Mathematical and Software Engineering, Vol. 7, No. 1-2 (2021), 7-28
Varεpsilon Ltd, http://varepsilon.com

A Survey on Ontology Operations Techniques

Mario Maroun

Department of Computer Science,
Varna Free University "Chernorizets Hrabar", Varna, Bulgaria

Abstract

Ontologies are increasingly seen as key factors in enabling interoperability between heterogeneous systems and semantic web applications and are emerging as representative techniques for overlapping complementary context domains. A single ontology is no longer sufficient to support the tasks predicted by a distributed environment such as the Semantic Web; several ontologies for many applications are necessary. Different ontological tools have different representations of data and concept operations with respect to their input. Its functions and informational structures also differ depending on its tools and processes. Currently very few investigation documents provide an in-depth discussion of these technologies and their applications. In this article, we discuss various sophisticated ontological tools with their various internal processes and algorithms. Mapping, aligning, or merging ontologies creates an identification layer that allows different applications to access the resulting ontology and then share its information, of course, while preserving the semantics it contains. For integrating large ontologies, automatic matches become an essential solution. However, the large ontology matching process presents high spatial and temporal complexities. Therefore, for a tool to efficiently and accurately match this large ontology within limited computing resources, it must have techniques that can significantly reduce the high spatio-temporal complexities associated with the existential matching process. These processes provide an important basis for many other processes such as translation, reconciliation, coordination and negotiation between ontologies.

Keywords: tools; algorithms; ontology operations; ontologies; techniques; ontology mapping; ontology aligning; ontology merging.

1 Introduction

Ontology and ontology matching techniques are a growing trend because ontology probably offers the most interesting opportunity to encode the meaning of information. The last decades have seen a period of intense research in this field. Nowadays, far from collapsing, activity seems to be increasing and new positions are constantly being posted, where the field of ontology mapping is approached.

The basic process that is performed when processing ontology is "mapping" which interprets a set of correspondences between similar concepts across two (or more) ontologies from the same application domain or similar domains. These mappings support two other interdependent processes, ontology alignment and merging, as well as many others such as translation, leveling, coordination, expression, negotiation, etc. These terminological and commercial differences make it difficult to identify problem areas and understand the solutions they offer.

These difficulties are due to the lack of comprehensive research, normative terminology, hidden assumptions, undiscovered technical details, and lack of evaluation metrics [1]. The process of alignment takes two ontological approaches and produces a set of links between concepts that correspond to one another linguistically. These matches are called “Mappings” [2] [3]. Ontology Merging, as its name suggests, combines concepts that are compatible with each other into a single concept, and then produces a single ontology from two ontologies. Semantic matches described in the designations may refer to equivalence (is-a), specialization and/or generalization (part of) relationships, as they may refer to other senses. There are many recent works and reports in the ontology literature on mapping, aligning, and integrating ontologies. These works adopt different approaches to identifying similar concepts (or mapping discovery). Several tools have been developed to match ontologies [4] [5]. However, with the increasing spread of ontology, challenges have arisen that need to address ontology matching tools to establish high-quality correspondence between ontologies in limited computing resources [6].

In this article, we focus on reviewing the tools and techniques developed for mapping, aligning, merging and matching, exploring the methods and describing the different criteria adopted by each process. This paper is organized as follows: in Section 2 we present the relevant completed work, in Section 3 general concepts of ontology operations. Next, we present a sample of the tools and techniques for operations found in the literature. The paper is ended by a conclusion and some perspectives in Section 4.

2 Related Work

Namyoun et al. have reported about the tools, systems, and related work of ontology mapping [7]. Three ontology mapping categories are explained as

i. Mapping between local ontologies
ii. Mapping between an integrated global ontology and local ontologies, and
iii. Mapping on ontology merging and alignment.

In their work, a comparison has been done on the evaluation criteria, input requirements, level of user interaction, type and the content of output, and in the five dimensions called structural, lexical, domain, instance-based knowledge, and type of result. Natalya has given a brief survey of the approaches to semantic integration developed by researchers in the ontology community [8]. They focused on distinguishing ontological research from other related fields. They also discussed different techniques for finding matches between ontologies and declarative ways of presenting and using these correspondences in different semantic integration tasks. Akrifi Katifori et al. introduced and categorized these techniques and their characteristics in order to support the choice of methods and to encourage future research in the field of ontology visualization [9]. Matteo Cristiani et al. provided a framework that analyses methodologies by comparing them using a set of general criteria [10]. A classification based on bottom-up or top-down directions was obtained to construct an ontology. The obtained classification is also claimed to be useful not only for theoretical purposes, but also in the practice of disseminating ontology in information systems, presented by Elena Simperi et al. [11]. An article based on empirical evidence and real results from methodologies and tools currently used to perform ontology reuse. They analyzed the most prominent case studies of reusing ontology in e-health and e-recruitment. Yannis research focuses on the current state of the art in ontological mapping [12]. Ravi et al. analyze the tools for mediating ontology [13]. They review modern methods, frameworks, techniques and tools. J L Hong et al. focused on current tools and their applications, but did not analyze existing tools [14]. Siham et al. conducted a research on mapping and the use of alignment and merging operations with mapping, but not other specific ontological operations. Sean M. Falconer and Natalia F. Noyet al. describes the techniques for matching ontology and their application, process technology and current tools, and their use. In this research, we provide a comparative overview of current algorithm-based ontology technologies and systems.

3 Ontologies

3.1 Role of Ontologies

Ontologies can be used in different areas of studies and can be used in different areas of computer science like security, intelligent systems, etc. In security, they can be used to secure the networks and to substitute with other methods of securing the networks and computer systems [15–19]. They can even be utilized for optimizing the life of the networks to further improve the methods discussed in [20]. As described earlier, the ontology of integration tasks can also be used in order to describe and define the semantics of information sources, explain the semantics of these resources, and clarify their context. Ontology can also be extended for other applications in other projects we describe as follows:

- Content Explanation: Ontology is presented as an explicit specification of visualization. They can be used to define linguistically relevant information concepts.
- Single Ontology Approaches: Use a universal ontology that provides a common vocabulary for specification of semantics. Single ontology

methods can be used to solve integration problems that have the same view of the field.

- Multiple Ontology Approaches: In this method, each information source has its own ontology, it is not necessary to engage in a global ontology. Each source ontology is developed regardless of other sources and their ontologies.
- Hybrid approaches: Similar to hybrid methods, the semantics of each source are determined by its ontology. Nevertheless, the local ontology is built from a common global vocabulary in order to be comparable to each other.

3.2 Ontology Operations

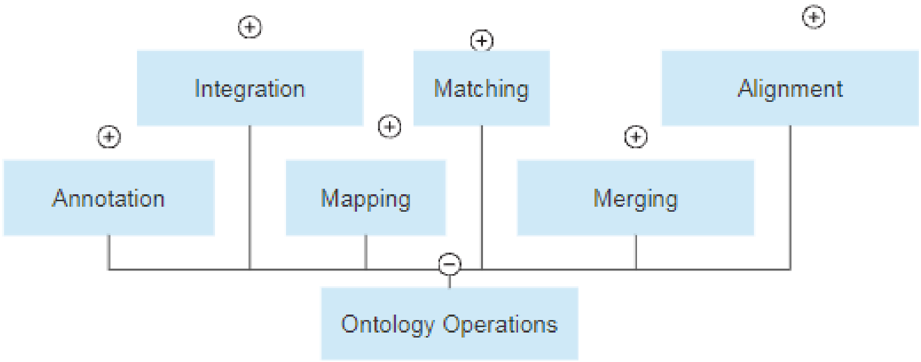

Figure 1. Ontology Operations.

Ontology Mapping. A formal expression describing the semantic relationship between two (or more) concepts belonging to two (or more) different ontologies.

Ontology Alignment. A set of correspondences between two (or more) ontologies in the same domain or in related fields. These correspondences are called "Mappings". In short, Mapping is the process of aligning two ontologies; a similar process used in database schema is called "Matching".

Ontology Merging. Once the mappings between the two source ontologies are defined, the mapped concepts are combined into one. This creates a new ontology from two ontological inputs. See figure 2:

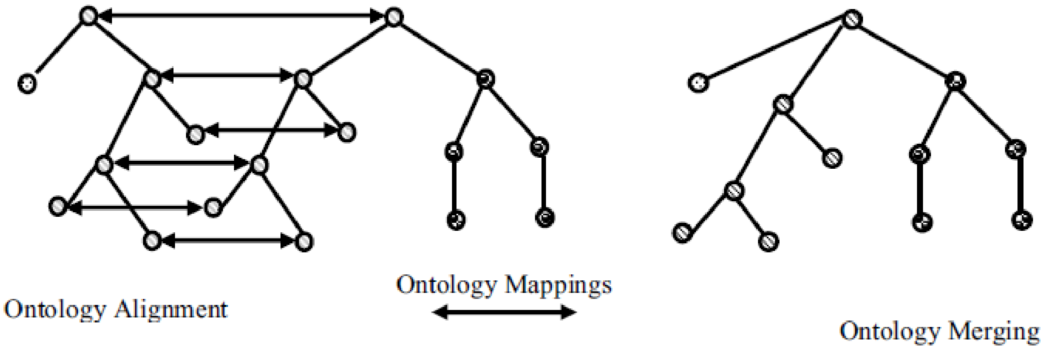

Figure 2. Ontology Mapping, Alignment and Merging [21].

Ontology Annotation. Ontology-based annotation refers to the process of creating metadata using ontology as a vocabulary.

Ontology Matching. It aims to find correspondences between linguistically related entities from different ontologies. These matches can symbolize equivalence as well as other relationships, such as outcome, assimilation, or separation between ontological entities.

Ontology Integration. It refers to the process of building a new ontology by reusing other already available ontologies; building an ontology by merging several ontologies into one that unifies them all; building an application using one or more ontologies.

Operation	Description
Mapping	relationship between concepts from different ontologies
Alignment	correspondences between ontologies in the same domain
Merging	mapped concepts combined into one new ontology
Annotation	process of creating metadata using ontology as a vocabulary
Matching	process of finding correspondences between linguistically related entities from different ontologies
Integration	process of building a new ontology by reusing other available ontologies

Table 1. Ontology Operations.

3.3 Algorithms and Tools

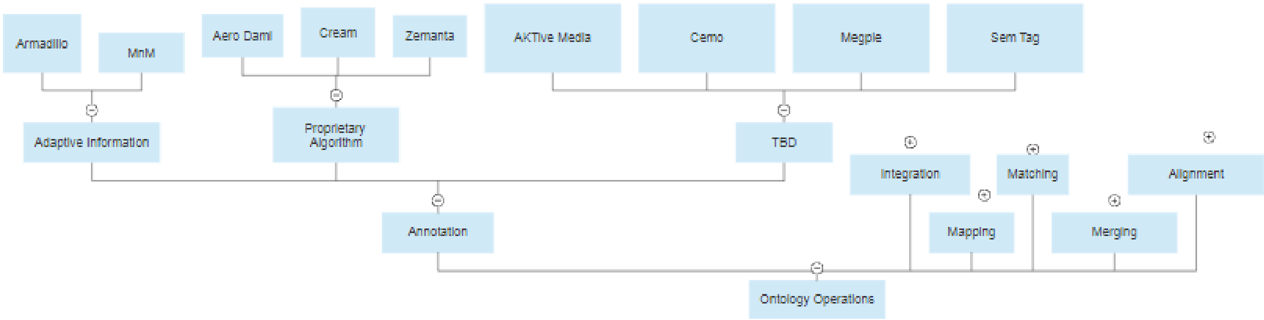

Figure 3. Algorithms and Tools for Ontology Annotation.

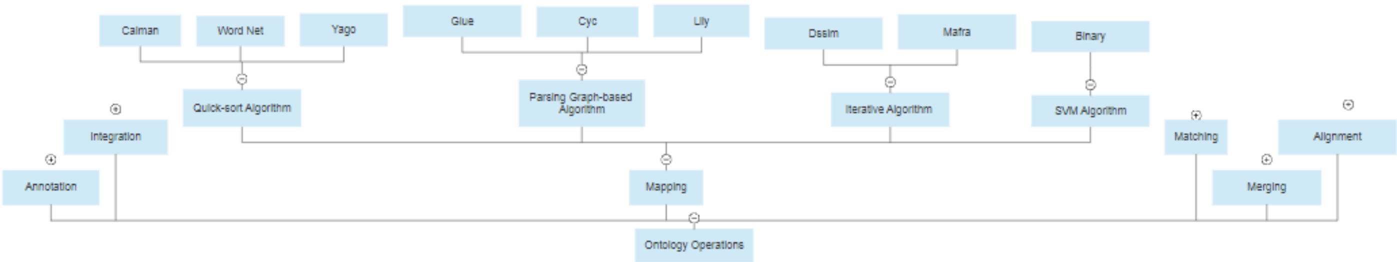

Figure 4. Algorithms and Tools for Ontology Mapping.

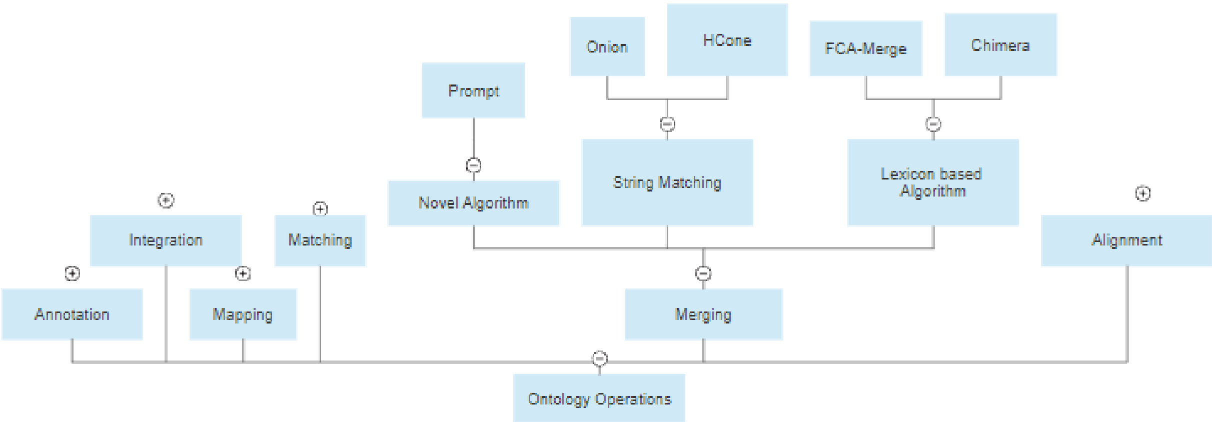

Figure 5. Algorithms and Tools for Ontology Merging.

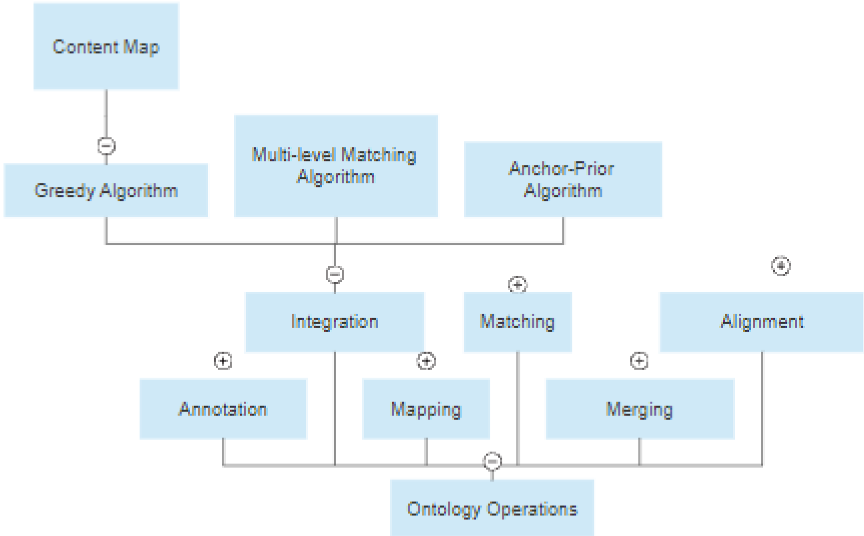

Figure 6. Algorithms and Tools for Ontology Integration.

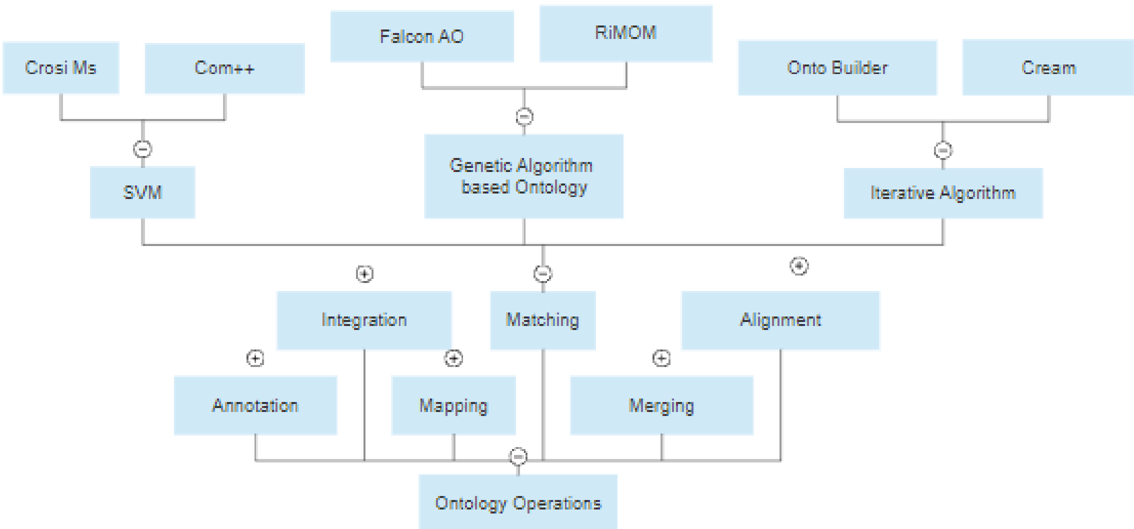

Figure 7. Algorithms and Tools for Ontology Matching.

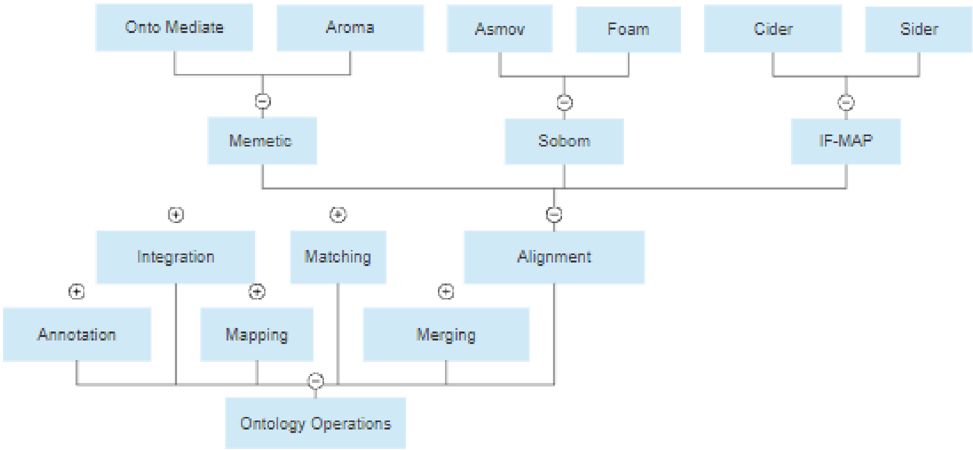

Figure 8. Algorithms and Tools for Ontology Alignment.

✔ Algorithms

Parsing graph-based algorithm: According to the similarity estimate of the complete analysis algorithm based on the ontology graphs, they first determine the ontology degree corresponding to the base as a small base value, and then add it to algorithm-based graph analysis. The problem is that fragmented data environments like the Semantic Web inevitably lead to data and information quality issues. In fact, applications that process this data deal with other algorithm comparisons with information that is not specific, inaccurate, or inconsistent in the field.

SVM: The support vector machine algorithm (SVM) was developed by Cortes and Vapnik in 1995. It is the most widely used kernel-learning algorithm. Achieves relatively high performance in model recognition using concepts well established in optimization theory. Despite this classical mathematics, the implementation of efficient SVM solutions differs from classical numerical optimization methods.

A Novel Algorithm for Fully Automated Ontology Merging: Merging ontology is the progression of creating a new solitary ontology that is understood from two or more dominant source ontologies associated with the same domain. The new ontology will replace the source ontology on this new algorithmic approach. The task of integrating is similar to the construction of the new ontology with this development of the new algorithm, which has comparisons with the implementation of computational analysis that begins with defining the domain and vocabulary of common form vocabulary. It forms the basis of a hierarchy of concepts, they are divided into categories and the corresponding existential relations are attached.

String matching algorithms: Ontology merging is a very effective way to solve the problem of ontology heterogeneity using a string-matching algorithm. This suggests that systems that integrate ontologies with a single mapping result will lead to defects in semantic density inertia, fusion inefficiency, confusion in architectural fusion, and so on.

Lexicon based algorithm for domain ontology merging: Similarity algorithms in ontology tend to have better accuracy in incorporating map-based ontology compared to the other two algorithms. Word Net is used to manage the lexicon-based algorithm that examines the relationship between the categories to which the terms belong.

Adaptive Information and Extraction Algorithm: A proprietary algorithm is a series of steps or rules that are implemented to complete a specific goal, which belongs to a commercial group registered by its owner. The search engine ranking algorithm is an example of a commercial web browser - some information may be readily available to the public other than the source to avoid misuse.

TBD: Track Before Detect (TBD) is a concept that is tracked before it is declared as a target. In this approach, temporary target data is combined over time and can provide detection in cases where it has detected data from any target user. The TBD approach can also be applied to pure detection when an uncertain target displays a very small amount of apparent motion and to actually track motions.

SVM algorithm: **GAOM: Genetic Algorithm based Ontology Matching:** The GAOM algorithm is used as a support for the multi-level matching technique and performs a search to find correspondences between entities in a particular ontology. The main advantage of this algorithm is the high quality of the matches you find. In addition, this algorithm is rapidly merging, making it comparable to SVM and iterative ontology techniques [22].

An Iterative Algorithm for Ontology Matching: This combines standard string-matching metrics with structural similarity metrics that rely on an iterative algorithm in vector representation. After finding the similarities between the concepts, the algorithm will be applied to obtain optimal matching criteria. It can also use existing third ontology mappings as training data to improve accuracy.

Memetic: Often, a harmonized ontology based on matching in combination with the Memetic algorithm has many variations, also known as heterogeneity. The reason for heterogeneity is due to the diversity in modeling an ontology based on different perspectives that have their own consideration in domain analysis. Heterogeneity in existential alignment cannot be avoided with distributed and open systems such as the semantic network. Furthermore, ontology can pursue different modeling goals.

IF-MAP: In order to meet the need to share knowledge within and across organizational boundaries about existential constraints, recent potential observations in both academia and industry have seen the call for the use of ontologies with implementation of the IF-map algorithm as a means to ensure shared understanding of common domains. But with the general use of large distributed environments such as the World Wide Web, many different ontologies have spread, even to the same or similar domain, creating a new need for sharing - that of sharing ontologies.

SOBOM: Modeling of lexical constraints on the IF-map algorithm using the SOBOM algorithm can occur at different levels of semantic basis for different meanings, while semiotic alignment on ontological alignment tends to analyze the fundamentals of individual interpretations of different existential relationships.

Greedy algorithm: The purpose of ontology integration is to create an independent link between different ontologies. Therefore, if we use a greedy algorithm, it tends to build the final flickering ontology and enrich the link between the ontologies using the query transformation algorithm. The approach, which is based on a discussion of different ad hoc relationships from different levels of algorithmic domains, categorizes generalization into an integral link, to further improve ontology storage, retrieval and design, and lays the foundations for cognitive reasoning.

A Multi-Level Matching Algorithm: Various similarity measures have been proposed to integrate the ontology to identify and suggest possible component matches in a semi-automated process. A Multiple Match Algorithm (MMA) can be used to effectively combine these actions, making it easier for users in such applications to identify the "perfect" matches that have been found. A multi-level extension corresponds to existential integration, which is similarly divided by the user and that there is a partial sequence of sections, also defined by the user.

Anchor-Prior algorithm: The main contribution of this approach is to reduce the complexity of the calculations and to improve the accuracy of the integration of the ontology with the implementation of the Anchor-Prior algorithm. The main idea of this approach is to start from the anchor (two identical concepts) and work on a set of identical pairs between its adjacent concepts by calculating the similarities between the concepts aggregated through the ontologies starting from the anchor.

✔ Tools

Binary Classification: Ming Mao dealt with ontology mapping problems with machine learning techniques. His approach has five steps:

i. Generated various domain independent features.
ii. Randomly generates training and testing sets for OAEI benchmark tests.
iii. Train an SVM model on the training set.
iv. Classify testing data on the trained SVM model.
v. Extract mapping results of testing data. Testing data are evaluated against ground truth.

Steps 2 to 6 are repeated 10 times and the average evaluation result is used to eliminate bias.

GLUE: Doan has developed a system, GLUE, that uses machine learning methods to discover maps. The system consists of three phases: the sharing estimator, the similarity estimator and the relaxation labeler. It takes the distribution estimator as input between the classifications O1 and O2 with their instances and applies machine learning to calculate the four possibilities. The similarity estimator applies a user-supplied function, such as a Jaccard or MSP parameter, and calculates the similarity value for each pair of concepts. The relaxation labeler takes the similarity values of classification concepts as input and searches for the best mapping configuration, taking advantage of user-specific domain constraints and inference.

CAIMAN: Martin Lacher proposed a system, CAIMAN, that uses machine learning to classify text based on existential mapping. They assumed that members of society organized their body of explicit knowledge (documentation) according to their personal classification scheme. For the concept node in personal ontology, the corresponding node is listed in the social ontology. CAIMAN offers two services to its users: publishing documents and finding related documents.

DSSIM: Miklos Nagy developed the Ontology Alignment System (DSSim). It takes a concept (or characteristic) from Ontology 1 and considers it a research part. The graph is built from that. It then takes grammatically similar concepts and properties and their synonyms for the search graph from Ontology 2 and the graph is generated. Various similarity algorithms are used to estimate the quantitative similarity values between the order nodes and the ontology fragment. The information is then combined using the Dempster's rule. Based on the collected evidence, they estimated the semantic similarity between search graph structures and ontology fragments and selected those in which they accounted for the highest belief function. The selected concepts are added to the alignment.

Mafra: Alexander proposed a framework for distributed ontological mapping. The MAFRA architecture consists of a set of modules organized in horizontal and vertical dimensions. The horizontal modules correspond to the five basic stages of lifting and normalization, similarity, semantic connection, implementation and post-processing. The vertical modules correspond to four phases; Namely, evolution, knowledge and domain limitations, harmonious co-architecture and graphical user interface. In the lifting and normalization phase, the ontology is imported. In the similarity phase, the similarities between the ontological entities are calculated. In the semantic bridging phase, linguistically similar entities are bridged. In the implementation phase, the tasks are exploited. The post-processing step depends on the results of the implementation. In the evolutionary step, changes in source and target ontologies are synchronized with the semantic bridges defined by the semantic bridge module. In the consensus building phase, the tool helps to establish a consensus among the various proposals of the people involved in the mapping task.

Wordnet: WordNet was developed in 1998 as a lightweight ontological method, nearer to thesauri, and it is a lexical database in English for the semantic similarity of words in Information recovery research. WordNet contains a vast quantity of information. WordNet represents nouns, adverbs, verbs and adjectives as a group of cognitive synonyms with their own individual concepts. A browser is used to control and navigate the individual component in WordNet. It categorizes English words into some groups, such as hypernyms, synonyms, and antonyms.

CYC: CYC is developed by Lenat as part of his research work for MCC Corporation. The ontology in CYC knowledge has 47,000 concepts and 306,000 facts brows able by the CYC web interface. CYC uses a mapping to identify the concepts of each word. For example, CYC provides part of the relationship between tree and leaves; every concept mapped to the terms will return either a true or false statement. Based on this return value, users can then decide the suitable actions for potential processing. CYC has been successfully applied to Terrorism Knowledge Based application and has been used as part of Cyclopedia database.

Babenet: BabeNet is developed to overcome the drawback of WordNet. BabeNet integrates the domain and knowledge base of these two systems, and could adequately supply the users with a higher-level ontology domain. In addition, BabeNet is also able to differentiate word sense disambiguation exactly using the information provided by Wikipedia domain knowledge.

Yago: Yet Another Great Ontology (YAGO) is developed by Fabian and it is a lightweight ontology with extensible functionalities for high data coverage and accuracy. YAGO achieved an accuracy of 95% on its test cases. YAGO extracted data from Wikipedia and combined it with WordNet, and provided the users with 1 million entities and 5 million facts.

Lily: Peng Wang had given an ontology mapping system, Lily. Lily realized four main functions:

i. Generic Ontology Matching method (GOM) is used for common matching tasks with small size ontologies.
ii. Large scale Ontology Matching method (LOM) is used for the matching tasks with large size ontologies.

iii. Semantic Ontology matching method (SOM) is used for discovering the semantic relations between ontologies.
iv. Ontology mapping debugging is used to improve the alignment results.

Prompt: PROMPT contains a set of tools that have made a significant impact in the field of ontology integration, alignment and release. The package includes an ontology merging tool, an ontology tool for finding additional points of similarity between ontologies such as iPROMPT, a PROMPT Diff and a complete ontology enrichment tool (PROMPTFactor) [23]. PROMPT takes two types of ontologies as inputs and directs the user in creating the combined ontologies as outputs. First PROMPT creates an initial list of matches based on the category names. Then the iterative cycle occurs: the user activates a process by selecting one of the PROMPT suggestions from the list or by using the ontology editing environment to directly select the desired process, and PROMPT automatically makes additional changes based on the type of operation. It generates a list of suggestions for the user based on the ontology structure around the arguments to the last operation, identifies the conflicts that the last operation has introduced into the ontology, and finds possible solutions to those conflicts.

Onion: Mitra has developed a scalable framework for ontology integration that uses a graph-oriented model to represent ontology. There are two types of ontologies, individual ontology (source ontology) and articulated ontology, which contain concepts and connections expressed as syntactic rules. Mapping between ontologies is done with ontological algebra. The ONION architecture consists of four components: data layers, a viewer, a query system, and an articulation engine. The data layer contains the shells of external sources and the expressive ontologies that form these bridges between sources. The viewer is a user interface that visualizes both the source and the ontology of the expression. The query engine translates the queries formulated in relation to the articulated ontology into a query execution plan and executes the query. The articulation engine takes the articulation rules proposed by SKAT and creates sets of articulation rules, which are sent to the expert for confirmation.

FCA-Merge: Gerd Stumme developed a framework for ontology merging (FCA-Merge) [24]. FCA Merge employs a bottom up approach. The process of FCA Merge consists of three steps, namely i. extracting instances and computing of two formal contexts K1 and K2, ii. deriving a common context and computing a concept lattice using FCA Merge core algorithm and iii. generating the final merged ontology based on the concept lattice. FCA Merge tool takes as input both ontologies and a set D of natural language documents. Instances are extracted from the document in D. The second step consists of the basic FCA Merge algorithm that merges two contexts and calculates a concept lattice from the merged context using FCA techniques. The final step requires human interaction, it is based on the pruned concept lattice and the sets of relations, the ontology engineer creates the concepts and relations of the target ontology in order to derive the merged ontology from the concept lattice.

Chimera: Deborah L. McGuinness developed a tool called Chimera to merge ontologies [25]. Chimaera is aimed to support:

i. merging multiple ontologies and
ii. diagnosing and developing ontologies.

It facilitates integration by allowing users to load existing ontologies into a new workspace. Chimaera proposes to combine candidates based on a number of features. It creates a list of resolution names that can be used as a guide during the merge task.

Hcone-merge: The purpose of HCONE approach is to confirm the mapping and to find the minimum set of axioms for the new merged ontology [26]. This approach relies on i. Capturing informal interpretations of concepts by mapping them to WordNet using lexical semantic indexing, and ii. Making use of formal semantics to define concepts with the help of description.

AeroDAML: AeroDAML is a cognitive learning tool that automatically generates annotations on web pages after applying natural language extraction techniques. AeroDAML assigns appropriate nouns and common relationships to classes and properties in DAML ontologies. AeroDAML has two different uses: i. The web-enabled version that supports annotation with a standard generic ontology of commonly found words, classes and relationships. ii. The client server version that supports annotation with customized ontologies.

Cream: CREAM is a framework for creating annotations, especially relational metadata. CREAM supports the creation of metadata during writing and after writing web pages. CREAM includes heuristic services, browser, document management system, fact browser / ontology guide, document editors / viewers, and meta ontology. Onto Annotate and OntoMat Annotizer are two different implementations of the CREAM framework.

Falcon AO: Wei Hu designed the Falcon-AO system to find, match, and learn ontologies, and eventually capture knowledge through an ontology-based approach. It is an automatic ontology matching method that helps enable interoperability between (semantic) web applications that use different but related ontologies. It consists of five components: the repository to store the data; the model pool to control the ontology and to create different models for different matches; the alignment set for the creation and evaluation of exported alignments; the central controller for configuring matching strategies and executing matching operations.

CROSICMS: Yannis Kalfoglu proposed the CROSI CMS architecture, a structure matching system. The modular architecture uses a multi-strategic system with four modules, namely feature generation, feature selection and processing, aggregator and evaluator. In this system, different input data features are generated and selected to launch different types of feature matching tools. The obtained similarity values are grouped with several comparison pools operating in parallel or recursive order.

RiMOM: Xiao Zhang proposed a structure RiMOM for ontology matching. The RiMOM consists of six major steps. The input ontologies are loaded into the memory and the ontology graph is constructed in Ontology Preprocessing and Feature Factors Estimation. In Single strategy execution the selected strategies are getting to find the alignment independently. Each strategy outputs an alignment result. In the Alignment combination phase RiMOM combines the alignment results obtained by the selected strategies. If the two ontologies have a high structure similarity factor, RiMOM employs a similarity propagation process to refine the found alignment and to find new alignment according to the structural information. Alignment-refinement refines the alignment results from the previous steps.

Anchor-prompt: Natalia F. Oyster has developed a tool in the prompt suite called Anchor-prompt for ontology merging. Anchor-PROMPT takes as input a set of pairs of related terms - anchors from the source ontology. Either the user selects anchor manually or the system creates them automatically. From this set, Anchor-PROMPT produces a set of new pairs of syntactically similar terms. To do this, Anchor-PROMPT crosses paths between anchors in the corresponding ontologies. The path traces the relationships between categories defined by relationships, or hierarchical slots, and their domains and ranges. Then Anchor-PROMPT compares terms along these paths to find similar terms.

COM++: Erhard Ram et al., developed COMA ++ as a tool for presenting and matching ontologies. The graphical user interface provides access to the five major components of COMA ++, the repository for storing all matching data, the model and mapping pool for managing patterns, ontologies and memory mapping, the match customizer for creating matches and match strategies, and the completing engine for performing operations on matches. Automatic matching processing is implemented in the execution engine as a three-step process, component selection, matching and execution of a set of similarities. The resulting mapping can be used as input to the next iteration for further modification.

OntoBuilder: Avigdor Gal suggested the OntoBuilder ontology matching tool. The OntoBuilder project supports extracting ontologies from web search interfaces. Finds the best mapping between two ontologies. It is a fully automated existential matching system. It contains many unique matching algorithms that can match concepts according to their data types, limitations in setting values and, above all, sorting concepts into forms.

ASMOV: Yves R. Jean-Mary et al. have developed ASMOV, an automatic ontology matching tool which has been designed in order to facilitate the integration of heterogeneous systems, using their data source ontologies [27]. The current ASMOV implementation produces mappings between concepts, properties, and individuals, including mappings from the object to data type properties and vice versa. The ASMOV algorithm iteratively calculates the similarity between entities for a pair of ontologies by analyzing four features: lexical description (id, label, and comment), external structure (parents and children), and internal structure (property restrictions for concepts; types, domains, and ranges for properties; data values for individuals), and individual similarity. The measures obtained by comparing these four features are combined into a single value using a weighted sum.

CIDER : Jorge Gracia et al. suggested a matching service called CIDER (Context and inference BaseD alignER) to measure semantic similarity. It is a pattern based matching system [28]. It consists of a 3-step process:

a- The first step is to calculate the linguistic similarity between the terms for the labels and descriptions.

b- The second step is to explore the structural similarity of terms, using their ontological contexts, and using vector space modeling.

c- The third step is to weight each contribution and a final degree of similarity is provided. Next, we get a matrix M with all the similarities. The final alignment A is then extracted, finding the highest-ranking associations between the terms and filtering out those that are below the given limit.

SPIDER: Marta Sabou gave the SPIDER system, which provides alignment with different types of maps. This system combines two specific subsystems. First, the CIDER algorithm is to perform parity mapping. Second, the alignment is extended by the depictions of inequalities made by Scarlet. CIDER is briefly explained in the section above. Scarlet selects and explores online ontologies automatically to discover the relationship between two given concepts. All relations are obtained using derivation rules that explore all the relations and not only the direct ones.

Onto Mediate: Gianluca Correndo proposed the Onto Mediate project to align ontologies and share mapping results. The system consists of three main subsystems: ontology and dataset manager; an environment to align the ontology; an environment for social interaction. Ontology and datasets manager allows users to register/deregister sets of data they intend to share with the community and the ontologies that describe their vocabulary. The ontology alignment environment provides an API for automated ontology alignment tools to be included. The function of the social interaction environment allows the members of the community to socially interact with one another.

AROMA: Jerome David proposed a method AROMA that is a hybrid, extensional and an asymmetric matching approach designed to find out the relations between entities from two textual taxonomies. AROMA is divided into three successive main stages:

i. the preprocessing stage allows representing each entity (classes and properties) by a set of terms,
ii. the second stage consists of the discovery of association rules between entities, and finally,
iii. the post processing stage aims to clean and enhance the alignment.

FOAM: Marc Ehrig has proposed a framework for Ontology Alignment and Mapping. This tool has six steps. As the first step in feature engineering, the tool selects the ontology for a specific domain. Next search step selection: it chooses two entities from the two ontologies to compare (e1, e2). Similarity assessment is the third step, indicating a similarity for a given description (feature) of two entities. Similarity aggregation step aggregates the multiple similarity assessments for one pair of entities into a single measure. To propose the alignment, a threshold and interpretation strategy is used. Finally, iteration, as the similarity of one alignment influences the similarity of neighboring entity pairs.

Content Map: Ernesto Jimenez-Ruiz et al. developed a system called Content Map A logic-based Ontology Integration Tool using Mappings [29]. The Content Map evaluates and repairs the logic consequences of merging two independent ontologies using mapping. The method is as:

i. Compute mapping M between O1 and O2 using a mapping algorithm, and filter those using the criteria.
ii. Compute logic differences and evaluate the impact by comparing the entailments holding before and after the integration.
iii. Detect unintended entailments and select them.
iv. Compute repair plans and execute the best one according to the user requirements.

ATOM: stands for Automatic Target-driven Ontology Merging [30]. It integrates source ontology into target ontology and aims to maintain a preference for a target ontology among source ontologies. The approach initially generates an intermediate merged result and then optimizes it based on some generic merge requirements (GMRs) to produce the final merged ontology. Target-based merging was assessed with full merging related to the number of concepts and terminal paths.

CODE: stands for Common Ontology Development [31]. It can be used to merge more than two ontologies at the same time. CODE merges entities based on four different scenarios in terms of the relationships between the entities involved between the source ontologies. CODE is evaluated by SPARQL to demonstrate knowledge preservation in the embedded ontology.

CreaDo: is a parameter-based ontology merging technique that selects a subset of mapping based on the merge purpose [32]. It allows the creation of a merged ontology only with relevant information for the specific purpose. To conduct this, CreaDo uses the ontology modularization technique to extract ontology modules (relevant information) from each source ontology. A merge parameter of CreaDo is a concept of the domain that should be represented in the merged ontology. Once the merged ontology is created, the detected errors will be reported to the user and no refinements take place for it. For the evaluation of the method, the authors reported basic statistics about a few common pitfalls related to the general design of the ontology.

AgreementMaker: is a system for matching pattern [33]. It Allows high customization of the matching process, including several alignment methods to be performed on inputs with different levels of detail, and allows you to determine the amount of user participation and the formats the input ontologies as well as the alignment results can be saved in.

Methodology	Operation	Tools Name	Algorithm	Input/ Output	Technology
Machine Learning	Mapping	Binary classification	SVM	Two ontologies and Evaluation Result	CROSS TASK
Machine Learning	Mapping	Glue	Parsing Graph based	Concepts in Taxonomy and similarity Measure	Jaccard Coefficient
Machine Learning	Mapping	Caiman	Quick-sort	Two Ontologies and Mapping result	
Machine Learning	Integration	Content Map	Greedy	Pre-Computed Mapping ontologies and visualizing the mapping results	
Machine Learning	Merging and Alignment	Prompt	Novel Algorithm for Fully Automated Ontology Merging	Two DAML Ontologies and Merged Ontology	OKBC
Structure Based	Matching	Falcon	GAOM	RDFS or Owl Ontology and RDF/XML Format	Jena
Structure Based	Alignment	ASMOV	SOBOM	OWL-DL	Jena ARP
Structure Based	Alignment	CIDER	IF-MAP	RDF/OWL input And RDF document	Alignment API
Structure Based	Alignment	SPIDER	SOBOM	RDF/OWL ontologies and RDF document	CIDER and Scarlet
Structure Based	Merging	ONION	String matching algorithms	RDF file and plain text	SKAT
Structure Based	Matching	CROSIC MS	SVM	OWL Ontologies and matched result	Jena JWNL
Structure Based	Matching	OntoBuilder	Iterative Algorithm for Ontology	OWL Ontologies and produce matching	
Structure Based	Mapping	DSSim	QOM	Two Ontologies And Mapping result	SKOS parser
Structure Based	Merging	FCA- merge	Lexicon based	Two ontologies + DL and produce merged ontology	Concept lattice
Semantic Based	Alignment and Mapping	OntoMediate	A Multilevel Matching Algorithm	OWL ontologies and produce result	Jena API
Semantic Based	Matching	RiMOM	GAMO	RDF and OWL ontologies and produce result	OWL API
Semantic Based	Matching	Anchor-prompt	SVM	Pair of anchors and generates new pairs with close similarity	Protégé 2000 plugin
Semantic Based	Mapping	MAFRA	QOM	RDF files and produce output	MAFRA Service Interface API
Semantic Based	Merging and diagnosing	Chimera	Lexicon based algorithm	RDF and DAML and merged output	
Semantic Based	Merging and alignment	HCONE	String matching algorithms	Two ontologies and result	Neo Classic Description Logic
Semantic Based	Mapping	Word Net	Quick-sort Algorithm	Different words and categorized words	LightWeight Ontological Tech
Semantic Based	Mapping	CYC	Parsing Graph based algorithm	Concepts Facts and result	Relationship between Tree and leaves
Semantic Based	Mapping	BabeNet	Parsing graph-based algorithm	Integrate domain and knowledge and high-level ontology	LightWeight Ontological Tech
Semantic Based	Mapping	YAGO	QUICK Sort Algorithm	High data and Accuracy Ontology	LightWeight Ontological Tech
Semantic Based	Annotation and Mapping	Aero DAML	Proprietary algorithm	URI and DAML Annotation result	NLET
Hybrid	Matching	COM++	SVM	XML, OWL files as input and generates output	OWL API
Hybrid	Alignment	AROMA	A Multilevel Matching Algorithm	OWL ontologies and produce output	OWL Libraries
Hybrid	Mapping	Lily	Parsing Graph based algorithm	RDF ontologies and generates output in text	LOM GOM and SOM
Similarity	Alignment and Mapping	Foam	Interpretation strategy	Two ontologies and result	
Target based	Merging	Atom	GMR	Two ontologies and result	
Hybrid	Merging	Code		Ontologies inputs and outputs	Sparql
Parameter Based	Merging	CreaDo	ontology modularization technique	Ontologies inputs and outputs	
Similarity and Structural based	Matching	AgreementMaker	Customized Matching	Ontologies inputs and outputs	OWL

Table 2. Comparison table that shows the essential features of some tools discussed in Section 3.

4 Conclusion and Future work

Choosing the right ontology application is one of the fundamental processes for any semantic web and all kinds of applications. Based on our reviews and results, we sincerely hope that Ontology Tools will have multilingual support forever. Our aim of this survey is describing the importance of ontology operations, their algorithms and applications. The comparison table shows the essential features of the tools that are being discussed. The result of this survey and analysis provides complete understanding of ontology operations, algorithms and tools developed in the ontology operations. This work can be extended with other tools/frameworks for subsequent ontology operations.

References

1. Y. Kalfoglou and M. Schorlemmer, "Ontology mapping, the state of the art". The Knowledge Engineering Review, 18(1):1-31. 2003.
2. N. F. Noy, "Semantic Integration: A Survey of Ontology-Based Approaches". SIGMOD Record, Vol. 33, No. 4. pp. 65-70.2004.
3. M. Benerecetti, P. Bouquet and C. Ghdini, "Contextual reasoning distilled". Journal of Théoritical and Experimental Artificial intelligence, 12(3):279-305. 2000.
4. Lorena Otero-Cerdeira, Francisco J. Rodriguez-Martinez, and Alma Gomez-Rodriguez. 2015. Ontology matching: A literature review. Expert Systems with Applications (2015), 949–971.
5. P. Shvaiko and J. Euzenat. 2013. Ontology matching: State of the art and future challenges. Knowledge and Data Engineering, IEEE (2013). 158–176.
6. Pavel Shvaiko and Jerome Euzenat. 2008. Ten challenges of ontology matching. In Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE'08). 1163–1181.
7. Namyoun Choi, Il-Yeol Song, and Hyoil Han, "A Survey on Ontology Mapping", SIGMOD, Vol. 35, No. 3, Sep. 2006.
8. Natalya F. Noy, "Semantic Integration: A Survey of Ontology-Based Approaches", SIGMOD, Vol. 33, No. 4, December 2004.
9. Akrivi Katifori and Constantin Halatsis, "Ontology Visualization Methods—A Survey", ACM Computing Surveys, Vol. 39, No. 4, Oct 2007.
10. Matteo Cristani and Roberta Cuel, "A Survey on Ontology Creation Methodologies", International Journal on Semantic Web & Information Systems 1(2), 49-69, June 2005.
11. Elena Simperi, "Reusing Ontologies on the Semantic Web: A Feasibility Study", Science Direct, Data and Knowledge Engineering, Vol: 68, issue 10, p: 905-925, Oct 2009.
12. Yannis, K. and S. Marco, "Ontology mapping: The state of the art", The Knowledge Engineering Review, 18: p: 1-31, 2003.
13. Ravi Loudrusamy and Gopinath Ganapathy, "Feature Analysis of Ontology Mediation Tools", Journal of Computer Science, 4(6), p: 437-446, 2008.
14. Tin-Yu Wu, Chi-Yuan Chen, Kai-Di Chang, Diego Chung and Han-Chieh Chao., Pave the Way to Future Smart Living Space-Cross-layer Enhanced

AAA for 4G Core Network, Multimedia and Ubiquitous Engineering, 2007. MUE '07, International Conference on., 1(2), 191-203, (2007).
15. Berenjian, Samaneh, Mehdi Shajari, Nadieh Farshid, and Majid Hatamian.” Intelligent automated intrusion response system based on fuzzy decision making and risk assessment.” In 2016 IEEE 8th International Conference on Intelligent Systems (IS), pp. 709-714. IEEE, 2016.
16. Naghizadeh, Alireza, Samaneh Berenjian, Ehsan Meamari, and R. Ebrahimi Atani. ”Structuralbased tunneling: preserving mutual anonymity for circular P2P networks.” International Journal of Communication Systems 29, no. 3 (2016): 602-619.
17. Naghizadeh, Alireza, Samaneh Berenjian, Behrooz Razeghi, Saghi Shahanggar, and Nima Razagh Pour.” Preserving receiver’s anonymity for circular structured P2P networks.” In 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp. 71-76. IEEE, 2015.
18. Berenjian, Samaneh, Saeed Hajizadeh, Majid Hatamian, and Reza brahimi Atani.” An Incentive Security Model to Provide Fairness for Peer-to-Peer Networks.” arXiv preprint arXiv:1906.09355 (2019).
19. Naghizadeh, Alireza.” Improving fairness in peer-to-peer networks by separating the role of seeders in network infrastructures.” Turkish Journal of Electrical Engineering & Computer Sciences 24, no. 4 (2016): 2255- 2266.
20. Hatamian, Majid, Seyed Sajad Ahmadpoor, Samaneh Berenjian, Behrooz Razeghi, and Hamid Barati.” A centralized evolutionary clustering protocol for wireless sensor networks.” In 2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6. IEEE, 2015.
21. Siham AMROUCH and Sihem MOSTEFAI, ‘Survey on the literature of ontology Mapping, Alignment and Merging.’, 2012.
22. Reuven Cohen, Guy Grebla and LiranKatzir., Cross-Layer Hybrid FEC/ARQ Reliable Multicast with Adaptive Modulation and Coding in Broadband Wireless Networks, *IEEE/ACM Transactions on Networking.*, 18(6), 1908–1920, (2010).
23. N. F. Noy and M. A. Musen, “The prompt suite: Interactive tools for ontology merging and mapping,” International Journal of Human-Computer Studies, vol. 59, no. 6, pp. 983–1024, 2003, ISSN: 10715819. DOI: http://dx.doi.org/10.1016/j.ijhcs.2003. 08.002.
24. G. Stumme and A. Maedche, “FCA-Merge: Bottom-up merging of ontologies,” in IJCAI, vol. 1, 2001, pp. 225–230.
25. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, “An environment for merging and testing large ontologies,” in KR, 2000, pp. 483–493.
26. K. Kotis, G. A. Vouros, and K. Stergiou, “Towards automatic merging of domain ontologies: The hcone-merge approach,” Journal of Web Semantics, vol. 4, no. 1, pp. 60–79, 2006, ISSN: 15708268. DOI: http://dx.doi.org/10.1016/j.websem.2005.09.004.

27. Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, “Ontology matching with semantic verification,” Journal of Web Semantics, vol. 7, no. 3, pp. 235–251, 2009.
28. X. Lin and N. B. Shroff., The Impact of Imperfect Scheduling on Cross Layer Rate Control in Wireless Networks, *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE.*, 3, 1804-1814, (2005).
29. E. Jiménez-Ruiz, B. C. Grau, I. Horrocks, and R. Berlanga, “Ontology integration using mappings: Towards getting the right logical consequences,” in ESWC, Springer, 2009, ch. 16, pp. 173–187, ISBN: 9783642021206. DOI: http://dx.doi.org/10.1007/978- 3-642-02121-3_16.
30. S. Raunich and E. Rahm, “Target-driven merging of taxonomies with ATOM,” Information Systems, vol. 42, pp. 1–14, 2014, ISSN: 03064379. DOI: http://dx.doi.org/10.1016/j.is.2013.11.001.
31. D. H. Fudholi, W. Rahayu, and E. Pardede, “Code (common ontology development): A knowledge integration approach from multiple ontologies,” in IEEE 28th International Conference on Advanced Information Networking and Applications, 2014, pp. 751–758, ISBN: 9781479936304. DOI : http://dx.doi.org/10.1109/AINA.2014.92.
32. S. P. Ju, H. E. Esquivel, A. M. Rebollar, M. C. Su, et al., “CreaDO–a methodology to create domain ontologies using parameter based ontology merging techniques,” in 10th Mexican International Conference on Artificial Intelligence, IEEE, 2011, pp. 23–28.
33. Cruz, I., Antonelli, F.P., Stroe, C.: AgreementMaker: efficient matching for large real-world schemas and ontologies. Proc. VLDB Endow. 2(2), 1586–1589 (2009) (p. 230).

In order to cite this article,

Maroun, M. (2021) A Survey on ontology operations techniques. Mathematical and Software Engineering, 7(1-2), 7-28. Varεpsilon Ltd, http://varepsilon.com.