applied sciences

Article

Ecological and Confined Domain Ontology Construction Scheme Using Concept Clustering for Knowledge Management

Khalid Mahmood [1,*], Rahmah Mokhtar [1,*], Muhammad Ahsan Raza [2], A. Noraziah [1,3] and Basem Alkazemi [4]

1 Faculty of Computing, College of Computing and Applied Sciences, University Malaysia Pahang, Pekan 26600, Malaysia
2 Department of Information Technology, Bahauddin Zakariya University, Multan 60000, Pakistan
3 Center for Software Development & Integrated Computing, Pekan 26600, Malaysia
4 Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia
* Correspondence: keeenkhan@gmail.com (K.M.); drrahmah@ump.edu.my (R.M.)

Abstract: Knowledge management in a structured system is a complicated task that requires common, standardized methods that are acceptable to all actors in a system. Ontology, in this regard, is a primary element and plays a central role in knowledge management, interoperability between various departments, and better decision making. The ontology construction for structured systems comprises logical and structural complications. Researchers have already proposed a variety of domain ontology construction schemes. However, these schemes do not involve some important phases of ontology construction that make ontologies more collaborative. Furthermore, these schemes do not provide details of the activities and methods involved in the construction of an ontology, which may cause difficulty in implementing the ontology. The major objectives of this research were to provide a comparison between some existing ontology construction schemes and to propose an enhanced ecological and confined domain ontology construction (EC-DOC) scheme for structured knowledge management. The proposed scheme introduces five important phases to construct an ontology, with a major focus on the conceptualizing and clustering of domain concepts. In the conceptualization phase, a glossary of domain-related concepts and their properties is maintained, and a Fuzzy C-Mean soft clustering mechanism is used to form the clusters of these concepts. In addition, the localization of concepts is instantly performed after the conceptualization phase, and a translation file of localized concepts is created. The EC-DOC scheme can provide accurate concepts regarding the terms for a specific domain, and these concepts can be made available in a preferred local language.

Keywords: concept clustering; domain ontology; knowledge mining; ontology construction; ontology localization; structured knowledge management; computational intelligence

Citation: Mahmood, K.; Mokhtar, R.; Raza, M.A.; Noraziah, A.; Alkazemi, B. Ecological and Confined Domain Ontology Construction Scheme Using Concept Clustering for Knowledge Management. *Appl. Sci.* **2023**, *13*, 32. https://doi.org/10.3390/app13010032

Academic Editor: Vincent A. Cicirello

Received: 5 October 2022
Accepted: 19 November 2022
Published: 20 December 2022

1. Introduction

The current era is the age of information technology and the implementation of new and advanced ideas in organizations; therefore, the large and varied amount of implicit, explicit, structured, and unstructured knowledge is increasing every day. Various knowledge management techniques have been introduced to effectively collect and manage such complex and varied knowledge. Typically, a knowledge management system is developed as a framework based on computational intelligence and used to share intellectual information. Recent research has shown that various companies and businesses invest funds to implement knowledge management systems to increase the efficiency of their organization operations or processes [1].

A structured knowledge management system involves multidisciplinary groups of users who resolve common issues at the domain level [2,3]. In this system, a domain provides well-organized knowledge for the management of and robust coordination between various autonomous departments. Typical examples of structured knowledge management are waste management systems, postal codes, the Semantic Web, flood management systems, map preparation, the Olympic Games, and geographic information management systems. Recently, research on these systems has focused on the integration of ontologies in knowledge management. However, the conceptualization and integration of ontologies should be based on organizational facts [4]. Ontology-supported knowledge systems enable a system and user to exchange knowledge-based information with each other due to the common knowledge of a specific field or domain.

Ontology is the most essential and primary component of knowledge-based applications. Ontologies are required in knowledge-based systems to serve as domain descriptions (i.e., formal models) that are machine-understandable. Knowledge management systems depend on ontologies to transform data into a machine-readable form to ensure reliable performance. Various ontologies have been introduced for general use, e.g., WordNet and Cyc [5], but most applications need ontologies related to some specific domains to define the relevant concepts and relations. As organizational knowledge is distributed in nature, knowledge-based systems need to integrate knowledge from different sources. This integration can only be achieved through the use of ontologies [6]. However, finding ontologies for this purpose is a big problem [1].

Ontology construction has been an important research domain in recent years [7–12]. Governments and industries have adopted ontology-based knowledge management systems and developed innovative applications in the deployment of semantic technology. A variety of ontology construction schemes have been presented by various scholars [13–28]. Studies of ontology construction schemes have mostly discussed the conceptualization, evaluation, domain analysis, implementation, and instantiation aspects (including domain ontology), but few of them have provided details about the adopted schemes [29]. Therefore, it is difficult to understand the schemes for the construction of ontologies for specific domains despite the need for well-designed schemes to perform the complex, time-consuming, and tedious task of ontology construction. An ontology construction method must comprise a detailed and clear discussion of the methodology, processes involved, and guidelines for the developers [30]. In this research, a scheme for ecological and confined domain ontology construction (EC-DOC) in structured knowledge management is proposed based on the ontology construction schemes of [6,31,32]. The proposed scheme provides a general structure for domain ontology construction that can be implemented in the development of any domain ontology-based structured knowledge management system. The main contributions of this research are as follows:

- The EC-DOC scheme provides support for the most important aspects of ontology construction including domain analysis, conceptualization, reusability, integration, and localization.
- The EC-DOC scheme introduces conceptualization and a Fuzzy C-Mean soft clustering mechanism. The conceptualization involves the identification of domain-relevant concepts, which are clustered via the Fuzzy C-Mean clustering technique.
- The scheme utilizes the concept of localization, i.e., localized concepts are extracted according to the clusters formed for key terms or concepts.
- The proposed scheme simplifies the implementation of activities regarding ontology construction in structured knowledge management.

The remaining sections of the research paper are arranged as follows. A summarized analysis of related work is presented in the literature review. The proposed EC-DOC scheme is discussed in detail in the methodology section. A detailed discussion of the proposed methodology is provided in the discussion section. The conclusion of this research is provided in the final section.

2. Literature Review

2.1. Background Work

The primary functionality of structured knowledge management is to share domain knowledge within an organization [32]. Furthermore, an ontology-based knowledge management system must have clear items that are stored in a knowledge base in various formats to enable the convenient reuse and localization of knowledge. This system can be divided into three stages: (1) knowledge mining, (2) knowledge representation, and (3) knowledge connection. The central idea of the whole process is to store knowledge in the ontology format. An ontology-based knowledge management system is represented in Figure 1.

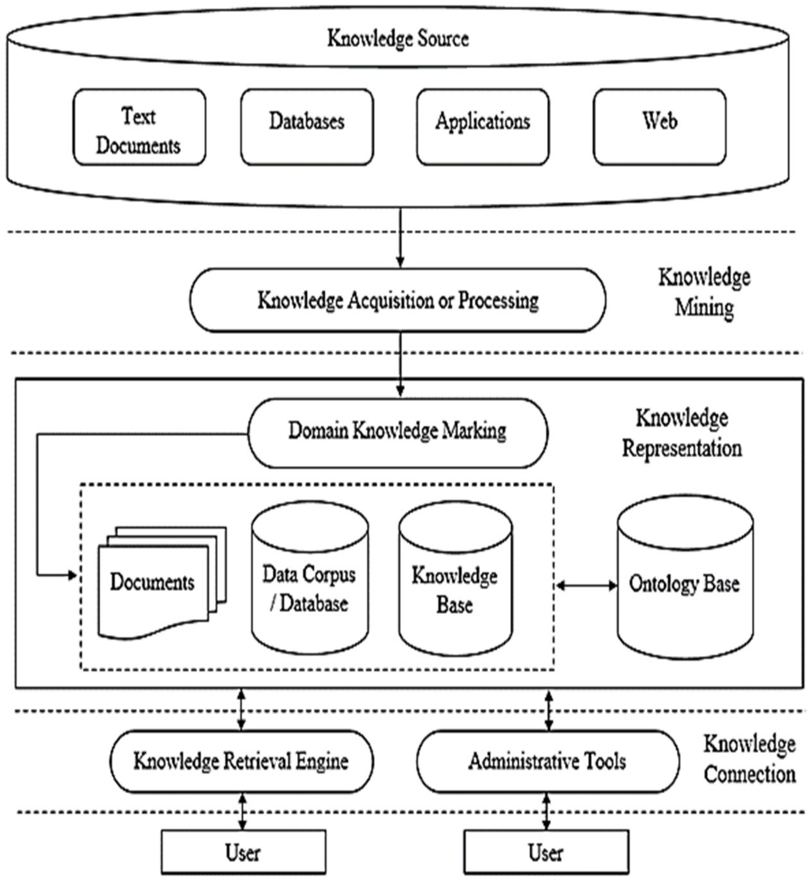

Figure 1. Ontology-based structured knowledge management system.

2.1.1. Knowledge Mining

The phase of knowledge mining or knowledge acquisition is considered the most significant part of a structured knowledge management system. In the knowledge mining phase, all the essential knowledge (unstructured and semi-structured information) is converted into structured information. This phase enables the sorting and placement of knowledge sources including documents, web, applications, databases, forums, and user feedback into a structured knowledge base. In this regard, the process of knowledge mining is the process of knowledge creation, not knowledge transformation or conversion.

2.1.2. Knowledge Representation

An ontology consists of classified concepts and relations related to domain knowledge [33]. In the knowledge representation phase, metadata are retrieved from the acquired knowledge sources, and knowledge objects are marked in terms of the domain ontology. The marked objects can also be stored in the local language or in multiple languages. The marked objects are placed in the knowledge base, which is promptly used for the effective search of knowledge objects.

There are variety of tools and techniques used for knowledge mining, but knowledge representation is a key consideration for all of them. A knowledge representation scheme must be selected to obtained a well-established description of a domain [14–16].

A. Decision Trees (Semantic Nets)

When using decision trees, knowledge is represented in graphs or hierarchical formats [34]. In such graphs, concepts are represented by nodes or vertices and the relations between these nodes (concepts) are represented by edges or lines.

B. Conditional Rules (Symbolic Rules)

IF–THEN conditional rules can be applied to represent domain knowledge. One or more logical operators such as "AND", "OR", and "NOT" are used to connect conditions in this type of technique [34].

C. Fuzzy Logic (Fuzzy Rules)

Fuzzy types of logic or rules can well-represent vague concepts in terms of "high" or "low". Fuzzy logic sets membership criteria by using a membership function in a range from 0.0 to 1.0, where these values represent absolutely false or absolutely true, respectively [35].

D. Case Based (4Rs Strategy)

Case-based techniques maintain the history of previous cases and their respective solutions. They can be used to find and implement a previous solution to solve similar cases, which they achieve through the implementation of the "retrieve", "reuse", "revise", and "retain" strategy [36].

E. Neural Networks (Connectionist)

The unique neural network approach consists of various processing units known as neurons connected to each other. These can efficiently produce conclusions, and this inference depends on numerical computations. A major drawback of neural networks is that they lack a natural representation of knowledge. Therefore, the knowledge produced by neural networks is not always comprehensible [37].

The merge or integration of two or more schemes can also solve domain knowledge representation problems in structured knowledge management. The authors of [17,31] argued that the best scheme (compared with single or integrated approaches) is one that satisfies the user requirements for a knowledge management system.

2.1.3. Knowledge Connection

In the knowledge connection phase, the knowledge is regulated, added, updated, edited, and saved by the managers of the knowledge base. This enables a system to competently and dynamically participate in confined usage and maintenance. Thus, users are able to find related data in different ways according to their requirements and preferences.

2.2. Related Work

An ontology construction scheme provides a developer's guidelines, processes, and techniques for ontology construction [38]. A variety of ontology construction schemes have been proposed in the literature [14,17–20,30,39,40]. It is important to notice that different authors have adapted domain ontology construction methodologies in different ways; therefore, there is no common or universal methodology for ontology construction [21]. Furthermore, different ontology construction methodologies focus on different aspects of domains in the process of ontology construction. For instance, some focus on the scope of the ontology and the analysis of the domain, while others focus on the validation of the ontology. In addition, we cannot find details regarding the performed activities and design descriptions for the ontology construction methodologies presented in the literature.

In a later section, we provide a comparative analysis of various ontology construction techniques regarding different criteria. The evaluation criteria were set following the observation of ontology construction trends in previous studies, and they cover various aspects of methodologies for the construction of domain ontologies. The criteria used for the comparison consisted of some coarse-grained level aspects and some fine-grained level aspects of ontology. The coarse-grained level includes reusability, localization, methodology roots, and integration. The fine-grained level includes technical aspects such as domain analysis, the availability of details, conceptualization, implementation, evaluation, maintenance, and documentation. The evaluation was carefully performed and prepared following an in-depth analysis of domain ontology construction methodologies. This study will obviously help researchers understand various ontology construction techniques and select the most suitable mechanism for the design of a chosen domain ontology.

A comprehensive comparison of some existing ontology construction schemes according to the decision criteria is given in Table 1, and the description of each criterion is as follows.

Table 1. A comparative study of domain ontology construction schemes.

Reference	Criteria											
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**
Ahmad et al. [13]	✓	×	✓	×	✓	✓	✓	×	✓	×	×	×
Fawei et al. [14]	×	×	×	×	✓	✓	✓	✓	✓	×	×	×
John et al. [15]	×	×	×	×	✓	✓	✓	×	✓	×	×	×
Li and S. Alian [16]	×	×	×	×	✓	✓	✓	✓	✓	×	×	×
Abdelghany et al. [18]	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	×
Alsanad et al. [19]	×	×	×	×	✓	✓	✓	✓	✓	×	×	×
Jacksi et al. [20]	×	×	×	×	✓	✓	✓	✓	×	×	×	×
Yunianta et al. [21]	×	×	×	✓	✓	✓	✓	✓	✓	×	✓	×
Trokanas et al. [23]	✓	×	×	✓	✓	✓	✓	✓	✓	×	×	×
Zhang et al. [24]	×	×	×	×	✓	✓	✓	✓	✓	×	×	×
Shaharin et al. [25]	✓	×	×	✓	✓	✓	✓	✓	✓	✓	×	×
Missikoff et al. [26]	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓	×
Dutta et al. [27]	×	×	×	×	✓	✓	✓	×	✓	×	×	×
Bautista et al. [28]	×	✓	×	×	✓	✓	✓	✓	×	×	×	×
Sattar et al. [31]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Figueroa et al. [40]	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	×
EC-DOC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

A. Ontology Reuse

When an ontology is too dependent on application, it can become costly to reuse. Upper-level ontology utilization helps to reduce ontology reuse costs by applying a common structure in domain ontologies.

B. Localization

Another aspect of ontology construction is ontology localization, which is the process of adapting an ontology into a local language. Here, considered whether the ontology construction approaches supported ontology localization.

C. Methodology Roots

We also analyzed whether the ontology construction approaches were developed on the basis of well-designed existing methodologies, as a domain ontology construction approach based on an existing method requires special attention for use in different applications of ontology representation semantics.

D. Integration

Integrating knowledge from multiple ontologies is a complex task. Ontology integration is reliant on the integration different ontologies with some similar representational aspects. The effectiveness of the integration of ontologies is computed in the form of accuracy and recall measures.

E. Domain Analysis

Domain analysis is a significant aspect of ontology construction during which the requirement of resources and the absence of domain-related guidance in the knowledge acquisition phase are studied. Domain analysis grows more complicated when knowledge from different fields is integrated.

F. Availability of Details

The availability of details regarding the technique and applied methodology in the design and construction of an ontology is also another aspect worth studying. With no details, it is difficult for future researchers to follow a methodology for ontology construction.

G. Conceptualization

This study was focused on whether a studied methodology implements or includes any conceptualization process in knowledge mining.

H. Implementation

The application of the conceptual methodology of an ontology requires a language for the representation of concepts.

I. Evaluation

The accurate and appropriate understanding of an ontology is the primary goal of ontology evaluation.

J. Maintenance

The evolution of an ontology is a continuous problem that must be addressed by ontology construction techniques. Changing an ontology is also a complex task that requires constant maintenance to ensure the reliability of its domain.

K. Documentation

In the context of maintenance, the methodology provided for domain ontology construction should be appropriately documented. The documentation of ontology construction aids the understanding, usability, reusability, and revisions of an ontology.

L. Concept Clustering

Concept clustering is a technique used to divide the concepts identified during the conceptualization phase into groups related to some key terms. This can be adapted to increase the accuracy of finding relevant concepts regarding key terms.

During the study of the literature, we found that most authors of schemes provided analyses of their domain, conceptualization, implementation, and evaluation along with a domain ontology example. However, most the considered studies did not provide detailed descriptions of their techniques, which could lead to problems in the design of other domain ontologies. However, the authors of some schemes did provide sufficient specific details [13–16,31].

The study of the literature also showed that most schemes did not support maintenance, reusability, documentation, integration, and localization. Maintenance is mandatory because changes in ontologies are sometimes required. Some researchers, such as those of [16,17,24,25,31], discussed maintenance support availability. Documentation is important because it aids the understanding of an ontology, and documentation was provided in [13,16,21,27,30,41].

The construction of an ontology is a difficult task. In this regard, the reusability of previously developed ontologies is recommended. A few schemes, including those of [24,25,30], provided evidence in support for ontology reuse. Translating ontologies into different languages (such as English, Urdu, Hindi, Malay, French and German) is a key part of localization support. Only two schemes [27,31] found during the study of the literature provided information on ontology localization. The scheme presented in [30,31] provided information about its root methodology, Design Science Research (DSR). The operative goal of an ontology-based knowledge management system in a distributed processing environment is to improve the performance and the effectiveness of computational intelligence [42]. A major requirement of knowledge management systems is that they provide a reliable service of relevant and accurate data retrieval and availability through the implementation of various techniques and algorithms regarding computational intelligence [33,43].

3. Methodology

The objective of the domain ontology construction methodology proposed in this research is to adapt the most preeminent practices that have already been introduced in the study of ontology construction. In the design and development of this methodology, the tasks were classified into various phases regarding cluster-based ecological and confined domain ontology construction in structured knowledge management. These phases are as follows: (1) Pre-conceptualization: this phase comprises activities that must be performed before the identification of terms for the construction of a domain ontology; (2) Conceptualization: this phase involves the identification of terms and/or concepts from a data source; (3) Pre-processing: in this phase, numerical values are computed for the textual terms/concepts available in the dataset; (4) Concept clustering: the aim of this phase is to cluster the instances and concepts through numerical values that were already computed in the pre-processing phase; and (5) Post-conceptualization: this phase consists of activities such as localization, verification, implementation, integration, evaluation and documentation. This ontology construction scheme comprises a combination of mechanisms, namely, adapted ontology development methodology [31], concept clustering, and concept extraction mechanisms (see Section 3.3). The proposed scheme is illustrated in Figure 2. A detailed description of the phases and activities involved in the proposed scheme is discussed below.

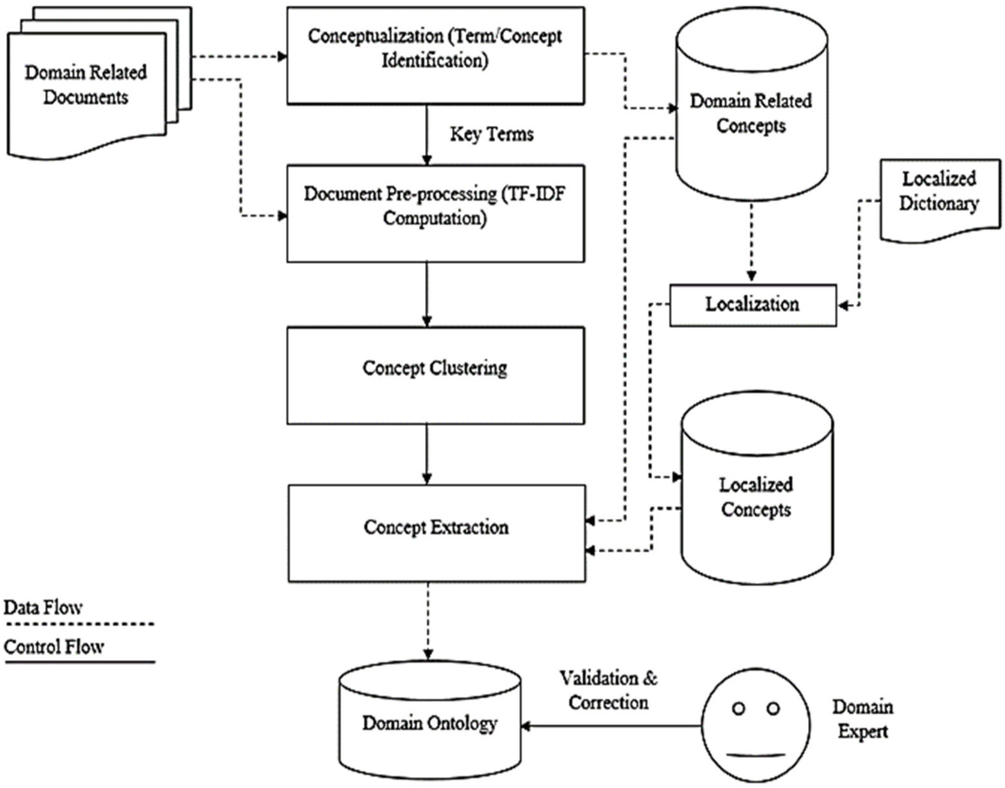

Figure 2. Phases in the proposed EC-DOC.

3.1. Phase 1: Pre-Conceptualization

The tasks involved in the first phase of pre-conceptualization must be performed before the startup of actions proposed in this article of domain ontology construction. These tasks are as follows.

3.1.1. Demarcation

A Requirement Specification Document for Domain Ontology Construction (RSD-DOC) is created in this phase. In this document, details on the objective, scope, implementation language, users, use, functional, and non-functional requirements that the ontology will satisfy are provided [31,44,45]. Ontology requirements that are content-centric are known as functional requirements. The general characteristics of an ontology that are not relevant to the ontology's contents fall into the category of non-functional requirements. Tools such as WebProtege, naming conventions, and the semantic representation language used for ontology construction are some examples of non-functional requirements. A major technique used for RSD-DOC construction is the answering of competency questions. An RSD-DOC provides an agreement between the owners of an ontology, engineers, domain experts, and users, and some competency questions aid the finding of functional requirements. An example RSD-DOC is shown in Table 2, which also lists a set of general competency questions according to related domains.

Table 2. A set of some general competency questions.

Requirement Specification	Description
Objective	Sharing knowledge and providing facilities for knowledge management in the main domain and its sub-domains are the major objectives of domain ontology.

Table 2. *Cont.*

Requirement Specification	Description
Scope	The implementation of the domain ontology in major domains and their sub-domains is the scope.
Language for Domain Ontology Implementation	Any language used for semantic web or OWL can be used for the implementation of the domain ontology.
Functional and Non-Functional Requirements	Functional Requirements The domain ontology must satisfy some questions starting from 'How to . . . ?'
	Non-functional Requirements Multilingual facility/support must present in the system International standards are mandatory to meet for the domain ontology
Uses	Keeping records of activities Maintaining the history

3.1.2. Identification of Tools and Techniques

The designers, translators, and developers of both ontologies and applications are responsible for performing the identification of tools and techniques after defining a domain ontology. In this phase, related and compatible modern tools for ontology construction are selected, and the latest application techniques are identified for knowledge capture.

The tools used for browsing, inspecting, coding and editing ontologies are called Ontology Construction Tools (OCTs) or Ontology Development Editors (ODEs). Some examples are SWOOP, IsaViz, WebProtege and Apollo [46]. WebProtege can also be used for the validation and verification of domain ontologies [47]. Some popular examples of ontology representation languages are the Knowledge Interchange Format (KIF), Agent Markup Language (AML) of DARPA, Web Ontology Language (OWL) and Resource Description Framework (RDF) [31,48].

3.1.3. Resource Identification

In the resource identification phase, the required resources for the construction of a domain ontology are identified; this is the responsibility of ontology designers and domain experts following the identification of knowledge sources. These resources are not necessarily useful in their already existing form, but they can be re-engineered or modified according to one's current needs. Resource selection should be performed after the verification, assessment, and comparison of ontology contents to decide the suitability of the considered resources.

3.2. Phase 2: Conceptualization

After the pre-conceptualization phase, the conceptualization phase is completed. The activities of this phase include knowledge acquisition and concept localization.

3.2.1. Knowledge Acquisition

Knowledge acquisition is performed by ontology designers and domain experts following knowledge source identification, during which non-ontological and ontological resources related to a specific domain are identified. In the knowledge acquisition phase, domain-related concepts and properties are defined and an ultimate lexicon of concepts and the properties of those concepts is created. The newly created glossary provides rich descriptions of domain-related concepts. An example glossary of some general concepts that can be replaced with specific domain-related concepts is provided in Table 3.

Table 3. A sample glossary containing some general concepts.

Concept	Description of the Concept
Country	A territory occupied by a nation having its own government.
State	A politically organized society under a government.
District	A zone with particular characteristics
Location	A particular place.
Feedback	Information about reactions.
Trip	A journey.
Firm	A business organization.
Contract	A written or spoken agreement.
Vehicle	A machine used for transportation.
Driver	A person who drives a vehicle.
Owner	A person who owns a property.
Waste	Any garbage.
Supervisor	A person who supervises an activity.
House	A building used for residence.
Property	An asset belonging to someone.

3.2.2. Localization

The availability of high-quality web contents in local languages is a reason for the success of the Semantic Web. A statistical analysis of internet users by language is provided by Internet World Stats (IW Statistics) [49], and Figure 3 illustrates this statistical analysis as a bar graph.

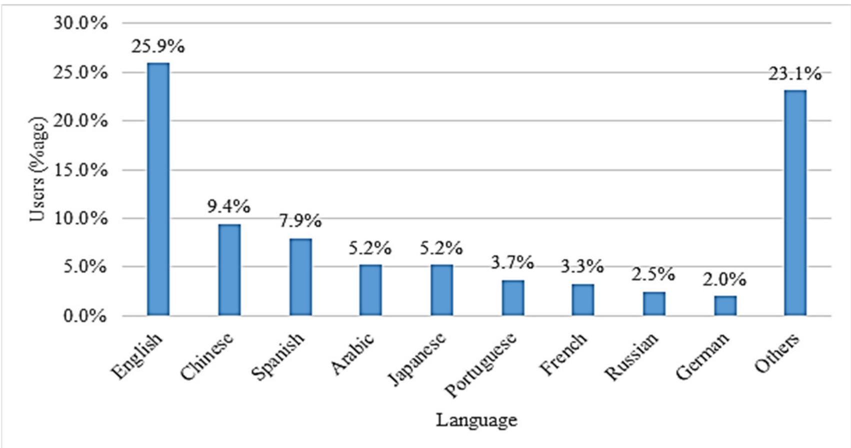

Figure 3. Internet user distribution by language [49].

In this regard, the translation of a domain ontology into multiple local languages is a good practice known as localization [16]. The translators of ontologies and domain experts perform the localization of ontology concepts during or after the conceptualization process. One task performed in domain ontology localization is the creation of a file containing translations of all concepts, as well as their properties and relationships. In this regard, a translation file is separately created for each local language.

One example of localization is the translation of the concept of "Country" into languages such as Arabic, Urdu, Malay, and Chinese [50]. Translations of this concept into these languages are illustrated in the diagram provided in Figure 4, and a sample translation file for a local language (Malay) is shown in Table 4. Any translation editor such as

POEDIT can be used for translation purposes. However, the process also requires some additional algorithms, implementation languages, and file structures for the manipulation and integration of translation files.

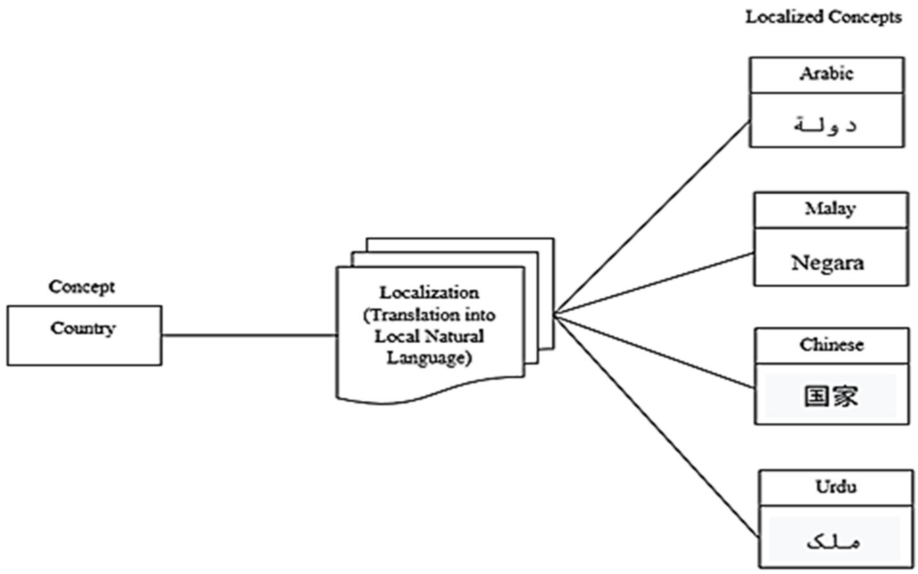

Figure 4. Translation of a concept into local natural languages.

3.3. Phase 3: Pre-Processing

A major problem faced by the development of natural language processing (NLP) models is the conversion of terms in textual data into numerical weights. The correlation of concepts is necessary for the implementation of such NLP models, including sematic information retrieval models and domain ontology construction methodologies. This task must be performed in the pre-processing phase because the models developed for machine learning and deep learning are not able to directly process text data, and efficient techniques are needed to convert textual data into numerical data. Term frequency and inverse document frequency (TFIDF) is one intelligent method that can be used to convert textual terms into numerical weights [51,52] and to calculate the significance of terms in a single document. TFIDF comprises three measures, as described below.

Table 4. A sample translation file (Malay language).

Source Concept (English)	Localized Concept (Malay)
Country	Negara
Vehicle	Kenderaan
Trip	Perjalanan
Owner	Pemilik
House	Rumah

3.3.1. Term Frequency (TF)

The term frequency measure is the computation of the frequency of term occurrences in a document. A mathematical representation of *TF* is shown in Equation (1).

$$TF_{ij} = \frac{Total\left(t_i occurences\ in\ d_j\right)}{Total\left(terms\ in\ d_j\right)}, \tag{1}$$

3.3.2. Inverse Document Frequency (IDF)

Inverse document frequency is the binary log of the fraction of the total documents in a repository and the total amount of documents possessing the term. Equation (2) is the general mathematical representation of *IDF*, where *N* represents the total number of documents in a corpus.

$$IDF_{ij} = \log_2\left(\frac{N}{Total\left(d_j\ with\ t_i\right)}\right) + 1, \tag{2}$$

3.3.3. Term Frequency and Inverse Document Frequency (TFIDF)

The combined weights of key terms can be calculated by multiplying the two already computed frequencies of *TF* and *IDF*. Equation (3) can be used to individually calculate the combined weights of key terms.

$$(TFIDF)_{ij} = (TF_{ij} \times IDF_{ij}), \tag{3}$$

It has already been discussed that machine learning and deep learning algorithms are unable to directly deal with textual data, so it is necessary to understand the mechanism of *TFIDF* to comprehend these algorithms developed for machine learning tasks. The task of assigning numerical values to any textual term is performed through the computation of *TFIDF*, and the numerical values computed through *TFIDF* support the formation of clusters of relevant documents after the indexing of terms or concepts and documents.

3.3.4. Domain Concept Extraction

Concept extraction is a major task in domain ontology construction. It is necessary to extract domain-related concepts from terms with TFIDF values. There are various domain concept extraction methods in the field of domain ontology construction. In the recent age of semantic information retrieval, semantic similarity-based manual or automatic filtering techniques are gaining popularity in the research community of domain extraction [53]. In this phase, domain experts perform the task of concept selection. Initially, a glossary is created, and that glossary is managed according to the relevant requirements. This initial glossary is created from a domain-related corpus, and the initial glossary is used as a customized dictionary [45]. The candidate domain concepts or key terms are selected from the domain-related literature. The most frequent terms are selected to calculate the *DRC* (Degree of Domain Relation) of a candidate concept, as described in Equation (4):

$$DRC = \frac{f_c}{F}, \tag{4}$$

where f_c represents the frequency of a candidate concept in the domain-related documents in a cluster and *F* represents the total number of domain concepts. If the calculated *DRC* of the concept is higher than any threshold value, then specific documents are selected, and the related concepts are extracted according to the *DRC*. These extracted concepts are then added to the ontology of that domain.

3.4. Phase 4: Concept Clustering

The purpose of the concept clustering phase is to cluster the concepts extracted from documents on the base of their numerical weights calculated through TF-IDF. The terms with the highest weights are selected as important terms for the clustering of concepts [54]. Here, a widely used and popular Fuzzy C-Mean soft clustering algorithm was considered for the formation of clusters, as Fuzzy C-Mean is used in most cluster-based semantic information retrieval systems. This clustering algorithm is used in a large variety of schemes to study the impact of parameter values on performance, dual expression-based fuzzy clustering, image clustering, membership scaling, pattern recognition with objective functions, and other data analysis techniques [54–61]. Fuzzy C-Mean is used in different cluster-based sematic information retrieval systems, data analysis methodologies, and cluster formation problems because of its flexibility, strengths, and significant impact on the effectiveness and efficiency of textual document, textual term, and concept clustering.

The Fuzzy C-Mean algorithm was first introduced in [62], where it was an enhanced version of an already proposed clustering technique. The Fuzzy C-Mean algorithm represented in Figure 5 has the ability to form clusters of data values represented in multidimensional space [51].

Algorithm : Fuzzy C-Means

Initialize matrix $U = [u_{ij}]$, $U^{(0)}$

Calculate the centers vectors at step k, $C^{(k)} = [C_j]$ *with* $U^{(k)}$, where

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} . x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Update $U^{(k)}, U^{(k+1)}$ where $u_{ij} = \frac{1}{\sum_{k=1}^{C}\left[\frac{\|x_i - C_j\|}{\|x_i - C_k\|}\right]^{\frac{2}{m-1}}}$

If $\left\|U^{(k+1)} - U^{(k)}\right\| < \mathcal{E}$, then EXIT,

Else GO TO second step

Figure 5. Fuzzy C-Mean algorithm [51].

The computation of a membership degree of between 0 and 1 is the major benefit of this soft clustering mechanism. Another of its advantages is that a single data value can be the member of various clusters. This flexibility is provided through the minimization of the objective function, as provided in Equation (5):

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \parallel x_i - c_j \parallel^2, \ 1 \leq m < \infty, \tag{5}$$

where m represents the fuzzy exponent variable value that is always greater than 1, u_{ij} represents the membership degree of concept x_i in jth cluster, and c_j represents the cluster centroid. The calculation of the similarity distance between a concept and the centroid is represented with $*$ in the equation.

In fuzzy partitioning, the objective function is an important task that is achieved by updating the centroid of cluster c_j and the degree of membership u_{ij} through an optimal number of iterations. A mathematical representation of the membership degree calculation is given in Equation (6).

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left[\frac{\|x_i - C_j\|}{\|x_i - C_k\|}\right]^{\frac{2}{m-1}}}, \tag{6}$$

The calculation of C_j is depicted in Equation (7):

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} . x_i}{\sum_{i=1}^{N} u_{ij}^{m}}, \tag{7}$$

where N represents the total number of concepts in a cluster. This procedure is continuously and iteratively performed until the condition expressed in the following mathematical expression (8) is met:

$$max_{ij}\{\| U^{(k+1)} - U^{(k)} \|\} < \mathcal{E}, \tag{8}$$

where k represents the iteration count, U is the centroid of each iteration, and $\mathcal{E}$ is the exit state between 0 and 1.

The sequence of operations performed in concept clustering is represented in Figure 6, which shows a flow chart of the Fuzzy C-Mean algorithm. The procedure starts after textual concepts are converted into numerical values, and various stages are involved. The number of clusters are decided on the basis of key concepts. The degree of the membership value of each key concept is first calculated as the distance between the centroids of each cluster and other concepts. Then, clusters are formed based on the measured distance. From the newly formed clusters, a new centroid is computed to include more relevant concepts. The process of computing a new centroid for each cluster and adding more relevant concepts is continued until new centroids are computed. After the completing the concept clustering procedure, further phases such as concept extraction and ontology construction are enacted. Because we use the soft clustering mechanism in this proposed model, a concept can be a member of more than one cluster.

Regarding overlapping clustering, popularly known as the combination of soft clustering and hard clustering, it is worth considering that in a soft clustering mechanism such as the Fuzzy C-Mean algorithm, data values or concepts can be members of multiple clusters depending on membership degree values while in a hard clustering mechanism, the typical K-Mean method is not suitable for the semantic clustering of concepts. For example, when clustering a dataset containing information on movies, one movie may belong to many genres. As an example, an action movie may also fit the historical and romantic genres [63]. Similarly, a concept may be related to multiple key concepts or terms. Accordingly, it is more optimal to use a Fuzzy C-Mean clustering mechanism than a hard clustering mechanism for such problems.

3.5. Phase 5: Post-Conceptualization

The clarification of concepts relevant to a specific domain is an important factor in ontology construction. The process of conceptualization plays a central role in the proposed scheme, and this task is performed at various stages. Like pre-conceptualization and conceptualization, post-conceptualization is a phase that includes activities such as validation, maintenance, and documentation. Each of these activities has their own importance at their level of operation. These activities are separately discussed in detail in the following sub-sections.

3.5.1. Validation

The validation process is used to ensure that an accurate and precise domain ontology is created [31,64]. In this phase, an RSD-DOC is validated by practically implementing or examining the domain ontology. If the RSD-DOC does not fulfill the requirements of its owner and the users of the ontology, then the domain ontology must be revised. Different approaches can be adopted to validate a domain ontology's specifications. These approaches include questionnaires, feedback recording, and interviews from the users and owners of the domain ontology. However, it is recommended that the validation process be conducted after completing the domain ontology construction process [65]. Furthermore, an ontology-based structured knowledge management application may assist in the validation of a domain ontology during its development process [66].

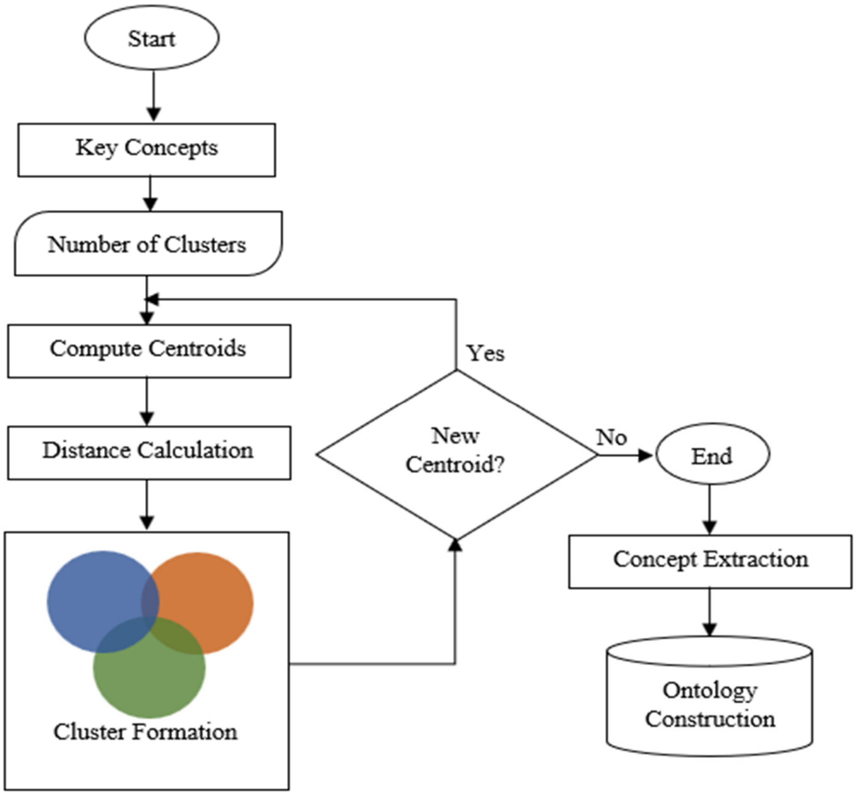

Figure 6. Flow chart of concept clustering using the Fuzzy C-Mean algorithm.

3.5.2. Maintenance

A developed domain ontology may have some errors or some missing domain knowledge. This missing knowledge is incorporated in the maintenance phase. The appropriate versioning of a domain ontology is maintained, and updated versions of the domain ontology are developed. Modifications of previous requirements or the implementation of new requirements are enacted in these updates.

3.5.3. Documentation

Technical writers, ontology designers, and domain experts are responsible for the documentation phase, during which all ontology construction activities are documented. These activities should be well-documented and -described [67]. The documentation of activities aids the understanding, usability, reusability and revision of a domain ontology, and ontology documents consist of ontology contents that are human-readable, documentation metadata that are machine-readable, and web-based versions of the documentation [68,69]. The documentation should also include detailed descriptions of all relevant statements regarding the ontology, purpose, modeling criteria, reusability, and base theories [69,70]. This documentation provides detailed information about the domain ontology construction process.

4. Implementation

The translation or transformation of a designed ontology's formal model into a formal language such as RDF, OWL and LINGO is performed in this phase. There are many reasons, mainly computations and machine-readability, for the transformation of the specifications of a domain ontology into the formal specifications of a domain ontology. Some automated tools such as WebProtege are very supportive for the transformation of a formal ontological model into a formal ontology language [31,66]. Importantly, the formal language should be selected based on the established scope and purpose, and the selection should be performed by the ontology developers. In some cases, the selected semantic formal language is does not meet the requirements for the formal design of a model; here, the ontology developers are responsible for changing the sematic formal language selection. In the proposed ontology construction methodology, we strongly suggest translating the designed domain ontology formal model into OWL because it provides strong representation for groupings and relationships between things.

Ontologies are used to represent the relationship between concepts in any complex domain. Regarding the implementation of this proposed scheme, the authors intent to implement it in the agriculture domain. As an example, a general representation of an ontology in the agriculture domain based on the grouping of crops by season, variety, and soil is provided in Figure 7.

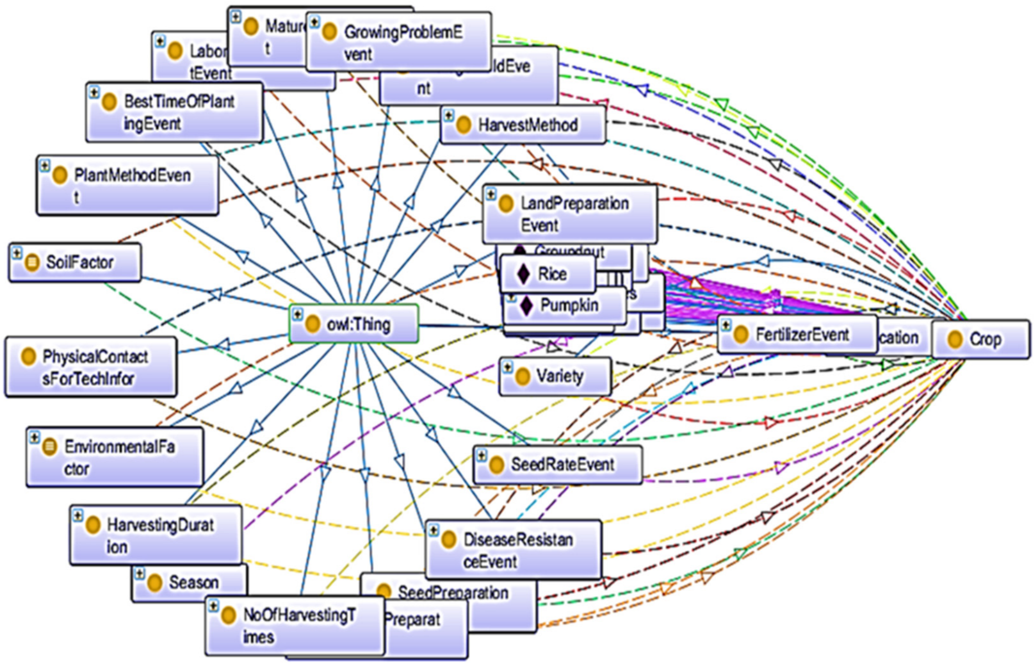

Figure 7. A general representation of an ontology in the agriculture domain.

Various inter-related events can also be represented in this ontology. Details regarding planting methodologies, climate change, and other environmental and harvesting factors are also represented in the graph shown in Figure 7.

5. Evaluation

The evaluation process of a domain ontology construction methodology needs special attention because it plays a critical role in the implementation of ontology-based structured knowledge management systems. Various strategies can be used for evaluation, including run time evaluation, collection-based evaluation, and search-based evaluation, and each strategy has its own advantages and disadvantages [71]. Ontology designers and developers, domain experts, and application developers perform evaluations and are responsible for adapting an evaluation strategy that best suits each specific case. In this evaluation process, the quality of a domain ontology is technically verified [44].

The verification of the quality and correctness of a domain ontology is the major objective of this phase. Other objectives of domain ontology evaluation include: (1) to cover each domain in the construction of a domain ontology; (2) to ensure a reliable design structure and the construction or development of the domain ontology in terms of quality; (3) to ensure that the developed ontology is suitable for ontology-based knowledge management systems [72]; and (4) to score and rate the developed domain ontology with help from other ontology designers who should be able to utilize this ontology domain in their ontologies [73]. The evaluation of the developed domain ontology should be performed on the basis of some competency questions and the consistency, content, and maintenance of the ontology [17].

6. Discussion

Most of the studied methodologies (discussed in the literature review section) for the domain ontology construction of knowledge management systems provided analyses of domain, conceptualization, implementation, and evaluation. However, most (with the exceptions of [13–16]) did not provide detailed descriptions of the tasks, activities and methodologies involved in the process of development. A lack of detailed descriptions can lead to difficulties in adapting a methodology for the design of a domain ontology.

Most of the proposed methodologies were not found to support documentation, maintenance, reusability, concept clustering, integration, or localization (see Table 1). The methodologies analyzed and discussed in the literature review section were specifically designed for specific domains.

The design and the development of the ontology of a specific domain has always been a complex task. We also observed that most of the proposed methodologies were based on many different schemes and did not allow for the satisfactory validation of results. Our analysis highlights some gaps in the studied methodologies:

- None of the studied methodologies implemented the clustering of concepts in the design and construction of their domain ontologies;
- According to a comparison based on criteria set for the maturity of an ontology (see the literature review section), all of the methodologies were not completely mature;
- The studied domain ontology construction methodologies did not provide adequate descriptions and details regarding the mechanisms, activities and tasks adapted in the development of their ontologies;
- A few methodologies provided support for localization, reusability and re-engineering.

Many methodologies use conventional approaches to establish their ontology concept, domain analysis, and integration. The implementation of the proposed methodology may demonstrate a better use of RSD-DOC principles in the construction of domain ontologies, which may provide help and support in minimizing the difficulties arising during ontology construction activities such as conceptualization, collaboration between domain experts and ontology engineers, the continuous analysis of a project's position, the involvement of domain experts in ontology construction, the emphasis of the most significant requirements, and real-time response for knowledge transformation. Our proposed methodology also aids the reusability of resources, conceptualization, localization, integration and merging of ontologies, appropriate versioning, concept clustering, and availability of ontology construction and root details.

7. Conclusions

A well-designed and well-organized methodology is needed for the construction of a domain ontology in structured knowledge management because ontology construction is a complex, time-consuming, and tedious job. In this research article, we propose a domain ontology construction methodology that is rooted in OntoWM and automated ontology construction for unstructured text document methodologies. An ontology-based structured knowledge management system is introduced and discussed. A comparative analysis of already proposed methodologies regarding domain ontology construction according to criteria described in the literature review section is provided. The proposed EC-DOC scheme provides an efficient and effective conceptualization mechanism for domains. A major strength of this domain ontology methodology is its ability to cluster similar concepts to increase the effectiveness of results. Another strength is that this methodology can be applied to any domain, as other domain ontology construction methodologies are applied to specific domains. This methodology also has the ability to customize an ontology through various factors such as the domain of interest, ontology size, and ontology complexity. The EC-DOC scheme can also support the approximation of human resources for ontology construction, the reusability of resources, localization, the integration and merging of ontologies, and appropriate versioning control. Finally, a validation of the effectiveness of the domain ontology scheme is presented. In future work, we recommend adopting EC-DOC in the development of forthcoming ontology-based applications and domain ontology construction methods in specific domains such as agriculture, education, tourism, and biomedicine to evaluate its validation and effectiveness in terms of precision, recall and F-measure, as well as to perform a detailed comparative analysis of the scheme with other related schemes based on evaluation metrics. This analysis will provide detailed discussions about the central algorithms used in various schemes.

Author Contributions: Conceptualization, K.M. and M.A.R.; methodology, K.M.; validation, K.M., R.M. and M.A.R.; formal analysis, A.N.; investigation, B.A.; resources, A.N. and B.A.; writing—original draft preparation, K.M.; writing—review and editing, M.A.R.; supervision, R.M.; project administration, R.M. and A.N.; funding acquisition, R.M. and B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Higher Education, Malaysia, under Fundamental Research Grant Scheme (FRGS) No. FRGS/1/2018/ICT04/UMP/02/1 (University reference RDU190112) and Universiti Malaysia Pahang, Faculty of Computing. This work was also funded by the Deanship of Scientific Research (DSR) at Umm Al-Qura University through grant number: 20UQU0074DSR.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahmad, M.N.; Zakaria, N.H.; Sedera, D. Ontology-based knowledge management for enterprise systems. In *Competition, Strategy, and Modern Enterprise Information Systems*; IGI Global: Hershey, PA, USA, 2013; pp. 184–212.
2. Gruber, T.R. A translation approach to portable ontology specifications. *Knowl. Acquis.* **1993**, *5*, 199–220. [CrossRef]
3. Colomb, R.M.; Ahmad, M.N. Merging ontologies requires interlocking institutional worlds. *Appl. Ontol.* **2007**, *2*, 1–12.
4. Colomb, R.M.; Ahmad, M.N. A perdurant ontology for interoperating information systems based on interlocking institutional worlds. *Appl. Ontol.* **2010**, *5*, 47–77. [CrossRef]
5. Tudorache, T. Ontology engineering: Current state, challenges, and future directions. *Semant. Web* **2020**, *11*, 125–138. [CrossRef]
6. Lee, C.S.; Kao, Y.F.; Kuo, Y.H.; Wang, M.H. Automated ontology construction for unstructured text documents. *Data Knowl. Eng.* **2007**, *60*, 547–566. [CrossRef]
7. Raza, M.A.; Raza, B. Comparative analysis of ontology extraction techniques from relational database. *Sci. Int.* **2016**, *28*, 3589–3595.
8. Kultsova, M.; Rudnev, R.; Anikin, A.; Zhukova, I. An ontology-based approach to intelligent support of decision making in waste management. In Proceedings of the 7th International Conference on Information, Intelligence, Systems & Applications (IISA), Chalkidiki, Greece, 13–15 July 2016. [CrossRef]
9. Chou, I.H.; Fan, C.F. An agent-based national radioactive waste management framework design. *Prog. Nucl. Energy* **2010**, *52*, 470–480. [CrossRef]
10. Sanfilippo, E.; Robert, I.M. Formal ontologies in manufacturing. *Appl. Ontol.* **2019**, *14*, 119–125. [CrossRef]
11. Khantong, S.; Ahmad, M.N. An ontology for sharing and managing information in disaster response: In flood response usage scenarios. *J. Data Semant.* **2020**, *9*, 39–52. [CrossRef]
12. Ashburner, M.; Ball, C.A.; Blake, J.A.; Botstein, D.; Butler, H.; Cherry, J.M.; Sherlock, G. Gene ontology: Tool for the unification of biology. *Nat. Genet.* **2000**, *25*, 25–29. [CrossRef]
13. Ahmad, M.N.; Badr, K.B.A.; Salwana, E.; Zakaria, N.H.; Tahar, Z.; Sattar, A. An Ontology for the Waste Management Domain. In Proceedings of the Pacific Asia Conference on Information Systems (PACIS), Yokohama, Japan, 26–30 June 2018; p. 12.
14. Fawei, B.; Pan, J.Z.; Kollingbaum, M.; Wyner, A.Z. A methodology for a criminal law and procedure ontology for legal question answering. In Proceedings of the Joint International Semantic Technology Conference, Awaji City, Japan, 26–28 November 2018; Springer: Cham, Switzerland, 2018; pp. 198–214. [CrossRef]
15. John, S.; Shah, N.; Stewart, C. Towards a Software Centric Approach for Ontology Development: Novel Methodology and its Application. In Proceedings of the 15th International Conference on E-Business Engineering (ICEBE), Xi'an, China, 12–14 October 2018; pp. 139–146. [CrossRef]

16. Li, J.; Alian, S. Design and development of a biocultural ontology for personalized diabetes self-management of American Indians. In Proceedings of the 20th International Conference on E-Health Networking, Applications and Services (Healthcom), Ostrava, Czech Republic, 17–20 September 2018; pp. 1–7. [CrossRef]
17. Suárez-Figueroa, M.C.; Gómez-Pérez, A.; Fernandez-Lopez, M. The NeOn Methodology framework: A scenario-based methodology for ontology development. *Appl. Ontol.* **2015**, *10*, 107–145. [CrossRef]
18. Abdelghany, A.S.; Darwish, N.R.; Hefni, H.A. An agile methodology for ontology development. *Int. J. Intell. Eng. Syst.* **2019**, *12*, 170–181. [CrossRef]
19. Alsanad, A.A.; Chikh, A.; Mirza, A. A domain ontology for software requirements change management in global software development environment. *IEEE Access* **2019**, *7*, 49352–49361. [CrossRef]
20. Jacksi, K. Design and implementation of e-campus ontology with a hybrid software engineering methodology. *Sci. J. Univ. Zakho* **2019**, *7*, 95–100. [CrossRef]
21. Yunianta, A.; Basori, A.H.; Prabuwono, A.S.; Bramantoro, A.; Syamsuddin, I.; Yusof, N.; Alsubhi, K. OntoDI: The methodology for ontology development on data integration. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 160–168. [CrossRef]
22. Brusa, G.; Laura Caliusco, M.; Chiotti, O. Towards ontological engineering: A process for building a domain ontology from scratch in public administration. *Expert Syst.* **2008**, *25*, 484–503. [CrossRef]
23. Trokanas, N.; Koo, L.; Cecelja, F. Towards a Methodology for Reusable Ontology Engineering: Application to the Process Engineering Domain. *Comput. Aided Chem. Eng.* **2018**, *43*, 471–476.
24. Zhang, L.; Chen, H.; Zhang, Y.; Chen, T. Fine construction of HIV protein ontology. In Proceedings of the 3rd International Conference on Information Systems Engineering (ICISE), Kota Bharu, Malaysia, 25–26 February 2018; pp. 82–88. [CrossRef]
25. Saad, A.; Shaharin, S. The methodology for ontology development in lesson plan domain. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*. [CrossRef]
26. De Nicola, A.; Missikoff, M.A. Lightweight methodology for rapid ontology engineering. *Commun. ACM* **2016**, *59*, 79–86. [CrossRef]
27. Dutta, B.; Chatterjee, U.; Madalli, D.P. YAMO: Yet another methodology for large-scale faceted ontology construction. *J. Knowl. Manag.* **2015**, *9*, 6–24. [CrossRef]
28. Bautista-Zambrana, M.R. Methodologies to build ontologies for terminological purposes. *Procedia-Soc. Behav. Sci.* **2015**, *173*, 264–269. [CrossRef]
29. Raza, M.A.; Mokhtar, R.; Raza, S.; Noraziah, A.; Hamid, R.A. A Methodology for Engineering Domain Ontology using Entity Relationship Model. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*. [CrossRef]
30. Sure, Y.; Staab, S.; Studer, R. Ontology engineering methodology. In *Handbook on Ontologies*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 135–152.
31. Sattar, A.; Ahmad, M.N.; Surin, E.S.M.; Mahmood, A.K. An improved methodology for collaborative construction of reusable, localized, and shareable ontology. *IEEE Access* **2021**, *9*, 17463–17484. [CrossRef]
32. Zhang, J.; Zhao, W.; Xie, G.; Chen, H. Ontology-based knowledge management system and application. *Procedia Eng.* **2011**, *15*, 1021–1029. [CrossRef]
33. Raza, M.A.; Mokhtar, R.; Ahmad, N.; Pasha, M.; Pasha, U. A taxonomy and survey of semantic approaches for query expansion. *IEEE Access* **2019**, *7*, 17823–17833. [CrossRef]
34. Negnevitsky, M.; Intelligence, A. *A Guide to Intelligent Systems*; Artificial Intelligence; Addison Wesley: Boston, MA, USA, 2005.
35. Dubois, D.; Prade, H. Fuzzy rules in knowledge-based systems. In *An Introduction to Fuzzy Logic Applications in Intelligent Systems*; Springer: Boston, MA, USA, 1992; pp. 45–68.
36. Kolodner, J.L.; Leake, D.B. A Tutorial Introduction to Case-Based Reasoning. In *Case-Based Reasoning: Experiences, Lessons & Future Directions*; AAAI Press/MIT Press: Menlo Park, CA, USA, 1996.
37. Gallant, S.I. *Neural Network Learning and Expert Systems*; MIT Press: Cambridge, MA, USA, 1993.
38. Pasha, M.; Sattar, A. Comparative analysis of ontology construction approaches from relational databases. In Proceedings of the 4th International Conference on Intelligent Networks and Intelligent Systems, Kuming, China, 1–3 November 2011; IEEE Xplore: New York, NY, USA, 2011; pp. 25–28. [CrossRef]
39. Blomqvist, E.; Presutti, V.; Daga, E.; Gangemi, A. Experimenting with eXtreme design. In Proceedings of the International Conference on Knowledge Engineering and Knowledge Management, Lisbon, Portugal, 11–15 October 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 120–134.
40. Suárez-Figueroa, M.C.; Gómez-Pérez, A.; Fernández-López, M. The NeOn methodology for ontology engineering. In *Ontology Engineering in a Networked World*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 9–34.
41. Gawich, M.; Badr, A.; Hegazy, A.; Ismail, H. A methodology for ontology building. *Int. J. Comput. Appl.* **2012**, *56*, 39–45. [CrossRef]
42. Ubaidillah, S.H.S.A.; Ahmad, N. Fragmentation Techniques for Ideal Performance in Distributed Database—A Survey. *Int. J. Softw. Eng. Comput. Syst.* **2020**, *6*, 18–24. [CrossRef]
43. Noraziah, A.; Fauzi, A.A.C.; Ubaidillah, S.H.S.A.; Alkazemi, B.; Odili, J.B. BVAGQ-AR for Fragmented Database Replication Management. *IEEE Access* **2021**, *9*, 56168–56177. [CrossRef]
44. Suárez-Figueroa, M.C.; Gómez-Pérez, A.; Villazón-Terrazas, B. How to write and use the ontology requirements specification document. In Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", Vilamoura, Portugal, 1–6 November 2009; Springer: Berlin/Heidelberg, Germany, 2012; pp. 966–982. [CrossRef]
45. Uschold, M.; King, M. *Towards a Methodology for Building Ontologies*; Artificial Intelligence Applications Institute, University of Edinburgh: Edinburgh, UK, 1995; pp. 1–13.

46. Rastogi, N.; Verma, P.; Kumar, P. Analyzing ontology editing tools for effective semantic information retrieval. *Int. J. Eng. Sci. Res. Technol.* **2017**, *6*, 40–47.
47. Knublauch, H.; Horridge, M.; Musen, M.A.; Rector, A.L.; Stevens, R.; Drummond, N.; Wang, H. The Protege OWL Experience. In Proceedings of the OWLED, Galway, Ireland, 11–12 November 2005.
48. Slimani, T. Ontology development: A comparing study on tools, languages and formalisms. *Indian J. Sci. Technol.* **2015**, *8*, 1–12. [CrossRef]
49. Internet World Statistics. Internet World Users by Language. Available online: https://www.internetworldstats.com/stats7.htm (accessed on 4 October 2022).
50. Hau, E.; Aparício, M. Software internationalization and localization in web-based ERP. In Proceedings of the 26th Annual ACM International Conference on Design of Communication, Lisbon Portugal, 22–24 September 2008; pp. 175–180. [CrossRef]
51. Mahmood, K.; Rahmah, M.; Fauziah, Z.; Azhar, A.N.; Noraziah, A.; Norshita, M.N. The Impact of Exponent Variable on the Performance and Effectiveness of FCM Algorithm for Ontology Construction in Structured Knowledge Management. In Proceedings of the International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), Pekan, Malaysia, 24–26 August 2021; pp. 673–678. [CrossRef]
52. Xu, H.; Sun, H. Application of Rough Concept Lattice Model in Construction of Ontology and Semantic Annotation in Semantic Web of Things. *Sci. Program.* **2022**. [CrossRef]
53. Li, X.F. Domain Concept Extraction and Ontology Visualization. *J. Phys. Conf. Ser.* **2021**, *1994*, 12–14. [CrossRef]
54. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; ACM Press: New York, NY, USA, 1999; p. 463.
55. Zhou, S.; Li, D.; Zhang, Z.; Ping, R. A new membership scaling fuzzy C-means clustering algorithm. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 2810–2818. [CrossRef]
56. Zhang, L.; Zhong, W.; Zhong, C.; Lu, W.; Liu, X.; Pedrycz, W. Fuzzy c-means clustering based on dual expression between cluster prototypes and reconstructed data. *Int. J. Approx. Reason.* **2017**, *90*, 389–410. [CrossRef]
57. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [CrossRef]
58. Zhang, L.; Luo, M.; Liu, J.; Li, Z.; Zheng, Q. Diverse fuzzy c-means for image clustering. *Pattern Recognit. Lett.* **2020**, *130*, 275–283. [CrossRef]
59. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
60. Askari, S. Fuzzy C-Means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: Review and development. *Expert Syst. Appl.* **2021**, *165*, 113856. [CrossRef]
61. Döring, C.; Lesot, M.J.; Kruse, R. Data analysis with fuzzy clustering methods. *Comput. Stat. Data Anal.* **2006**, *51*, 192–214. [CrossRef]
62. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [CrossRef]
63. McJones, P. Eachmovie collaborative filtering data set. *DEC Syst. Res. Cent.* **1997**, 249–257.
64. Raza, M.A.; Rahmah, M.; Qaswar, F.; Shafqat, A.; Rauf, M.; Raza, S. User Interest Driven Semantic Query for Effective Web Search. *Int. J. Adv. Soft Compu. Appl.* **2021**, *13*, 11–24.
65. Cantador, I.; Fernández, M.; Castells, P. Improving ontology recommendation and reuse in WebCORE by collaborative assessments. In Proceedings of the Workshop Social Collaborative Construction of Structured Knowledge, 16th International World Wide Web Conference, Banff, AB, Canada, 8 May 2007; pp. 1–11.
66. Raad, J.; Cruz, C. A survey on ontology evaluation methods. In Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Lisbon, Portugal, 12–14 November 2015. [CrossRef]
67. Looser, D.; Ma, H.; Schewe, K.D. Using formal concept analysis for ontology maintenance in human resource recruitment. In Proceedings of the Ninth Asia-Pacific Conference on Conceptual Modelling, Adelaide, Australia, 29 January–1 February 2013; Volume 143, pp. 61–68.
68. de Almeida Falbo, R. SABiO: Systematic Approach for Building Ontologies. In Proceedings of the CEUR Workshop Proceedings, Leipzig, Germany, 2 September 2014.
69. Garijo, D. WIDOCO: A wizard for documenting ontologies. In Proceedings of the International Semantic Web Conference, Vienna, Austria, 21–25 October 2017; Springer: Cham, Switzerland, 2017; pp. 94–102.
70. Pinto, H.S.; Martins, J.P. A methodology for ontology integration. In Proceedings of the 1st International Conference on Knowledge Capture, Victoria, BC, Canada, 22–23 October 2001; pp. 131–138. [CrossRef]
71. Horridge, M.; Gonçalves, R.S.; Nyulas, C.I.; Tudorache, T.; Musen, M.A. Webprotégé: A cloud-based ontology editor. In Proceedings of the 2019 World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 686–689. [CrossRef]
72. Shaukat, S.; Shaukat, A.; Shahzad, K.; Daud, A. Using TREC for developing semantic information retrieval benchmark for Urdu. *Inf. Process. Manag.* **2022**, *59*, 102939. [CrossRef]
73. Fernández, M.; Overbeeke, C.; Sabou, M.; Motta, E. What makes a good ontology? A case-study in fine-grained knowledge reuse. In Proceedings of the Asian Semantic Web Conference, Shanghai, China, 6–9 December 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 61–75. [CrossRef]