

An Efficient Ontology Comparison Tool for Semantic Web Applications

James Z. Wang

*Department of Computer Science
Clemson University, Box 340974
Clemson, SC 29634-0974, USA
+1-864-656-7678
jzwang@cs.clemson.edu*

Farha Ali

*Department of Mathematics and Computing
Laura Lander Hall 317, Lander University
320 Stanley Avenue, Greenwood, SC 29649
+1-864-388-8330
fail@lander.edu*

Abstract

With the growing access to heterogeneous and independent data repositories, determining the semantic difference of two ontologies is critical in information retrieval, information integration and semantic web applications. In this paper, we propose an ontology comparison tool based on a novel senses refinement algorithm, which builds a senses set to accurately represent the semantics of the input ontology. The senses refinement algorithm automatically extracts senses from the electronic lexical database WordNet (locally installed or online), removes unnecessary senses based on the relationship among the entity classes of the ontology, and specifies relations and constraints of the concepts in the refined senses set. The senses refinement converts the measurement of ontology difference into simple set operations based on set theory, thus ensures the efficiency and accuracy of the ontology comparison. Our experimental studies show that the proposed senses refinement algorithm outperforms the naive senses set construction algorithm in terms of efficiency and accuracy

1. Introduction

Among all essential components of the semantic web, ontology plays the most important role since it makes the extraction and formalization of semantics possible. There are several approaches for building ontologies as discussed in [1, 2]. The concepts in ontologies are represented in natural language words. Yet meaning and understanding of words vary in different communities. This heterogeneity causes problems in interoperability of knowledge resources. With the growing access to heterogeneous and independent data repositories, determining the semantic similarity or difference of two ontologies is critical in information retrieval and information integration. Furthermore, when semantic web applications (especially P2P semantic web services) becomes popular, it is necessary to provide a tool to

efficiently measure the similarity of ontologies so that the semantic web agents can use the tool to make query decisions based on the ontology similarity or difference of the semantic P2P web services. Currently there is no such tool available.

To fill the void, we implement an online ontology comparison tool which can give a numeric measurement of the difference between two ontologies. This measurement tool is based on a senses refinement (SR) algorithm which makes use of the concepts and senses retrieved from the electronic lexical database WordNet (locally installed or online) [3]. The rest of this paper is organized as follows. We first discuss the background and existing approaches to the problem of measuring ontology similarity in section 2. Then we give the formal definition of ontology difference based on set theory in section 3. We propose our senses refinement (SR) algorithm and discuss its advantages in section 4. In section 5, we use experimental studies to prove the efficiency and accuracy of our proposed senses refinement algorithm. Finally we give our conclusion and discuss the future work in section 6.

2. Background and existing approaches

Recent studies in semantic web applications have emphasized on using ontologies and semantic similarity functions as mechanisms for directing queries across heterogeneous information repositories. Several approaches have been proposed to deal with the heterogeneity of ontologies. One approach is the ontology integration by mapping different ontologies into a more generic ontology [4, 5] or by vocabulary heterogeneity resolution [6, 7] of various ontologies.

There are two problems in ontology integration approaches. First, building a shared ontology is a very complicated process which is not suitable for online semantic web query processes. Second, these methods are designed to compare entity classes within the ontologies, yet no method has been proposed to measure the semantic similarity of two ontologies. Determining

the semantic similarity of two ontologies is as important as measuring the semantic similarity of entity classes within the ontologies. Measuring the semantic similarity between two ontologies can help peer grouping and query routing in P2P semantic web applications, as well as identifying potential collaboration in research areas such as GIS and bioinformatics.

Another popular approach as suggested by SUMO (Suggested Upper Merged Ontology) [8] is to develop a merged ontology by sharing ideas from all available ontologies and mapping the entries of merged ontologies with WordNet entries. The main idea of this approach is that ontology designers will look for the SUMO entries in WordNet corresponding to the concepts they use, so that two different ontology designers will use the same term for the same concept. SUMO helps reducing the complexity of concept mapping, yet it does not address the requirement of comparing two different ontologies.

Contrary to shared ontology approaches, some researchers try to create a computational model to assess semantic similarity among entity classes from different and independent ontologies without constructing *a priori* a shared ontology [9]. This approach uses a matching process to establish links among ontologies while keeping them autonomous. However it focuses on the semantic similarity of entity classes and does not allow deep processes due to the complexity of matching process. Thus using this approach to measure the semantic similarity of two ontologies is not practical.

In this paper, after giving a formal definition of ontology difference based on set theory, we propose an efficient ontology comparison algorithm that uses a novel senses refinement algorithm to convert ontology semantic difference measurement into set operations.

3. Ontology Difference

There are many ways of measuring the difference of two given objects. For two numeric data values, their difference can be calculated by using dissimilarity formulas. Yet for nonnumeric objects, it is necessary to correlate nonnumeric data to numeric values so that the difference can be quantified. Tversky defined a similarity measurement model [10] based on set theory so that difference in characteristics between objects can be evaluated by set operations. This similarity measurement model is also in agreement with an information-theoretic definition of similarity [11].

In this paper, we define our ontology difference measurement formula based on the normalization of Tversky's model to give a numeric measurement of ontology difference. To facilitate set operations, we use senses set to summarize the semantics of the ontology. A senses set for an entity class is a set of synonym words denoting the concept of the entity class. A senses set for

an ontology is obtained by extracting synonym words related to the ontology semantics from the senses sets of all concepts in the ontology. Assume the senses set of Target ontology is T and the senses set of Source ontology is S . The difference of set T from set S , denoted by $T-S$, is defined as

$$T-S = \{x \mid x \in T \wedge x \notin S\}$$

We use cardinality of the senses set to correlate the nonnumeric ontology semantics into numeric value. The cardinality of set $T-S$ indicates how many distinct synonym words existing in Target senses set T are not in Source senses set S . The cardinality of set T represents the number of distinct synonym words in Target senses set T . Thus the semantic difference between two ontologies can be defined by function $D(T,S)$ in following equation:

$$D(T,S) = \frac{|T-S|}{|T|} \quad (1)$$

Based on Equation 1, we have $0 \leq D(T,S) \leq 1$. When there is no common element between sense sets T and S , i.e., $|T-S| = |T|$, $D(T,S) = 1$. On the other hand, if set T is a subset of set S ($T \subseteq S$), i.e., $|T-S| = 0$, then $D(T,S) = 0$.

This measurement formula for ontology difference is not forced to satisfy symmetry property which is preserved by semantic distance based models [12]. That is, the semantic difference from ontology A to ontology B may not be the same as the semantic difference from ontology B to ontology A. Employing such an asymmetric measurement is important because we must ensure the ontology difference evaluations sensible to human judgments, in which cognitive properties of similarity play key roles. For instance, assume the senses set of ontology A and B are S_A and S_B respectively. If $S_A \subset S_B$, then $D(S_A, S_B) = 0$ and $D(S_B, S_A) > 0$. $D(S_A, S_B) = 0$ means semantics existing in ontology A is also in ontology B. On the other hand, $D(S_B, S_A) > 0$ means that ontology B includes some concepts that are not present in ontology A. Thus allowing the asymmetry in semantic difference of ontologies has significant importance in information retrieval and semantic web applications.

4. Efficient Ontology Comparison

Our proposed ontology measurement method tries to correlate the nonnumeric ontology semantics into numeric cardinality of senses sets. Using only the concept labels of the entity classes to form the senses sets can not yield accurate ontology comparison results, because the same concept may be represented by different words in different ontologies. It is necessary to discover the senses of the concepts to ensure accurate

presentation of the ontology semantics using senses set. Thus how to efficiently build senses set that can accurately represent the semantics of the ontology becomes critical in ontology comparison. We propose a senses refinement algorithm that satisfies both efficiency and accuracy criteria.

4.1 Senses Refinement Algorithm

The entity classes in an ontology may be associated with many different concepts. Each concept may have many senses because the evolution of natural language has produced polysemy that the same word denotes more than one meaning. However not all senses of a concept should be included in the senses set for the ontology. Besides senses, the relations (“is-a” or “part-whole” relation) of concepts within the ontology also contributes to the semantics of the ontology. To build a senses set to accurately represent the semantics of the ontology, we should obtain the senses sets for all concepts of ontology, decide which of them should be included in or excluded from the refined senses set for the ontology, and determine a way to represent relations among the concepts in the senses set.

We have implemented a Java programming interface to WordNet to automatically extract and store the senses of concepts in an ontology. Once all senses of the concepts in an ontology are extracted out of WordNet, a naive algorithm to build the senses set for the ontology is to union the senses sets of all concepts in the ontology. However this naive approach has some problems. First, not all senses of a concept should be included in the senses set for the ontology due to polysemy. Having unrelated senses in the ontology senses set will diminish the accuracy of measuring the ontology difference. Second, having too many unnecessary senses in the senses set hinders the efficiency of ontology comparison because larger number of elements in senses set incurs higher computation cost for set operations. Third, relations among entity classes in the ontology have to be included in the senses set so that the semantics of the ontology can be accurately represented by the senses set. The naive algorithm for senses set construction does not make any attempt to include relations in the senses set.

To address the aforementioned problems, we propose a senses refinement algorithm that refines the senses set of the ontology based on the semantic relationships between the parent concepts and the children concepts. There are two kinds of semantic relationships between the parent concept and the child concept according to WordNet. Hyponymy, i.e., “is-a” relation, is the most common relation used in ontologies. The “is-a” relation is transitive and asymmetric, and defines a hierarchical structure in which concepts inherit the entire characteristics from their superordinate concepts.

Meronymy is the “part-whole” relation in which the child concept is part of the parent concept. These relations determine whether a particular sense of a concept should be included in the senses set of the ontology. Our senses refinement (SR) algorithm is based on “is-a” relation since it is the dominating relationship in ontologies. Our senses refinement algorithm is depicted in Figure 1.

```

Algorithm SR(Ontology O)
begin
  Q = {};
  P = {p | p ∈ O && p is a parent in ontology O}
  for any p ∈ P
    P_flag = false;
    Sp = Senses set of p from WordNet;
    C = { c | c is a child of p in ontology O }
    for any c ∈ C
      C_flag = false;
      Sc = Senses set of c from WordNet;
      for any s ∈ Sc
        H = Hypernym set of s from WordNet;
        for any h ∈ H
          if ( h ∈ Sp )
            C_flag = true; P_flag = true;
            if ( h == p )
              x = c + “_is-a_” + p;
              Q = Q ∪ { x };
            else
              Q = Q ∪ { s };
            endif
          endif
        endfor
      endfor
    endfor
  if(!C_flag)
    Q = Q ∪ { c };
  endif
endfor
if (!P_flag)
  Q = Q ∪ { h };
else
  Q = Q ∪ { p };
endif
endfor
return Q;
end

```

Figure 1: Senses Refinement Algorithm

```

Algorithm OntoCmp(OS, OT)
begin
  S = SR(OS)
  T = SR(OT)
   $D = \frac{|T - S|}{|T|}$ 
  return D
end

```

Figure 2: Ontology Comparison Procedure

4.2 Ontology Comparison Procedure

Using the proposed senses refinement algorithm, we design a simple ontology comparison procedure shown in Figure 2. This procedure takes two ontologies as the input parameters and returns their semantic difference in numeric value. To further demonstrate the execution flow of our ontology comparison procedure, we apply the procedure on some simple ontologies and show the steps of senses refinement and ontology comparison.

Assume we have two ontologies, OntoBeverage and OntoPL, defined by “is-a” relation hierarchy. OntoBeverage shown in Figure 3 is a simple ontology representing two beverages, Java and Beer. OntoPL depicted in Figure 4 is a simple ontology representing programming language Java. We assume OntoBeverage is the target ontology and OntoPL is the source ontology.

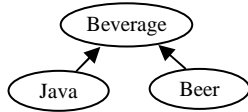


Figure 3: OntoBeverage

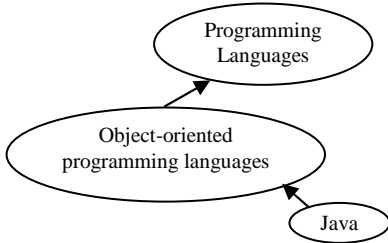


Figure 4: OntoPL

Table 1: Senses and Hypernyms for OntoBeverage

Concepts	Senses	Hypernyms	In Sense Set?
Beverage	beverage, drink, drinkable, potable	food, nutrient	Yes
Java	1. Java	Island	No
	2. coffee, java	beverage, drink, drinkable, potable	Yes
	3. Java	object-oriented programming language, object-oriented programing language	No
Beer	Beer	brew, brewage	No

To compare those two ontologies, we need to get the refined senses set \mathbf{T} for target ontology OntoBeverage and the refined senses set \mathbf{S} for source OntoPL respectively. First we need to get the concepts and their senses with associated hypernyms for those two

ontologies. In this example, we only consider first-level hypernyms. For more accurate results we can use hypernyms of higher levels. Concepts and senses with their associated hypernyms obtained from WordNet for ontology OntoBeverage are depicted in Table 1.

To get the refined target senses set \mathbf{T} , we examine all concepts in the ontology starting from the root concept Beverage. First the senses set is empty, i.e., $\mathbf{T} = \{ \}$. Then we determine what senses of the concepts should be included in set \mathbf{T} using our senses refinement algorithm. Looking at the parent concept Beverage and the child concept Java, the hypernyms of the second senses set of the child concept Java have common elements with the only sense of beverage. Hence the synonyms for the second sense of Java and for the only sense of beverage should be included in the refined sense set. For the first and the third senses sets of the child concept Java, their Hypernyms have no common element with the senses of the parent concept Beverage, thus those senses for child concept Java can not be included in set \mathbf{T} . Now $\mathbf{T} = \{ \textit{beverage, drink, drinkable, potable, coffee, java} \}$.

In addition to excluding the unrelated senses of the concepts, our senses refinement algorithm also specifies senses to reflect the relationship of child and parent concepts. Sometimes the synonyms sets for different senses of a concept contain the same word as the concept label itself. For example, “java” is the word used for child concept label in OntoBeverage. Three different senses sets for concept java can be extracted from WordNet. Among those three senses sets for java, only the second senses set can be included in the senses set for the ontology and the concept label “java” is in this senses set. In the meantime, the related parent concept label “beverage” is in the hypernyms set of java. This can be used to identify the “is-a” relation between concept “java” and “beverage”. The “is-a” relation can also be used to differentiate “java” from other senses. To specify the “is-a” relationship in the senses set for ontology OntoBeverage, we rename sense “java” as “java_is-a_beverage”. So the senses set for ontology OntoBeverage is $\mathbf{T} = \{ \textit{beverage, drink, drinkable, potable, coffee, java_is-a_beverage} \}$.

Finally, if a concept does not have a single sense that matches with one of its parents’ senses or a parent does not have a single sense that matches with hypernyms of all the senses of its children, we just include the concept label in the senses set of the ontology. Based on this rule, “beer” is added into the refined senses set \mathbf{T} for OntoBeverage. Thus, $\mathbf{T} = \{ \textit{beverage, drink, drinkable, potable, Coffee, java_is-a_beverage, beer} \}$.

Similarly we can get the concepts and the senses with their associated hypernyms for source ontology OntoPL from WordNet. They are presented in Table 2. Using our proposed senses refinement algorithm, we can get the

refined senses set for source ontology OntoPL. That is, $S = \{programming\ language, programming\ language, object-oriented\ programming, language, object-oriented\ programming\ language, java_is-a_object\ oriented\ programming\ language\}$. Using Equation 1, we get,

$$D(T, S) = \frac{|T - S|}{|T|} = 1$$

Table 2: Senses and Hypernyms for OntoPL

Concepts	Senses	Hypernyms	In Senses Set?
Programming Language	Programming language, programming language	Artificial language	Yes
Object-oriented Programming Language	Object-oriented Programming Language, Object-oriented Programming Language	Programming language, programming language	Yes
Java	1. Java	Island	No
	2. coffee, java	beverage, drink, drinkable, potable	No
	3. Java	object-oriented programming language, object-oriented programming language	Yes

Now let us change the source ontology to another ontology, OntoDrink, depicted in Figure 5. OntoDrink is a simple ontology representing some drinks. We want to use this ontology as the source ontology to compare with the target ontology OntoBeverage shown in Figure 3. The concepts and the senses with the associated hypernyms obtained from WordNet for ontology OntoDrink are presented in Table 3. Using our senses refinement algorithm, we can get the refined senses set $S = \{beverage, drink, drinkable, potable, coffee_is-a_drink, java, cola\}$ for this new source ontology OntoDrink.

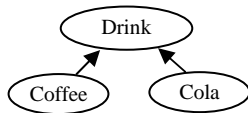


Figure 5: OntoDrink

As discussed before, we have already got the refined target senses set $T = \{beverage, drink, drinkable, potable, coffee, java_is-a_beverage, beer\}$. Using equation 1, we get,

$$D(T, S) = \frac{|T - S|}{|T|} = 0.2857$$

Table 3: Senses and Hypernyms for OntoDrink

Concepts	Senses	Hypernyms	In Sense Set?
Drink	1. drink ,	small indefinite quantity, small indefinite amount	No
	2. drink, drinking, boozing, drunkenness, crapulence	intemperance, intemperateness	No
	3. beverage, drink, drinkable, potable	food, nutrient	Yes
	4. drink	body of water, water	No
	5. swallow, drink, deglutition	consumption, ingestion, intake, uptake	No
Coffee	1. coffee, java	beverage, drink, drinkable, potable	Yes
	2. coffee, coffee tree	tree	No
	3. coffee bean, coffee berry, coffee	Seed	No
	4. coffee, deep brown, umber, burnt umber	brown, brownness	No
Cola	1. cola, genus cola	dilleniid dicot genus	No
	2. cola, dope ,	soft drink	No

Our ontology comparison algorithm shows that the semantic difference between ontology OntoBeverage and OntoPL is 1. It means even though they are using the same concept label for one of their concepts, they are representing very different data. Conversely, our ontology comparison algorithm reveals that the difference between Ontologies OntoBeverage and OntoDrink is just 0.42857. So these two Ontologies represent some similar concepts, although they have used different concept labels.

5. Performance study

We have done some experimental studies to evaluate our proposed senses refinement algorithm in terms of accuracy and efficiency. Because there is no precedent work that measures the semantic difference of two ontologies (existing studies focus on measuring the semantic similarity of entity classes in ontologies), we will compare our senses refinement algorithm with the naive senses set construction algorithm discussed in section 4.1.

5.1 Efficiency of the proposed ontology comparison tool

To evaluate the efficiency of our proposed ontology comparison, we run the evaluated ontology comparison tools on some simple ontologies presented in Figure 6. We choose these ontologies because most of their concepts have a lot of senses. Thus the time spent in senses refinement and senses specification should be noticeable when our senses refinement algorithm is used to construct the senses sets for the ontologies.

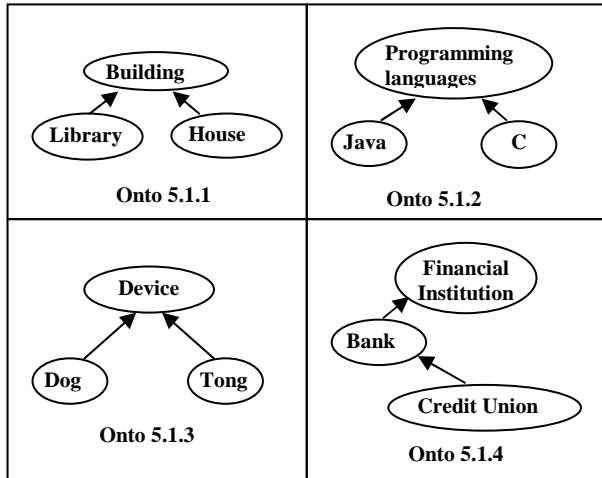


Figure 6: Ontologies for evaluating efficiency of ontology comparison tool

We use J2SE to develop the ontology tool that employs either our senses refinement algorithm or the naive senses set construction algorithm. Without the loss of generality, we compare each ontology with itself using different ontology comparison tools which have different senses set construction algorithms (senses refinement or naive). Comparing an ontology to itself ensures that both algorithms will generate the same ontology difference value 0, so that the efficiency of the algorithms are comparable. We observe the processing times of the ontology comparison under different algorithms (senses refinement or naive) for constructing senses sets. The experimental studies are conducted on a desktop computer equipped with 1.3GHz Intel Pentium M processor and 512 MB RAM. The evaluated ontology comparison tools run as Java application under Windows XP. The experimental results are depicted in Table 4.

We have ignored the time spent on fetching the senses and the hypernyms of senses from WordNet because this time varies significantly due to the dynamic change of network traffic on internet and workload in WordNet server. The processing times reported in Table 4 only include the times for constructing the senses sets and the times for calculating ontology difference. Ignoring the

time for fetching the senses and the hypernyms of senses from WordNet can give better assessment on the efficiency of the ontology comparison tool.

Table 4: Processing times under different senses set construction algorithms

Ontology	Processing Time (milliseconds)	
	SR Algorithm	Naive Algorithm
Onto 5.1.1	0.06	0.26
Onto 5.1.2	0.12	0.29
Onto 5.1.3	0.14	3.21
Onto 5.1.4	0.08	0.66

Performance of our SR algorithm in terms of processing time depends upon the number of senses of the concepts in the ontology. If the concepts have many senses, then the SR algorithm will reduce the number of senses and refined senses set will be considerably smaller as compared to the senses set used by naive algorithm which is generated by union of all senses of the ontology. Thus the time for calculating the ontology difference is reduced using our senses refinement algorithm. The results shown in Table 4 have proven that the processing time of ontology comparison using our senses refinement algorithm is less than that using the naive algorithm. We must note here that if the compared ontologies have concepts that do not have many senses, then naive algorithm may produce better results because the time saving in comparing the refined senses sets may not compensate the time spending on senses refinement.

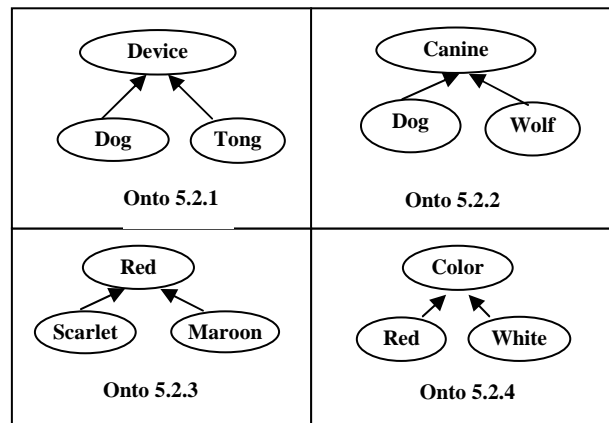


Figure 7: Ontologies for evaluating accuracy of ontology comparison tool

5.2 Accuracy of the proposed ontology comparison tool

To evaluate the accuracy of the proposed ontology comparison tool in measuring the semantic difference of ontologies, we run the ontology comparison tools on some simple ontologies presented in Figure 7.

We choose these ontologies because they are very simple. Thus the accuracy of the ontology comparison can be easily judged by human observation. The experimental results are depicted in Table 5. The ontology comparison results show that the ontology comparison tool using our senses refinement algorithm is more accurate than the ontology comparison tool based on the naive senses set construction algorithm. The accuracy evaluation is based on our subjective judgment since the compared ontologies are very simple. Overall, the experimental results prove that the senses sets built by our senses refinement algorithm accurately represent the semantics of the input ontology.

Table 5: Ontology differences under different senses set construction algorithms

Target Ontology	Source Ontology	Ontology Difference Value	
		SR Algorithm	Naive Algorithm
Onto 5.2.1	Onto 5.2.2	1.0	0.120
Onto 5.2.2	Onto 5.2.1	1.0	0.435
Onto 5.2.3	Onto 5.2.4	0.857	0.3076
Onto 5.2.4	Onto 5.2.3	0.857	0.3076

6. Conclusion and future studies

In this paper, we implemented a tool for determining ontology difference based on our proposed senses refinement algorithm, which builds senses sets to accurately represent the semantics of input ontologies. The senses refinement algorithm automatically extracts senses from the electronic lexical database WordNet (locally installed or online), removes unnecessary senses based on the relationship among the entity classes of the ontology, and specifies relations and constraints of the concepts in the refined senses set. The senses refinement converts the measurement of ontology semantic difference into simple set operations based on set theory, thus ensures the efficiency and accuracy of the ontology comparison. Our experimental studies show that the proposed senses refinement algorithm outperforms the naive senses set construction algorithm in terms of efficiency and accuracy.

We are currently extending the senses refinement algorithm so that it can integrate the “part-whole” relations and semantic features of entity classes into the senses set construction for ontology comparison.

7. References

- [1] Resource Description Framework (RDF), *W3C Semantic Web Activity*. <http://www.w3.org/RDF/>
- [2] Web Ontology Language (OWL), *W3C Semantic Web Activity*. <http://www.w3.org/2004/OWL/>
- [3] Christine Fellbaum (ed.), *WordNet: An Electronic Lexical Database*. The MIT Press, May 1998.
- [4] P. Weinstein and P. Birmingham, Comparing Concepts in Differentiated Ontologies, in *12th Workshop on Knowledge Acquisition, Modeling, and Management*. 1999, Banff, Canada.
- [5] H. Stuckenschmidt, and I. J. Timm, Adapting communication vocabularies using shared ontologies. In *Proceedings of the Second International Workshop on Ontologies in Agent Systems, Workshop at 1st International Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, 15-19 July 2002.
- [6] V. Kashyap and A. Sheth, Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context, and Ontologies, *Cooperative Information Systems: Trends and Directions*, pp. 139-178, M. Papazoglou and G. Schlageter (eds.), Academic Press, London, UK. 1998.
- [7] E. Mena, A. Illaramendi, V. Kashyap, and A. Sheth, OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation across Preexisting Ontologies. *Distributed and Parallel Databases*, 8(2): pp. 223-271, 2000.
- [8] A. Pease, I. Niles, and J. Li, The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, Edmonton, Canada, July 28-August 1, 2002.
- [9] M. A. Rodriguez and M. J. Egenhofer, Determining semantic similarity among entity classes from different ontologies; *IEEE Transactions on Knowledge and Data Engineering*. 2003.
- [10] A. Tversky, Features of Similarity. *Psychological Review*, 84(4), pp. 327-352, 1977.
- [11] D. Lin, An Information-Theoretic Definition of Similarity (eds.), *International Conference on Machine Learning (ICML'98)*, Madison, WI, 1998.
- [12] R. H. Rada, H. Hili, E. Bicknell, and M. Blettner, Development and Application of a Metric on Semantic Nets. *IEEE Transactions on System, Man, and Cybernetics*, 19(1), pp. 17-30, 1989.