COMET: An Ontology Extraction Tool based on a Hybrid Modularization Approach

Bernabé Batchakui[1][a], Emile Tawamba[2][b] and Roger Nkambou[3]
[1]National Advanced School of Engineering, University of Yaoundé 1, Yaoundé, Cameroon
[2]University Institute of Technology, University of Douala, Douala, Cameroon
[3]Department of Computing, University of Quebec at Montreal, Montreal, Canada

Keywords: Structural Modularization, Semantic Modularization, Segmentation Algorithm, Stuckenschmidt Approach, Cuenca Grau Approach, Wu and Palmer Measurement, OWL Module Extractor, Ontology Segmentation.

Abstract: The design of ontologies is a non-trivial task that can simply be reduced to the reuse of one or more existing ontologies. However, since an expert in knowledge engineering would only need a part of the ontology to perform a specific task, obtaining this partition will require the modularization of ontologies. This article proposes a tool named COMET, based on hybrid modularization, composed of existing structural and semantic modularization techniques, that, from an ontology and a list of input terms, generates, according to an integrated segmentation algorithm, a module which in fact is a segment consisting only of concepts deemed relevant. The segmentation algorithm is based on two parameters which are hierarchical deep and semantic threshold.

1 INTRODUCTION

The development of Web technologies has brought about an increased interest in the research for knowledge sharing and integration in a distributed environment. The Semantic Web tends not only to render information accessible but also readable and usable through data processing applications. The latter provides the tools which permit a better organization of information through extracting, sharing and reusing the specific knowledge in a domain. In this perspective of sharing and interoperability, the proliferation and availability of ontologies are crucial. It goes without saying that ontologies have become an inevitable pattern for the representation and reasoning on domain knowledge. Though essential to the system's management on knowledge basis, it does not refute the fact that the conception, the reuse and integration of ontologies remain complex tasks. In the present situation, the authors quickly realize that modularization will be an effective approach which will make it possible to provide answers to a good number of problems of expert knowledge. The authors will examine the modularization as a process during which, a set of pertinent concepts of a given domain is identified. Modularisation of ontologies is an important field of research with many recent publications such as (Leclair and al, 2019; De Giacomo and al, 2018; Xue and Tang, 2017; Khan and Keet, 2015; Algergawy and al., 2016; Babalou and al, 2016; Movaghati and Barforoush, 2016). Each of these works addresses modularisation either on the structural or on the semantic view. The notion of modularization is based on the principle of «divide and rule» generally applied in software engineering which is about developing an application whose structure depends essentially on autonomous components easily conceivable and reusable. The module thus represents a component of software which executes a precise task and interacts with others. In ontological engineering, modularization can be perceived in two ways. Firstly, it can be seen as a process leading to the decomposition of a large ontology into ontological modules of small sizes. Secondly, it can equally be perceived as a stage in the ontology construction process which is done through the conception of a set of ontological modules

[a] https://orcid.org/0000-0002-5287-4207
[b] https://orcid.org/0000-0003-1486-9922

Batchakui, B., Tawamba, E. and Nkambou, R.
COMET: An Ontology Extraction Tool based on a Hybrid Modularization Approach.
DOI: 10.5220/0010688300003064
In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) - Volume 2: KEOD*, pages 74-83
ISBN: 978-989-758-533-3; ISSN: 2184-3228

independent from one another. Modularization therefore is summarized in selecting concepts and pertinent relationships based on the task and the application for which the modeling of an ontology is decided. To do this, the researchers may:

- Reuse all the ontology in entirety. All the concepts and relationships of domain ontology will be included in the ontology which is conceived. This results in a great weighing on the ontology application which will be as big as if the domain ontology is large. In such a case, the authors would discover concepts and definitions which are not necessarily relevant to the task at hand.
- Build a new ontology. This will represent a fastidious work. The conception of an ontology remains a complex task in the sense that it is expected of the expert to define all the concepts and relationships necessarily.
- Reuse of part of the initial ontology. It is a compromise between the two preceding alternatives. In this case, the application ontology will therefore represent a part or a module of the domain ontology. There exist two principal modularization techniques of ontologies (Algergawy A. and al, 2016.). The authors distinguish the techniques based on the partitioning of ontology and the techniques based on the extraction of ontological module. The partitioning subdivides an ontology into a set of sub structures; autonomous or dependent on each other called partitions, meanwhile the module extraction consists in extracting a sub ontology based on a precise signature. Whether it deals with the ontology partitioning or module extraction, it is remarked that these techniques are based on structural and semantic approaches.

The structural extraction module approaches bring the problem of modularization to the extraction of a graph which in essence is a sub structure containing pertinent concepts in relation to a previously fixed signature. While the semantic approaches are based essentially on the very usage of ontology, putting in evidence the way concepts are linked one to the other. That is called the semantic proximity. However, they quickly realized that each of these approaches present limitations for which they will delve in to provide preliminary solutions in proposing a hybrid approach name COMET. The latter draws from the better of the two worlds in combining at the same time the structural and semantic approaches so as to consolidate the strengths of these.

The remaining parts of this paper are organized into five sections as follows: the first section is the state-of-the-art on existing approaches to modularization techniques. The authors present the particularities and limitations of these approaches by demonstrating the need to develop a new approach. The next section concerns the COMET modularization approach. The third section is the experimentations and results of COMET where validation protocol is presented and the results of the experimentations. The fifth section is concerned with the analysis of the results. The conclusion of this paper will be presented in the last section.

2 STATE OF THE ART

An ontology O is defined by the following formula (Palmisano, Tamma, Payne, & Doran, 2009):

$O = (Ax(O), Sig(O))$ where Ax(O) represents the set of axioms made of concepts, relationships and functions (sub class, equivalence, instantiation, etc…). Sig(O) is the signature of the O which represents the set of entity names which are found in the axioms. In other words, it refers to the O vocabulary. The ontology modularization of O permits the definition of a module M as:

M = (Ax(M), Sig(M))

where M is part of O, $(\mathrm{Ax}(M))^I \subseteq (Ax(O))^I$ and Sig (M) ⊆ Sig (O), I is the interpretation. In fact, it is a basis of the description semantic logic. It is symbolize by I = (Δ^I, $\bullet^I$) where Δ^I is the domain interpretation and $\bullet^I$ the interpretation function.

There are two main approaches to modularizing ontologies: approaches based on ontology partitioning and approaches based on ontology module extraction.

Partitioning is the process during which an ontology O is fractioned into a set of M modules (not necessarily disjoint) such that the union of interpretation of all the partitions thus creates the equivalence of the initial ontology interpretation O. Formally, the ontology partitioning can be defined as follows:

$$M = \{M_1, M_2, \ldots, M_n\} \mid \{(Ax(M_1)^I U\, Ax(M_2)^I U \ldots U\, Ax(M_n)^I\} = \{\mathrm{Ax}(O))^I\}.$$

the authors distinguish two principal partitioning approaches:

- The approach of the Stuckenschmidt and al. is based uniquely on the hierarchical structure of classes (Stuckenschmidt & Klein, 2004). This

approach is founded on the hypothesis that the dependence between the concepts could be derived from the structure of the ontology. The latter is represented by a weighted graph O = < C, D, W > where the nodes (C) represent the concepts and the crests (D) represent the relationships between concepts, and the weight (W) varies as a function of the dependence.

- The approach of the Cuenca Grau and al. on the other hand is based on ε-connections (Σ) (Grau, Parsia, Sirin, & AdityaKalyan, 2005). These are used as the language of definition and ontologies combination instantiation OWL-DL. The partitions generated by Cuenca Grau and al. are at the same time structurally (Σ ~ O) and semantically compatible (Σ ≈ O). The structural compatibility has as objective to guarantee that no entity or axiom should be added, removed or modified during the partitioning. The role of the semantic compatibility is to guarantee the preservation of the interpretation module.

The extraction of the ontology module is a process during which an M module covering a specific signature is extracted from an ontology O, as in Sig(M) ⊆ Sig(O). M is the pertinent part of O which covers the entire elements defined by Sig (M). M module is an ontology of its own, as such, other modules can be extracted from it. Formally, the module extraction can be defined as follows: extracting

$$(O, Sig(M)) \longrightarrow \{M|(Ax(M))^I \subseteq (Ax(O))^I\}$$

The most evident way to visualize an ontology is to represent it in a graph where the peaks consist of concepts and individuals and the crests are the relationships (hierarchies and semantics) (Dupont, Callut, Dooms, Monette & Deville, 2006). The problem of module ontology extraction will then be brought to the extraction of a sub graph containing the most pertinent concepts in relation to the list of terms entered by the user. There is the necessity to evaluate the pertinence of concepts one from the other. The authors can refer to the notion of distance in the sense that the most pertinent will be the closest to the starting concepts in a certain predefined range by the user, and for which the inter-concepts distance could be evaluated by the calculation of the shortest path of Dijkstra. This approach will be realizable, most assuredly since there exist a Java library called JUNG, which enables the transformation of an ontology into a graph.

The advantages of this structural approach are many. Firstly, there already exist algorithms effective in extraction of sub graphs. Supposing that the ontology from which the module to be extracted is of a satisfactory quality, this approach does not re-structure the ontology in the sense that the extracted module conserves the same internal structure as the original ontology. And should the graph be any least dense, the complexity of the algorithm is reduced. On the contrary, JUNG considers by default that all the crests of the graph are equivalent. On this basis, it becomes impossible to correctly evaluate the nodes and arcs, even as in an ontology, the relationships between different concepts are never equivalent. More so, this method ignores the concepts distance from the beginning concepts but which can also be pertinent.

The second way of seeing an ontology is to consider it through its utility which is the representation of knowledge. In other words, when it comes to the definition of concepts which are useful in relation to the others, in order to understand the domain which the ontology describes, the authors speak of semantic domain. In making a total abstraction of the structure of ontology, the authors shall be taking concepts two by two in order to see if they are semantically close (Jiang & Conrath, 1997). Different from the structural approach, the authors make allusion here to a semantic distance which will allow us evaluate the semantic proximity of ontology concepts in relation to those corresponding to the terms entered by the user.

The advantages of the semantic approach are the pertinence and precision, because, what constitutes the principal strength of an ontology remains its semantics (Ghosh, Abdulrab, Naja & Khalil, 2017b). Suffice for a concept to be linked to a bad concept for the quality of ontology to be altered, and so, can led us to an ontology which does not tie anymore with reality. The semantic approach permits us to put ontology from start and to test the semantic similarity of all the sets of concepts of ontology. Meanwhile, seeing that the authors work with large ontologies, to calculate the semantic distance of all the sets of concepts seems very much utopic in terms of the algorithmic complexity and the execution time. It therefore becomes primordial to turn to other alternatives which will permit us to still consider the ontology structure instead of completely ignoring it.

3 COMET MODULARIZATION APPROACH

In order to cope with the current limitations of existing modularization approaches, the authors

propose the hybrid COMET approach that integrates both the structure and semantics of the ontology. The modularization process with COMET is done in 2 steps: the segmentation of the ontology and the extraction of the module.

Segmentation is the first phase of the extraction process. It is based on the assumption that when a concept is pertinent, then its sub concepts are equally pertinent (Stuckenschmidt & Schlicht, 2006)(Ghosh, Abdulrab, Naja & Khalil, 2017a). The segmentation phase therefore consists in determining the limits of the module. To do this, the user must first of all enter a list of terms, then define a semantic threshold and a hierarchical depth. Once all these parameters have been defined, the algorithm will first of all search for the concepts corresponding to the list of terms and will save them in a formerly defined stack: this is the initialization. This stack containing the starting concepts is the initial core of the module. The purpose of this approach is to extend the base over time as the semantically close concepts are discovered. Thus for each concept, it is verified that it possesses semantic relationships stemming from the object type. In the case where there is, then for each of these relationships, the concept to which it is semantically linked will be searched and the distance between these two concepts will be evaluated. In a scenario where this distance is greater than or equal to the fixed semantic threshold, the new concept thus discovered will be added to the stack containing the list of pertinent concepts. Also, depending on the user's defined hierarchical depth, the offspring of the concept judged pertinent will equally be added to the stack. In the event where the semantic distance is less than the threshold and cannot find a semantically linked concept to the pertinent concept already present in the stack, the algorithm will call upon the hierarchy of ontology to verify if the initial concept has sub concepts. If sub concepts exist, then for each of them, the algorithm will start searching again for semantic relationships. And, if there exists one, it will evaluate the semantic inter-concept distances in relation to the threshold. In a case where there exists no semantic relationships or if the threshold is still not attained, the algorithm will search even further in the hierarchy until it finds a concept with satisfactory semantic relationships. The calculation of the semantic distance is done using Wu and Palmer's distance, which is a practical and intuitive measure based on the length of the path between two concepts of the same hierarchy. It calculates the distance separating two concepts in the hierarchy according to their position relative to the root (Wu & Palmer, 1994). It is obvious that two concepts found at the same depth in the hierarchy will have a higher similarity than concepts at different levels of the hierarchy. The measurement of Wu and Palmer is defined as follows:

$$Sin(C_1 + C_2) = \frac{2 + \text{dept(C)}}{depth_c(C_1) + depth_c(C_2)}$$

where c, the most accurate common subsuming, depth(c) is the length of the path between c and the root of the hierarchy, $depth_c\ (c_i)$ is the number of arcs between c_i and the root passing through c. This measure ranges between 0 and 1. The segmentation algorithm is presented by the following codes.

```
Algoritm COMET_Segmentation
Require: Ontology O
Require: domain : OntClass
Ensure: relevantClasses : Vector<OntClass>
1: procedure computeRelevantClasses(domain)
2: relations ← getOutgoingObjectProperties(O, domain)
3: if (relations.size() > 0) then
4: for each op ∈ relations do
5: range ← op.getRange()
6: if range 6= 0 then
7: if SemanticDistance(domain, range) > threshold then
8: if relevantClasses.contains(range) then
9:   continue
10: else
11: relevantClasses.add(range)
12: end if
13:       else
14:       if domain.hasSubclass() then
15:       relevantSubClasses ← domain.listSubClasses()
16:       for each c ∈ relevantSubClasses do
17:   computeRelevantClasses(c)
18:       end for
19:       end if
20:       end if
21:       else
22:           relevantClasses.remove(range)
23:        end if
24: end for
25: else
26: if domain.hasSubclass() then
27: relevantSubClasses ← domain.listSubClasses()
28:    for each c ∈ relevantSubClasses do
29:       computeRelevantClasses(c)
30:    end for
31:  end if
32: end if
33:  return relevantClasses
34: end procedure
```

After segmentation, the second phase of the modularization process follows, it is the extraction of the sub ontology that has been identified and stored in memory. To do this, the authors proceed with the pruning of the graph (tree) representing the ontology in looking through its entirety in order to delete non-pertinent concepts. It is important to indicate that the authors work with two sets of data, one containing all

the relevant concepts and the other containing all the concepts judged irrelevant. When a concept is deleted from the ontology, all the semantic and hierarchical relationships that index it are equally deleted. Once pruning is completed, the new ontology will be exported into a file. Pruning is by far the most suitable technique for the extraction phase, in that it would have been clearly more complex to save both the relevant concepts and their differing properties in data structures (list type) and uniquely from these, to generate the ontological module.

Figure 1, illustrate the entire modularization process from COMET, with input parameters: a list of terms composed of a and b, a semantic threshold fixed at 0.4, and a hierarchical depth of 1. At the start, the authors have an ontology of 15 concepts and after modularization, the authors obtain a module of 7 concepts. The authors note that the concepts d, g and i are identified and added via hierarchical relationships according to the depth fixed in relation to a pertinent concept already present in the stack.

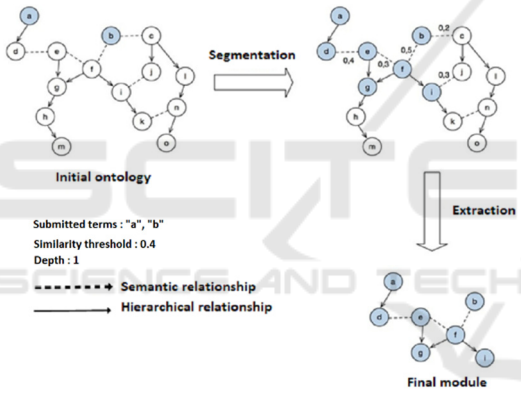

Figure 1: Modularization process with COMET.

4 EXPERIMENTATIONS AND RESULTS OF COMET

A module is an ontology, and as such, the quality assessment approaches of ontologies apply equally to it. Although some evaluation techniques analyze the structure of the ontology represented by the content of the ontology, other techniques focus mainly on the comparison of the content of the ontology itself, either with another existing ontology whose quality would have been judged satisfactory, or with another alternative representation of the domain of knowledge in question, such as a corpus of documents. This comparison is made in order to determine the extent to which this content modulates all relevant aspects of the domain to be described. Based on these principles and in order to finalize their validation protocol, the authors will first of all define their evaluation criteria and then establish the procedure by which the different extracted modules will be evaluated.

4.1 The Validation Protocol

The validation protocol is essentially based on a comparative analysis of the modules obtained by COMET to those generated by the OWL Module Extractor1 and SegmentationApp tools. It is in fact a Gold standard type evaluation having as reference the ontology from which the different modules are extracted (Dellschaft & Steffen, 2006). The aim of our validation protocol is to evaluate both the lexico-semantic aspect and the structural aspect of the different modules. To do this, in addition to OntoMetrics API, the authors also use the OntoEval API to calculate the precision and recall of modules as a function of those obtained by the reference ontology. Each of these APIs take two parameters for input: the reference ontology and the generated or calculated ontology which, in this case is the extracted module.

The API evaluation process is therefore carried out in three phases. In the first, the API evaluates the source ontology according to Alani and al. This evaluation consists first of all in calculating the quality metrics of the source ontology, then in assigning the maximum score which is 100% to each returned metric result. This systemic attribution of a maximum score is justified by the fact that our source ontology is considered as an ontology of reference, that is, an ontology of satisfactory quality.

In the second phase, the modules are evaluated in turn using these same APIs. The metrics are calculated for each module. The results obtained in the calculations of the module's quality metrics are compared with those of the reference ontology. Based on the scores attributed to those of the reference ontology, the scores of the metrics in the module are calculated. In order to illustrate this approach of evaluation, let us consider an ontology O and a module M for which the density DEM is calculated. At the end of the calculation of the density of the initial ontology O, the authors obtain a result of DEMO = 12.22; while the calculation of the density of the module gives a result of DEMM = 5.21.

In making a connection between the densities of the module to that of the reference ontology, the authors obtain the score of the module density, which in this case, is of 42.63%.

$$Score_{DEM} = \frac{DEM_M}{DEM_O}(\%)$$

The last phase of the evaluation consists simply of comparing the metric scores of the modules generated by COMET, OWL Module Extractor and SegmentationApp amongst them in order to determine which one is most relevant depending on the metrics used.

To summarize the validation protocol, the authors compare each module generated from the different extraction tools with the source ontology to obtain scores. Then, compare the scores of the naked modules with each other in order to determine the highest score. It is the modules with the best metric scores or the best total score that allows the judgment of the efficiency of an extraction tool.

It is expected that the module that will maximize the characteristics of the reference ontology will end up with the highest scores. The metric scores depend on factors such as the number of concepts, the number of properties and the taxonomy of the module or the ontology of reference to name just a few. Table 1and 2 shows all ontologies modularized, the signature of the module (the list of terms to be covered), the extraction tools together with the size of ontologies and the different modules extracted from them. As for the tests, the authors have fixed the hierarchical depth of COMET at 1 while the semantic threshold S is variable.

Table 1: Ontologies and terms.

Ontologies	#C	Terms
sweto simplified	115	Person, Event
cmt	29	Co-author, Review
confOf	38	Participant, Conference
iasted	140	Delegate, Activity, Place, Fee, Deadline, Submission
Conference	59	Submitted contribution, Topic
edas	103	Author, Paper, Workshop, Program, SocialEvent, Call, Country, Document, ReviewRating, Rejectedpaper
paperdyne	45	Conference, Location
OpenConf	62	Paper Review, People
ekaw	73	Review, Possible Reviewer
sigkdd	49	Author, Award, Sponzor
travel	34	Destination, AccommodationRating
PPOntology	88	Case, Biotic Disorder, Organism, Disorder, Mineral, Plant Observatio, Pesticide, Abnormality, Bacterium, Roots
OTN	179	Feature, Service, Road Element, Manoeuvre, Temperature, Face, Tourism, Accident, Forest, Junction

Table 2: Extraction of modules using approaches from Seidenberg and al., Cuenca and al. and COMET.

			COMET				CUENCA	
	SEIDENBERG		S = 0.2		S = 0.0		Bottom	
O	#C	Ratio	#C	Ratio	#C	Ratio	#C	Ratio
A	4	3,48%	21	18,26%	21	18,26%	4	3,48%
B	7	24,14%	3	10,34%	24	82,76%	7	24,14%
C	10	26,32%	5	13,16%	5	13,16%	10	26,32%
D	48	34,29%	56	40%	56	40%	54	38,57%
E	10	16,95%	21	35,59%	23	38,98%	11	18,64%
F	20	19,42%	46	44,66%	47	45,63%	24	23,30%
G	18	40%	17	37,78%	18	40%	2	4,44%
H	22	35,48%	7	11,29%	29	46,77%	58	93,55%
I	4	5,48%	15	20,55%	15	20,55%	4	5,48%
J	16	32,65%	20	40,82%	21	42,86%	16	32,65%
K	1	2,94%	15	44,12%	15	44,12%	25	73,53%
L	17	19,32%	33	37,50%	41	46,59%	87	98,86%
M	25	13,97%	59	32,96%	59	32,96%	52	29,05%
Average		21,11%		29,77%		39,43%		36,31%

Legende: column O = Ontologies.
A = sweto simplified, B =cmt, C= ConfOf, D = Conference, E= iasted, F = edas, G=paperdyne, H=OpenConf, I=ekaw, J = sigkdd, K = travel, L =PPOntology, M = OTN, #C = Size.

4.2 Analysis of the Results

In order to determine the most efficient extraction tool, the authors evaluated each module extracted from the ontologies listed in Table 1 according to the metrics of Alani and al. (Alani & Shadbolt, 2006). Independent of the ontology to modularize and the modules generated, it is a question of calculating the score of each metric used, then comparing the scores of the modules obtained between them. Each of the metrics is intended to evaluate a specific aspect of the ontological module.

Class Matching *(CMM)* assesses the ability of an ontology to cover a given set of terms. In this case, it is intended to calculate the percentage of both exact and partial matching between the classes of the module and those of the reference ontology, in order to find the degree of representation of the classes of the initial ontology in the extracted module. It is obvious that the larger the module, the more likely it is to cover the set of concepts (corresponding to the list of searched terms) presented in the initial ontology. After analyzing of Table 1, it is noted that in most cases of ontologies studied, the COMET modules obtain a better score from the *CMM*, which suggests that the modules they generate are more representative of the initial ontology.

The density (DEM) expresses the degree of precision of a given concept, that is, the richness of

its attributes. This definition implies that an adequate representation of a concept must be able to offer as much information as possible about it. The density depends essentially on the number of subclasses, the number of attributes associated with this concept or betters still, the number of related concepts. In the present case, it is observed that the modules generated using COMET have a significantly higher DEM score than those extracted from the other tools (Figure 3). This is explained by the fact that COMET crosses the ontology horizontally through semantic relationships in order to search for the relevant concepts and once they have been identified, the hierarchy is gone through in order to add their descendants to the module. However, it should be remembered that in the case of the COMET tool, the hierarchic depth is a parameter which, like the semantic threshold, is defined by the user. It is through this parameter that the expert is able to determine the limit at which the algorithm should stop during the course of the taxonomy. The hierarchical depth goes in peer with the level of specialization, which is an important factor in the calculation of density. From there, the authors can deduce that the deeper the hierarchical depth of the COMET, the more the score of the DEM module increases.

The density does not depend on the number of concepts in the module. This hypothesis is verified in the cases of ontologies 8, 11 and 12, where it is noted that, although the size of the top-modules extracted with OWL Module Extractor represent about 75% to 95% of the size of source ontologies, their score remains less than that of the modules extracted with COMET whose average size remains largely inferior to 50% (Figure 3).

The semantic similarity (SSM) calculates the proximity between the classes corresponding to the terms searched in the ontology. Thus, concepts that correspond to these terms must be linked either by hierarchical relationships or by semantic relationships. This measurement is based on the calculation of the shortest path between pairs of concepts. It can be seen that in the majority of cases, the modules generated with COMET have a SSM elevated score, which is in line with the COMET approach where the detection of relevant concepts is done through the path of object type properties. The more semantic relationships there are in the module, the more differing paths there will be between peer of concepts in the module. Similarly, the more the concepts of a module are interconnected by semantic and hierarchical relationships, the higher the SSM score of this module will be.

The centrality (CEM) measures the degree of representation of all the concepts corresponding to the terms searched for in an ontology. It is based on the calculation of the shortest route through each concept of ontology. The most solicited concepts during the ontology process have a higher centrality than those of other concepts. In view of the results shown in figures 2, 3, 4 and 5 below, it is noticed that in 7 of the 13 ontology cases studied, the modules generated by COMET have a significantly higher CEM score than those of modules generated by OWL Module Extractor and SegmentationApp. The authors can justify this result by the fact that these last two tools have produced modules with few semantic relationships, thus reducing the number of paths between pairs of concepts. This observation leads to the conclusion that the structure's course in the calculation of the CEM of the OWL Module Extractor and SegmentationApp modules is mainly done through hierarchical relationships. It is important to remember that the density, the centrality and the semantic similarity are structural measures which are founded essentially on the degree of interconnection of concepts in the module.

In Scenario 1, the results of which are shown in Figure 2, the authors consider both class matching and density to be priorities. To do this, the authors distribute the weights as follows: 0.4CMM, 0.4DEM, 0.2SSM, 0.0CEM. In this case, the authors estimate that centrality has no impact on the global quality of the module.

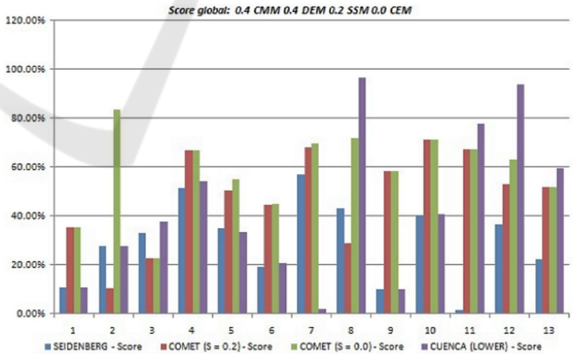

Figure 2: Module score as a function of the metrics of Alani and al.: scenario 1.

In scenario 2, the weights are assigned equally to each metric, resulting in the following distribution: 0.25CMM, 0.25DEM, 0.25SSM, 0.25CEM. The authors estimate that no one metric is more important than another. The results obtained are shown in Figure 3.

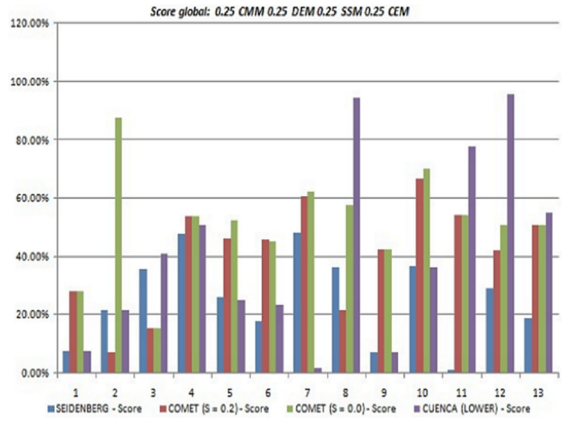

Figure 3: Module score as a function of the metrics of Alani and al.: scenario 2.

In Scenario 3, priority is given to the correspondence of classes, followed by density, then similarity and finally centrality. The weights are assigned as follows: 0.4CMM, 0.3DEM, 0.2SSM, 0.1CEM. The results obtained are shown in Figure 4.

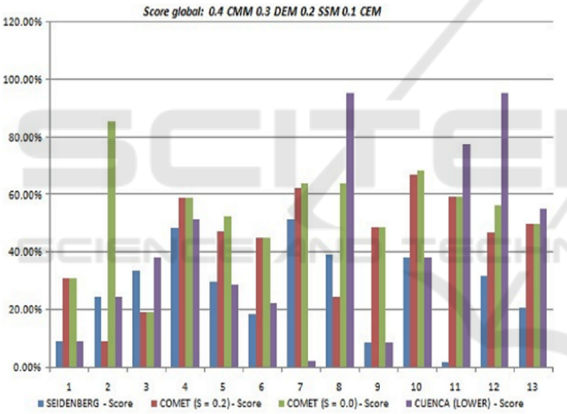

Figure 4: Module score as a function of the metrics of Alani and al.: scenario 3.

In scenario 4, whose results are shown in Figure 5, semantic similarity is considered to be the most important metric, while centrality remains the least important, resulting in the following distribution: 0.2CMM, 0.3DEM, 0.4SSM, 0.1CEM.

After analyzing Figures 2, 3, 4 and 5, it is noted that in the majority of ontologies studied, more precisely in 8 of the 13 cases of ontologies used, COMET produced the modules whose quality is the most satisfactory according to Alani and al. measures. The hierarchic depth and semantic threshold are essential parameters that allow us to control the size of the modules. The more an ontology is rich in object type relationships, the more likely the COMET algorithm will return larger modules.

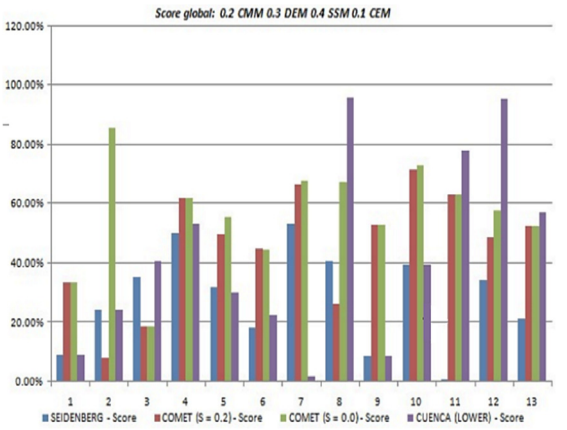

Figure 5: Module score as a function of the metrics of Alani and al.: scenario 4.

Hence the need for the user to define a semantic threshold, in order to control the proportions of the modules.

5 CONCLUSION

In this paper, the authors propose a new technique of modularization of ontologies, which is a combination of existing structural and semantic approaches. Based on this technique, the authors have implemented a prototype module extraction tool: COMET. Tests were carried out on different modules generated using extraction tools, including COMET, in order to demonstrate that the authors could derive benefits from both the structure and semantic of an ontology, in order to produce modules of satisfactory quality.

COMET is a tool that, from an ontology and a list of terms, generates a module. The latter represents a segment that is made up only of concepts judged relevant according to a segmentation algorithm. This algorithm is based on two parameters which are the hierarchical depth and the semantic threshold; essential parameters allowing to regulate the taxonomy path of source ontology and to control the proportions of the module to extract.

Potentially relevant concepts are observed through semantic relationships. As for the hierarchy, it intervenes in two cases, either when searching in a taxonomy for concepts having object type properties, more specifically, concepts linked to others by a relationship whose weight is superior or equal to the threshold; or when it is to add the derivative of the relevant concepts to the module. After the process of segmentation of the ontology, a second algorithm proceeds to its pruning in deleting all concepts

deemed irrelevant in order to generate the final module.

The authors have also set up a validation protocol to evaluate the quality of the different extracted modules using a number of metrics. The validation of the tools is based on a comparative study of the modules generated compared to a reference ontology, which this case is the source ontology.

On the strength of the results obtained during the tests, it was observed that density is an essential characteristic for it represents the level of completeness of an ontology. A dense ontology is an ontology rich in semantic relationships, that is, an ontology whose classes are clearly defined. From the series of tests, it can be concluded that the more dense an ontology is, the more the module returned by COMET is as well. However, the authors believe that, given the current limitations of COMET and with a view to future development, improvements could be made both at the algorithm as well as at the tool implementation levels. Thus, the following points can be addressed:

- Choose a better semantic distance. It would be interesting to look at another measure such as a semantic distance based on WordNet, because in addition to being a database containing the lexical semantic content, WordNet equally presents an ontology. This representation can be used to evaluate the semantic distance between two concepts not according to their position in the ontology to modularize, but rather according to their position in the WordNet taxonomy.
- Propose an empirical approach which can set the semantic threshold and the hierarchical depth. The expert must carry out a certain number of tests in order to find the ideal threshold, hence the necessity to elaborate a protocol by which these tests are to be conducted.
- To be inspired by methods of graph exploration based on heuristics or incremental deepening during the course of the ontology and the addition of the derivative of the relevant concepts to the module. Indeed, it would be a question of exploring the nodes of the graph representing ontology according to the weights associated with them. Depending on the depth set by the user, the algorithm cannot systematically add the concepts derivative identified, but rather add concepts belonging to this derivative based on their weights.

REFERENCES

Alani, H., Brewster, C., Shadbolt, N. (2006). Ranking ontologies with aktiverank. In Proceedings of the 5th International Conference on The Semantic Web (ISWC'06), Springer-Verlag, pages1–15.

Alani, H., Brewster, C. (2005). Ontology ranking based on the analysis of concept structures. In Proceedings of the 3rd International Conference on Knowledge Capture, ACM, pages 51–58.

Algergawy, A., Babalou, S., and Konig-Ries, B. (2016). Anew metric to evaluate ontology modularization. In SumPre@ ESWC.

Babalou, S., Kargar, M. J., and Davarpanah, S. H. (2016). Large-scale ontology matching: A review of the literature. In Web Research (ICWR), 2016 Second International Conference on, pages 158–165. IEEE.

Cuenca, B., Grau, Parsia, B., Sirin, E., AdityaKalyan. (2005). Automatic partitioning of owl ontologies using e-connections. In Proceedings of the 2005 International Workshop on Description Logics (DL-2005).

Dupont, P., Callut, J., Dooms, G., Monette, J., Deville, Y. (2006). Relevant subgraph extraction from random walks in a graph. Technical report, Catholic University of Louvain, UCL/INGI.

Ghosh M., Abdulrab H., Naja H., and Khalil M. (2017a). Ontology Learning Process as a Bottom-up Strategy for Building Domain-specific Ontology from Legal Texts.in proceedings of 9th International Conference on Agents and Artificial Intelligence. PP 473-480.

Ghosh M., Abdulrab H., Naja H., and Khalil M. (2017b). Using the Unified Foundational Ontology (UFO) for Grounding Legal Domain Ontologies. In proceedings of 9th International Conference on Knowledge Engineering and Ontology Development. Madeira, Portugal, Hal-01644015.

Khan, Z. C. and Keet, C. M. (2015). Toward a framework for ontology modularity. In Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists, page 24. ACM

Klaas Dellschaft, K., Staab, S. (2006). On how to perform a gold standard based evaluation of ontology learning. In The Semantic Web-ISWC 2006, Springer, pages 228–241.

Jiang, J. J., Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of 10th International Conference on Research in Computational Linguistics (ROCLING X), Computing Research Repository CoRR, pages 19–33.

Leclair, A., Khedri, R., Marinache, A. (2019). Toward Measur ing Knowledge Loss due to Ontology Modularization. In Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019, pages 174-184 ISBN: 978-989-758-382-7. SCITEPRESS – Science and Technology Publications, Lda.

Movaghati, M. A. and Barforoush, A. A. (2016). Modularbased measuring semantic quality of ontology.

In Computer and Knowledge Engineering (ICCKE), 2016 6th International Conference on, pages 13–18. IEEE

Palmisano, I., Tamma, V., Payne, T., & Doran, P. (2009). Task oriented evaluation of module extraction techniques. In The Semantic Web-ISWC, Springer, pages 130–145.

Stuckenschmidt, H., Klein, M. (2004). Structure-based partitioning of large concept hierarchies. In The Semantic Web–ISWC2004, Springer, pages 289–303.

Stuckenschmidt, H., Schlicht, A. (2009). Structure-based partitioning of large ontologies. In Modular Ontologies, volume 5445 of Lecture Notes in Computer Science, Springer-Verlag, pages 187–210.

Xue, X. and Tang, Z. (2017). An evolutionary algorithm based ontology matching system. Journal of Information Hiding and Multimedia Signal Processing, 8(3):551–556. High heterogeneity; Matcher combination; Matching system; Ontology matching; Optimal model; Recall and precision;Semantic correspondence;State of the art.

Wu, Z., Palmer, M. (1994). Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pages 133–138.