Design of A Machine Learning Model for Automatic Generation of Domain-Specific Ontologies

Sivarama Krishnan[a], R Guruvayur[b], R.Suchithra[c]
[ab]Department of Computer Science, Jain University, Karnataka, India
[c]Department of Computer Science, Jain University, Karnataka, India

Abstract

Presently, the development of an Ontology for a particular domain is influenced by a knowledge engineer's intervention and extent of knowledge in that domain. In general, ontologies are created by domain experts who use domain-specific approaches to generate taxonomies from different knowledge sources. Therefore, due to the manual aspects of Ontology creation, validation and updation, and the absence of a comprehensive and automated standard methodology for Ontology Engineering, there is significantly low and ineffective adoption of Ontologies in emerging AI applications. This paper describes on-going research to utilise Machine Learning Algorithms for domain-specific automatic generation and continuous updation of Ontologies. The proposed approach involves the development of four novel algorithms for automatic generation of ontology which offer cost- and time-efficient ways of automatically creating and maintaining Ontologies.bstract should be times new roman with 9 fount single spacing. The main focused of Watermarking is developing and introducing new techniques for watermark embedding and detection. Experimental results show that the embedded watermark is transparent and quite robust in face of various watermark images at high compression ratios and provides good results in terms of imperceptibility.

KEYWORDS – Machine Learning, Domain- Specific Ontology, Algorithms, Automatic Ontology Generation

INTRODUCTION

An ontology, in computer science, is "a formal, explicit specification of a shared conceptualization" [1]. An ontology can be employed for domain modelling and to support entity analysis [2].A closer examination of the elements of the definition reveals that a conceptualization is basically a depiction of a particular domain with regard to theories and associations, which facilitates analysis of the domain. Moreover, a concept is not simply a term or tag of an entity. Instead, it offers a description of the entity which permits recognition of whether or not an object or occurrence is an illustration of the entity being considered. Also, a concept specifies the entity's associations with other entities in the setting and the rules that are applicable for all occurrences of a specific entity. Explicit specification indicates that the definition of concepts is performed through a group of statements that are comprehensible to both machines and humans. Thus, an ontology has an implication and can be utilised independent of the system for which it was developed. In practice, formal implies that a logic-based language is used to encode the specification thus allowing automatic inquiry and evolution of novel information. A hierarchy of concepts comprises the nucleus of a formal ontology. The final aspect of the definition, shared, signifies that the principal stimulus for the definition of ontology is to enable sharing and reuse of knowledge [3]. The typical elements of an ontology are individuals (or instances), concepts (or classes), attributes, and relationships [4].

Ontologies are of great significance in the Semantic Web as they are used for knowledge representation. On the other hand, ontology engineers are greatly helped by Ontology learning (OL) to develop their own ontologies for a specific domain. OL encompasses different activities, namely: “ontology import, extraction, pruning, refinement, and evaluation” [5]. Ontologies also play a significant role in emerging technology trends such as Artificial Intelligence, Natural Language Processing, Internet of Things, and Machine-to-Machine (M2M) integration. Moreover, they have a crucial role in user engagement and dialogue management platforms, such as Chatbots, etc.

I. Types of Ontologies

Researchers have provided different types of ontologies. For instance, Guarino[6] suggests four types of ontologies: top-level, domain, task, and application. Top-level ontologies are domain independent indicate common sense knowledge or very general concepts. On the other hand, domain ontologies contain vocabulary associated with a broad domain (e.g., physics, medicine, etc.). Task ontologies indicate vocabulary associated with a common activity or task (e.g., selling). Application ontologies signify knowledge that depends on both task and domain. This is typically the concentration of both task and domain ontologies.

Van Heijst, Schreiber, and Wielinga[7]provided another categorisation of ontologies based on the conceptualisation, for example, on the extent and type of structure (information, knowledge modelling and terminological) and on the subject (application, domain, generic, and representation) (Figure 1).

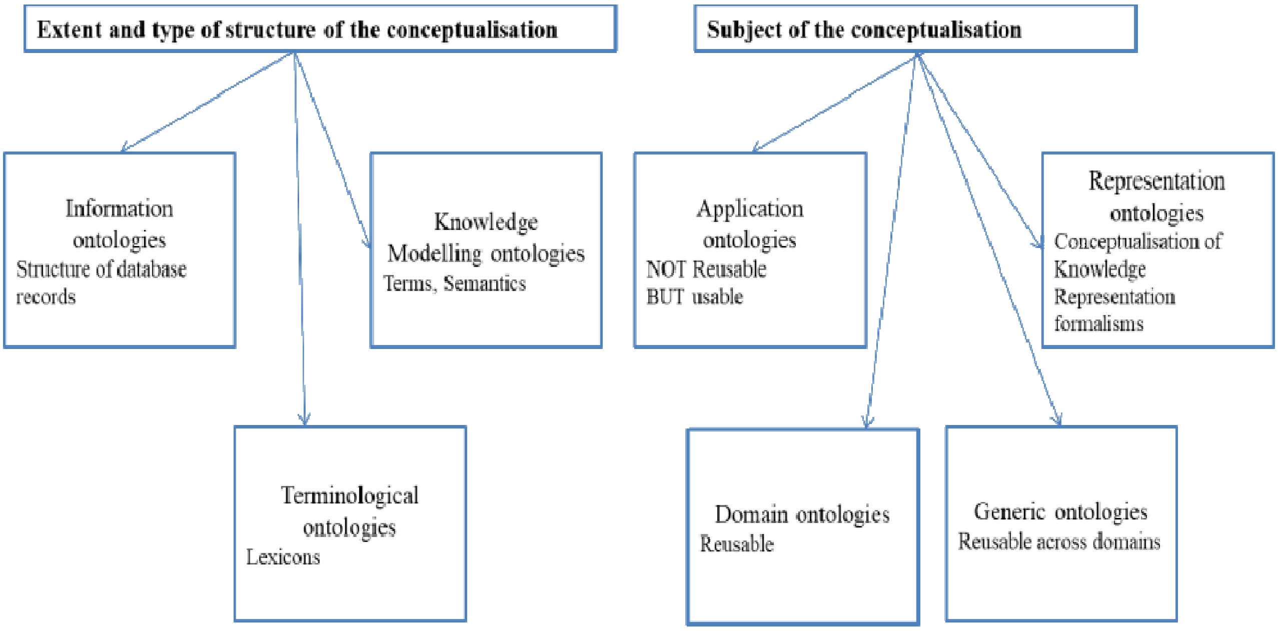
Extent and type of structure of the conceptualisation
Subject of the conceptualisation
Information ontologies
Structure of database records
Knowledge Modelling ontologies
Terms, Semantics
Application ontologies
NOT Reusable
BUT usable
Representation ontologies
Conceptualisation of Knowledge Representation formalisms
Terminological ontologies
Lexicons
Domain ontologies
Reusable
Generic ontologies
Reusable across domains

Figure 1. Different types of ontologies[7]

Another classification of ontologies is by the level of formality: informal, formal, and semi-formal [4]. Further, ontologies can be classified on the basis of purpose as classification and descriptive ontologies. The document stores in classification ontologies are immense and grow with time on the Internet. Suitable hierarchies are used to classify the documents on the basis of relations between terms. Classification ontologies facilitate the searching for a document using author, title, and subject. On the other hand, descriptive ontologies can be employed to describe all existing entities in the real-world. These ontologies are utilised to describe DERA (i.e., Domain, Entity, Relation, and Attribute) [8].

2.1.Importance of Domain Specific Ontologies

A domain specific ontological study gives clarification to knowledge structure. In any domain, the ontology is the core of the system of representing information for that domain. Without the existence of the ontology, or the conceptualizations which form the basis of knowledge, a vocabulary of knowledge representation cannot exist. Therefore, the first stage of creating an effective system for representation of knowledge and vocabulary is an effective ontological study of the domain. A weak analysis will result in knowledge bases that are incoherent[9].

For building a language of knowledge representation based on analysis, an association of terms along with ontological concepts and relations, and creation of a syntaxfor knowledge encoding pertaining to the concepts and relations is necessary. This language of knowledge representation can be shared with those having similar requirements for representing knowledge in that domain, hence negating the requirement for repeating the process of knowledge analysis. Ontology sharing can hence create the basis for domain specific information-representing languages. When compared with the earlier generation of such languages (say, KL-1), these languages have rich content besides having a many terms embodying complex constituent of the domain[9].

Domain specific ontologies find use in the Semantic Web, Artificial Intelligence, Systems Engineering, Biomedical Informatics, Software Engineering, Enterprise Bookmarking, Library Science, and Information Architecture in a type of representation of knowledge about the domain or a part of it. Creating domain ontology is also key to defining and using a framework of enterprise architecture[10]. There are 4 groups of ontologies: static, dynamic, social and intentional[11]. A static ontology elaborates items in existence; their attributes and the relationship with them. Dynamic ontology explains the domain as states and their transitions including processes. Social ontology describes social scenarios, permanent structures in an organization or changing networks of independencies and alliances. Intentional ontology includes the domain of agents, things wanted, believed in, proved, disproved, and discussed about[12].

2.2.Advantages of ontologies

Rani and colleagues [4] summarised the advantages of creating an ontology as gleaned from various sources (e.g., [13]–[16]):

- An ontology provides a common vocabulary for a domain;
- Ontology metadata enables easy merging and expansion of ontologies.
- Content is defined unambiguously by an ontology.
- Domain knowledge can be separated from operational knowledge.
- An ontology enables re-use of its content.
- An ontology offers ordering and structuring of its content.
- Rules can be added to ontologies to infer new knowledge.
- Ontologies integrate content from heterogeneous sources.

- Ontologies provide successful information distribution, and storage, and recovery of information (text corpus).
- Content sharing in an ontology takes place through agent interactions.

2.3.Challenges in Ontology Creation

However, several challenges underlie the creation and updation of ontologies. For example, domain-specific ontologies are characterised by the need for the manual intervention of domain experts. Moreover, the restrictions imposed by current technology adoptions reduce the feasibility of automatic creation and updation of ontologies.

With regard to text processing and Knowledge acquisitions, Ontology engineering and modelling engines encounter the uphill task of dealing with vast Unstructured texts, Multiple senses of word, Unstructured text, Ambiguity in language in question, Multiple senses of a word, Multiple parts of speech, Lack of closed domain of lexical categories, Noisy texts, Requirement of very large training text sets for Machine learning algorithms, etc.

The theoretical formalism supported by the existing mechanisms of ontology creation does not support data integration dueto heterogeneous data formats from various sources, absence of relevancy and context sensitivity of the data. With the current approaches, keywords extracted from various sources can be utilized to infer the corresponding domain. However, the same keyword may lead to a different context or domain (e.g., Balance). This issue can be addressed through application of machine learning algorithms into Ontology Engineering to determine the correct context and hence domain. Manual generation of machine learning from a predefined concept hierarchy is a difficult and tedious task that often requires expert interpretation. Consequently, automatic generation of concept hierarchy and machine learning based Ontology Engineering from heterogeneous datasets for a domain is highly desirable.

II. PROBLEM STATEMENT

Ontologies are an important part of the Semantic Web and numerous emerging AI applications. Using an ontology, both the client and the framework can interact with each other in a machine-to-machine environment with the common understanding of the domain. Despite the fact that ontology has been proposed as a vital means for representing the real world knowledge for the construction of database designs, most ontology developments are not performed automatically. However, the underlying challenges in creating and updating these domain-specific ontologies such as need for manual intervention of domain experts and the restrictions imposed by the current technology adoptions make the tasks of automatic creation and updation of ontologies less feasible.

Moreover, data frameworks progressively rely upon ontology to structure information in a machine readable configuration and guarantee fast performance. Some existing ontologies, for example, WordNet[17], Dublin Core [18], and Cyc[19] are accessible, yet most applications require a particular domain ontology to depict concepts and relations in that domain. Automatic Ontology Generation is a challenging task due to the absence of structured database or domain taxonomy. Ontology development generally relies upon domain experts, but this is lengthy and costly. While numerous ontology tools, such as OntoEdit[20], Protege-2000 [21] and Ontolingua[22], exist and are available to assist ontology development, the involvement of domain experts is still necessitated. Therefore, the automatic generation of ontology gains significance in Semantic Web and emerging AI applications.

The fully automatic derivation of ontologies from Web sources without human review is to date a challenging research issue as is the inability of enterprises to grow their competitive advantage by uncovering hidden knowledge that is too complex for human cognition. In this connection, various problems exist as mentioned below:

1. Knowledge acquisition using semantic web technologies: The available information is diverse in terms of format, language, domain, quality, accuracy, context, etc. However, there is no standard approach to normalize and harmonize the knowledge representation across domains for the purpose of building relevant ontologies.

2. Knowledge Analysis and Automatic attribute extraction: No common ontology learning framework is presently available to extract concepts, attributes, values and relations automatically across domains. This is due to the lack of common ontology models and related tools and methods for ontology learning.

3. Knowledge Representation and Building relationships between entities: There is presently no robust mechanism to automatically draw relationships and / or roles between attributes to build domain ontology.

4. Knowledge updation (Validate and update Ontology): Before the ontology is updated, entities and relationships (triples) need to be validated for accuracy, logical consistency and persistency. Today, this process is dependent on experts and extremely difficult to automate.

3.1. Domain specific ontologies

Domain specific ontologies are created for the following purposes [23]:

1. Sharing common understanding of structures of knowledge among software agents or users
2. Enabling reuse of knowledge of the domain
3. Making explicit domain assumptions
4. Separating operational and domain knowledge
5. Analysing knowledge of the domain

3.1.1. Challenges in Building Domain Specific Ontologies

The following are the challenges in building domain specific ontologies:

1. Insufficient coverage for limited domains: Typically, the coverage for domains having lesser web presence is lesser than that of domains that are more popular.
2. Relational identification: Generally, the extraction relations are not spelled out in advance. Hence, for domain specific ontologies, identifying relations needs expertise in the domain.
3. Resolving entities: The issue of identification and grouping/linking various manifestations/occurrences of the same real-world item is a difficult task.
4. Disambiguation of Entities: A phrase or word may imply more than one entity. Entity disambiguation involves association of the phrase/word with the most relevant entity.
5. Problem of temporal knowledge base: There are facts which vary with time, hence mapping the phrase/word with the relevant entity can be an issue.

6. Extracting values: Ontologies are generally represented in the form f triplets, i.e. <Entity, Relationship, Entity>. The system is required to learn all possible formats of entity/relationship to enrich the ontology.
7. Confidence of facts: There are many disputable facts which depend on the source of information. Hence, it is often difficult to locate a unique entity if there is a conflict.

3.2. General ontology generation

Despite the existence of large-scale ontologies, ontology engineers, still, are required for constructing the knowledge base and ontology for a given domain, and to update and maintain the ontology for relevancy and currency. Ontologies constructed manually are time-consuming, error-prone, and labour-intensive. Also, any major delay in the updation of ontologies that results in currency issues would go to hinder the development of the ontologies.

Ontology learning is gathering interest as an offshoot of ontological engineering owing to the sporadic increase of web information and the advanced approaches shared by the data retrieval, ML, and AI communities. Most current ontologies have been manually generated. Ontology generation in this way has been the norm undertaken by a majority of ontology engineers. An ontology could be generated in different ways, depending on the situation. It could be created from zero, from available ontologies, from a corpus of data, or from a combination both methods. Several levels of automation can be deployed to generate ontologies, i.e. fully manual, partially-automated, or fully automated. Currently, a fully automated technique functions well only for light ontologies, that too, in a few situations only. Typical approaches or generating ontologies are bottom-up (from specific to general) and top-down (from general to specific).

In general, the following steps are involved in automatic domain-specific ontology extraction (Figure2).

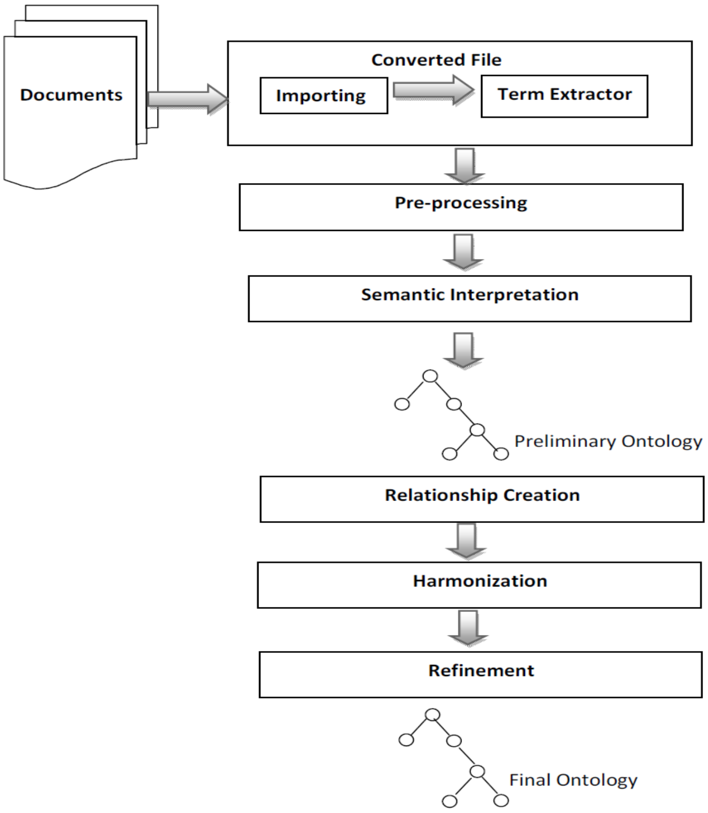

Figure 2. Automatic domain-specific ontology extraction

Table 1 describes the steps in detail.

Table -1 Automatic domain-specific ontology extraction

Step	Description
Preprocessing	Here, the documents are made ready for the extraction. Several sub-phases, described below, come comprise this phase. 1) Converting formats: Documents conversion to a more appropriate one (say, XML) takes place. 2) Stemming: Here, terms in the analysed document are reduced to their root form using a combination of various algorithms. 3) Tagging parts of speech: Here, terms are marked in the document (also multi-word terms) in a text matching with a specific part of speech (e.g. nouns, adjectives, verbs, etc.). 4) Listing stopwords: Here, unnecessary terms not relevant to domain are removed (e.g. conjunctions, articles, and verbs). 5) Identifying synonymy and extracting terminology
Creating the Ontology	A basic draft version of the ontology is created based on primitive terms having simple and compound concepts.
Concept and Relationship Mapping	Various statistical and ML algorithms for data mining are implemented for identifying the concepts and relationships in the created ontology. There are three major types of ML

Step	Description
	algorithms: unsupervised, supervised and semi-supervised.
Harmonizing	This is considered optional and required when a user wishes to harmonize the ontology extracted with the knowledge bases available. Two or more ontologies are merged in one unique ontology for improving the available knowledge base.
Refining and Validating	Here, the target ontology is tuned and its evolving nature supported. The adaptation and refining of the ontology, keeping in view user requirements, plays an important role in the development of the specific application and also its continued development. The pruning of unrelated concepts from the ontology extracted is a major step.

3.3. Automatic Generation of Ontology

Ontologies can be built manually by knowledge engineers and domain experts. However, this can result in long and cumbersome stages of development, growing into a knowledge acquisition bottleneck [5]. Accordingly, an important area of research is ontology learning. Ontology learning signifies the group of approaches and tools utilized for developing an ontology from the basics, enhancing or modifying a present ontology in a semi-automatic manner through the use of numerous sources of information and knowledge [24].

Figure 3 depicts a classification of methods to ontology learning from various perspectives [25].

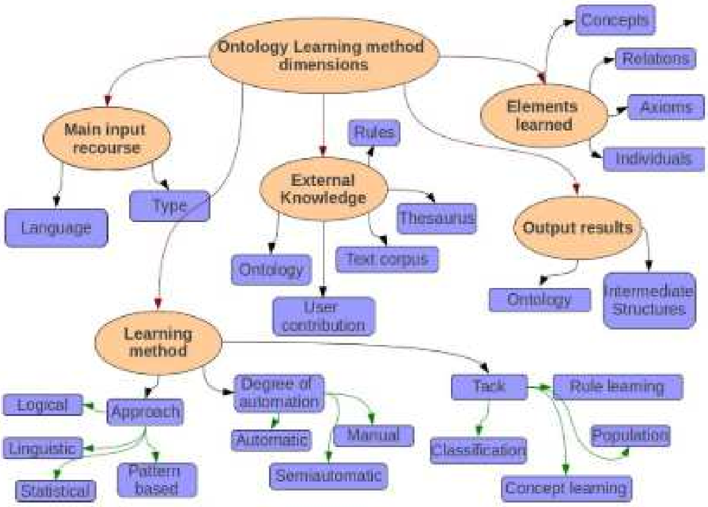

Figure 3. Classification of approaches to ontology learning [25]

The lifecycle of ontology development has been considered differently by different researchers. For example, Maedche and Staab[5] submitted that the ontology learning process (Figure 4) was composed of: "ontology import, extraction, pruning, refinement, and evaluation." This framework combines machine learning with knowledge acquisition.

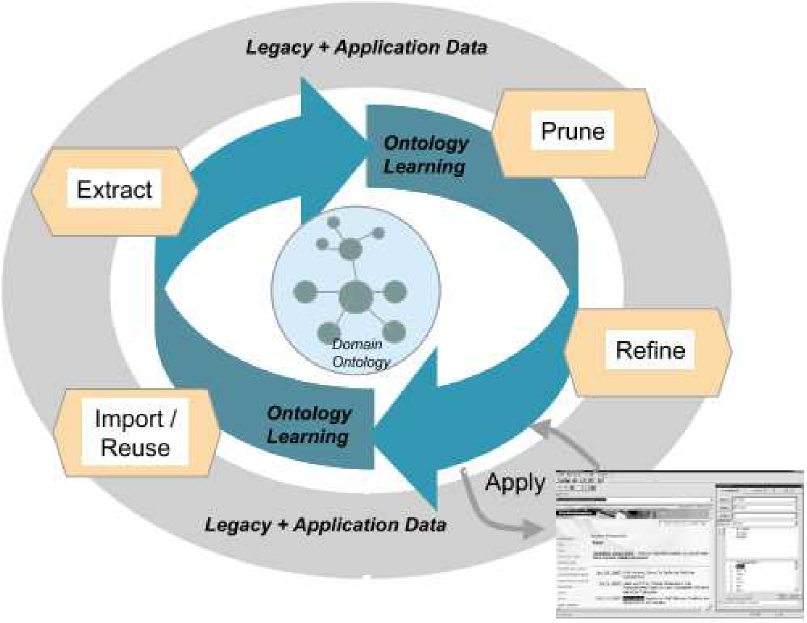

Figure 4. Ontology learning process [5]

On the other hand, Weng and colleagues [26] placed emphasis on the methods of extraction, taking four categories into account namely, “dictionary-based, text clustering, association rules, and knowledge base.” Further, Buitelaar and colleagues [27] described an ontology-building process based on the “named cake model.” This model regards ontology building as an overlay, that is, where every layer parallels a task that permits the obtaining of an ontology element (Figure 5). Following a bottom-up approach, the layers are organized as terms, synonyms, concepts, concept hierarchy, relations, relation hierarchy, and rules.

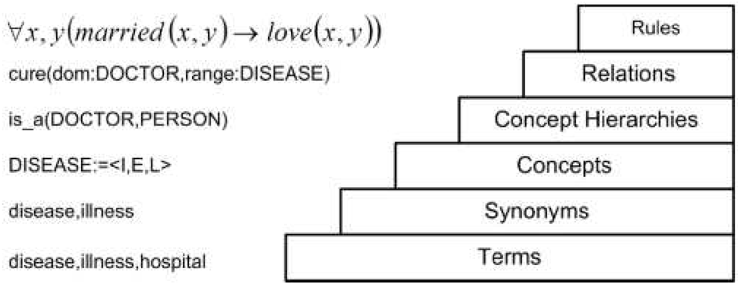

Figure 5. Ontology Learning Layers[27]

Wang and colleagues [28] classified the approaches that can perform these tasks into four groups based on lexical-syntactic patterns, information extraction, machine learning, and co-occurrence analysis. Natural Language Processing (NLP) techniques are generally utilized to recognize appropriate terms and their associations. A processing phase is required by text, where tasks such as, 1) plain text extraction, 2) text splitting into sentences, 3) stopwords elimination, 4) sentence tagging, and 5) sentence parsing are used.

Fernandez &Ponnusamy[29] proposed an effective approach for automatic ontology generation using behaviour of students while using the Internet as the underlying basis. They used the individualised feedback data of students to discover their learning behaviour. They developed a novel fuzzy based ontology approach by combining gravitational search optimization algorithm with fuzzy rules for automatic ontology generation.

Bhatia & Dixit [30] observed that hidden web pages could be automatically and efficiently extracted using an ontology and a database that archives semantic information about objects and their associations.They proposed and implemented a novel technique for the creation of ontology using form pages.

Rani et al. [4] explored two topic modelling algorithms (LSI & SVD and Mr.LDA) with the objective of determining the statistical association between a document and the terms it contains to build, with minimum human involvement, a topic ontology and an ontology graph. The effectiveness of the proposed approach was demonstrated through experimental results and was in terms of building richer topic- specific knowledge and semantic retrieval.

Balakrishna et al.[31] presented a comprehensive and enhanced process to extract deep semantic information automatically from text resources and speedily develop semantically-rich domain ontologies while limiting the human involvement to a minimum. They also presented evaluation outcomes for the intelligence and financial ontology libraries created semi-automatically by their suggested methodologies using textual resources freely-available from the Web.

3.3.1. Domain specific ontology development

Domain specific ontology development is a fast growing technique for representing knowledge, and subsequently utilizing it. Huge amounts of data exist as tables, textual documents, spreadsheets, etc. However, this data is typically underutilized due to the fact that modern data processing methods are not applied to it. In countries like India, decision taking still is based primarily on human intervention. Corroboration by facts using existing data is still lacking. It is a task to extract terms and establish relationships from existing texts with the use of minimal domain specific knowledge for the creation of an ontology[32].

Extraction of relationships in ontological generation and population, which is the inclusion of new ideas to the ontology, is being researched for the last decade. This task presents many challenges because different types of methods are required, even for extracting the same relation from the text. As a result, extraction of relationships has received a lot of attention in research work of the last decade [33]. Relation extraction methodologies, generally, fall into three categories:

Knowledge-based techniques: Such methods use rules and patterns created by experts for extracting relationships from domain-specific data. A major shortcoming of such knowledge-based techniques is that, being extremely domain-specific, their applicability in other areas is limited. But then, such methods are effective and yield good results for well-defined input data.

Supervised techniques: Such methods utilize machine-learning (ML) approaches and training examples to extract relationships from texts specific to domains. Depending on the requirement, various algorithms are available in this category, such as bootstrapping, kernels, logistic regression, augmented parsing methods, etc.

Self-supervised techniques: These approaches are distinguished by their capabilities of pattern extraction in establishing relationships automatically. Open Information

Extraction and distant learning are some examples in this category. The former identifies entity sets and patterns (possible relationships) that occur between such entities in the domain text, the latter uses certain knowledge bases to identify patterns for establishing relationships.

3.4. Ontologies and Knowledge Graphs

Knowledge graphs (KGs) are graph structured knowledge bases (KBs) that “store factual information in form of relationships between entities” [34]. Numerous knowledge graphs have been developed in the recent past each containing several million nodes and several billion edges. Examples include, YAGO [35], DBpedia[36], Nell [37], Freebase [38], and the Google Knowledge Graph [39].

Information is modeled in knowledge graphs in the shape of entities and the associations among them. This type of representation of relational knowledge has been long utilized in logic and artificial intelligence [40], for instance, in semantic networks [41] and frames [42]. A more recent use has been in the Semantic Web community with the objective of generating a “web of data” that is machine-readable [43]. However, this vision has not yet been completely achieved [34].

Ehrlinger and Wöß[44] draw attention to the increased emphasis in knowledge graph research since 2012. This has resulted in several descriptions and definitions of the concept (Table 2).

Table -2Selected knowledge graph definitions

Author(s)	Definition
Paulheim[45]	“A knowledge graph (i) mainly describes real world entities and their interrelations, organized in a graph, (ii) defines possible classes and relations of entities in a schema, (iii) allows for potentially interrelating arbitrary entities with each other and (iv) covers various topical domains.”
Kroetsch andWeikum [46]Journal of Web Semantics (Special Issue on Knowledge Graphs)	“Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities.”
Blumauer[47]	“Knowledge graphs could be envisaged as a network of all kind things which are relevant to a specific domain or to an organization. They are not limited to abstract concepts and relations but can also contain instances of things like documents and datasets.”
Färber et al. [48]	“We define a Knowledge Graph as an RDF graph. An RDF graph consists of a set of RDF triples where each RDF triple (s, p, o) is an ordered set of the following RDF terms: a subject s ∈ U ∪ B, a predicate p ∈ U, and an object U ∪ B ∪ L. An RDF term is either a URI u ∈ U, a blank node b ∈ B, or a literal l ∈ L.”
Pujara et al. [49]	“[...] systems exist, [...], which use a variety of techniques to extract new knowledge, in the form of facts, from the web. These facts are interrelated, and hence, recently this extracted knowledge has been referred to as a knowledge graph.”

Ehrlinger and Wöß[44] highlighted issues about present research associated with knowledge graphs. Specifically, two basic issues are reported. First, Google's Knowledge Graph blog entry is referred to as if it offers a suitable explanation for creating a knowledge graph. Second, there is interchangeable use of the terms knowledge graph and knowledge base. This second issue results in the ambiguous belief that the terms knowledge graph and knowledge base are synonymous, which leads to confusion as knowledge base itself is used as a synonym for ontology. For example, the creators of both Knowledge Vault and Google's Knowledge Graph have referred to these as large-scale knowledge bases [50]. YAGO is a further example, which as its name indicates is an ontology, but is called a knowledge base [39, 50] and also a knowledge graph [51,52].

Likewise, employees of Yahoo [53] do not clearly differentiate between knowledge graph, knowledge base, and ontology. They assert that their knowledge base is constructed by associating new entities, associations, and information with their general ontology. Thus, partial, variable, and probably incorrect information is transformed into a powerful, combined, established knowledge graph. This indicates that their awareness of a knowledge graph relates to the prepared knowledge base that is their ontology population (e.g., instances).

Thus, it is evident that the terms must be clarified explicitly to be distinguishable. Akerkar and Sajja[54] submitted that a system that is knowledge-based utilizes artificial intelligence (AI) to resolve problems and is composed of two components: an inference engine and a knowledge base. In contrast, as stated before, an ontology is a "formal, explicit specification of a shared conceptualization" [55] that is typified by high semantic expressiveness necessitated for enhanced complexity [56].

Ontological representations permit knowledge to be semantically modelled, and thus are typically utilized as knowledge bases in AI applications. Usage of an ontology as knowledge base assists validation of semantic associations and drawing of inferences from known facts [56]. Ehrlinger and Wöß[44] emphasized explicitly that an ontology is not different from a knowledge base. Ontologies can comprise not only classes and properties but also instances.

Size has been frequently mentioned as a critical feature of knowledge graphs. Consequently, a knowledge graph could be pronounced to be an extremely large ontology. Nevertheless, other contributors have highlighted the superiority of knowledge graphs to ontologies as they offer extra features [47]. Therefore, the dissimilarity between a knowledge graph and an ontology could be understood either as a question of quantity or of extensive needs. The second understanding results in the belief that a "knowledge graph is a knowledge-based system that contains a knowledge base and a reasoning engine" [44]. Placing emphasis on present automatically created "knowledge graphs," additional crucial features can be identified: "collection, extraction, and integration of information from external sources extends a pure knowledge-based system with the concept of integration systems" [44].

Figure 6 depicts the merging of these assumptions, which results in an abstract architecture for a knowledge graph. A knowledge graph can thus be defined as follows:

> A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge[44].

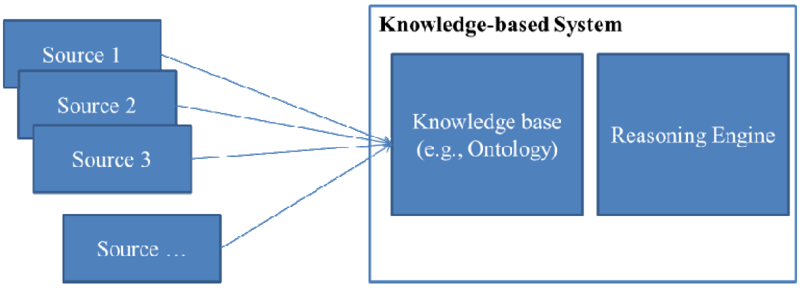

Figure 6. Knowledge Graph Architecture[44]

This definition corresponds to the assumption regarding the superiority and complexity of a knowledge graph in contrast to a knowledge base (e.g., ontology) as it uses a reasoning engine to create new knowledge and assimilates single or multiple sources of information.

3.5. Machine Learning Methods and Ontologies

A fundamental research area in artificial intelligence is machine learning. The preliminary motivation was to fit a computer system with an individual's capacity to learn to achieve artificial intelligence. A system without the capacity to learn cannot be considered to be intelligent. Tian et al. [57]conceded that the "generic form of machine learning is a knowledge acquisition and manipulation process mimicking the brain (p. 1).

Different types of algorithms are utilised in machine learning [58]. These are summarised in Table 3.

Table -3. Types of machine learning [58]

Sl #	Type of machine learning	Description
1	Supervised learning	The algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behaviour of) a function which maps a vector into one of several classes by looking at several input- output examples of the function.
2	Unsupervised learning	Models a set of inputs: labelled examples are not available.
3	Semi- supervised learning	Combines both labelled and unlabelled examples to generate an appropriate function or classifier.
4	Reinforcement learning	The algorithm learns a policy of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm.
5	Transduction	Similar to supervised learning, but does not explicitly construct a function: instead, tries to predict new outputs based on training inputs, training outputs, and new inputs.
6	Learning to learn	The algorithm learns its own inductive bias based on previous experience.

Machine Learning is a discipline that can contribute in a major way in improving ontology creation and learning. A classical explanation of machine learning (ML) is as follows: an experience E teaches the system in accordance with a performance indicator P if it enhances its performance measured by P after going through the experience E. ML, traditionally, is based on developing inductive pattern extracting methods from the provided data. It goes beyond statistical research for model building and works with much more complex algorithms to obtain precise and larger models for the data, which may not be comprehensible to humans any more. ML finds wide use in areas involving prediction, adaptation, and pattern recognition and extraction.

The relation between ontologies and types of ML algorithms used for learning of these ontologies is presented in Figure7. The classification method in the figure is based on update speed and ontology size.

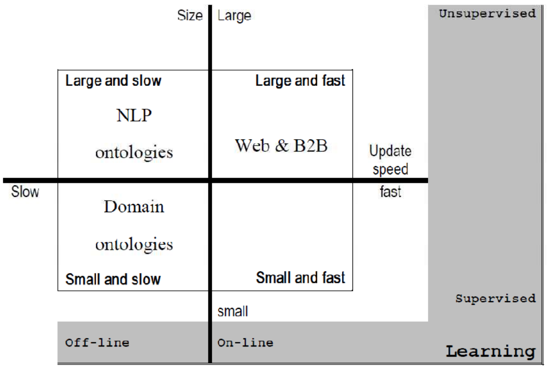

Figure 7. Ontology vs. ML type[58]

Table 4 presents the comparison of the kinds of ML algorithms with types of ontologies. It is seen that each kind of ontology can be learned by a specific ML technique.

Table -4 Ontology type vs. ML algorithm used[58]

Type of learner	**Stage of Learning**	**Problem-solving stage**	**Type of Ontology**	**Comments**
Supervised off-line	Slow; needs significant training data; highly tuned resulting knowledge base	Fast	Small and slow (domain ontologies)	Widespread current research
Unsupervised off-line	Slow; mostly training data is not needed; suitable for clustering tasks	Fast	Large and slow (NLP ontologies)	Widespread current research
Supervised on-line	Fast; needs significant training data to be available online; resulting knowledge base is moderately good	Relatively slow		Widespread current research
Unsupervised on-line	Fast; no need for training data; quality of results not known	Not known	The only case for large Web ontologies	Insufficient research

3.5.1. Algorithms for Machine Learning

This section briefly summarises different algorithms for machine learning.

Supervised ML algorithms

These algorithms predict on a given set of samples. Supervised ML algorithms seek patterns inside the value labels given to the data points. Some commonly used such algorithms are presented below.

Naïve Bayes Classifiers

It is impossible to manually classify a document, web page, email or some such lengthy text note. Naïve Bayes Classifier ML algorithms help for such applications. A classifier allocates an element value of a population from an available category. It is a commonly used ML model grouped by similarities which is based on the Bayes Theorem. It finds applications in sentiment Analysis, classification of articles on technology, document categorization, sports, entertainment, etc.

Support Vector ML algorithm

This is used in regression or classification problems in which datasets teach classes to the SVM so that it classifies any new given data. Data is classified into different classes by estimating a hyperplane that groups the training data into classes. Among the many linear hyperplanes, SVM algorithms try to keep the spacing between the involved classes at a maximum. The probability of good generalization with new data increases when the line maximizing the class distance is located. These algorithms are popular in applications like stock market forecasting.

Decision Tree ML algorithm

It is a graphical representation utilizing branching methods to illustrate all possible conditional decision outcomes. The internal nodes serve as tests on attributes, tree branches represent the test outcomes, and leaf nodes serve as specific class labels (decisions made after calculating all attributes). The rules for classification are indicated by the route from root to the leaf node. These algorithms find application in finance (option pricing), Remote sensing for pattern recognition, banking loan default detection, etc.

Artificial Neural Networks (ANNs)

ANNs are created using many elements having inputs of magnitude much greater computational elements having typical architectures. The artificial neurons are connected in categories that use mathematical modelling to process information utilizing a connectionist computation approach. The ANNs keep the neurons sensitive for item storage. They can be utilized for storage of many cases containing vectors of high dimensions, and the storage is tolerant to distortion.

Unsupervised ML algorithms

In these algorithms, labels and data points have no association. These ML algorithms arrange the data in clustered groups for describing their structures and make complicated data appear manageable and organized for study. Unsupervised ML activity involves the derivation of a function which defines the structure of unclassified or uncategorized (unlabeled) data. As the illustrative data given to the learning algorithm is not labeled, no forthright method exists for evaluation of the accuracy of the algorithm-generated structure. This feature is a characteristic that distinguishes unsupervised from supervised learning.

Clustering involves grouping a set of items in a way that items in the same cluster have more similarity to one another than to items in other clusters. Cluster analysis finds wide use in market research in analyzing multivariate data obtained from surveys. Cluster analysis is utilized by market researchers to partition consumers into various market segments for understanding the relationships among various classes of customers, both existing and potential. Also, product positioning studies, developing new products, pattern recognition, and choosing test markets are some typical investigations aided by cluster analysis.

Automatic Knowledge Extraction

For automatic knowledge acquisition in inductive learning, Akgöbek et al.[59] presented an algorithm, REX-1. Instead of the decision tree approach, this algorithm makes use of direct rule extraction and employs a group of examples to generate broad rules.

Pham &Dimov[60] presented a new algorithm which extracted IF-THEN rules from examples. An efficient rule searching method is used by the algorithm along with a simple metric to assess rule generality and accuracy.

Attributes extraction

Liang et al.[61] designed a framework to automatically extract attributes from query interfaces. Each attribute was extended into a candidate attribute expressed by a hierarchy tree and described the semantic relation of the attributes. They performed their experiments in the real-world domain. The outcomes of the study demonstrated the validity of the query translation framework.

An et al. [62] developed a three-stage algorithm to automatically extract the attributes for different Web data sources. For a given set of Web data sources, the inner identifiers are used to obtain the Programmer Viewpoint Attributes (PVAs). Next, the free text within the query interface is used to obtain the User Viewpoint Attributes (UVAs). Lastly, the an ontology (WordNet) is utilised to determine the final attributes (FAs) of each Web data source based on PVAs and UVAs. It must be noted that the extraction of PVA and UVA, and determination of FA are all accomplished in an automatic manner.

K-means clustering algorithm

K-means clustering refers to a non-hierarchical method of arranging items various clusters/groups [63]. A user can define the number of clusters/groups based on the use case and data under consideration. The K-means algorithm “is an algorithm for putting N data points in an I-dimensional space into K clusters. Each cluster is parameterized by a vector $m^{(k)}$ called its mean” [64]. Clustering of data points in the K-means algorithm is achieved by decreasing the aggregate of the sum of squared distances connecting the data points and their centroids. The central point to a set of data points in the data set is referred to as a centroid.

There are several approaches to select the initial centroid. However, in many cases it is performed through the use of random allocation. The K-means algorithm functions as follows [64]:

Initialization. Set K means $\{\mathrm{m}^{(\mathrm{k})}\}$to random values.

Assignment step. Each data point n is assigned to the nearest mean. The guess for the cluster $\mathrm{k}^{(\mathrm{n})}$ is denoted as the point $\mathrm{x}^{(\mathrm{n})}$ belongs to by $\hat{\mathrm{k}}^{(\mathrm{n})}$.

$$\hat{\mathrm{k}}^{(\mathrm{n})} = \underset{\mathrm{k}}{\mathrm{argmin}} \{\mathrm{d}(\mathrm{m}^{(\mathrm{k})}, \mathrm{x}^{(\mathrm{n})})\}$$

Update step. The model parameters (i.e., the means) are adapted to correspond to the sample means of the data points they are concerned with.

$$\mathrm{m}^{(\mathrm{k})} = \frac{\sum_{\mathrm{n}} \mathrm{r}_{\mathrm{k}}^{(\mathrm{n})} \mathrm{x}^{(\mathrm{n})}}{\mathrm{R}^{(\mathrm{k})}}$$

Where $\mathrm{R}^{(\mathrm{k})}$ is the total responsibility of mean k,

$$\mathrm{R}^{(\mathrm{k})} = \sum_{\mathrm{n}} \mathrm{r}_{\mathrm{k}}^{(\mathrm{n})}$$

The **assignment and update steps are repeated** until the assignments do not change.

Ortega et al. [65] summarised the advantages and disadvantages of the K-means algorithm. Two advantages were evident. First, "The process, which is called "k-means", appears to give partitions which are reasonably efficient in the sense of within-class variance… corroborated to some extent by mathematical analysis and practical computational experience… Also, the k-means procedure is easily programmed and is computationally economical, so that it is feasible to process very large samples on a digital computer" [66]. Similarly, Barbakhand Fyfe [67] described the benefits of K-means stating that the algorithm is "one of first which a data analyst will use to investigate a new data set because it is algorithmically simple, relatively robust and gives 'good enough' answers over a wide variety of data sets" (p. 1).

Moreover, Ortega and colleagues [65] summarised the limitations of the algorithm. These are:

- The sensitivity of the algorithm sensitivity to preliminary conditions, i.e., number of partitions, initial centroids, etc. (p. 89)
- The algorithm's convergence to a local rather than a global optimum (p. 87)
- The algorithm's efficiency (p. 88)
- The algorithm's sensitivity to outliers and noise (p. 89)

The application of the algorithm is limited to numerical variables due its definition of "means" (p. 89)

Maximization-Expectation Algorithms

The Expectation and Maximization (EM) algorithm indicates the application of alternating maximization to the likelihood function for a mixture of distributions model. EM is performed at each iteration first through expectation and then by maximization. Expectation indicates the finding of posterior probabilities for observations to belong to individual clusters given parameters of the mixture and individual density functions. On the other hand, maximization indicates finding parameters of the mixture maximizing the likelihood function given posterior probabilities [68].

Welling and Kurihara[69] introduced a new class of “maximization expectation” (ME) algorithms where they maximized over hidden variables while marginalizing over random parameters. In other words, they reversed the roles of maximization and expectation in the classical EM algorithm.

A probabilistic model, $p(x, z, \theta)$ is considered in the ME algorithm, where x and z are respectively observed and hidden random variables, and $\theta$is a parameter set, also assumed to be random [69].

Bayesian K-Means Algorithm

Welling and Kurihara[69] discussed a top-down “Bayesian k-means” algorithm as an example of the maximization-expectation algorithm. Shouman et al.[70] described the integrated k-means clustering and naïve Bayes algorithm:

K-means clustering

1. Identify the attributes that will be used in clustering
2. Identify the number of clusters
3. Apply one of the initial centroid selection methods (Inlier method, Outlier method, Range method, Random attribute method, Random row method)
4. Using Euclidean distance, assign each of the data instances to the cluster which it is nearest to the centroid
5. Recalculate the centroids of the k clusters
6. Repeat steps 4 and 5 until there is no change in the centroids.

Naïve Bayes:

1. For each cluster

a. Calculate prior probability for the target attribute

b. Calculate conditional probability for the remaining attributes

3.5.2. Machine Learning Based Ontology

Greenbaum et al. [71] utilised contextual autocomplete to facilitate data entry by nurses regarding the reason a patient visited the Emergency Department. They demonstrated a method that encapsulates structured data for almost all patients. They concluded that enhanced structured data capture, ontology usage compliance, and data quality resulted from the implementation of a contextual autocomplete system.

Nyberg[72], in her master’s thesis, explored the manner in which the contents of documents can be used to automatically classify them. She created an RDF schema for representing documents, sentences and words to prepare the data for the machine learning analysis. Nyberg [72] concluded that the classification accuracy of the model is enhanced by the addition of ontology information. Lukyanenko et al.[73] proposed that conceptual modelling can be utilised to overcome some of the challenges of using machine learning effectively.

III. CONCEPTUAL FRAMEWORK FOR THE PROPOSED METHODOLOGY

The proposed approach uses four different kinds of algorithms for automatic knowledge acquisition, automatic attribute extraction, automatic relationship between entities, and automatic entity validation. The data collected from various sources is first stored in a data lake in their native format before being sent, in the form of tables, to the Intelligent Knowledge Acquisition (IKA) algorithm. Data lakes are present fashion in the field of data warehousing and analytics. Essentially, a data lake is a location where data are accumulated in an unprocessed format and can be utilised for analyses [74]. The IKA algorithm uses three techniques to process the data: normalization, harmonization, and decision tree construction. During normalization, the graphical data tables are organized logically and redundant information is also eliminated. After the completion of the normalization process, the graphical data tables are sent to the harmonization process. During harmonization, the possibility of combining data from heterogeneous sources is created. After completing the pre-processing steps, the graphical data tables are further converted to .arff file format for injecting knowledge to the ontology. The decision tree is constructed through its attributes and relationship values from the .arfffiles.

Next, the obtained decision tree is sent to the Bayesian K-means algorithm which clusters and classifies the decision tree. Initially, the clustering of the decision tree is based on the mean value of the parent nodes. Subsequently, the clustered data is sent for classification using the Naïve Bayes algorithm. The Naïve Bayes algorithm is chosen to classify the clustered data as it provides higher accuracy rate than other existing classification algorithms. The classified result depicts the perfect classification of attribute for the ontology.

The classified attributes are then injected to the Automated Entity Relationship (AER) algorithm. The algorithm obtained from the entity relationship based on the classified attributes. The AER algorithm uses a mapping technique to combine the entity relationship from the classified attribute. Finally, the obtained entity-relationship model is sent to the Automated Entity Relationship Validation (AERV) algorithm. The AERV checks whether or not the obtained model violates modelling rules. Also, it checks the correctness of the syntax, whether positional conventions were followed by the model, and finally whether or not the assertion conditions are met. The proposed validation techniques are highly useful for further knowledge updating process. Finally, the outcome is developed into an efficient ontology.

The proposed framework is depicted in Figure 8.

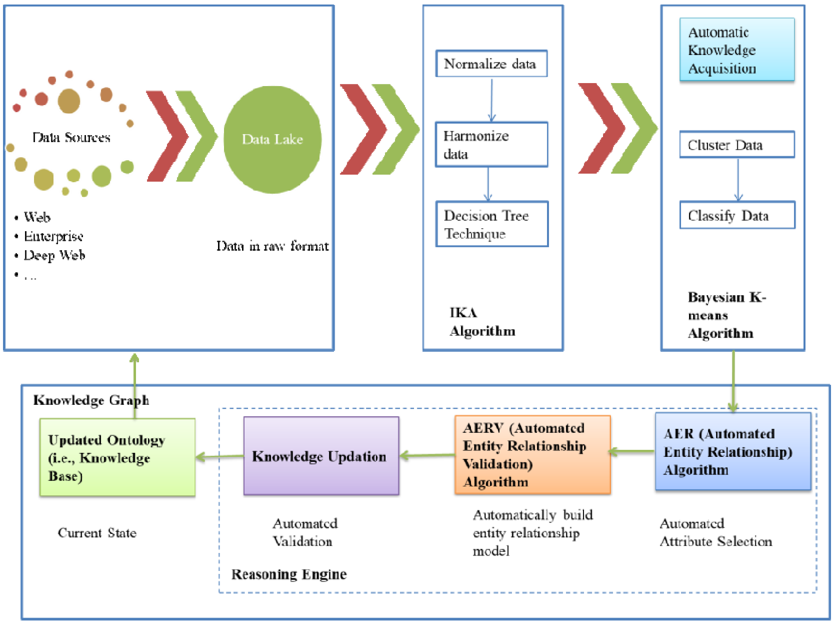
Data Sources
Data Lake
• Web
• Enterprise
• Deep Web
• ...
Data in raw format
Normalize data
Harmonize data
Decision Tree Technique
IKA Algorithm
Automatic Knowledge Acquisition
Cluster Data
Classify Data
Bayesian K-means Algorithm
Knowledge Graph
Updated Ontology (i.e., Knowledge Base)
Current State
Knowledge Updation
Automated Validation
AERV (Automated Entity Relationship Validation) Algorithm
Automatically build entity relationship model
AER (Automated Entity Relationship) Algorithm
Automated Attribute Selection
Reasoning Engine

Figure 8. Proposed Framework Diagram

The outcomes anticipated from the use of the methodology are summarised in Table 5.

Table -5: Anticipated outcomes

Techniques	**Knowledge Acquisition**	**Knowledge Analysis**	**Knowledge Representation**	**Knowledge Updation**
Existing	Knowledge Base accuracy (35%)	Classification accuracy (80%)	No automatic knowledge representation (20%)	Knowledge updation accuracy (45%)
Proposed	Knowledge Base accuracy (98%)	Classification accuracy (90%)	Automatic knowledge representation (80%)	Knowledge updation accuracy (90%)

IV. Conclusion

This paper described the activities undertaken in the early stages of a study which resulted in a preliminary version of a framework for automatically generating a domain-specific ontology based on existing machine learning algorithms.

Several advantages are evident in the proposed approach. The most fundamental is the use of four algorithms for the various stages of the ontology generation. Secondly, the perfect classification of attributes for the ontology is anticipated from the methodology. Thirdly, the validation techniques are proposed are anticipated to be of great use for subsequent knowledge updating processes.

References

[1] T.R. Gruber. “Toward principles for the design of ontologies used for knowledge sharing,” International Journal of Human-Computer Studies, 43(5-6), 907-928, 1995.

[2] Y. Chao and P. Zhang. One General Approach For Analysing Compositional Structure Of Terms In Biomedical Field. Master's Thesis, Jönköping University, 2013.

[3] M. Zemenova. Exploiting ontologies and higher order knowledge in relational data mining. (Doctoral Thesis, Czech Technical University in Prague), 2012.

[4] M. Rani, A.K. Dhar and O.P. Vyas. "Semi-automatic terminology ontology learning based on topic modelling". Engineering Applications of Artificial Intelligence, 63, 108-125, 2017.

[5] Maedche and S. Staab. "Learning ontologies for the semantic web," in Proceedings of the Second International Conference on Semantic Web-Volume 40 (pp. 51-60). CEUR-WS. Org, 2001.

[6] N. Guarino. "Formal ontology in information systems," In Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy (Vol. 46). IOS press, 1998.

[7] G. Van Heijst, A.T. Schreiber and B.J. Wielinga. "Using explicit ontologies in KBS development". International Journal of Human-Computer Studies, 46(2-3), 183-292, 1997.

[8] F. Giunchiglia, B. Duttaand and V. Maltese. From knowledge organization to knowledge representation. Technical Report, Università di Trento, 2013.

[9] B. Chandrasekaran, J.R. Josephson and V.R. Benjamins. "What are ontologies, and why do we need them?" IEEE Intelligent Systems and their applications, 14(1), 20-26, 1999.

[10] D. H. Deshmukh and S.D. Deshpande. "A Review of Ontology based information retrieval". International Journal, 1(7), 2013.

[11] J. Yan, D. Hu and L. Zhao. "An ontology-based approach for bank stress testing". In System Sciences (HICSS), 2013 46th Hawaii International Conference on (pp. 3407-3415). IEEE, 2013.

[12] Jurisica, J. Mylopoulos and E. Yu. "Ontologies for knowledge management: an information systems perspective," Knowledge and Information Systems, 6(4), 380-401, 2004.

[13] S. Cakula and A.B.M. Salem. "E-learning developing using ontological engineering". WSEAS transactions on Information Science and Applications, 1(1), 14-25, 2013.

[14] D. Dzemydiene and L. Tankeleviciene. "On the development of domain ontology for distance learning course". In 20th International Conference EURO Mini Conference: Continuous Optimization and Knowledge-Based Technologies, EurOPT 2008. Vilnius Gediminas Technical University, 2008.

[15] N. Jekjantuk. E-learning content management: an ontology-based approach (Doctoral dissertation, SIU THE SOT-MSIT 2006-03), 2006.

[16] J. Jovanović, D. Gašević, C. Knight and G. Richards. "Ontologies for effective use of context in e-learning settings," Journal of Educational Technology & Society, 10(3), 2007.

[17] D.B. Lenat. "CYC: A large-scale investment in knowledge infrastructure," Communications of the ACM, 38(11), 33-38, 1995.

[18] S. Weibel. The Dublin Core: a simple content description model for electronic resources. Bulletin of the Association for Information Science and Technology, 24(1), 9-11, 1997.

[19] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K.J. Miller. “Introduction to WordNet: An on-line lexical database,” International Journal of Lexicography, 3(4), 235-244, 1990.

[20] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke. “OntoEdit: Collaborative ontology development for the semantic web”. In International Semantic Web Conference (pp. 221-235). Springer, Berlin, Heidelberg, 2002.

[21] N.F. Noy, M. Sintek, S. Decker, M. Crubézy, R.W. Fergerson and M.A. Musen. “Creating semantic web contents with protege-2000,” IEEE intelligent systems, 16(2), 60-71, 2001.

[22] Farquhar, R. Fikes and J. Rice. “The ontolingua server: A tool for collaborative ontology construction,” International journal of human-computer studies, 46(6), 707-727, 1997.

[23] Singh and P. Anand. “State of art in ontology development tools”. International Journal, 2(7), 2013.

[24] D.S. Ruenes. Domain ontology learning from the web. (PhD thesis, UniversitatPolitecnica de Catalunya) 2007.

[25] M. Shamsfard and A.A. Barforoush. Learning ontologies from natural language texts. International journal of human-computer studies, 60(1), 17-63, 2004.

[26] S.S. Weng, H.J. Tsai, S.C. Liu and C.H. Hsu. “Ontology construction for information classification”. Expert Systems with Applications, 31(1), 1-12, 2006.

[27] P. Buitelaar, P. Cimiano and B. Magnini. “Ontology learning from text: An overview”. In Buitelaar, P., Cimiano, P., &Magnini, B. (Eds.), Ontology learning from text: Methods, evaluation and applications/ Frontiers in artificial intelligence and application, volume 123. IOS Press, Amsterdam, 2005.

[28] W. Wang, P.M. Barnaghi and A. Bargiela. “Probabilistic topic models for learning terminological ontologies”. IEEE Transactions on Knowledge and Data Engineering, 22(7), 1028-1040, 2010.

[29] F. Fernandez and R. Ponnusamy. “An Efficient Technique for the Automatic Generation of Ontology Based on Students Behavior Analysis Using Optimal Fuzzy-GSO Algorithm,” Journal of Computational and Theoretical Nanoscience, 14(9), 4488-4495, 2017.

[30] K.K. Bhatia and A. Dixit. “Automatic Generation of Ontology for Extracting Hidden Web Pages”. In Big Data Analytics (pp. 127-139). Springer, Singapore, 2018.

[31] M. Balakrishna, D. I. Moldovan, M. Tatu, M., and M. Olteanu. “Semi-Automatic Domain Ontology Creation from Text Resources”. In LREC, 2010.

[32] N. Kaushik and N. Chatterjee. “Automatic relationship extraction from agricultural text for ontology construction,” Information Processing in Agriculture, 5(1), 60-73, 2008.

[33] N. Konstantinova. “Review of relation extraction methods: What is new out there?” In International Conference on Analysis of Images, Social Networks and Texts_x000D_(pp. 15-28). Springer, Cham, 2014.

[34] M. Nickel, K. Murphy, V. Tresp and E. Gabrilovich. “A review of relational machine learning for knowledge graphs,” Proceedings of the IEEE, 104(1), 11-33, 2016.

[35] F.M. Suchanek, G. Kasneci and G. Weikum. “Yago: a core of semantic knowledge”. In Proceedings of the 16th international conference on World Wide Web (pp. 697-706). ACM, 2007.

[36] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, J., Cyganiak, R., & Ives, Z. “Dbpedia: A nucleus for a web of open data”. In The Semantic Web (pp. 722-735). Springer, Berlin, Heidelberg, 2007.

[37] Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. HruschkaJr and T.M. Mitchell. “Toward an architecture for never-ending language learning”. In AAAI, Vol. 5, p. 3, 2010.

[38] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor. “Freebase: a collaboratively created graph database for structuring human knowledge”. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 1247-1250). AcM, 2008.

[39] Singhal. Introducing the Knowledge Graph: things, not strings. Retrieved from http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html, 2012.

[40] R. Davis, H. Shrobe and P. Szolovits. “What is a knowledge representation?” AI magazine, 14(1), 17, 1993.

[41] J.F. Sowa. “Semantic networks”. Encyclopedia of Cognitive Science, 2006.

[42] M. Minsky. A framework for representing knowledge.MIT-AI Laboratory Memo 306, 1974.

[43] T. Berners-Lee, J. Hendler and O. Lassila. The Semantic Web. Retrieved from http://www. scientificamerican.com/article/the-semantic-web, 2001.

[44] L. Ehrlinger and W. Wöß. “Towards a Definition of Knowledge Graphs,” In SEMANTiCS (Posters, Demos, SuCCESS), 2016.

[45] H. Paulheim. “Knowledge graph refinement: A survey of approaches and evaluation methods,” Semantic Web, 8(3), 489-508, 2017.

[46] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy ... and W. Zhang. “Knowledge vault: A web-scale approach to probabilistic knowledge fusion”. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 601-610). ACM, 2014.

[47] Blumauer. From Taxonomies over Ontologies to Knowledge Graphs. Retrieved from https://semantic-web.com/2014/07/15/from-taxonomies-over-ontologies-to-knowledge-graphs/, 2014.

[48] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger. “Linked data quality of dbpedia, freebase, opencyc, wikidata, and YAGO,” Semantic Web, (Preprint), 1-53, 2016.

[49] J. Pujara, H. Miao, L. Getoor, and W. Cohen. “Knowledge graph identification,” in International Semantic Web Conference (pp. 542-557). Springer, Berlin, Heidelberg, 2013.

[50] M. Kroetsch and G. Weikum. Journal of Web Semantics: Special Issue on Knowledge Graphs. Retrieved from http://www.websemanticsjournal.org/index.php/ps/ announcement/view/19, 2016.

[51] M. Färber and A. Rettinger. “A Statistical Comparison of Current Knowledge Bases,” In SEMANTiCS (Posters & Demos) (pp. 18-21), 2015.

[52] Tonon, M. Catasta, R. Prokofyev, G. Demartini, K. Aberer and P. Cudre-Mauroux. “Contextualized ranking of entity types based on knowledge

graphs". Web Semantics: Science, Services and Agents on the World Wide Web, 37, 170-183, 2016.

[53] R. Blanco, B.B. Cambazoglu, P. Mika and N. Torzec. "Entity recommendations in web search". In International Semantic Web Conference (pp. 33-48). Springer, Berlin, Heidelberg, 2013.

[54] R. Akerkar, and P. Sajja. Knowledge-based systems, 2010.

[55] R. Studer, V.R. Benjamins and D. Fensel. "Knowledge engineering: principles and methods". Data and knowledge engineering, 25(1), 161-198, 1998.

[56] Feilmayr and W. Wöß, W. "An analysis of ontologies and their success factors for application to business," Data & Knowledge Engineering, 101, 1-23, 2016.

[57] Y. Tian, Y. Wang, M.L. Gavrilova and G. Ruhe. "A formal knowledge representation system (FKRS) for the intelligent knowledge base of a cognitive learning engine". International Journal of Software Science and Computational Intelligence (IJSSCI), 3, 1-17, 2011.

[58] Omelayenko. Machine learning for ontology learning. Report for the course, International Jyvaskyla Summer School, Finland, 2000.

[59] Ö. Akgöbek, Y. S. Aydin, E. Öztemel, and M. S. Aksoy, "A new algorithm for automatic knowledge acquisition in inductive learning," Knowledge-Based Systems, vol. 19, no. 6, pp. 388–395, 2006.

[60] D.T. Pham and S.S. Dimov. "An efficient algorithm for automatic knowledge acquisition," Pattern Recognition, 30(7), 1137-1143, 1997.

[61] H. Liang, F. Ren, W. Zuo and F. He. "Ontology based automatic attributes extracting and queries translating for deep web," JSW, 5(7), 713-720, 2010.

[62] Y.J. An, J. Geller, Y.T. Wu, and S. Chun, S. "Semantic deep web: automatic attribute extraction from the deep web data sources". In Proceedings of the 2007 ACM symposium on Applied computing (pp. 1667-1672). ACM, 2007.

[63] S. Gopalani and R. Arora. "Comparing apache spark and map reduce with performance analysis using k-means," International Journal of Computer Applications, 113(1), 2015.

[64] D.J. MacKay. Information theory, inference and learning algorithms. Cambridge University Press, 2003.

[65] J. P. Ortega, M. Del, R.B. Rojas, and M.J. Somodevilla. "Research issues on k-means algorithm: An experimental trial using Matlab," in CEUR Workshop Proceedings: Semantic Web and New Technologies, 2009.

[66] J. MacQueen. "Some methods for classification and analysis of multivariate observations," in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297), 1967.

[67] W. Barbakh and C. Fyfe. "Local vs global interactions in clustering algorithms: Advances over K-means". International Journal of Knowledge-based and Intelligent Engineering Systems, 12(2), 83-99, 2008.

[68] B. Mirkin. Clustering: A Data Recovery Approach (2nd Edition). Chapman and Hall/CRC, 2013.

[69] M. Welling, and K. Kurihara. "Bayesian K-means as a "maximization-expectation" algorithm". In Proceedings of the 2006 SIAM international conference on data mining (pp. 474-478). Society for Industrial and Applied Mathematics, 2006.

[70] M. Shouman, T. Turner and R. Stocker. “Integrating Naive Bayes and K-means clustering with different initial centroid selection methods in the diagnosis of heart disease patients”. CS & IT-CSCP, 125-137, 2012.

[71] N.R. Greenbaum, Y. Jernite, Y. Halpern, S. Calder, L.A. Nathanson, D.A. Sontag and S. Horng. “Contextual Autocomplete: A Novel User Interface Using Machine Learning to Improve Ontology Usage and Structured Data Capture for Presenting Problems in the Emergency Department,” bioRxiv, 127092, 2017.

[72] K. Nyberg. Document classification using machine learning and ontologies. (Master's Thesis, Aalto University, School of Science), 2011.

[73] R. Lukyanenko, J. Parsons and V.C. Storey. “Modeling Matters: Can Conceptual Modeling Support Machine Learning?” AIS SIGSAND, 1-12, 2018.

[74] Hejmalíček. Hadoop as an Extension of the Enterprise Data Warehouse. (Master’s thesis, Masaryk University, Faculty of Informatics), 2015.