InFra_OE: An Integrated Framework for Ontology Evaluation

Narayan C. Debnath*, Archana Patel*, Debarshi Mazumder*, Phuc Nguyen Manh*, Ngoc Ha Minh*
Eastern International University, Binh Duong, VIETNAM

Abstract

Nowadays, ontologies are used everywhere to share information semantically, and hence it is crucial to evaluate before using them. Ontology evaluation becomes more important when we have more than one ontology for a domain and we have to choose one ontology from them. The existing ontology evaluation approaches focus on only a few ontology evaluation criteria. Therefore, they cannot determine the overall quality of the ontology. This paper aims to propose an integrated framework for the evaluation of ontologies. The proposed framework uses a knowledge representation approach, criteria-based approach, software engineering approach, and layer-based approach to evaluate the quality of the ontologies based on various criteria.

Keywords: Ontology, ontology evaluation, semantic, knowledge representation, OOPS!

1 Introduction

Knowledge representation and reasoning is a field of 'Artificial Intelligence' that encodes knowledge, beliefs, actions, feelings, goals, desires, preferences, and all other mental states in the machine. Nowadays, ontology is prominently used to represent knowledge. Earlier than ontologies, semantic network and semantic frame emerged, but they were lacking in formal semantics despite the fact that they had semantic in the name [30]. Ontologies offer the richest machine-interpretable (rather than just machine-processable) and explicit semantics and are being used today extensively for semantic interoperability and integration. Ontology is a knowledge representation formalism that reduces the problem of big semantic loss in the process of modelling knowledge [24]. Ontology does not only provide sharable and reusable knowledge, but it also provides a common understanding of the knowledge; as a result, the interoperability and interconnectedness of the model make it priceless for addressing the issues of querying data. Ontology work with concepts and relations that are very close to the working of the human brain. It also provides a way to represent any data format like unstructured, semi-structured, structured, and enables data fetching with semantics. The key requirement of ontology is the development of suitable languages for the representation and extraction of information. Varieties of ontology languages have been developed, and the most operable and standard language is web ontology language (OWL) [18]. Ontology query language plays a very important role in extracting and processing the information. SPARQL is one of the most widely used ontology query languages [25]. By using these semantic technologies (Ontology, SPARQL, OWL), users and systems can interact and share information with each other in an intelligent manner.

Ontology can be developed either from scratch or by modifying an existing ontology [28]. Many well-known ontology repositories are available that contain more than a thousand ontologies about a domain, such as-

- OBO Foundry: It contains biological science-related ontologies
- Bio portal: It is a comprehensive repository of biomedical ontologies
- Agro portal: It is a vocabulary and ontology repository for agronomy related domains
- OLS: It provides single-point access to the latest version of biomedical ontologies

Users use these repositories to choose the ontology for an application. Ontology evaluation is a way that determines the relevance and importance of the ontology in a specified domain [35]. Ontology evaluation is an essential process for the development and maintenance of an ontology. Mainly, we need to evaluate the ontology because of two reasons:

1. The developed ontology needs to be evaluated to check the quality of the ontology. The ontology evaluation is also a phase of the ontology development life cycle.
2. For the reusability purpose, we work with the existing ontologies. However, when more than one ontology is available for a domain, then it is hard to choose one ontology among them. In such a case, we need to evaluate the ontology to find the best-suited ontology according to the need.

The work of ontology evaluation is focused on five questions [28]. What should be evaluated? Why should it be evaluated? When should it be evaluated? What should be the base of the evaluation? What are the possible criteria to evaluate the design and implementation of an ontology? The various tools

* Department of Software Engineering, Email: narayan.debnath@eiu.edu.vn, archana.patel@eiu.edu.vn, debarshi.mazumder@eiu.edu.vn, phuc.nguyenmanh@eiu.edu.vn, ngoc.ha@eiu.edu.vn.

are proposed in the literature to evaluate the ontology; some are available on the web. These tools are grouped into two categories: domain-dependent ontology evaluation tools and domain-independent ontology evaluation tools. These tools evaluate the design of ontologies based on various criteria like, accuracy, adaptability, cognitive adequacy, completeness, clarity, expressiveness, conciseness, consistency, and grounding. In contrast, computational efficiency, precision, recall, and practical usefulness are used to evaluate the implementation of an ontology. Many studies are available for the evaluation of the ontology; however, they have some limitations:

1. The existing studies usually show only a set of criteria and questions, and they do not show the guidelines to evaluate the ontology.
2. The effort to evaluate the ontology is very high as no tool available that concisely evaluates the ontology.
3. The evaluation heavily depends on the evaluator's expertise to understand the evaluation criteria and questions.
4. The evaluation is still very subjective.

The proposed work presents an integrated framework for ontology evaluation called InFra_OE. The framework takes into account four fundamental principles: (a) It supports five roles of knowledge representation proposed by Davis [10] (b) It is based on the evaluation of the effectiveness of the coding standard (a software engineering approach) (c) It integrates the OOPS! tool for anomalies detection (d) It uses a layer-based approach. The main contributions of this work are: (a) to propose an integrated framework for ontology evaluation that integrate the features of four approaches, namely five roles of knowledge representation, approach for evaluation of software development methodology, ontology pitfall scanner (OOPS!) tool, and different layers of ontologies. (b) to propose an algorithm for the step-by-step execution of the proposed framework. The rest of the paper is organized as follows: Section 2 shows the definitions of important concepts that are required to understand the proposed framework. Section 3 discusses the available literature of ontology evaluation. Section 4 describes the proposed framework and algorithm. The last section concludes the paper.

2 Background

Ontology Scope: The first step of ontology development is to determine the scope of an ontology. In ontological engineering, ontology scope shows the specification and design aspects for the representation of the knowledge [12]. The scope of an ontology can be classified into three aspects: Domain scope (determines that the scope of the ontology is relevant to the task for which ontology is designed), conceptual scope (determines that ontology will represent the hierarchical and taxonomical concepts), technical scope (shows that the specifications and requirements for ontologies are integrated smoothly and correctly in terms of ontology integration and application in practice).

Ontology Layers: The structure of the ontology is complex, and it is hard to evaluate the whole ontology once. So, it is required to evaluate the ontology according to its different layers. The layer-based ontology evaluation approach allows users to use different techniques for different layers [19]. Mainly, ontology has four different layers: Lexicon/vocabulary layer (This layer evaluates the ontology with respect to knowledge representation and conceptualization of ontologies like naming criteria for concepts, instances, and facts.), structure/ architecture layer (this layer evaluates the hierarchical and taxonomic elements of ontology like hierarchical relations among concepts). Representation/semantic layer (this layer evaluates the ontology with respect to the semantic elements), context/application layer (This layer evaluates the ontology according to the context and application where the ontology would be used. Typically, evaluation looks at how the outcomes of the application are affected by the use of ontology).

Type of Ontology: Mainly, four types of ontologies are available, namely upper ontology, domain ontology, task ontology, and application ontology. The *upper ontology* occupies general concepts or terms like matter, time, and space. The aim of the upper ontologies is to support broad semantic interoperability among domain ontologies by providing a common platform for the formulation of the definitions. The *domain ontology* is developed to capture and relate the content of specific domains (e.g., medical, electronic, digital domain) or part of the world. Domain ontologies use the services of upper ontologies. The *task ontology* contains fundamental concepts according to a general activity or task. It is a specification of element relationships of tasks to explain how tasks can exist and be used in a specific environment. Task ontology serves as a foundation for using tasks in certain fields, like in the field of management, and it defines what element it has and what type of relationships can be established with other tasks. The *application ontology* is a specialized ontology focused on a specific application. It has a very narrow context and limited reusability because it depends on the particular scope and requirements of a specific application. Application ontologies are typically developed ad hoc by the application designers.

Ontology Development Methodology: In the literature, various authors have developed ontologies for the semantical analysis of data [9]. However, the major problem for ontology developers is to choose the right methodology that builds correct, complete, and concise ontology as per requirement. Ontology development methodology describes the step-by-step process for ontology development. The most famous used methodologies are TOVE, Enterprise Model Approach, METHONTOLOGY, and KBSI IDEF5. These methodologies have various steps for ontology development, and some steps are common among them. Figure 1 shows the relationship among these methodologies via arrows (double-headed arrows) [11].

The most important step of ontology development is to identify the purpose and fix the boundary/scope of the ontology. This can be achieved by writing competency questions and the ontology and impose constraints on the classes and their

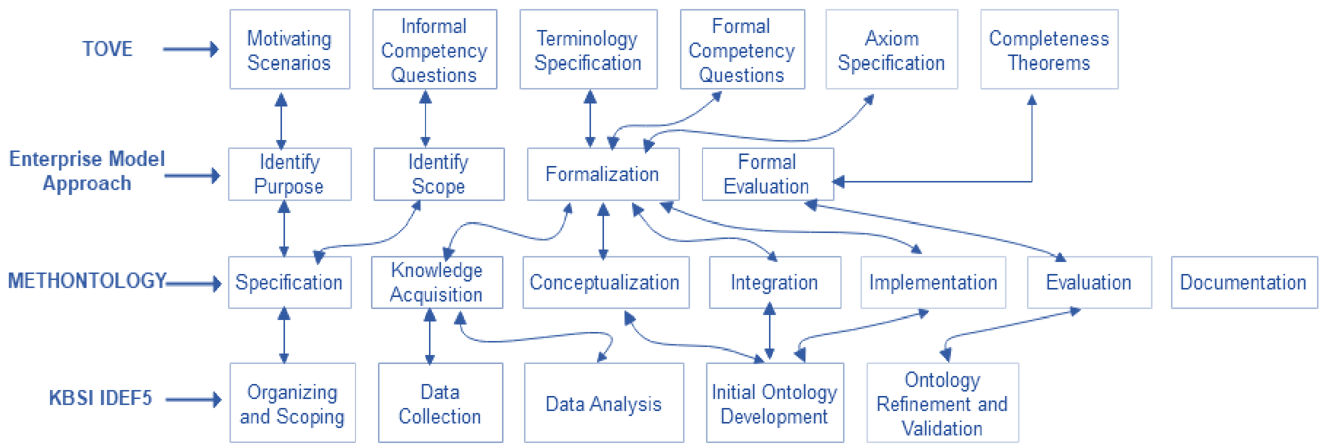

Figure 1: Most popular and extensively used ontology development methodologies

properties as required. Ontology evaluation is the vital step of ontology development methodologies. It shows the quality of the developed ontology based on various crietria like completeness (ontology must contain all the required information as per domain need), and accuracy (ontology must be free from anomalies and is able to infer the correct answer). The last step of ontology development methodology is to document the ontologies that can become the base of other activities. Apart from these methodologies, three ontology development methodologies, namely Neon, YAMO, and SAMOD are also available in the literature [11].

Ontology Evaluation Quality Criteria: Ontology evaluation checks the quality and quantity of an ontology based on various criteria [23]. The essential criteria are:

- Accuracy: It is determined by the definitions, descriptions of entities like classes, properties, and individuals. This criterion states that the ontology is correct.
- Clarity: Clarity evaluates how well an ontology communicates the intended meaning of specified concepts. Clarity is determined by a number of factors. First and foremost, definitions must be objective and independent of social or computational circumstances. Social events or computational needs may motivate the definition of a notion. Second, ontologies should utilize definitions for classes rather than descriptions. Third, entities should be adequately documented and completely labeled in all essential languages, among other things. Most of these criteria are best examined using criterion-based techniques such as OntoClean.
- Completeness: It states that the ontology covers complete information about a specified domain. Sometimes some important information about the entities is missing in the ontology, which leads to an ambiguity problem and hampers the results of reasoning. Precision and recall are measured to check the incompleteness problem in the ontology. Precision shows the ability of an ontology to present only relevant items, whereas recall shows the ability of an ontology to present all relevant items. The completeness of entities depends on the level of granularity agreed to in the whole ontology.
- Adaptability: It measures the adaptability of an ontology and shows how far the ontology anticipates its uses. An ontology should offer the conceptual foundation for a range of anticipated tasks.
- Consistency/Coherence: Consistency highlights the fact that the ontology does not include or allow for any inconsistencies. An ontology should be coherent, which means that it should allow inferences that are compatible with the definitions. The defining axioms should be logically consistent. Coherence should also apply to imprecisely defined ideas, such as those given in natural language documentation. An example of a contradiction is the element Lion's description, "A lion is a giant cat that lives in pride," while possessing a logical axiom ClassAssertion(ex: Type of chocolate ex: Lion). Consistency can be evaluated using criteria-based techniques that focus on axioms. It can also be recognized depending on the ontology's performance in a specific job.
- Reusability: The feature indicates that good ontologies immediately lead to increased data reuse and improved collaboration across application and domain boundaries. The reusability of an ontology may be determined by analysing the various metadata that is available for it or by investigating the interactions within its specific community.
- Computational efficiency: It shows the flexibility of an ontology with the tools, specifically focusing on the speed of the reasoner that infers the information from the ontology.
- Extendibility: Extendibility defines the high-level needs of an ontology design that must be specialized enough for usability, extendable for upgrades, abstract enough for

reusability, and compatible with current applications even after future modifications. An ontology is not anticipated to contain all of the potential information, attributes, and constraints of the domain it represents. As a result, several versions addressing the same body of information are feasible as far as the ontology design meets specified standards. In general, a good ontology holds adequate knowledge to develop expertise in solving significant issues.

- Minimal encoding bias: This criterion represents that encoding bias should be minimized because knowledge-sharing agents may be developed with a variety of libraries and representation styles. An encoding bias occurs when representation choices are selected purely for the sake of notation or implementation. This quality highlights the platform-independent representation of knowledge.
- Conciseness: It examines the usefulness and preciseness of the stored information in an ontology. An ontology is called precise if it does not store any useless or unnecessary definitions; if any explicit redundancies of definitions of entities and between definitions of entities do not exist.
- Minimal ontological commitment: This criterion evaluates that the ontology should define just those terms that are necessary for communicating information compatible with that theory. Nonrelevant information should not be part of the top-level ontology. Top-level ontology can be further specialized by the respective community.
- Expandability: This shows the effort that needs to be put to add new definitions and more knowledge to an entity of an ontology without altering the set of well-defined properties already guaranteed.
- Sensitiveness: This examines how small changes in a definition of an entity alter the set of well-defined properties already guaranteed.
- Organizational fitness: It investigates how ontology is easily deployed within the organization.
- Agreement: Measured through the proportion of agreement that experts have with respect to ontology elements, that is, by measuring the consensus of a group of experts.

Table 1: Most commonly identified ontology evaluation criteria

	Thomas Gruber	**Gómez Pérez**	**Denny Vrandecic**	**Gangemi**
Clarity	√		√	
Consistency		√	√	
Coherence	√		√	
Extendibility	√			
Completeness		√	√	
Minimal encoding bias	√			
Conciseness		√	√	
Minimal ontological commitment	√			
Expandability		√		
Sensitiveness		√		
Accuracy				
Adaptability			√	
Computational efficiency			√	
Organizational fitness			√	
Agreement				√
User Satisfaction				√
Task				√
Topic				√
Modularity				√

- User Satisfaction: This can be evaluated by dedicated research or reliability assessment
- Task: This deals with measuring an ontology according to its fitness to some goals, postconditions, preconditions, options, constraints, and others.
- Topic: This measures the ontology according to its fitness for a repository of existing knowledge.
- Modularity: Modularity measures fitness to a repository of existing reusable components.

3 Related Work

There are a lot of studies available for ontology evaluation. These studies are grouped into two categories: domain-dependent ontology evaluation studies and domain-independent ontology evaluation studies. The domain-dependent ontology evaluation studies show the evaluation of domain dependant ontologies and discuss the available tool for the same. The domain-independent ontology evaluation studies show the evaluation of domain-independent ontologies and discuss the available tool for the same [25]. Figure 2 shows the domain dependent and independent tools.

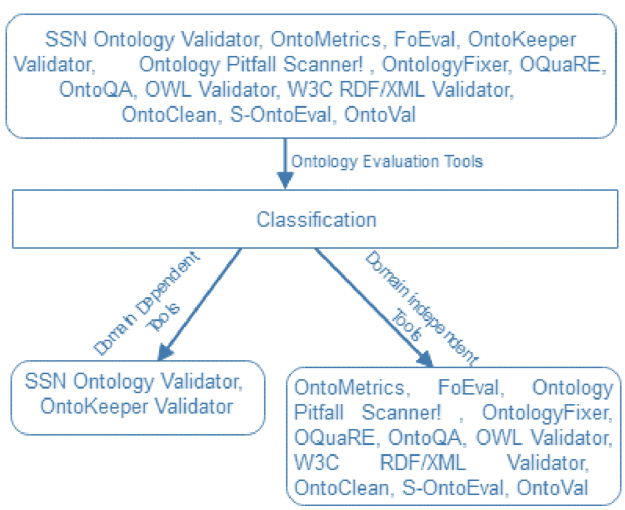

Figure 2: Ontology evaluation tools

SSN Ontology Validator [21] is used to validate the new ontology of the IoT domain. The validator gives a validation report that reports inconsistencies in an ontology. The SSN validator receives the ontology and compares it with the SSN ontology and other ontologies associated with it. *OntoKeeper Validator* [1] evaluates biodiversity ontologies with different levels of granularity. It analyzed the ontology files using semiotic metrics (semiotics has three branches, namely, syntactic, pragmatic, and semantic) and detects the quality score of the biodiversity ontology based on various parameters, namely richness, interpretability, comprehensiveness, and accuracy. *OWL Validator* [17] aims to ensure that all concepts and properties in the ontology are specified as per the W3C standard. OWL validator shows an error message with the detailed report when an input ontology does not support the selected profile. *W3C RDF/XML validator* [33] validates the RDF document by tracking the RDF issues and shows a warning message when an error occurs. W3C RDF validator shows the number of tuples (subject-object-predicate) that are encoded in the ontology as well as its graphical representation. This validator aims to ensure that the document is syntactically valid.

Ontology Pitfall Scanner! (OOPS!) [34] tool shows the pitfalls or anomalies of an ontology. OOPS! shows the 41 types of pitfalls and groups them into three categories, namely, minor pitfalls (these pitfalls are not serious and no need to remove), important pitfalls (not very serious pitfalls but need to remove), critical pitfalls (these pitfalls hamper the quality of an ontology and need to remove them before using the ontology). *OntoMetrics tool* [22] calculates the statistical information about an ontology. It has five types of metrics, namely, Base metric, Schema metrics, Knowledge base metric, Class metric, and Graph metrics. *OntologyFixer tool* [31] allows the detection and correction of errors. It uses various metrics to measure the different aspects of the quality of an ontology. These metrics are ANOnto (shows annotation richness), CBOnto (shows the coupling between objects), CROnto (shows class richness), INROnto (shows the number of relations per class), LCOMOnto (shows lack of cohesion in methods), NOMOnto (shows the number of properties per class), RCOnto (shows the distribution of instance across class), RFCOnto (shows response measure for a class), and RROnto (shows relationship richness). To detect the pitfalls of an ontology, OntologyFixer integrated OOPS! tool. *OntoVal tool* [3] evaluates the OWL ontologies by nontechnical domain specialists and allows users to provide textual feedback for each evaluated term and evaluate the correctness of the developed ontology via an integrated engine. OntoVal starts the ontology evaluation by collecting information about the participant (name; age; domain experience level, ranging from 0 to 10; ontology experience level, ranging from 0 to 10). The ontology evaluation process of OntoVal is divided into three stages, namely, class evaluation, property evaluation, and overall evaluation. The *Semiotic-based Ontology Evaluation (S-OntoEval) tool* [13] aims to evaluate the quality of the ontology by taking three metrics, namely, syntactic, semantic, and pragmatic, that are considered different aspects of the ontology quality. In essence, there are three types of evaluation levels, namely, structural level, functional level, and usability related level.

OQuaRE framework [14] evaluates the quality of an ontology on the basis of both quality models and quality metrics. OQuaRE adapts and reuses five characteristics from the SQuaRE, namely, Structural (it specifies the formal and semantic important properties of an ontology), functional adequacy (it includes the degree of accomplishment of functional requirements), Reliability (it checks the level of performance under stated conditions), Operability (it shows the effort needed for building an ontology and individual assessment), Maintainability (it shows the ability of ontologies to be modified by changes in environments). The *Ontology Quality Analysis (OntoQA) approach* [32] evaluates the design and representation of knowledge of ontology, and the placement of instances within the ontology and its effective usage. The OntoQA categorizes the quality of the ontology into three groups, namely, Schema metrics, Knowledgebase metrics, and Class metrics.

OntoClean Methodology [16] validates the adequacy of the ontology hierarchy based on general ontological notions, namely, essence, unity, and identity. These notions are used to characterize relevant aspects of the intended meaning of classes, properties, and relations. It checks the correctness of the ontology hierarchy via the principles of metaproperties, namely, rigidity (this property is essential to all their instances), unity (refers to being able to recognize all parts that form an individual entity), identity (refers that all instances are identified in the same way), and dependence (captures a meta-property of certain relational roles).

Full Ontology Evaluation (FoEval) model [6] is a ranking and selection tool that has three features, namely: it allows the user to select a set of metrics that help in the evaluation process, this tool enables the user to evaluate the locally stored and searched ontologies from different search engines or repositories, it captures the structural and semantic information of a domain. It includes a rich set of metrics, namely, coverage, richness, comprehensiveness, and computational efficiency.

4 Proposed Framework

The proposed framework consists of three phases, namely Input phase, Processing Phase, and Output phase, whereas the processing phase contains four modules (a) Evaluation based on Role of Knowledge Representation, (b) OOPS! (c) Ontology Code Effectiveness, and (d) Layer Based Evaluation. The human interaction is required in two modules, namely evaluation based on the role of knowledge representation and layer-based evaluation. All the modules of the processing phase are executed parallelly, and the final result is calculated by taking the mean value of obtained results from all four modules. Figure 3 shows the proposed integrated framework for ontology evaluation.

I. Input Phase: It consists of knowledge base and expert person. The knowledge base consists of domain ontologies, core ontologies, and upper ontologies. Any type of these ontologies can be input into the processing phase. The expert assigns the appropriate value of parameters of these ontologies as per need. The expert input is required in two modules of the processing phase. The proposed framework is called semiautomatic because of the involvement of the experts.

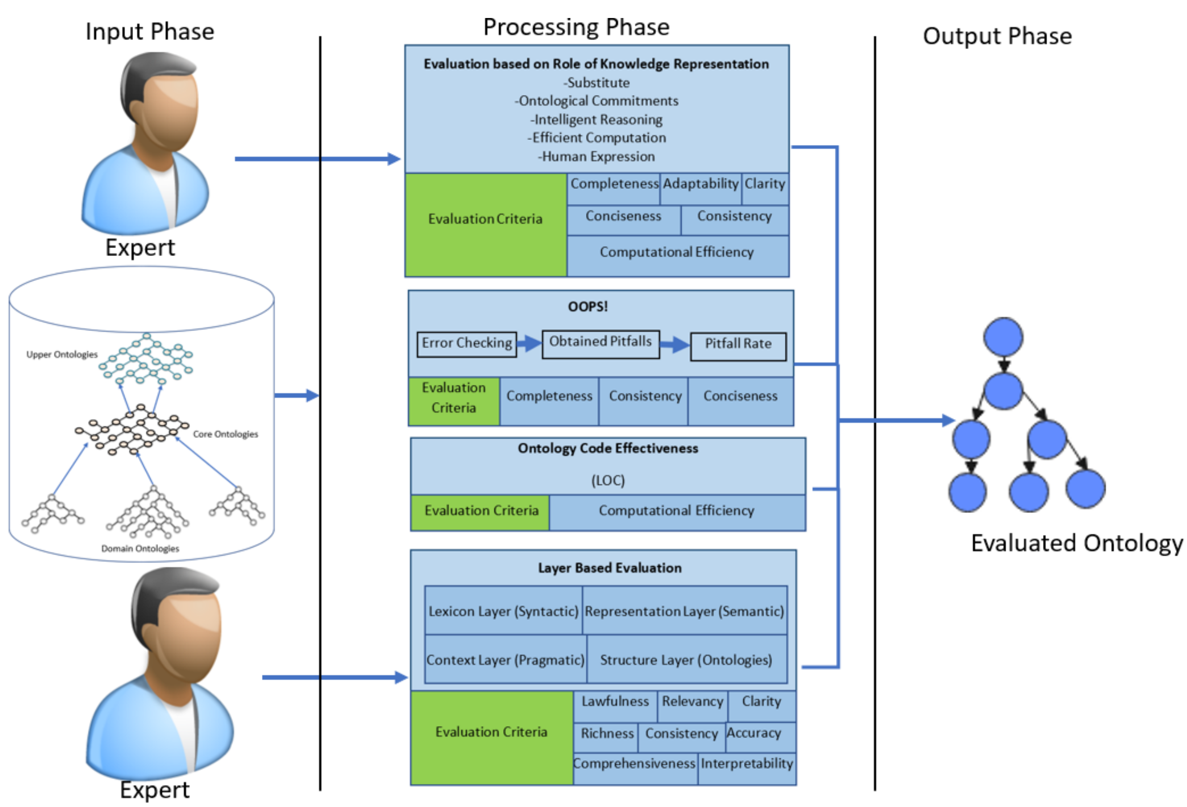

Figure 3: InFra_OE: An integrated framework for ontology evaluation

II. Processing Phase: This phase takes ontologies from the knowledge base and then executes all the four modules of the processing phase over it. These modules evaluate the ontologies from different aspects and generate different results for the ontology evaluation.

a) Evaluation based on Role of Knowledge Representation: Davis [10] has presented five roles for the discussion of knowledge representation. These roles clearly describe What is Knowledge Representation? We use these roles for the evaluation of ontology [4]:

- Substitute: This role shows how ontology approaches the real world. It focuses on which concepts should be represented and which should be omitted. For example, vehicle, rim, tires, and handlebars are the concepts of the bicycle. The concepts which are close to the real world need to be represented to fulfill this role of knowledge representation.
- Ontological Commitments: This role shows how the ontology is closer to the real world. This role will be fulfilled better if the representation is more consistent. For example, the more consistent representation is *a bicycle is a vehicle* and *a vehicle is an object* as compared to *bicycle is an object*.
- Intelligent Reasoning: This role represents how the ontology correctly infers the real world. This role will be fulfilled by the concise representation of the relations and attributes. For example, the vehicle is a bicycle because it has two tires, thin wheels, and handlebars.
- Efficient Computation: This role represents how the machine can think about a domain and extract the information within minimum time (i.e., computational time). Suppose, all the websites have a bicycle domain ontology, and the user is searching for a bicycle which has a red color, two tires of size x, and manufactured by company y, then the machine must be able to find this bicycle in a few seconds.
- Human Expression: This role represents how easy it is to understand the modelling. This role will be achieved by a clear declaration/representation of the concepts and relations. For example, the concept of bicycle is represented by bicycle, not any other words like bi or bic.

Each of these roles is fulfilled by some questions (shown in Table 2) and shows different criteria of ontology evaluation.

Table 2: Connection between role, questions, ontology development phases, and ontology evaluation criteria

Role	Questions	Ontology Development Phases	Ontology Evaluation Criteria
Substitute	Q1. Address the document that specify the scope and objective of the ontology.	Scope Determination, Concept Extraction, Encoding	Completeness, Adaptability
	Q2. Address the coherence between the document of Q1 and modelling of the ontology.		
	Q3. Address the reusability of the concepts that model the real world.		
Ontological Commitments	Q4. Address about the representation scheme for a specific domain	Concept Extraction and Encoding	Conciseness, Consistency
	Q5. Address about the representation scheme for an abstract domain		
	Q6. Address the coherence with the real world.		
Intelligent Reasoning	Q7. Address the reasoning power of ontology	Evaluation	Consistency
Efficient Computation	Q8. Address computational performance in term of successfully executed queries.	Evaluation	Computational efficiency
	Q9. Address computational performance in term of reasoner speed.		
Human Expression	Q10. Address the easy and precise understanding of the modelling.	Encoding	Clarity

The evaluation of an ontology with respect to five roles are calculated by the below mentioned equation [9].

$$\frac{\exp\{-0.44 + 0.03(Cov_S \times Sb)E + 0.02(Cov_C \times Co)E + 0.01(Cov_R \times Re)E + 0.02(Cov_{Cp} \times Cp)E - 0.66LExp_E - 25\,(0.1 \times Nl)E\}}{1 + \exp\{-0.44 + 0.03(Cov_S \times Sb)E + 0.02(Cov_C \times Co)E + 0.01(Cov_R \times Re)E + 0.02(Cov_{Cp} \times Cp)E - 0.66LExp_E - 25\,(0.1 \times Nl)E\}} \quad (1)$$

The parameter E indicates the evaluators/experts that assign the value of all the parameters and evaluates the quality of the ontology based on the five roles of knowledge representation.

- Cov_S shows the mean value of role 1 i.e. Substitute by rating all the three questions corresponding to role 1.
- Cov_C shows the mean value of role 2 i.e. Ontological Commitments by rating all the three questions corresponding to role 2.
- Cov_R shows the mean value of role 3 i.e. Intelligent Reasoning by rating the question corresponding to role 3.
- Cov_P shows the mean value of role 4 i.e. Efficient Computation by rating all the two questions corresponding to role 4.
- $LExp_i$ shows the value of the evaluator/expert experience, its value will be 1 if the expert has good knowledge about the ontologies otherwise 0.
- Nl value will be 1 if the expert will not be able to answer all the questions of the goal.
- The value of Sb = 1, Co = 1, Re = 1, Cp = 1, if total quality will be calculated otherwise it will be 0 according to the absence of any role which indicates the partial ontology evaluation.

(b) *OOPS!*: It is a web-based tool that shows the pitfalls or anomalies of an ontology. OOPS! shows the 41 types of different pitfalls starting from P01 to P41. Basically, OOPS! groups the pitfalls under three categories, namely minor pitfalls (these pitfalls are not serious and no need to remove), important pitfalls (not very serious pitfalls but need to remove), critical pitfalls (these pitfalls hamper the quality of an ontology and need to remove them before using ontology). The pitfall describes the number of features that could create problems during reasoning. We have calculated the pitfall rate by using the following equation [27]

$$A = \frac{\sum_{i=1}^{n} P_i}{N} \quad (2)$$

P_i represents the total number of pitfall cases according to the pitfall type P_i, and N is the total number of tuples (ontology size). The high value of the pitfall rate implies a more significant number of anomalies and vice-versa. To calculate the quality of the ontology, we subtract the obtained pitfall rate (A) by 1. It focuses on the completeness, consistency, and conciseness criteria of ontology evaluation.

(c) *Ontology Code Effectiveness*: Ontology code effectiveness evaluates the size and complexity of the ontology's code. Basically, it shows the conciseness of the developed ontology by identifying the similarity in an ontology code and identification of the duplicate in a code (known as clones). In software engineering, the quality of the code is determined by various techniques like Line of code (LOC), function point, etc. However, we evaluate the effectiveness of the ontology coding standard by the Goal-Question-Metric (GQM) approach proposed by Basili et al. [5]. Figure 4 shows the GQM approach. We calculate the ontology size by LOC, which shows the number of tuples stored in the ontology. The size of the ontology does not depend on the annotation properties as they serve metadata information about the entities. We have ignored nine annotation properties, namely- backwardCompatibleWith, comments, deprecated, incompatibleWith, isDefinedBy, label, priorVersion, seeAlso, and versionInfo. These properties are supported by the protégé tool.

(d) *Layer Based Evaluation*: The structure of the ontology is complex, and it is hard to evaluate the whole ontology at once. Hence, it is good practice to evaluate the ontology based on the layer approach. Mainly, ontology has four different layers [7]:

- Lexicon/Vocabulary layer: This layer evaluates the ontology with respect to knowledge representation and conceptualization of ontologies like naming criteria for concepts, instances, and facts. Ex- Bicycle or bic.
- Structure/ Architecture layer: This layer evaluates the hierarchical and taxonomic elements of ontology, like hierarchical relations among concepts. Ex- human must be a superclass of Male and Female.
- Representation/ Semantic layer: This layer evaluates the ontology with respect to the semantic elements. Ex- Mouse should be able to explain itself either mouse is a device or mouse is an animal.
- Context/Application layer: This layer evaluates the ontology according to the context and application where the ontology would be used. Typically, evaluation looks at how the outcomes of the application are affected by the use of ontology.

The advantage of the layer-based ontology evaluation approach is that it allows users to use different techniques at different ontology layers. We use the syntactic approach at Lexicon/Vocabulary layer; wordnet (lexical database of semantic relations between words) at the Structure/Architecture Layer; Semantic approach at the Representation/semantic layer; and Pragmatic approach at the Context/Application layer.

Syntactic Approach: It measures the quality of the ontology based on syntax and the way it is written. It focuses on syntactic correctness (checks how ontology language's rules are compiled), Richness (shows the number of ontological entities) [8]. The overall syntactic quality is calculated by below mentioned equation-

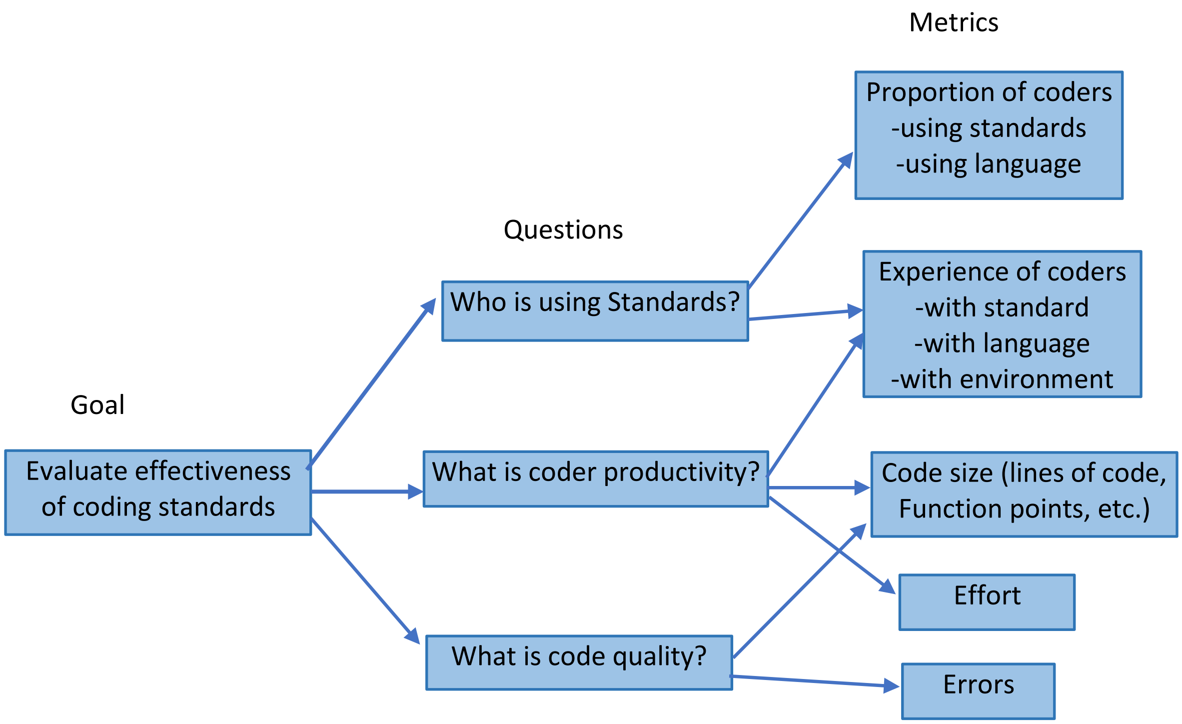

Figure 4: Goal-questions-metrics approach

$$Le = w_1 \times La + w_2 \times Ri \quad (3)$$

La= Number of violated axioms over the total number of axioms

Ri= Used number of ontological features over the total number of ontological features

- Semantic Approach: It deals with the meaning of the entity that is derived from the syntax via logic. It focuses on Interpretability (reveals that every ontological term shows correct meaning in everyday usage), Consistency (shows the ontological terms are uniformly defined and lack duplicate terms), Clarity (shows that all terms are unambiguously represented) [3]. The overall Semantic quality is calculated by below mentioned equation-

$$Se = w_3 \times In + w_4 \times CO + w_5 \times Cl \quad =(4)$$

In: Terms which has at least one-word sense over the total number of terms

CO: Number of duplicated terms divided by the total number of terms

Cl: the average word sense per term over the number of terms

- Pragmatic Approach: It deals with inferential meaning, not merely logical inference, but the subtler aspects of communication expressed through indirection [20]. It focuses on comprehensiveness (shows the coverage of domain), accuracy (shows the truthfulness of the statement), and relevancy (shows the relevancy of ontology in particular applications).

$$Con = w_6 \times Com + w_7 \times Ac + w_8 \times Re \quad (5)$$

Com= percentage number of instances, classes, and properties of the ontology to a group of ontologies.

AC= truth statement over the total statement.

Re= varies and depends on the possible use-case of the ontology.

III. Output Phase: This phase shows the numeric value for the evaluated results by taking the mean of all the four modules of process6ing, namely Evaluation based on Role of Knowledge Representation, OOPS!, Ontology Code Effectiveness, and Layer based approach. The obtained value lies between [0,1]. The highest value shows that ontology has good quality.

InFra_OE has mainly four functions KR(), OOPS(), OCE(), and LBE (). The function KR() corresponds to the five roles of knowledge representation (module 1 of phase 2) and returns an integer value calculated by equation 1. The function OOPS() is used for OOPS! tools [29] and shows the working of module 2 of phase 2. It returns an integer value by subtracting 1 from the pitfall rate (A). The function OCE() shows the ontology code effectiveness by calculating the size of the ontology (excluding annotation properties) [15] and shows the working of module 3

Algorithm for InFra OE

Input:

Expert E, KB, number of ontologies L

Function- KR(), OOPS(), OCE(), LBE ()

Integer- OE_1, OE_2, OE_3, OE_4, OE, KR, OOPS, LE, OE, KRR, FKRR, Ano, NoAno, Cov_S, Cov_C, Cov_R, Cov_{CP}, TPi, TP, N, Le, Se, Co, St, TLE, La, Ri, In, CO, Cl, Com, Ac, Re, Con, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8

Boolean- $LExp_i$, Nl, Sb, Co, Re, Cp

Integer- TP=0; *Range*- {0.25, 0.50, 0.75, 1}

Processing:

$L \leftarrow Pick\ ontology\ from\ the\ KB$ // L is the total number of ontologies stored in the KB

For ($Li = 1;\ Li \le L; Li++$) {

{

$OE_1 \leftarrow KR$ ()

$OE_2 \leftarrow OOPS$

$OE_3 \leftarrow OCE$

$OE_4 \leftarrow LBE$

}

Int KR () // function KR() corresponding to 5 role of KR

{

For ($E = 1;\ E \le Ex;\ E++$) // Ex is the total number of experts

{

$Cov_S \leftarrow mean\ of\ rating\ given\ to\ questions\ Q1, Q2, and\ Q3$

$Cov_C \leftarrow mean\ of\ rating\ given\ to\ questions\ Q4, Q5, and\ Q6$

$Cov_R \leftarrow mean\ of\ rating\ given\ to\ questions\ Q7$

$Cov_{Cp} \leftarrow mean\ of\ rating\ given\ to\ questions\ Q8, and\ Q9$

$$KRR \leftarrow \frac{\exp\{-0.44+0.03(Cov_S \times Sb)E+0.02(Cov_C \times Co)E+0.01(Cov_R \times Re)E+0.02(Cov_{Cp} \times Cp)E-0.66LExp_E-25\ (0.1 \times Nl)E\}}{1+\exp\{-0.44+0.03(Cov_S \times Sb)E+0.02(Cov_C \times Co)E+0.01(Cov_R \times Re)E+0.02(Cov_{Cp} \times Cp)E-0.66LExp_E-25\ (0.1 \times Nl)E\}}$$

} FKRR ← FKRR + KRR

return FKRR }

Int OOPS () // function OOPS() corresponding to OOPS! tool

{

run OOPs! tool on the selected ontology Li

Pitfall ← calculate total number of cases of obtained pitfalls

{

For ($Pi = 1;\ Pi \le Pitfall;\ Pi++$)

{

TPi ← cases of Pi

TP ← TP + TPi

}

$Ano \leftarrow \frac{TP}{N}$ // N is the total number of tuples

NoAno ← 1- Ano

} return NoAno }

Int OCE() ← calculate size of ontology, excluding annotation properties // function OCE() corresponding ontology's size

Int LBE () {

For ($E = 1;\ E \le Ex;\ E++$) // Ex is the total number of experts

{

$Le \leftarrow w_1 \times La + w_2 \times Ri$

$Se \leftarrow w_3 \times In + w_4 \times CO + w_5 \times Cl$

$Con \leftarrow w_6 \times Com + w_7 \times Ac + w_8 \times Re$

$St \leftarrow check\ semantic\ relations\ between\ words$

$$LE \leftarrow \frac{Le + Se + Co + St}{4}$$

} return LBE} }

Output:
$OE \leftarrow \frac{OE_1 + OE_2 + OE_3 + OE_4}{4}$ // OE is the value of evaluation results that lies between $0 \leq OE \leq 1$

of phase 2. This function also has an integer value. The last function LBE() evaluates the ontology with respect to the four layers of ontology and uses different formulas at different layers like the syntactic approach is used at the lexicon layer, wordnet is utilized at the structure layer, etc. This function shows the working of module 3 of phase 2. The expert involvement occurs in two modules (module 1: evaluation based on 5 roles of knowledge representation; module 4: layer-based ontology evaluation), and they assign the value to the various parameters ranging {0.25, 0.50, 0.75, 1}. This range is defined by Bandeira et al. [4] after experimentation (we have mapped the range provided by Bandeira et al. [4] on a 0 to 1 scale for consistency purpose). The final value of evaluation is determined by taking the mean value of the four modules.

Evaluation of InFra_OE Framework: For the evaluation of InFra_OE framework, we have taken four Covid-19 Ontologies namely CODO, COKPME, COVID19, and LONGCOVID [26]. The details of these ontologies are mentioned below-

1. An Ontology for Collection and Analysis of COviD-19 Data (CODO): The CODO ontology is a data model that publishes Covid-19 data on the web as a knowledge graph.
2. The CODO aims to show the patient data and cases of Covid-19. The latest version of COVID19 was released in Sept 2020.
3. COKPME: This ontology is used to analyse the precautionary measures that help in controlling the spread of Covid-19. COKPME ontology is able to handle the various competence questions. The latest version of COKPME was released in Sept 2021.
4. COVID-19 Surveillance Ontology (COVID19): This ontology supports surveillance activities and is designed as an application ontology for the Covid-19 pandemic. The developed COVID-19 surveillance ontology ensures transparency and consistency. The latest version of COVID19 was released in May 2020.
5. Long Covid Phenotype Ontology (LONGCOVID): It is RCGP RSC Long Covid Phenotype ontology. The latest version of LONGCOVID was released in Oct 2021.

We have examined the pitfalls of these ontologies by OOPS! tool and the obtained results are mentioned in Table 3. The numbers (e.g. 1, 2, 4,..) that are contained in Table 3 denotes the total number of cases in accordance with the given pitfalls, like CODO ontology contain 1 case of pitfall P04. The sign × indicates no pitfall case is available in the respective ontology.

Table 3: Obtained pitfalls of covid-19 ontologies

Ontologies → Pitfalls ↓	CODO	COKPME	COVID19	LONG COVID
Minor Pitfalls				
P04	1	2	4	1
P07	×	×	×	×
P08	58	14	×	12
P13	38	15	×	×
P20	3	×	×	×
P21	×	1	×	×
P22	1	1	×	×
P32		×	×	×
Important Pitfalls				
P10	1	1	1	×
P11	58	14	×	×
P24	4	×	×	×
P25	4	×	×	×
P30	2	×	×	×
P34	7	×	×	×
P38	1	×	1	1
P41	×	×	1	1
Critical Pitfalls				
P19	×	3	×	×

We have calculated the pitfall rate by using equation (2). The value for the parameters of five role of knowledge representation is mentioned Table 4 and these values are used according to equation (1) to calculate the quality of the ontology based on the five role of knowledge representation. The Figure 5 shows the comparison among the three ontology evaluation methods namely OOPS! tool, Five role of knowledge representation (KR), and the proposed InFra_Ont framework. The Figure 5 depicts that-

- OOPS! tool- The LONGCOVID ontology has 0.454 pitfall rate, which is the highest as compared to other Covid-19 ontologies and COVID19 ontology has 0.042 pitfall rate, which is the lowest as compared to other Covid-19 ontologies.
- Five Role of Knowledge Representation (KR)- The COKPME ontology has 0.3947 pitfall rate, which is the highest as compared to other Covid-19 ontologies and CODO ontology has 0.2950 pitfall rate, which is the lowest as compared to other Covid-19 ontologies.
- InFra_Ont Framework- The LONGCOVID ontology has 0.42405 pitfall rate, which is the highest as compared to other Covid-19 ontologies and CODO ontology has 0.1905 pitfall rate, which is the lowest as compared to other Covid-19 ontologies.

Table 4: Value for parameters of five role of knowledge representation

Ontologies → Parameters ↓	CODO	COKPME	COVID19	LONGCOVID
COV_S	0.75	0.75	0.833	0.833
COV_C	0.833	0.75	0.5	0.5
COV_R	0.75	0.5	0.75	0.75
COV_{CP}	0.875	0.75	0.625	0.625
$LExp_E$	0.75	0.75	0.75	0.75

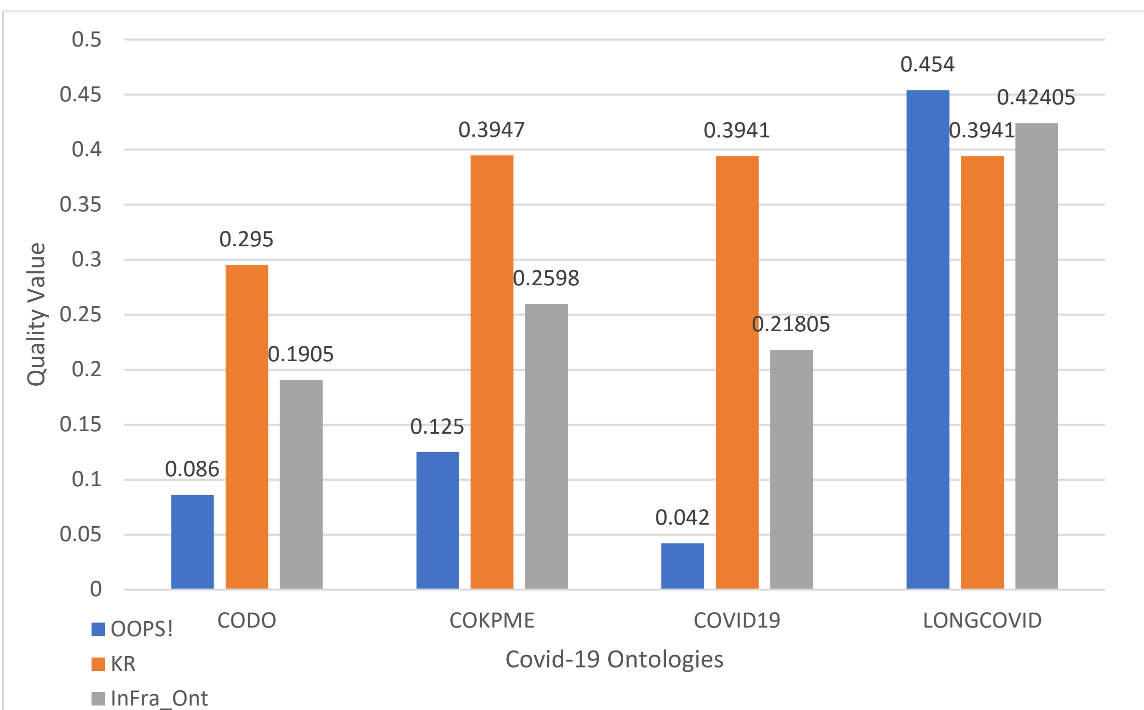

Figure 5: Ontology evaluation via OOPS! tool, KR, and InFra_Ont

The obtained results show that proposed INFra_Ont framework is better as compared to the existing frameworks as it successfully examines the good quality ontology among the available ontologies.

5 Conclusion and Future Research

Ontology provides a way to encode human intelligence so that machines can understand and make decisions by referring to this intelligence. For this reason, ontologies are used in every domain, and now, it becomes important to know the quality of the ontology. Ontology evaluation provides a set of methods and approaches that determine the quality of the ontology based on various criteria. The proposed integrated framework for ontology evaluation uses four well-known approaches to accommodate various criteria of evaluation. The proposed framework needs expert involvement in two approaches, namely evaluation based on the five roles of knowledge representation and evaluation based on the layer approach. The proposed algorithm shows the step-by-step execution of the InFra_OE and the evaluation of InFra_Ont framework shows that it is better as compared to the existing frameworks. The future work of this paper will be based on the evaluation of the proposed framework over different types of ontologies

References

[1] M. Amith, F. Manion, C. Liang, M. Harris, D. Wang, Y. He, and C. Tao, "OntoKeeper: Semiotic-Driven Ontology Evaluation Tool for Biomedical Ontologists." *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, pp. 1614-1617, December 2018.

[2] M. Amith and C. Tao, "Modulated Evaluation Metrics for Drug-Based Ontologies," *Journal of Biomedical Semantics*, 8(1):1-8, 2017.

[3] C. V. S. Avila, G. Maia, W. Franco, T. V. Rolim, A. D. O. da Rocha Franco, A. D. O., and V. M. P. Vidal, "OntoVal: A Tool for Ontology Evaluation by Domain Specialists," *ER Forum/Posters/Demos*, pp. 143-147, November 2019.

[4] J. Bandeira, I. I., Bittencourt, P., Espinheira, and S. Isotani, "FOCA: A Methodology for Ontology Evaluation," arXiv preprint arXiv:1612.03353, 2016.

[5] V. R. Basili and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, 6:728-738, 1984.

[6] A. B. Bouiadjra and S. M. Benslimane, "FOEval: Full Ontology Evaluation," *2011 7th International Conference on Natural Language Processing and Knowledge Engineering*, IEEE, pp. 464-468, November 2011.

[7] J. Brank, M. Grobelnik and D. Mladenic, "A Survey of Ontology Evaluation Techniques," *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, Slovenia: Citeseer Ljubljana, pp. 166-170, October 2005.

[8] A. Burton-Jones, V. C. Storey, V. Sugumaran, and P. Ahluwalia, "A Semiotic Metrics Suite for Assessing the Quality of Ontologies," *Data & Knowledge Engineering*, *55*(1):84-102, 2005.

[9] M. Cristani and R. Cuel, "A Survey on Ontology Creation Methodologies," *International Journal on Semantic Web and Information Systems (IJSWIS)*, 1(2):49-69, 2005.

[10] R. Davis, H. Shrobe, and P. Szolovits, "What is a Knowledge Representation?" *AI magazine*, 14(1):17-17, 1993.

[11] N. C. Debnath, A. Patel, D. Mazumder, P. N. Manh, and N. H. Minh, "Evaluation of Covid-19 Ontologies through OntoMetrics and OOPS! Tools," International Conference on Expert Clouds and Applications (ICOECA), *Lecture Notes in Networks and Systems*, Springer (ISSN: 2367-3370), 2022.

[12] V. Devedzić, "Understanding Ontological Engineering," *Communications of the ACM*, 45(4), 136-144, 2002.

[13] R. Dividino, M. Romanelli, and D. Sonntag, "Semiotic-Based Ontology Evaluation Tool (S-OntoEval)," *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pp. 2687- 2692, May 2008.

[14] A. Duque-Ramos, J. T. Fernández-Breis, M. Iniesta, M. Dumontier, M. E. Aranguren, S. Schulz, and R. Stevens, "Evaluation of the OQuaRE Framework for Ontology Quality," *Expert Systems with Applications*, 40(7):2696-2703, 2013.

[15] H. Fujita and I. Zualkernan, "Evaluating Software Development Methodologies Based on Their Practices and Promises," *Proceedings of the Seventh Somet New Trends in Software Methodologies, Tools and Techniques 08*, *182*:14, 2008.

[16] N. Guarino and C. A. Welty, "An Overview of OntoClean" *Handbook on Ontologies*, pp. 151-171, 2004.

[17] A. Gyrard, M. Serrano, and G. A. Atemezing, "Semantic Web Methodologies, Best Practices and Ontology Engineering Applied to Internet of Things," *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, IEEE, pp. 412-417, December 2015.

[18] D. Kalibatiene, and O. Vasilecas, *Survey on Ontology Languages. International Conference on Business Informatics Research*, Springer, Berlin, Heidelberg, pp. 124-141, October 2011.

[19] D. D. Kehagias, I. Papadimitriou, J. Hois, D. Tzovaras, and J. Bateman, "A Methodological Approach for Ontology Evaluation and Refinement," *ASK-IT Final Conference. June. (Cit. on p.)* pp. 1-13, June 2008.

[20] S. Kim and S. G. Oh, "Extracting and Applying Evaluation Criteria for Ontology Quality Assessment," *Library Hi Tech*, Emerald Publishing, 37(3):338-354, 2019.

[21] S. Kolozali, T. Elsaleh, and P. M. Barnaghi, "A Validation Tool for the W3C SSN Ontology based Sensory Semantic Knowledge," *TC/SSN@ ISWC*, pp. 83-88, October 2014.

[22] B. Lantow, "OntoMetrics: Putting Metrics into Use for Ontology Evaluation," *KEOD*, pp. 186-191, November 2016.

[23] M. McDaniel and V. C. Storey, "Evaluating Domain Ontologies: Clarification, Classification, and

Challenges," *ACM Computing Surveys (CSUR)*, 52(4):1-44, 2019.

[24] K. Munir, and M. S. Anjum, "The Use of Ontologies for Effective Knowledge Modelling and Information Retrieval," *Applied Computing and Informatics*, 14(2):116-126, 2018.

[25] A. Patel, and N. C. Debnath, "Development of the InBan_CIDO Ontology by Reusing the Concepts Along with Detecting Overlapping Information," *Inventive Computation and Information Technologies*, Springer, Singapore, pp. 349-359, 2022.

[26] A. Patel, N. C. Debnath, A. K. Mishra, and S. Jain, "Covid19-IBO: A Covid-19 Impact on Indian Banking Ontology Along with an Efficient Schema Matching Approach," *New Generation Computing*, *39*(3):647-676, 2021.

[27] A. Patel, N. C. Debnath, and P, K. Shukla, "SecureOnt: A Security Ontology for Establishing Data Provenance in Semantic Web, *Journal of Web Engineering*, 21(4):1347-1370, 2022.

[28] A. Patel, and S. Jain, "A Partition based Framework for Large Scale Ontology Matching," *Recent Patents on Engineering*, 14(3):488-501, 2020.

[29] M. Poveda-Villalón, A. Gómez-Pérez and M. C. Suárez-Figueroa, "Oops!(Ontology Pitfall Scanner!): An On-Line Tool for Ontology Evaluation," *International Journal on Semantic Web and Information Systems (IJSWIS)*, *10*(2):7-34, 2014.

[30] P. Q. Rashid, *Semantic Network and Frame Knowledge Representation Formalisms in Artificial Intelligence*, Doctoral Dissertation, Eastern Mediterranean University (EMU)-Do˘gu Akdeniz Universitesi (DA ¨ U)), 2015...

[31] G. R. Roldan-Molina, J. R. Mendez, I. Yevseyeva, and V. Basto-Fernandes, "Ontology Fixing by Using Software Engineering Technology," *Applied Sciences*, 10(18):6328, 2020.

[32] S. Tartir, I. B. Arpinar, M. Moore, A. P. Sheth, and B. Aleman-Meza, "OntoQA: Metric-Based Ontology Quality Analysis," IEEE ICDM 2005 Workshop on Knowledge Acquisition, 2005.

[33] T. Uddin Haider, "Natural Language Text to RDF Schema Conversion and OWL Mapping for an e-Recruitment Domain," *Applications of Advanced Computing in Systems*, Springer, Singapore, pp. 49-57, 2021.

[34] M. Poveda Villalon, A. Gómez-Pérez, and M. C. Suárez-Figueroa, "OOPS!(Ontologty Pitfalls Scanner!): An On-Line Tool for Ontology Evaluation," *International Journal on Semantic Web and Information Sysems (IJSWIS)*, 10(2):7-34, 2012.

[35] J. Yu, J. A. Thom, and A. Tam, "Requirements-Oriented Methodology for Evaluating Ontologies," *Information Systems*, 34(8):766-791, 2009.

Narayan C. Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University (EIU), Vietnam. He is also serving as the Head of the Department of Software Engineering at EIU, Vietnam. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA), USA since 2014. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years where he also served as the Chairperson of the Computer Science Department for 7 years. Dr. Debnath earned a Doctor of Science (D.Sc.) degree in Computer Science and also a Doctor of Philosophy (Ph.D.) degree in Physics. In the past, he served as the elected President and Vice President of ISCA, and has been a member of the ISCA Board of Directors since 2001. Professor Debnath has made significant contributions in teaching, research, and services across the academic and professional communities. Dr. Debnath is an author or co-author of over 500 publications in numerous refereed journals and conference proceedings in Computer Science, Information Science, Information Technology, System Sciences, Mathematics, and Electrical Engineering. Dr. Debnath is an editor of several books published by Springer, Elsevier, CRC Press, and Bentham Science Press on emerging fields of computing. He also served as a guest editor of the Journal of Computational Methods in Science and Engineering (JCMSE) published by the IOS Press, the Netherlands.

Archana Patel is working as a faculty of the Department of Software Engineering, School of Computing and Information Technology, Eastern International University, Binh Duong Province, Vietnam. She has completed her Postdoc from the Freie Universität Berlin, Berlin, Germany. She has filed a patent titled "Method and System for Creating Ontology of Knowledge Units in a Computing Environment" in Nov 2019. She has received Doctor of Philosophy (Ph.D.) in Computer Applications and PG degree both from the National Institute of Technology (NIT) Kurukshetra, India in 2020 and 2016, respectively. She has qualified GATE and UGC-NET/JRF exams in year 2017. Dr.

Patel has also contributed in research project funded by Defence Research and Development Organization (DRDO), for the period of two year and her contribution in teaching is also remarkable. Dr. Patel is an author or co-author of more than 30 publications in numerous referred journals and conference proceedings. She has been awarded best paper award (four times) in the international conferences. She has served as a reviewer in various reputed journal and conferences. Dr. Patel has received various awards for presentation of research work at various international conferences, teaching and research institutions. She has edited three books, served as a guest editor in three reputed journals and served as a keynote at various reputed international conferences. Her research interests are Ontological Engineering, Semantic Web, Big Data, Expert System and Knowledge Warehouse.

Debarshi Mazumder is presently a lecturer in School of Computing & Information Technology at Eastern International University, Binh Duong, Vietnam. He received his M. Tech from Jadavpur University, Kolkata, India in 2013, and B. Tech from West Bengal University of Technology, Kolkata, India in 2008. His areas of research interests are Image Processing, Pattern Recognition, Machine Learning, IoT.

Manh-Phuc Nguyen received his undergraduate from Eastern International University in 2016. He finished his master's degree from the University of Information Technology, Ho Chi Minh National University, Viet Nam. He has been a full-stack software developer for 5 years and turned to be a lecturer of Software Engineering at Eastern International University. His research interests include computer vision, ontology learning, and software engineering. Recently, his work is focused on methods of object recognition and object tracking in traffic context and detect traffic violations. He would like to build a computer vision-based system to improve safety on the road.

Minh Ngoc Ha has 20 years of experience in the software development field, with over 15 years as a leader and 10 years as a technical director for some software firms. He turned to be a lecturer in Software Engineering at Eastern International University for nearly 10 years. His research interests include Data Structures and Algorithms, Software Engineering Methods, and teaching methodology. He loves to share his knowledge, skills, and experience with thousands of CIT students to promote them in their careers.