enabling precise reasoning and retrieval. Unlike keyword-based or purely statistical models, C-ONTO captures explicit relationships among concepts, problems, and methods, allowing the system to interpret queries and documents in a context-aware manner.

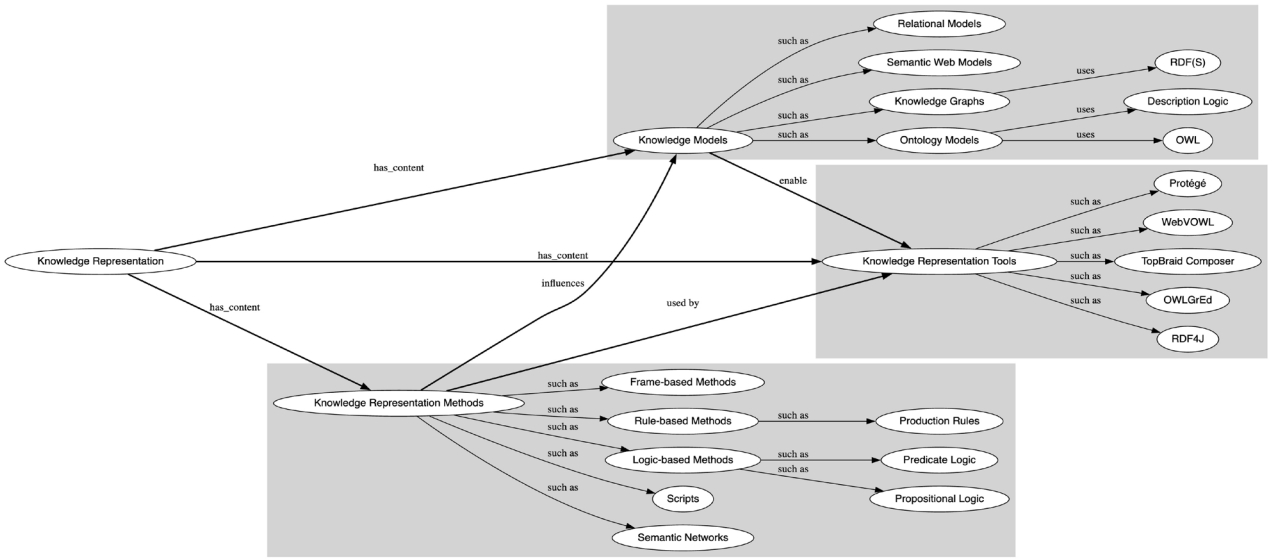


Fig. 4. Conceptual graph on the topic of knowledge representation.

### *E. Document Representation*

*1) Representing Documents Using Concept Graphs(CG)*: Definition: A CG is a set consisting of three components ($G_K$, $E$, $b$), where:

- $G_K \subset K$: Is a finite, non-empty set of keyphrases or concepts defined in the C-Onto model, serving as the graph's vertex set. Each vertex corresponds to a keyphrase or concept extracted from documents, representing an important semantic element.
- *E:* Is a finite set of elements in the set $G_K \subset G_K$, is called the set of edges of the graph. Each edge denotes a semantic relation between two vertices, capturing connections between keyphrases or concepts, whether semantic or structural.
- *b:* $E \rightarrow Relation_{KK}$*:* b is a labeling function for the edges of a graph such that: an edge $e$ is labeled by $b(e) \in \text{Relation}_{KK}$ is a relation between two keyphrase vertices adjacent to $e$. The function b labels the relations between vertices, with $b(e) \in R_{KK}$ representing different semantic relations, such as "belongs to", "consists of", "such as", etc.

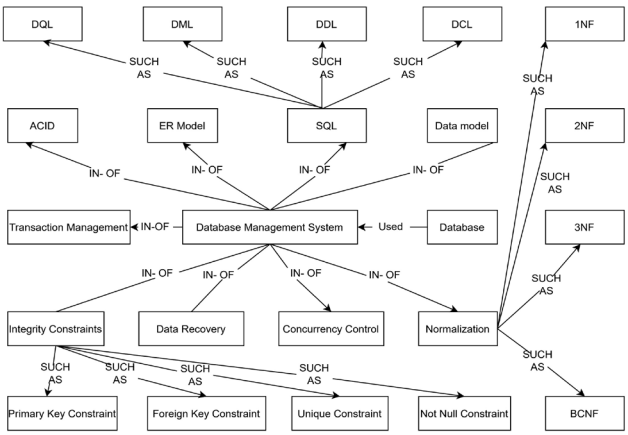


Fig. 5. Document representation for the ebook "An Introduction to Database Systems" using a Concept Graph (CG), assuming the keyphrases and concepts extracted from the document.

This representation method was chosen because mapping standardized keyphrases directly to ontology concepts ensures semantic consistency, minimizes lexical ambiguity, and improves retrieval accuracy. By aligning document content with the structured knowledge base, the system can better match user queries with relevant resources as Fig. 5.

*2) Determine the semantic similarity between the two concept graphs*

The semantic similarity of two concept graphs—one for a Topic ($T$) and one for a Document (D)—yields a value between 0 and 1, as defined by Eq. (1) in [1]:

$$Rel(T, D) = Max\{v(\Pi) | \Pi \text{ is a partial mapping from } T \text{ to } D\} \quad (1)$$

where:$v(\Pi)$ is a valuation model for projections $\Pi(f, g)$ from graph $T$ to graph $D$, is defined by Eq. (2) in [1]:

$$v(\Pi) = \frac{\sum_{k \in KT} tf(g(k),D) \times \alpha(k,g(k)) \times ip(g(k),D) + \sum_{r \in RT} \beta(r,f(r))}{(|KT| + |RT|)} \quad (2)$$

where:

- $\alpha$ and $\beta$ have been mentioned in [1].
- *tf*: The frequency of a keyphrase in a document, denoted as *tf*($k$, $d$), represents how often keyphrase $k$ appears in document $d$ and is computed using by Eq. (3) in [1]:

$$tf(k, d) = \frac{n_k}{\sum_{i \in d} n_i} \quad (3)$$

- ip: The positional importance of a keyphrase, denoted as *ip*($k$, $d$), reflects the significance of keyphrase $k$ in document $d$ based on its occurrence in specific sections (title, summary, content, tags). It is computed by Eq. (4) in [2]:

$$ip(k, d) = \frac{\sum_i w_i n_i}{\sum_i n_i} \quad (4)$$

where:

- $w_j$: $w_j$ is the importance of the $i$th appears in the document ($w_i \in [0, 1]$ and $\sum_i w_i = 1$).

- $n_j$: Denotes the number of occurrences of keyphrase $k$ in document $d$.

We adopted this similarity computation approach because it leverages both the ontology's structural relationships and semantic weighting, combining them with lexical matching to achieve a more accurate assessment of relevance. This hybrid method overcomes the limitations of relying solely on statistical or keyword-based measures.

### F. Problems and Algorithms for Searching Documents by Topic

Searching for topics in ebook and paper documents to serve students' learning and research process is one of the important functions in the system. From reputable scientific conference websites and experts in the field of artificial intelligence as well as real users, a list of topics related to the field under consideration and users will be created. will select topics and the system will produce results of documents related to the selected topic.

The task of topic-based document retrieval is defined as: given a collection of ebooks and papers in the field of Information Technology and a topic $T$.

The semantics of a topic is represented through a set of content specific to the topic. An ebook and paper document includes a set of keyphrases specific to a field under consideration. Therefore, the topic proposes a formula to calculate the correlation between a document and a topic based on the idea of finding the correlation between keyphrases in the topic and keyphrases in the document, specifically as Eq. (5) in [1]:

$$\frac{\sum_{j=1}^{n} relevance(k_j, A)}{n} \tag{5}$$

Here, $n$ denotes the number of keyphrases in topic $T$, and $relevance(k_j, A)$ represents the degree of correlation between keyphrase $k_j$ and document $A$, which is computed using Eq. (6) in [1].

$$relevance(k, A) = \text{Max}(\text{tf}(k_i, A) \times \alpha(k, k_i) \times ip(k_i, A) \mid k_i \; is \; the \; keyphrase \; in \; A) \tag{6}$$

Example: Suppose we have the topic "Knowledge representation" with the following typical content set:

Knowledge representation = {($F_1$: Knowledge representation methods), ($F_2$: Tools for Knowledge Representation), ($F_3$: Knowledge models)} Document $D$ with the following set of keyphrases:

{(Knowledge representation methods, $tf$ = 0.8, $ip$ = 0.9), (Tools for Knowledge Representation, $tf$ = 0.9, $ip$ = 0.9), (Knowledge models, $tf$ = 0.9, $ip$ = 0.85), (*Artificial Intelligence*, $tf$ = 0.7, $ip$ = 0.6)}. Degree of relationship between keyphrases: $\alpha$(Knowledge representation methods, Knowledge representation methods) = 1, $\alpha$(Tools for Knowledge Representation, Tools for Knowledge Representation) = 1, $\alpha$(Knowledge models, Knowledge models) = 1. Relevance between documents $D$ and the topic "Knowledge representation" is calculated as Eq. (8) in [1]:

$$\frac{relevance(F1,D) + relevance(F2,D) + relevance(F3,D)}{3} = 0.8 \times 0.9 \; + \; 0.9 \times 0.9 \; + \; 0.9 \times 0.85 = \; 0.75 \tag{8}$$

In summary, the proposed C-ONTO model integrates a concept model, a knowledge graph, and semantic similarity computation to enable precise and context-aware retrieval. The knowledge base model (Section III.C) provides a structured representation of domain concepts and their relationships. Document representation (Section III.D) maps standardized keyphrases to ontology concepts, ensuring semantic consistency and reducing lexical ambiguity. Semantic similarity (Section III.E) then leverages both the ontology's structural relationships and keyphrase semantics to measure the relevance between queries and documents. Together, these components form a coherent processing pipeline that supports semantically enriched and highly accurate topic-based document retrieval.

## IV. Result and Discussion

Based on knowledge base model (C-ONTO) represented in Section III.C, topic model represented in Section III.D, document representation in Section III.E and algorithms for searching documents by topic in Section III.F. This study showcases experimental findings across key areas of Information Technology, such as Databases, PHP programming, Data Structures and Algorithms, and Artificial Intelligence.

The knowledge base of the databases, the PHP programming language, the data structures and algorithms, and artificial intelligence in the field of information technology will be modeled by C-ONTO. It includes eight components.

- *Set of keyphrases:* For examples: SQL query optimization, ACID properties, PHP and MySQL integration, Laravel framework Supervised learning, Neural networks, Natural Language Processing, Graph traversal algorithms, Dynamic programming, Time and space complexity, etc.
- *Set of concepts:* For examples: list, tree, query, queue, stack, table, row, column, primary key, program, programming language, symbol, name, foreign key, library, keyword, SQL, transaction, 1NF, 2NF, 3NF structure of program, float, agent, action, machine learning, deep learning, datatype, numeric data type, logical data type, character data type, integer, etc. Each concept has the structure and organization following the definition in Section IV.
- *Set of the hierarchies:* For examples: a stack *is_a* linear data structure, a queue *is_a* linear data structure, a binary tree *is_a* non-linear data structure, a machine learning algorithm *is_a* AI technique, supervised learning *is_a* machine learning, a primary key *is_a* constraint, a table *is_a* database object**,** a numeric data type *is_a* a data type, long int *is_a* int, a linked list *is_a* dynamic data structure**.**
- *Set of Relations*: For examples: Search algorithm *is used to* solve problems, Programming Language

*supports to* programmer, Programming Language *use to make* application. Function is *a part of* program, variable *is inside* a function, foreign key *refers to* primary key, etc.

- *Set of operators:* For examples: plus (+), substraction (-), multiplication (*), division (/), modules (%), increment (++), decrement (--); equal (==), not equal (!=), greater than (>), less than (<), greater than or equal (>=), less than or equal (=<), etc.
- *Set of Rules*: For examples:
  (1) $r_1$: rules of *symmetric* {*A*: relation, *B*: relation, *C*: relation}, $\{A \cup B = C\} \rightarrow \{B \cup A = C\}\{A \cap B = C\} \rightarrow \{B \cap A = C\}$
  (2) $r_2$: *associate rules* {A: relation, B: relation, C: relation, R:relation}, $\{A \cup (B \cup C) = R\} \rightarrow \{(A \cup B) \cup C = R\}\{ A \cap (B \cap C) = R\} \rightarrow \{(A \cap B) \cap C = R\}$
  (3) $r_3$: *implication rules* {Os: set of objects, As: attribute set, Bs:attributeset, Cs: attribute set, Ds: attribute set, | As, Bs, Cs, Ds ⊆ Os}, $\{Bs \subseteq As\} \models \{As \rightarrow Bs\}\{As \rightarrow Bs\} \models \{AsCs \rightarrow BsCs\}\{As \rightarrow Bs, Bs \rightarrow Cs\} \models \{As \rightarrow Cs\}\ \{As \rightarrow Bs, As \rightarrow Cs\} \models \{As \rightarrow BsCs\}\ \{As \rightarrow BsCs, As \rightarrow Bs\} \models \{As \rightarrow Cs\}\ \{As \rightarrow Bs, BsCs \rightarrow Ds\} \models \{AsCs \rightarrow Ds\}$
- *Problems set:* Common exercises include implementing stacks and queues, building binary trees, performing graph traversals, and writing sorting algorithms.
  (1) Managing and manipulating linear data structures.
  (2) Tree structures and tree operations.
  (3) Graphs and related algorithms.
  (4) Sorting and searching algorithms.
  (5) Dynamic programming and optimization problems.

Each problem has the structure and organization following the definition in Section III.C.

- *methods set:* Some approaches to solve these problems include:
  (1) Use appropriate data structure operations.
  (2) Recursive and iterative tree algorithms.
  (3) Graph traversal and shortest path algorithms.
  (4) Implement standard algorithmic techniques.
  (5) Dynamic programming and memoization.

Each method has the structure and organization following the definition in Section III.C.

The theoretical models described in Sections III.C, III.D, and III.E are directly embedded into the proposed knowledge retrieval system to ensure practical effectiveness. The knowledge base model (C-ONTO) serves as the semantic backbone, organizing and linking domain concepts so that queries can be interpreted in a contextually accurate and semantically consistent manner. The document representation technique indexes and maps standardized keyphrases to ontology concepts, thereby enhancing the precision of query–document alignment. The semantic similarity computation exploits both ontology structure and keyphrase semantics to deliver more accurate and meaningful rankings of relevant documents. As confirmed by the evaluation results, the seamless integration of these theoretical components into the system's architecture significantly improves retrieval accuracy, precision, recall, and topic coverage when compared to baseline systems.

### A. Expeimental Data Collection Process

To construct a semantically rich experimental dataset suitable for supporting a document retrieval system, we composed a four-step process. This process ensures that the keyphrases and concepts extracted from academic IT materials are not only technically accurate but also semantically consistent. The following flowchart provides an overview of this processing workflow as Fig. 6.

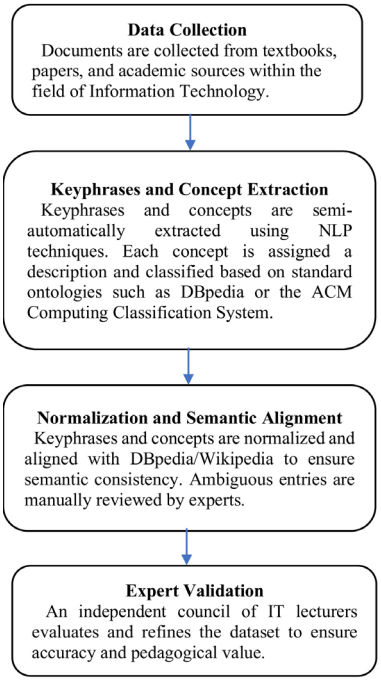


Fig. 6. The four-step process for constructing and validating the experimental dataset.

Upon completing the construction and validation process, the resulting dataset forms a comprehensive knowledge base for semantic retrieval in the IT domain. It comprises 26,150 keyphrases, 15,241 concepts, 2512 topics, 315 problems, and 400 methods, representing a well-structured and domain-relevant corpus of core IT knowledge. This dataset serves as the foundation for evaluating the proposed system's accuracy, topic coverage, and semantic retrieval capabilities.

### B. Testing on Knowledge Domain of Data Structures and Algorithms

The intelligent querying system is designed to search for documents by topic. The search results include ebooks and papers in the domain of data structures and algorithms in information technology field. Fig. 5 illustrates the user interface of the developed intelligent querying system. Users can enter queries in English, and in addition to the

search results, the system also provides other knowledge related to the topics of those results as intellectual tags.

*Example:* query_sentence_user = *"Lookup:concepts related to Tree"*.

*Step 1:* Classify the query sentence into the structure query language on knowledge-based of C-ONTO model following Algorithm 1 and Algorithm 2.

**Algorithm 1: Find documents related to the topic**

**Input:** Docs document set, T topic
**Output:** Documents related to topic T
**Function** GetDocuments(Keyphrases, T)
keyphrasesInTopic: = GetKeyphrasesTopic(T);
resultDocs: ={};//list of documents related to topic T
foreach (d in Docs){
// Calculate the semantic similarity between a document and a topic
relevance = ComputeRelevance(d, keyphrasesInTopic);
if (relevance > δ)
resultDocs.Add(d);
End End
**End**

**Algorithm 2: The algorithm for processing the query sentence**

**Input:** query sentence user, Knowledge of C-ONTO K C;
**Output:** KCE // (Ks, Cons, Ent)
Ks:= {};
Cons:={};
Ent:={};
***for*** **c** in {C, Ops, Funs, Rules, Problems, Methods} *do*
***for*** e in c *do*
***for*** p in Paragraphs(e) *do*
*if* name(p) in query_sentence_user *then*
Ent: = Ent ∪ {name(p)};
End
**End** **End**
**End**
**for** rela in {Relation$_{CC}$, Relation$_{Orther}$} do
*if* Name(rela) in query_sentence_user *then*
Cons := Cons union {Name(rela)};
**End**
**return** KCE;
Ks = {*parent, subtree, height, leaf*}.
Cons = {*related to*}.
Ent = {*Tree*}.

*Step 2:* The system will handle the structure query language on knowledge-based of C-ONTO model (Keys, Conditions, Entities) following Algorithm 3.

**Algorithm 3: Calculation of semantic similarity between the document and the topic**

**Input:** A document D, List of keyphrases of topic T
**Output:** Document relevance D with topic T
//Calculate the relevance of keyphrases in topic T to document D
Weight:={};//contains contrast value relationship between //keyphrase in topic T with a document D;
*//Browse the list of keyphrases in topic T*
**foreach** (k in keyphrasesInTopic)
*//Calculate the correlation between k and document D*
relevance(k, D ):= ComputeRelevant(k, D );
*//Add relevance(k, D ) to Weight;*
Weight.Add(relevance(k, D )); *// using Eq. (7) in [2]*

$$\text{relevance}(T,D) = \frac{\sum_{j=1}^{n} Weight_j}{n} \quad (7)$$

**End**

*Step 3:* The results of the knowledge retrieval system that can show to user as Fig. 7.

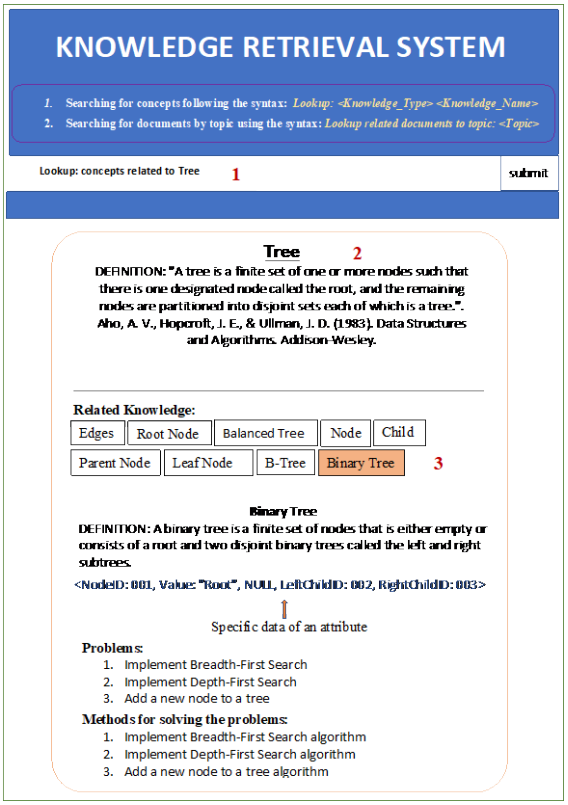


Fig. 7. The user interface for an intelligent querying system in the knowledge domain of data structures and algorithms.

## C. Testing on Knowledge Domain of Artificial Intelligence

In addition to the knowledge of data structures and algorithms domain, the paper also focuses on the implementation of the Artificial Intelligence knowledge domain within the field of information technology.

*Example:* query_sentence_user = "*find the documents related to knowledge representation methods*".

*Step 1:* classify the query sentence into the structure query language on knowledge-based of C-ONTO model following Algorithm 1 and Algorithm 2.

Ks = {*frames, symbolic representation, ontology, fuzzy logic*}. Cons = {*related to*}. Ent = {*knowledge representation methods*}

*Step 2:* The system will handle the structure query language on knowledge-based of C-ONTO model (Keys, Conditions, Entities) following Algorithm 3.

*Step 3:* The results of the knowledge retrieval system that can show to user as Fig. 8.

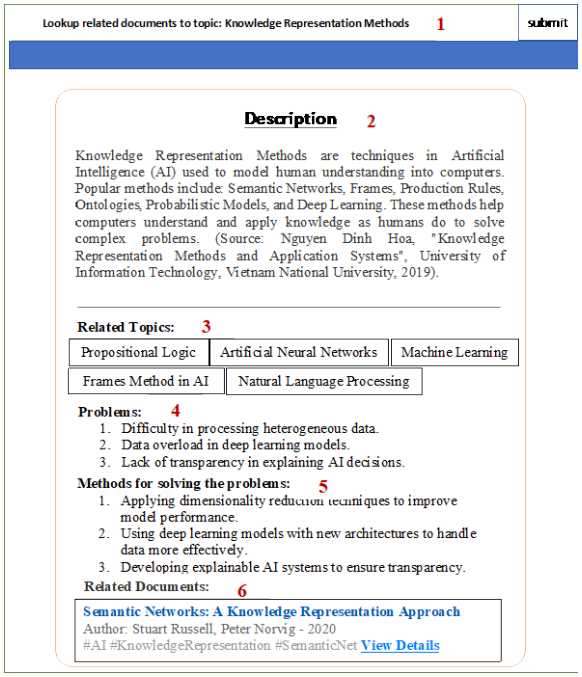


Fig. 8. The user interface for an intelligent querying system in the knowledge domain of artificial intelligence.

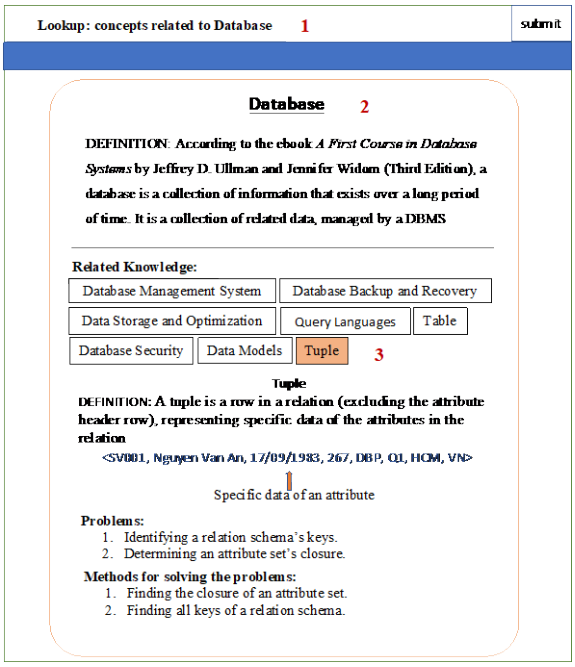


Fig. 9. The user interface for an intelligent querying system in the knowledge domain of database.

## D. Testing on Knowledge Domain of Database

Next, we conduct an experimental implementation on the knowledge domain of Database within the field of information technology.

*Example:* query_sentence_user = "*Lookup: concepts related to Database*".

*Step 1:* Classify the query sentence into the structure query language on knowledge-based model following Algorithms 1 and 2.

*Ks*:: = {relational model, sql, normalization, transaction}. Cons:: = {related to}. Ents:: = {Database}

*Step 2:* The system will handle the structure query language on knowledge-based of C-ONTO model (Keys, Conditions, Entities) following Algorithm 3.

*Step 3:* The results of the knowledge retrieval system that can show to user as Fig. 9.

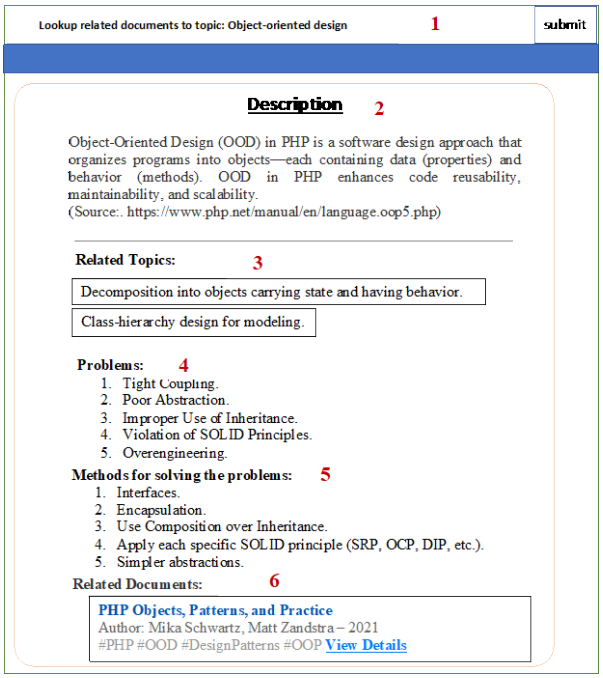


Fig. 10. The user interface for an intelligent querying system in the knowledge domain of of PHP programming language.

## E. Testing on Knowledge Domain of PHP Programming Language

Finally, we will conduct testing on the *PHP* Programming Language knowledge domain within the field of information technology.

*Example:* query_sentence = "*find the documents related to Object-oriented design*".

*Step 1:* classify the query sentence into the structure query language on knowledge-based of C-ONTO model following Algorithm 1 and Algorithm 2.

Ks = {*Class, Object, Encapsulation, Inheritance*}.

Cons = {*related to*}.

Ent = {*Object-oriented design*}

*Step 2:* The system will handle the structure query language on knowledge-based of C-ONTO model (Keys, Conditions, Entities) following Algorithm 3.

*Step 3:* The results of the knowledge retrieval system that can show to user as Fig. 10.

## F. Experimental Results

A total of 425 requirements were collected from 73 students studying information technology at FPT University. These requirements have been classified according to Table XIV.

Table XV is the detailed table dividing the 425 search requirements of students into 4 knowledge domains: Data Structures and Algorithms, Artificial Intelligence, Database, and PHP Programming Language, while also classifying these requirements by difficulty (easy and difficult) as Table XV.

TABLE XIV. THE TABLE OF CLASSIFIED REQUIREMENTS

| No. | The classification of requirements | Quantity |
|---|---|---|
| 1 | The query requirements include the knowledge about a topic. | 125 |
| 2 | The query requirements include the knowledge about a concept, a rule, a property, a problem, a method, the syntax of the statement, a function, a library, an operator, and an exercise. | 95 |
| 3 | The query requirements include the combined knowledge through operator AND, OR, and NOT. | 80 |
| 4 | The query requirements include the knowledge to explain statements, functions, operators. | 20 |
| 5 | Other requirements | 105 |
| | Total | 425 |

TABLE XV. NUMBER AND DIFFICULTY OF STUDENT REQUIREMENTS IN 4 KNOWLEDGE DOMAINS

| Knowledge Domain | Total Number of Requirements | Easy | Difficult |
|---|---|---|---|
| Data Structures and Algorithms | 100 | 60 | 40 |
| Artificial Intelligence | 120 | 70 | 50 |
| Database | 105 | 65 | 40 |
| PHP Programming Language | 100 | 60 | 40 |
| Total | 425 | 255 | 170 |

### G. Compare and Evaluate

In this study, all algorithms are evaluated using the same IT education dataset under identical preprocessing, parameter settings, and evaluation metrics, ensuring that the comparison of topic extraction results is meaningful and valid. To further ensure a fair and unbiased evaluation, all systems were assessed under identical experimental conditions using this common dataset. Specifically, the evaluation corpus comprised 425 topic-based queries along with a standardized academic knowledge base in the Information Technology domain, including 26,150 key phrases, 315 problems, 400 methods, and 15,241 semantic concepts. This unified setup provides a consistent foundation for objectively comparing the performance of our ontology-based retrieval model against several baseline systems. Despite sharing the same input data, our proposed approach demonstrates a marked improvement in both accuracy and topic coverage, yielding consistently superior performance across test scenarios, as evidenced by higher accuracy, precision, recall, and F1-Score in Table XVI.

TABLE XVI. PERFORMANCE COMPARISON OF DOCUMENT RETRIEVAL SYSTEMS IN THE FIELD IT

| The systems | Topic-based search using the Ontology model | Number of topics | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|---|
| Knowledge Querying System in [5] | Yes | 125 | 55.60 | 19.91 | 54.0 | 54.80 |
| Search engine for the knowledge in [8] | No, but it references the Ontology model | 125 | 57.50 | 16.91 | 56.1 | 56.70 |
| Information querying system in [12, 17] | No, but it references the Ontology model | 125 | 69.60 | 20.47 | 68.0 | 68.80 |
| Intelligent searching system in [22] | Yes | 125 | 67.50 | 36.91 | 65.2 | 66.35 |
| Our system | Yes | 125 | 81.18 | 70.15 | 79.5 | 80.30 |

To provide a clearer comparison, the results in Table XV are illustrated in Fig. 11. As shown, the proposed ontology-based system achieves the highest performance in all four metrics—Accuracy, Precision, Recall, and F1-Score—demonstrating its superiority over existing approaches as Fig. 11.

*Evaluation on the field of mathematics*—To explore the potential generalizability of the proposed approach, we conducted a preliminary experiment on a Mathematics knowledge base consisting of 12,450 concepts, 8920 keyphrases, and 210 problem–solution pairs. The same set of algorithms and semantic similarity computations used in the IT domain were applied without modification. Using 150 topic-based queries, the system achieved an accuracy of 78.65%, a precision of 76.20%, a recall of 79.45%, and an F1-score of 77.80%. These results, although slightly lower than those obtained in the IT domain, indicate that the proposed ontology-based model is adaptable to other domains with minimal configuration changes, demonstrating promising generalizability and robustness as Table XVII.

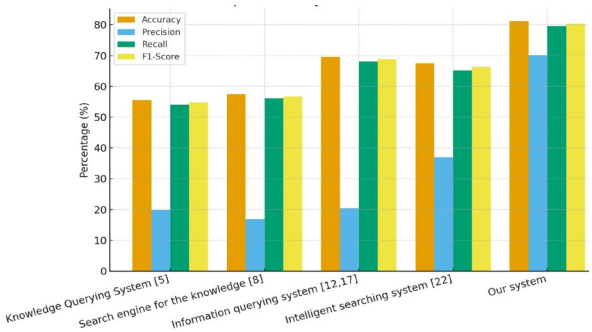


Fig. 11. Comparison of systems on IT domain dataset.

TABLE XVII. PERFORMANCE COMPARISON OF DOCUMENT RETRIEVAL SYSTEMS IN THE FIELD OF MATHEMATICS

| The systems | Topic-based search using the Ontology model | Number of topics | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|---|
| Knowledge Querying System in [5] | Yes | 50 | 68.45 | 67.90 | 69.20 | 68.54 |
| Search engine for the knowledge in [8] | No, but it references the Ontology model | 50 | 69.15 | 67.10 | 70.80 | 68.91 |
| Information querying system in [12, 17] | No, but it references the Ontology model | 50 | 70.30 | 68.40 | 71.50 | 69.93 |
| Intelligent searching system in [22] | Yes | 50 | 72.10 | 70.25 | 73.40 | 71.80 |
| Our system | Yes | 50 | 78.65 | 76.20 | 79.45 | 77.80 |

Compared with existing systems as Fig. 12, the proposed ontology-based model achieves the highest accuracy and balanced precision–recall performance in the mathematics domain, despite no domain-specific optimization. This demonstrates its adaptability and robustness across knowledge.

*Evaluation on the field Road Traffic Law*—To further examine the generalizability of the proposed approach, we performed a preliminary evaluation using a Road Traffic Law knowledge base containing 12,450 concepts, 8920 keyphrases, and 210 problem–solution pairs. The same algorithms and semantic similarity computations employed in the IT domain were applied without any modifications. Using 50 topic-based queries, the system achieved an accuracy of 79.10%, precision of 77.25%, recall of 80.05%, and an F1-Score of 78.63%. The results, although slightly lower/higher than those obtained in the IT domain, demonstrate that the proposed ontology-based model can adapt effectively to other domains with minimal configuration adjustments, highlighting its potential robustness and scalability as Table XVIII.

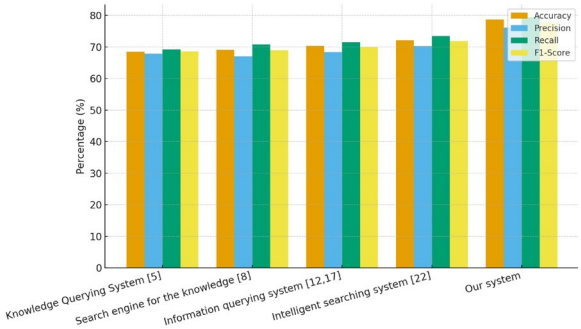


Fig. 12. Comparison of systems on mathematics domain dataset.

TABLE XVIII. PERFORMANCE COMPARISON OF DOCUMENT RETRIEVAL SYSTEMS IN THE FIELD ROAD TRAFFIC LAW

| The systems | Topic-based search using the Ontology model | Number of topics | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|---|
| Knowledge Querying System in [5] | Yes | 50 | 74.25 | 72.10 | 75.35 | 73.69 |
| Search engine for the knowledge in [8] | Yes | 50 | 69.80 | 68.45 | 70.25 | 69.34 |
| Information querying system in [12, 17] | No, but it references the Ontology model | 50 | 71.15 | 69.90 | 2.40 | 71.13 |
| Intelligent searching system in [22] | No, but it references the Ontology model | 50 | 68.95 | 67.50 | 69.85 | 68.66 |
| Our system | Yes | 50 | 79.10 | 77.25 | 80.05 | 78.63 |

To provide a clearer comparison, the results in Table XVIII are illustrated in Fig. 13. As shown, the proposed ontology-based system consistently outperforms existing approaches across all four metrics—Accuracy, Precision, Recall, and F1-score—demonstrating its robustness and adaptability in the Road Traffic Law domain.

The average accuracy estimates of all mentioned systems are presented in Table I [21], Table V [13, 22], and Table II [30]. These systems [31–34] outperform our system because they support Vietnamese input. However, our system excels in handling ability 425 query sentences), with an accuracy of 81.18%. System [35] also achieves high accuracy but does not support combined requirements that include operators.

Besides, we also tested the document search function based on several topics on the interface, as shown in Figs. 8 and 9, compared to existing application interfaces. The results returned by our system, with documents related to the topic, have higher accuracy than those from current application interfaces, as shown in Table XIX. In addition to the clearly designed topic-based search function, the system also provides a knowledge search feature related to concepts, problems, and methods, which have not been addressed by existing application.

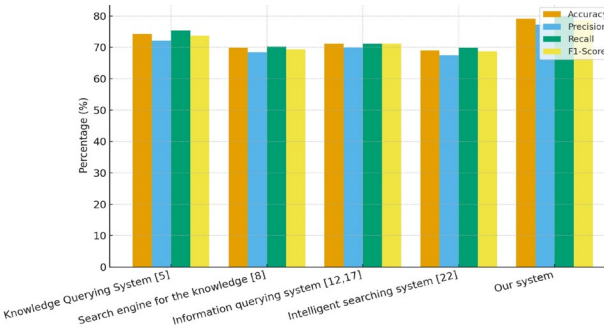


Fig. 13. Comparison of systems on road traffic law domain dataset.

TABLE XIX. CONDUCTING EXPERIMENTS ON TOPIC-BASED DOCUMENT RETRIEVAL USING EXISTING APPLICATION INTERFACES

| Systems | Main Function | Topic-based Search Function | Concept-based Search Function | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|---|---|
| Our application | Search for documents by topic, concept, problem, and method | Yes, designed clearly with high accuracy results | Yes, searches knowledge related to concepts and methods | 81.18 | 80.50 | 81.80 | 81.15 |
| ChatGPT | Natural language interaction, answering questions | Does not directly support topic-based search | Does not support concept-based search | 70.25% | 69.80 | 70.50 | 70.15 |
| Google Dataset Search | Search for scientific datasets | Yes, but mainly focuses on searching datasets | Does not have a concept-based search feature | 65.15 | 64.20 | 65.50 | 64.85 |
| OpenAIRE Explore | Search for research and scientific papers | Yes, but focuses on research and literature | Yes, but mainly for scientific research | 69.25 | 68.90 | 69.40 | 69.15 |

The C-ONTO framework was developed and tested mainly for Information Technology education, but its design is not limited to this domain. By updating the domain ontology, concept graphs, and keyphrase sets to match the knowledge structure of another field, the model can be applied to topic-based document retrieval in areas such as healthcare, engineering, or law. Because it combines topic names, descriptions, learning objectives, and application scenarios in a structured way, the system can maintain strong semantic alignment with expert understanding in any subject. This flexibility makes C-ONTO a promising solution for semantic, topic-oriented document retrieval across many different disciplines.

## V. Limitations and Future Work

IT domain and cross-domain evaluations, several limitations remain that warrant further attention. First, the current ontology model is designed exclusively for the Information Technology domain. While preliminary experiments in the Mathematics and Road Traffic domains confirm its adaptability, extending the ontology to new domains still requires manual adjustments to certain model components, which can be time-consuming and dependent on expert knowledge. Second, the topic model is currently optimized for IT-related subjects; broader coverage of other domains would require the incorporation of additional domain-specific ontologies and concept representations. Third, the semantic similarity computation, although effective, has room for optimization to improve both accuracy and computational efficiency, especially in large-scale or real-time applications.

Future work will focus on three main directions. First, we plan to develop automated or semi-automated methods for ontology construction and maintenance to reduce manual effort and enhance scalability. Second, we aim to integrate multi-domain ontology alignment techniques to enable more seamless cross-domain semantic search. Third, we will explore advanced keyphrase extraction and semantic similarity algorithms—potentially leveraging deep learning or graph-based methods—to further improve retrieval performance and better capture nuanced relationships between concepts. In addition, multilingual support, including Vietnamese and other languages, will be incorporated to expand accessibility and usability in diverse educational contexts.

## VI. Conclusion

In this paper, we have improved topic model, concept model, knowledge models (C-ONTO), document representation techniques, and the calculation of semantic similarity between documents and topics, thereby developing a topic-based document retrieval application known as the knowledge retrieval system. In addition, this paper introduces structured keyphrases that are extracted and represented through Sections III.C, ensuring semantic consistency and clarity for document understanding. This system has demonstrated improved retrieval performance over existing approaches, confirming its effectiveness and practical value.

The developed knowledge retrieval system is capable of retrieving information that aligns with the specific topic and meaning of the entered query. Additionally, it can gather knowledge related to the content of the search topic, assisting information technology students in discovering valuable study and research resources based on their topics of interest.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

TDN, VND and HTT conducted the research; TDN designed and performed the experiments derived the models, and analyzed the data; In consultation with TDN, VND and HTT wrote the manuscript; all authors had approved the final version.

## Acknowledgment

The authors wish to thank FPT University for providing the necessary support and resources for this research.

## References

[1] D. N. Nhon, N. D. Hien, and N. H. Long, "Some techniques for intelligent searching on ontology-based knowledge domain in e-learning," in *Proc. of the 12th International Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol. 2, 2020, pp. 313–320.

[2] N. T. Dien, D. V. Nhon, and T. H. Tung, "A model of topic for document retrieval systems in the field of artificial intelligence for information technology students," in *Proc. International Conf. on Intelligent Information Technology (ICIIT)*, 2024, pp. 376–385.

[3] N. T. Dien, D. V. Nhon, and T. H. Tung, "Ontology-based solution for designing knowledge retrieval systems in the field of artificial intelligence," in *Proc. International Conf. on Intelligent Information Technology (ICIIT)*, 2024, pp. 558–563.

[4] D. V. Nhon, M. T. Thanh, and N. H. Long, "A knowledge-based model for designing the knowledge querying system in education," in *Proc. of International Conf. on Research Innovation and Vision for the Future*, Ho Chi Minh, 2022, pp. 524–529.

[5] O. M. L. Denis, "Identification of topics from scientific papers through topic modeling," *Open Journal of Applied Sciences*, pp. 541–548, 2021.

[6] S. Tahereh, D. Mohamad, and S. Tayebeh, "Artificial intelligence for sustainable energy: A contextual topic modeling and content analysis," *Sustainable Computing: Informatics and Systems*, vol. 35, 100699, 2022.

[7] P. Sneh and S. Neetu, "Question tags or text for topic modeling: Which is better," *Procedia Computer Science*, vol. 218, pp. 2172–2180, 2023.

[8] Y. Dejian, F. Anran, and X. Zeshui, "Topic research in fuzzy domain: Based on LDA topic modelling," *Information Sciences*, vol. 648, 119600, 2023.

[9] W. Xiaobao, N. Thong, and L. T. Anh, "A survey on neural topic models: methods, applications, and challenges," *Artificial Intelligence Review*, vol. 57, no. 2, 18, 2024.

[10] H. Nguyen, T. T. N. Le, H. Nguyen *et al.,* "Design a knowledge chatbot system in education based on ontology approach," in *New Trends in Intelligent Software Methodologies, Tools and Techniques*, IOS Press, 2024.

[11] N. D. Hien, D. Truong, S. Vu *et al.,* "Knowledge management for information querying system in education via the combination of rela-ops model and knowledge graph," *Journal of Cases on Information Technology (JCIT)*, vol. 25, no. 1, pp. 1–17, 2023.

[12] A. Velu and M. Thangavelu, "Ontology based ocean knowledge representation for semantic information retrieval," *Journal of Coputers Materials & Continua*, pp. 4707–4724, 2022.
[13] N. T. T Sang, D. H. T. Ho, and N. N. T Anh, "An ontology-based question answering system for university admissions advising," *Journal of Intelligent Automation & Soft Computing*, vol. 36, no. 1, 2022.
[14] N. R. Muhammal, M. W. Iqbal, S. K. Shahzad *et al.,* "Ontological model for cohesive smart health services management," *Journal of Computers Materials & Continua*, vol. 74, no. 2, 2023.
[15] M. Fiaz, A. Ahmad, M. A. Hassan *et al.,* "Ontology-based crime news semantic retrieval system," *Journal of Computers Materials & Continua*. vol. 77, no. 1, 2023.
[16] T. L. H. Nghia and L. T. Tran, *Students' Experiences of Teaching and Learning Reforms in Vietnamese Higher Education*, 1st ed. Oxon, U.K.: Routledge, 2021, pp. 15–36.
[17] P. Patel, M. S. Batcha, and M. Ahmad, "Impact of Web 2.0 technologies on academic libraries: A survey on affiliated colleges of Solapur University," *J. Acad. Librarianship*, vol. 47, no. 1, pp. 1–8, 2022.
[18] Y. Dejian and X. Bo, "Discovering topics and trends in the field of artificial intelligence: Using LDA topic modeling," *Expert Systems with Applications*, vol. 225, 120114, 2023.
[19] S. Vaid, S. Puntoni, and A. R. Khodr, "Artificial intelligence and empirical consumer research: A topic modeling analysis," *Journal of Business Research*, vol. 166, 114110, 2023.
[20] C. M. Lewis and F. Grossetti, "A statistical approach for optimal topic model identification," *Journal of Machine Learning Research*, vol. 23, pp. 1–20, 2022.
[21] P. X. Thien, T. V. Tran, V. T. Nguyen-Le *et al.,* "Build a search engine for the knowledge of the course about introduction to programming based on ontology ReLa-model," in *Proc. International Conf. on Knowledge and Systems Engineering*, Can Tho, 2022.
[22] N. D. Hien, H. Huynh, T. Mai *et al.,* "Design an ontology-based model for intelligent querying system in mathematics education," *Journal of Interdisciplinary Mathematics*, vol. 26, no. 3, pp. 449–473, 2023.
[23] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
[24] D. V. Nhon, *Knowledge Representation and Reasoning*, Hanoi, Vietnam: Science and Technics Publishing House, 2005.
[25] K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th ed. New York, NY, USA: McGraw-Hill, 2011.
[26] K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York, NY, USA: McGraw-Hill, 2019.
[27] J. D. Ullman and J. Widom, *A First Course in Database Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2007.
[28] J. D. Ullman and J. Widom, *A First Course in Database Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
[29] J. D. Ullman and J. Widom, *A First Course in Database Systems*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 1997.
[30] P. T. Mien, K. Nguyen, V. T. Nguyen-Le *et al.,* "An intelligent searching system for academic courses of programming based on ontology query-onto," *International Journal of Intelligent Systems Design and Computing*, vol. 1, no. 1, 10048574, 2022.
[31] N. D. Hien, T. V. Tran, X. T. Pham *et al.,* "Ontology-based integration of knowledge base for building an intelligent searching chatbot," *Sensors and Materials*, vol. 33, no. 9, 2021.
[32] H. T. Thanh, "Discovering community interests approach to topic model with time factor and clustering methods," *Journal of Information Processing Systems*, vol. 17, pp. 163–177, 2021.
[33] R. B. Kaliwal and S. L. Deshpande, "Study on intelligent tutoring system for learner assessment modeling based on Bayesian network," in *Proc. International Joint Conf. on Advances in Computational Intelligence*, Singapore, 2021, pp. 389–397.
[34] N. H. Thinh, N. D. Hien, P. T. Vuong *et al.,* "Legal-Onto: An ontology-based model for representing the knowledge of a legal document," in *Proc. the 17th International Conf. on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, 2022, pp. 426–434.
[35] H. T. T. Thuong, N. P. T. An, and D.V. Nhon, "A method for designing domain specific document retrieval systems using semantic indexing," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 10, 2019.