

Using OpenCyc and Domain Ontologies for Ontology Learning from Concept Maps

Amhed Suárez-Rodríguez¹, Alfredo Simón-Cuevas¹, José A. Olivas²

¹Instituto Superior Politécnico José Antonio Echeverría, Cujae, La Habana, Cuba.

²Universidad de Castilla-La Mancha, Ciudad Real, España.

Abstract

In this paper, an extended method for ontology learning from concept maps is presented. Concept maps are a flexible and informal knowledge representation, while ontologies are semantically formalized representations oriented to be processed by intelligent systems. The mapping between them is a formal transformation based on semantics inference in concept maps. OpenCyc and domain ontologies are used in a combined way with WordNet for increasing the coverage in the semantics inference in the concept map and the method's applicability. The novel proposal is experimentally evaluated using concept maps from published works with satisfactory and promising results.

Keywords: concept maps, semantic inference, and ontology learning

1. Introduction

In most scientific domains, information needs sometimes to be analyzed and processed by machines. In the knowledge representation oriented to the semantic analysis and processing by machines, context in which a certain degree of formalization is required, the development and use of ontologies is increasingly common. Several methods, methodologies and tools for ontology constructions have been reported; arising the ontology engineering as a new discipline [3]. However, the processes for design and creation of ontologies, the tools available, such as Protégé [14], and specification languages are still complex for non-experts in this subject. This suggests the use of a knowledge representation form that can be used naturally by humans and integrated with ontologies in such a way that the latter can be obtained automatically; such as the Concept Maps (CMs).

CMs are graphical tools for organizing and representing knowledge [13], were defined for application in the learning process and they are easy to be created, flexible and intuitive for people. They include concepts, linking words or linking phrases, to specify the type of relationships between concepts, and propositions, which contain two or more concepts, connected using linking phrases to form a meaningful statement.

CMs are one of these human-friendly knowledge representations, very useful for the ontology engineering, mainly as informal representation of a conceptualization

in the ontology construction process [6],[10],[17], and in methods to automatically ontology construction [1],[4],[7],[9],[18]. Many of these approaches show limited in the semantic inference capacity on the CM, without reducing its flexibility. This paper presents a novel proposal for increasing these capabilities, extending an ontology learning method with other knowledge bases as OpenCyc and domain ontologies. Coverage is the metrics defined to measure the capacity in terms of semantics inference implicit in the CM. The proposal is evaluated experimentally using 50 selected CMs from scientist papers and about environmental domain.

2. Concept Mapping in the Ontology Engineering

CMs are a graphically rich technique for organizing and representing knowledge and proposed by Novak and Gowin [13]; especially defined for application in the learning process. They are easy to be created, flexible and intuitive for people, and are especially useful for conceptual-knowledge management. They include *concepts*, labeled relationship by *linking-words* and *propositions*. CMs' propositions contain two or more concepts connected using linking-words (*lw*) and sometimes are called semantic units, or units of meaning.

CMs usually have a hierarchical structure and express the most significant understanding of a knowledge domain, but they can integrate concepts from different domains. They have been considered a very useful for ontology engineering processes, mainly as informal representation of a conceptualization in ontology construction process [6],[10],[17], and as starting point in methods to automatically ontology construction [1],[4],[7],[9],[18]. The mapping between CMs and ontologies creates the bases for the collaborative development of ontologies in a more intuitive, friendlier manner for humans and allows the reuse of the knowledge represented in CMs by knowledge management systems

The automatically ontologies construction from CMs is pursued through structural mapping between both representations [8],[17]. Knowledge in ontologies is formalized using classes, properties, and instances [17], while in CMs this semantic aspects must be inferred. Knowledge in CMs is not formalized and its semantics is implicit, because they have been defined to be used and interpreted by people and not by computer systems.

Some contributions are based on a conceptualization represented in CM form and carry out the formalization

and an encoding in OWL of that knowledge, as a two-part process: semantic inference in CMs and codification using an ontology formal language. Several difficulties are present in these methods related to: insufficient semantic inference from CMs, use CMs formalizations on reducing its flexibility, and they don't treat CMs concepts ambiguity, on which propose improvements in [18]. However, implicit semantic inference from CMs, in which flexibility in its construction is kept, should be increased to obtain better ontologies.

In this paper is proposed and experimentally demonstrated these capabilities increase, including the ontology learning method, using already established knowledge resources. We have developed an extension of the method reported in [18], with OpenCyc upper ontology and mechanisms for processing an ontology corpus, related to a specific domain. This extension is based on utility of reusing existing knowledge for creating new knowledge, using especially well formalized knowledge.

3. Knowledge Bases to Semantic Inference in Concept Maps

The use of knowledge bases for semantic inference associated to concepts and propositions in CMs for ontology learning have been previously considered [1],[4][17]; particularly WordNet [12]. The reuse of existing knowledge for creating new knowledge is frequently used today.

WordNet is a lexical knowledge base, whose basic structure is the *synset*. The *synset* defines the meaning of a word, which, in the case of polysemy, can be found in various synsets. *Synsets* form a semantic network and are interconnected among themselves by several types of relations, some of which are used for semantic inferring in the CM [1][17], such as *hypernymy-hyponymy* (to infer class/subclass relations between concepts) and *meronymy-holonymy* (to infer part/whole relations between concepts). WordNet can be used as ontology if its links are associated to a formal semantics.

However, WordNet provides satisfactory results in conceptualizations of general domain, whose concepts are simple terms, and not in specific domains of knowledge. Therefore, the only used of WordNet as knowledge base in the required semantic inference in the CM is considered insufficient and the covering of the method spreads to be low. In this paper, we propose extending and combine the knowledge bases of the ontology learning method with the upper ontology OpenCyc and the use of domain ontologies contained in a corpus for reducing the exposed limitations.

The upper ontology OpenCyc was proposed to extend the general domain terminology and use other taxonomic and semantics relations not included in WordNet. At the same time, domain ontologies allows using knowledge bases of specific domain in the ontology learning from CM, therefore it would be useful for coverage increasing in specific domain CM and extending the applicability of the method. The use other knowledge bases (in addition

to WordNet) for automatic ontology construction from CM has not been reported.

3.1. OpenCyc

OpenCyc is the free version of the Cyc knowledge base [10], one of the most important results of Cyc Project. OpenCyc represents an upper ontology, which included a taxonomy of concepts, relations, properties, restrictions and instances codified in CycL are included in OpenCyc, for describing objects and events related to the daily life, and using 250 000 words and 15 000 predicates. OpenCyc has been used in several contexts, such as: word sense disambiguation, ontology mapping, and in the metadata semantic integrations, but not in the ontology learning from CMs. This upper ontology can be used for semantics inference about classes, subclasses and instances in the CM, through the concept identification in predicates:

- *genls*: for representing class - subclass relations, e.g.: $\#\$genls C' C$, where C' is subclass of C .
- *isa*: for representing an instance of a collection, e.g. $\#\$isa I C$, where I is an instance of C .

The most of information in this upper ontology is represented through these types of predicate.

3.2. Domain Ontologies Processing

Domains ontologies allow identify semantic information related to concepts and relations between them in the CM, such as: class-subclass, class-instance, class-property, among others, mainly in specific domain contexts. An Ontology Corpus (OC) is defined for containing the domain ontologies to be used, which may have one or more ontologies according to the necessities and the users' readiness. The use of OWL language [19] for codifying all domain ontologies is required.

In OC, not all domain ontologies provide the same semantic information; therefore the occurrence of semantic conflicts is possible. A procedure based on selecting the most relevant domain ontologies (*reference ontologies*) for the CM that is being processed was proposed, as a way for reducing the problem previously referred. Moreover, knowledge contained in the corpus can be use more efficiently. Figure 1 shows a graphical view of the reference ontologies selection procedure.

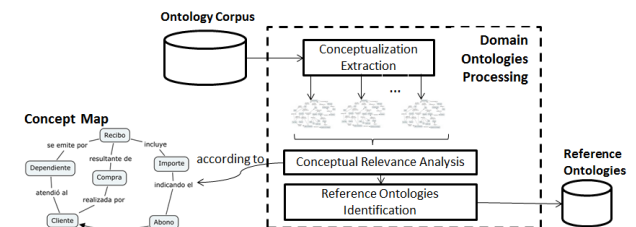


Fig. 1: Procedure for selecting reference domain ontologies.

In the selection of relevant domain ontologies to specific CM the comparison between the knowledge represented in both knowledge models is required, but this is not directly appropriate. Therefore, the use of a common representation model, specifically CM model, was considered as solution.

The procedure is defined in three steps: *conceptualization extraction*, *conceptual relevance analysis*, and *reference ontologies identification*. In first step, the conceptualization of domain ontologies is extracted and represented in CM form, which is carried out using the method reported in [8] for automatically obtaining a CM form an ontology. In second step, the conceptual relevance analysis of the domain ontologies is carried out. The analysis is based on the identification of conceptual connections between a CM and the domain ontologies (whose knowledge is resented in CM form) through apply the CMs comparison algorithms reported in [2]. This algorithm allows obtaining a *similarity value* between two CMs, using fewer computational resources in its implementation, and it is one of the most used for this purpose. When all conceptualizations of the domain ontologies are analyzed and an ordered list (by *similarity value*) is obtained the step is finished. In the third step, the 25% most similar conceptualization to the CM being processing, according results of previous step, are identified and whose corresponding reference ontologies are selected as knowledge bases for semantic inference process.

4. Ontology Learning Method

The method for obtaining the OWL ontology from a CM is organized in three phases: *preprocessing*, *semantic mapping* and *OWL codification* [18], as shown in Figure 2. Four components are defined for the implementation of the system: *parser*, *disambiguator*, *semantic interpreter* and an *OWL codifier* [18].

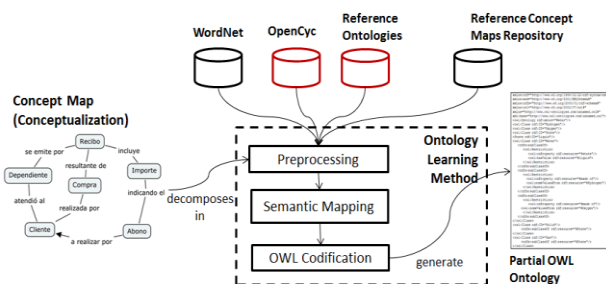


Fig. 2: Ontology learning method.

4.1. Preprocessing Phase

In the *preprocessing* phase, the *parser* analyzes the CM, identifying propositions and their parts (*concepts* and *linking-phrases*), creates a proposition set (*PS*) and a concepts set (*CS*). PS have $(C_o, \textit{linking-phrase}, C_d)$ as basic structure, where C_o is the origin-concept and C_d is the destination-concept in the proposition. The *disambiguator* determines the most rational sense of the concepts, using

the algorithm reported in [16]. Relevant semantic information is retrieval from WordNet, OpenCyc, and the reference ontologies (preciously selected according to procedure presented in section 3.2). Several sets are created, as follows:

- A $C_{\textit{class/subclass}}$ set with the pair of concept (C, C') , if: the *synsets* of C and C' are directly related by a *hyperonymy* or *hyponymy* relation in WordNet or a connection between them using these relations can be created; or a $(\#\$ \textit{genls } C' C)$ predicate is included in OpenCyc, or a connection between them using these predicate can be created; or C is coded as *owl:class* and in C' code's the $\langle \textit{rdf:SubClassOf... "C"} \rangle$ in one of reference ontologies is specified.
- A $C_{\textit{class/instance}}$ set with the pair of concept (C, C') , if: a $(\#\$ \textit{isa } C' C)$ predicate is present in OpenCyc; or the $\langle C' \textit{rdf:ID} = "C" \rangle$ specification in one of the reference ontologies is included.
- A $C_{\textit{mero/holo}}$ set with the pair of concept (C, C') , if the *synsets* of C and C' are directly related by a *meronymy* or *holonymy* relation in WordNet.
- A $C_{\textit{mero/holo-type}}$ set with the pair of triplets $(C, C', \textit{relation type})$, if the *synsets* of C and C' are directly related by some *meronymy type* or *holonymy type* relation in WordNet; in *relation type* the *meronymy* or *holonymy type* is registered.
- A $C_{\textit{prop/object}}$ set with the triplets (C, Pr, C') , if Pr is specified as *owl:ObjectProperty* in one of the reference ontologies, where C and C' are specified as its *rdfs:domain* and *rdfs:range*, respectively.
- A $C_{\textit{restr/prop}}$ set with the triplets (C, Pr, C') , if C is specified as *owl:class*, and Pr is specified as its property, with the value C' as restriction.

4.2. Semantic Mapping Phase

In the *semantic mapping*, the *semantic interpreter* analyzes the CM identifying several semantic specifications from concepts and the propositions (*P*) included in PS, according to OWL DL descriptions [19], such as: *classes* and *relations between them* (simple classes), *union and intersection classes* (complex classes), *instances*, *properties* and some *restriction*, and *property characteristics*. A set of heuristic rules grouped by type of semantic inferring to form a *semantic-inference engine* are defined. These rules use the set of predefined linking-phrases (*l-p*) included in the several categories [17], and the sets generated in the *preprocess* phase. A *saliency* value is attributed to each rule to define their activation order. Some rules have been modified, respect to previous version [18], for using the recovered information from OpenCyc and the reference ontologies. When all propositions of the MC are analyzed and all rules are activated the phase is finished. Rules are formally defined as follow:

Rules to infer simple classes:

R1 (salience = 8):

- If $P = (C_o, l-p, C_d) \in PS \wedge l-p \in CC$ then $PS = PS - \{P\}$, $S_CS = S_CS \cup \{(C_o, C_d)\}$.
- If $P = (C_o, l-p, C_d) \in PS \wedge l-p \in CC^{-1}$ then $PS = PS - \{P\}$, $S_CS = S_CS \cup \{(C_d, C_o)\}$.

R2 (salience = 8): If $P = (C_o, l-p, C_d) \in PS \wedge l-p \notin CC \wedge l-p \notin IC \wedge (C_o, C_d) \in C_{class/subclass}$ then $PS = PS - \{P\}$, $S_CS = S_CS \cup \{(C_o, C_d)\}$, $CC = CC \cup \{l-p\}$.

Rules to infer complex classes

R3 (salience = 6): If $U = (UC, \{C_i / (UC, C_i) \in S_CS\}) \wedge U \notin Union$ then $Union = Union \cup \{U\}$.

R4 (salience = 6): If $I_C = (IC, \{C_i / (C_i, IC) \in S_CS\}) \wedge I_C \notin Intersection_{classes}$ then $Intersection_{classes} = Intersection_{classes} \cup \{I_C\}$.

R5 (salience = 2): If $I_{C-Pr} = (IC, C / ((C, IC) \in S_CS, (C', IC) \notin S_CS), Pr / (IC, Pr, V) \in S_CPV_{hasValue}, V) \wedge I_{C-Pr} \notin Intersection_{class-property}$ then $Intersection_{class-property} = Intersection_{class-property} \cup \{I_{C-Pr}\}$.

Rule to infer instances

R6 (salience = 8):

- If $P = (C_o, l-p, C_d) \in PS \wedge l-p \in IC \wedge (C_o, C_d) \notin C_{class/subclass}$ then $PS = PS - \{P\}$, $S_CI = S_CI \cup \{(C_o, C_d)\}$.
- If $P = (C_o, l-p, C_d) \in PS \wedge l-p \notin IC \wedge [(C_o, C_d) \in C_{class/instance}]$ then $PS = PS - \{P\}$, $S_CI = S_CI \cup \{(C_o, C_d)\}$.

Rules to infer properties

R7 (salience = 8): If $P = (C_o, l-p, C_d) \in PS \wedge (l-p \in PC \vee (C_o, C_d) \in C_{mero/holo}) \vee (P \in C_{prop/object})$ then $PS = PS - \{P\}$, $S_CP = S_CP \cup \{(C_o, l-p, C_d)\}$, $PC = PC \cup \{l-p\}$.

R8 (salience = 8):

- If $P = (C_o, l-p, C_d) \in PS \wedge l-p \in PVC$ then $PS = PS - \{P\}$, $S_CPV = S_CPV \cup \{(C_o, l-p, C_d)\}$;
- If $P = (C_o, l-p, C_d) \in PS \wedge (C_o, C_d, type) \in C_{mero/holo-type}$ then $PS = PS - \{P\}$, $S_CPV = S_CPV \cup \{(C_o, prop-erty^l, C_d)\}$, $PVC = PVC \cup \{l-p\}$.
- If $P = (C_o, l-p, C_d) \in PS \wedge (P \in C_{restr/prop})$ then $PS = PS - \{P\}$, $S_CPV = S_CPV \cup \{(C_o, l-p, C_d)\}$.

Rule to infer properties restrictions

R9 (salience = 6): If $(C, Pr, I) \in S_CPV \wedge \exists (C', I) \in S_CI$ then $S_CPV_{hasValue} = S_CPV_{hasValue} \cup \{(C, Pr, I)\}$.

Rules to infer properties characteristics

R10 (salience = 2): If $(C, Pr, V) \in S_CPV_{hasValue} \wedge (\forall C' \forall (C, l-p, C') \in PS_KB[(f-e = Pr \vee Synonym(l-p, Pr)) \rightarrow (C' = V \vee Synonym(C', V))])$ then $Pr_{functional} = Pr_{functional} \cup \{Pr\}$.

R11 (salience = 2): If $(C, Pr, V) \in S_CPV \wedge (\forall V' \forall (V, l-p, C') \in PS_KB[(f-e = Pr \vee Synonym(l-p, Pr)) \rightarrow (C' = C \vee Synonym(C', C))])$ then $Pr_{symetric} = Pr_{symetric} \cup \{Pr\}$

4.3. OWL Codification Phase

In this final phase, the *OWL codifier* uses the sets generated by the *semantic interpreter* and writes out the corresponding OWL constructs according to W3C Recommendation [19], considering the mapping conventions shown in Table 1.

Table 1. Conventions for mapping between inferred sets and OWL constructs

Inferred sets	Basic structure	OWL constructs
S_CS	(C, C')	C is coded as <i>owl:class</i> . C' as <i>owl:class</i> , and the specification <i>rdf:subClassOf</i> C is included.
S_CI	(C, I)	C is coded as <i>owl:class</i> and I as an instance of C.
S_CP	(C, l-p, Pr)	An <i>owl:ObjectProperty</i> is coded, whose name is formed by the concatenation of the label <i>l-p</i> , and the label <i>Pr</i> . C is coded as <i>owl:class</i> and <i>rdfs:domain</i> of the <i>Object Property</i> . <i>Pr</i> is coded as <i>owl:class</i> and <i>rdfs:range</i> of the <i>Object Property</i> .
S_CPV	(C, Pr, C')	C is coded as <i>owl:class</i> . <i>Pr</i> is coded as <i>owl:onProperty</i> (of C). C' is coded as <i>owl:someValueFrom</i> (of <i>Pr</i>) restriction in the specification of C.
S_CPV _{hasValue}	(C, Pr, I)	C is coded as <i>owl:class</i> . <i>Pr</i> is coded as <i>owl:onProperty</i> (of C) with <i>owl:hasValue</i> restriction of C'. C' is coded as <i>owl:class</i> and <i>owl:hasValue</i> (of <i>Pr</i>).
Pr _{symetric}	Pr ₁ ... Pr _n	<i>owl:SymmetricProperty</i> construct in the specification of each <i>Pr_i</i> property is included.
Pr _{functional}	Pr ₁ ... Pr _n	<i>owl:FunctionalProperty</i> construct in the code of each <i>Pr_i</i> is incorporated.
Intersection _{classes}	(C, {C' ₁ ... C' _n })	C and each C' _i are coded as <i>owl:class</i> . The collection of C' _i is coded as <i>owl:intersectionOf</i> .
Intersection _{class-property}	(C, C', Pr)	C is coded as <i>owl:class</i> . C' is coded as <i>owl:class</i> . <i>Pr</i> is coded as <i>owl:onProperty</i> (of C), with restriction <i>owl:hasValue</i> of C'. The collection of C' and <i>Pr</i> is coded as <i>owl:intersectionOf</i> .
Union	(C, {C' ₁ ... C' _n })	C is coded as <i>owl:class</i> . The C' _i are coded as <i>owl:class</i> . Collection of C' _i as <i>owl:unionOf</i> .

A partial OWL ontology is obtained at the end of this phase, which formalizes the semantic information inferred from concept and propositions represented in the CM. Resulting ontology can be refined and completed through Protégé [14].

5. Experimental results

The described method has been applied to 54 CMs on environmental knowledge domain (extending the experi-

¹ The *l-p* corresponding to a WordNet's relation in PVC category (e.g. "made of" in case of *has_mero_madeof*).

mental CMs repository reported in [18]) in an experimental process. CMs were taken mainly from published scientific articles, because there isn't a reference collection of CMs for this testing type and the quality of the conceptualizations was required. The CMs selection of environmental knowledge domain allows making comparisons with previous versions, and identifying improvements.

CMs considered in the experimental repository had an average of 23 concepts and 26 propositions. Independent conceptualizations are represented in each CM and a partial OWL ontology was automatically obtained from each one. However, in [17] a method which allows automatically processing and integrating conceptualizations from different approaches was reported.

Metrics of *precision*, *recall* and *coverage* have been defined to quantify the results of this type of proposed method [18]. However, the *coverage* of the semantic inference in the CM for ontology learning was greater interest for evaluating the new proposal presented. *Coverage* means how much knowledge; through the number of codified propositions in OWL language was formalized. It is mathematically formalized as follows:

$$Coverage = \frac{CCP + ICP}{CP} \quad (2)$$

where *CCP* is the number of correctly coded propositions, *ICP* is the number of incorrectly coded propositions, and *CP* is the number of propositions should be encoded. A proposition is considered as coded if the semantics associated with the elements that compose it (concepts and the relationship between them) were formally described in the resulting OWL ontology.

Three fundamental aspects were tested in this process:

1. The effects of OpenCyc in the semantic inference for coverage increasing.
2. The effects of the domain ontologies in the semantic inference for coverage increasing.
3. The coverage improvement of the proposed method (combined OpenCyc and domain ontologies with WordNet), in comparison with previous version reported [19].

The second aspect was evaluated with an ontology corpus resulting from the SWEET (Semantic Web for Earth and Environmental Terminology) Project [15], which contains about 100 OWL ontologies on environmental domain; this is one of the most recognized in this domain.

The coverage measurement results obtained in ontologies construction from CMs are graphically presented as follow. Figure 3 shows results of method's coverage using only OpenCyc (related with the first aspect), and Figure 4 shows results of coverage using only SWEET ontology corpus (related with the second aspect). Results

were compared with the obtained by previous version reported in [4], using the same CMs.

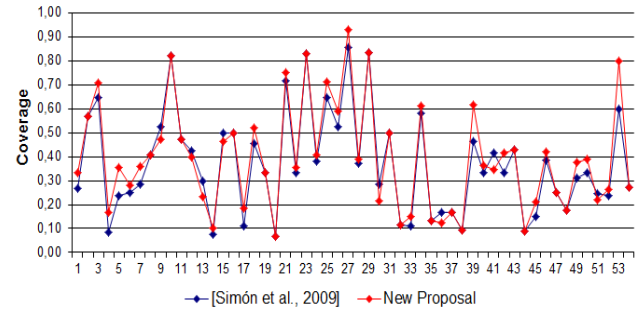


Fig. 3: Comparative results of *coverage* using OpenCyc.

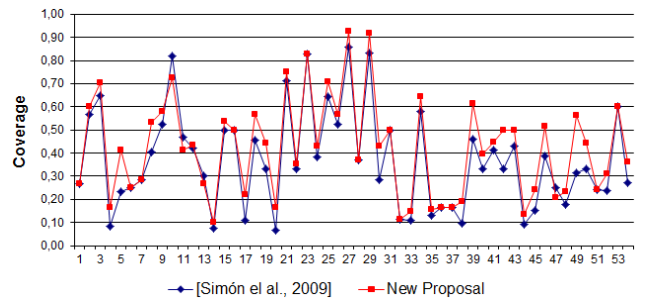


Fig. 4: Comparative results of *coverage* using SWEET Ontology Corpus.

According the first experiment, 12 % of all inferred semantics in the CMs and coded in the resulting ontologies were inferred using OpenCyc, demonstrating their contribution in this process. In general, the resulting coverage of the proposed method using only OpenCyc was of 40 %, increasing the obtained by the version reported in [18] in 3%.

In second experiment, the resulting coverage using the SWEET ontology corpus was of 45 %, 8% higher than the method reported in [18]. In this case, 19 % of all inferred semantics in the CMs were inferred using domain ontologies included in the selected corpus. Although results can be considered positive, the significance of the reference ontologies (selected in preprocessing phase) used in the semantic inference process (in the CM) is an important aspect that can influence in these results. Significance was considered here as the quantity of semantic information that can contribute the reference ontologies to the inference process, according their thematic relationship with the processed CM. This aspect has been analyzed through the definition of *relevance level* (RL) metric, which evaluates the similarity measures between CM and the conceptualization (represented in CM form) of corresponding reference ontologies. It is mathematically formalized as follows:

$$RL = \frac{\sum_{i=0}^n S(CM_p, O_i)}{n} \quad (2)$$

where $S(CMp, O_i)$ is the similarity value between CM to be processed (CMp) and the conceptualization of reference ontology i (O_i) represented in CM form, and n is the number of reference ontologies selected of the corpus. Figure 5 shows coverage trend respect the relevance level of the reference ontologies used in each processed CM (seen like a red line).

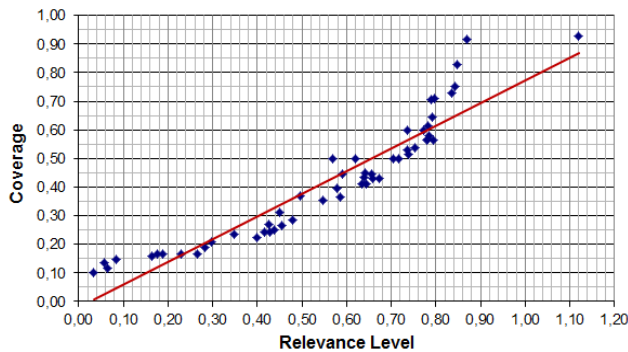


Fig. 5: Coverage trend respect the relevance of reference ontologies.

An important conclusion of this partial experiment is that while more thematic relationship exists between the knowledge represented in the CM to be processed and the conceptualization of the domain ontology to use as knowledge source better results in ontology learning form the CM can be obtained. It is due because these ontologies can provide more semantics information for implicit semantic discovering in the CM. Experimental results did not report contradictions between inferred semantic from external knowledge bases and the rest of the included criteria in the method. The proposed procedure for reference ontologies selecting from the corpus, based on relation measurement between represented knowledge (conceptualization) with the knowledge in CM also helps to reduce ambiguities and contradictions in the semantic inference process.

Finally, to compare the whole proposed method with the one reported in [18] and for responding third aspects tested, another experiment was carried out, which OpenCyc and domain ontologies were included. Figure 6 shows the experiment results. The coverage results were higher in 14 % with the extended method than the version reported in [18]. The contributions of OpenCyc and domain ontologies for ontology learning from CMs evidencing were demonstrated in these experiments; allowing coverage improvement of the semantic inference process in the CMs.

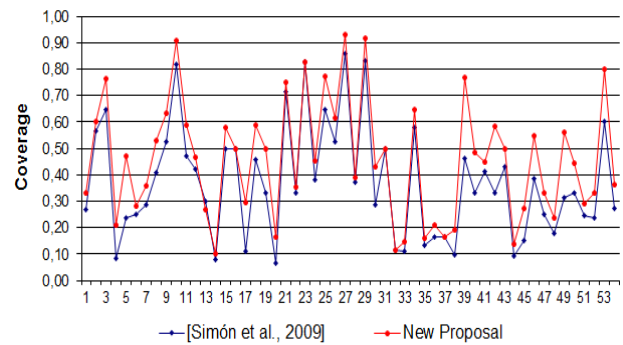


Fig. 6: Comparative results of coverage of extended method and previous version [18].

In general, were satisfactory the obtained results; however, we consider that the coverage needs to be higher. A cause possible for not obtaining higher results was that the 55% of analyzed propositions had at least one concept not included in any knowledge bases used; therefore, could not infer semantics from these propositions. To solve this limitation, a new rule was added in the semantic mapping phase, through which the semantic of all concepts included in not codified propositions and semantically formalized as class OpenCyc or in any reference ontologies was inferred as class too (if it has not been coded before in another way). Figure 7 shows comparative results of this refined of the proposed method, concluding that the coverage in the improved proposal was increasing in 21 %, which is a promising results.

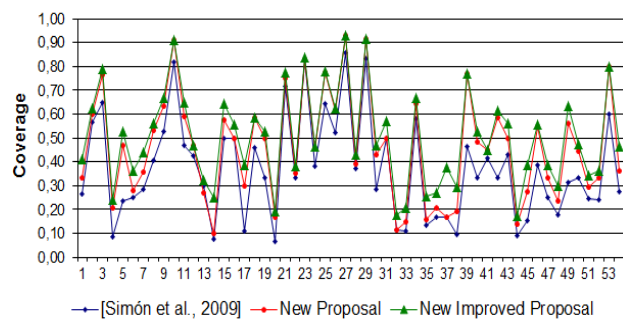


Fig. 7: Comparative results of coverage among the extended method, its previous version [18], and the refined new proposal.

6. Conclusions

The work presented here is a contribution for the ontology learning from an informal knowledge representation, such as concept maps, especially useful in context which users with little technical background are requiring generate their own ontologies and collaborate in the construction of distributed knowledge bases. In this paper, the extension and improvement of a method for ontology learning from a concept map have been presented.

The proposed method combine the analysis of the semantics represented in concept maps, mechanisms of natural language processing based on a concept-sense-

disambiguation algorithm, and an extension of the knowledge bases, with OpenCyc and domain ontologies in a novel way, which significantly distinguishes from the existing literature. The use of OpenCyc and domain ontologies (used in a combined way with WordNet) has contributed to increase the coverage in the semantics inference in the concept map (while keeping its flexibility) and the applicability of the method proposed.

Finally, the mapping between concept maps and OWL ontologies creates the bases for the collaborative development of ontologies in a more intuitive, friendlier manner for humans and allows the reuse of the knowledge represented in concept maps by knowledge management systems.

7. References

- [1] Brilhante V., Macedo G., and Macedo S., "Heuristic Transformation of Well-Constructed Conceptual Maps into OWL Preliminary Domain Ontologies", *Proc. of 2nd Workshop on Ontologies and their Applications*, Brazil, 2006.
- [2] Cañas A., J. Leake, D., and Maguitman A. "Combining Concept Mapping with CBR: Towards Experience-Based Support for Knowledge Modeling", *Proc. of FLAIRS Conference 2001*, AAAI, pp. 286-290, 2001.
- [3] Corcho O., Fernández M., and Gómez A., "Ontological Engineering: What are Ontologies and How Can We Build Them?", in *Semantic Web Services: Theory, Tools and Applications*, J. Cardoso ed., IGI Global, pp. 44-70, 2007.
- [4] Cury D., and Menezes C. S., "A Portal of Knowledge Based on Concept Maps", *Proc. of 5th Int. Conf. on Concept Mapping*, 2012.
- [5] García A., Norena A., Betancourt A., García L., and Sequeda J. F., "CMAPS supporting the development of OWL ontologies". *Proc. Of Int. Semantic Web Conf. (ISWC'08)*, 2008.
- [6] Giouvanakis T., Frigidis G., Kehris E., and Samaras H., "Exploiting Concept Mapping in a Semantic Web Environment", *Proc. of the Fifth IEEE Int. Conf. on Advanced Learning Technologies*, IEEE Computer Society Press, pp. 432-434, 2005.
- [7] Gómez H., Díaz B., and González A., "Two Layered Approach to Knowledge Representation Using Conceptual Maps Description Logic", *Proc. of 1st Int. Conf. on Concept Mapping*, 2004.
- [8] Graudina V., and Grundspenkis J., "Concept Map Generation from OWL Ontologies". *Proc of 3rd Int. Conf. on Concept Mapping*, pp. 173-180, 2008.
- [9] Hayes P. J., Eskridge T. C., Saavedra R., Reichherzer T., Mehrotra M., and Bobrovnikoff, D., "Collaborative Knowledge Capture in Ontologies", *Proc. of 3rd Int. Conf. on Knowledge Capture*, Canada, pp. 99-106, 2005.
- [10] Lenat, D. B., "Cyc: A Large-Scale Investment in Knowledge Infrastructure," *Communications of the ACM*, vol. 38, pp. 33-38, 1995.
- [11] McGuinness D., Fox P., Cinquini L., West P., Benedict J., and Garcia J., "Current and future uses of OWL for Earth and Space science data frameworks: successes and limitations", *AGU Fall Meeting 2007*, 2007.
- [12] Miller G., Beckwith R., Fellbaum C., Gross D., and Miller K.J. "Introduction to WordNet: An On-line Lexical Database", *Int. Journal of Lexicography*, 3(4), pp. 235-244, 1993.
- [13] Novak J. D., and Gowin D. B., "Learning how to learn", *Cambridge University Press*, 1984.
- [14] Noy N. F., Sintek M., Decker S., Crubezy M., Ferguson R. W., and Musen, M. A., "Creating Semantic Web contents with Protege-2000". *IEEE Intelligent Systems*, 16(2), pp. 60-71, 2001.
- [15] Raskin R., Pan M., and Mattmann C., "Semantic Web for Earth and Environmental Terminology (SWEET)". *Proc. of Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, 2004.
- [16] Simón-Cuevas A., Ceccaroni L., Rosete-Suárez A., and de la Iglesia M., "A Concept Sense Disambiguation Algorithm for Concept Maps", *Proc. of 3rd Int. Conf. on Concept Mapping*, pp. 14-21, 2008.
- [17] Simón-Cuevas A., Ceccaroni L., Rosete-Suárez A., Suárez-Rodríguez A., and Victoria-Oquendo R., "A Support to Formalize a Conceptualization from a Concept Map Repository", *Proc. of 3rd Int. Conf. on Concept Mapping*, pp. 68-75, 2008.
- [18] Simón-Cuevas A., Ceccaroni L., Rosete-Suárez A., and Suárez-Rodríguez A., "Formal Modeling Method applied to Environmental-Knowledge Engineering", *Proc. of Int. Conf. on Complex, Intelligent and Software Intensive Systems*, pp. 1132-1137, 2009.
- [19] Smith M. K., Welty Ch., and McGuinness, D., "OWL Web Ontology Language Guide", W3C, 2004.