A Hybrid Ontology Mediation Approach for the Semantic Web

Saravanam Muthaiyah, George Mason University, USA

Larry Kerschberg, George Mason University, USA

ABSTRACT

This article introduces a hybrid ontology mediation approach for the Semantic Web. It combines both syntactic and semantic matching measures to provide better results for matching data labels. Although ontologies are meant to provide a shared conceptualization of the world, the development practices, lack of standards, and subjective naming conventions today, create data heterogeneity problems among ontologies. This is a significant problem particularly for ontologies of similar domains. Mediation techniques at present focus mainly on syntactic matching and our premise is that a hybrid approach would be a better solution to this problem. We provide empirical evidence with hypothesis tests and also provide several new measures such as relevance, reliability, and precision to validate our approach. We also introduce a detailed mapping algorithm.

Keywords: human cognitive response (HCR); hybrid measure (SRS); ontology; semantic matching (SEM); Semantic Web; syntactic matching (SYN)

INTRODUCTION

The Semantic Web is the brainchild of Tim Berners-Lee, who is also the founder of the World Wide Web (WWW). The WWW has changed the way we communicate, shop, and carry out business transactions. Just like the WWW, the Semantic Web is anticipated to impact our lives in many ways. Berners-Lee defines the Semantic Web as "a web of data that is directly or indirectly processed by machines." The Web today is capable of handling lots of data and presenting it to humans in human readable format. However, it does not understand such data well enough to display what is most relevant or significant in a given context and this is prevalent among search engines today (Barbir, 2002).

The Semantic Web is meant to overcome this limitation via a shared conceptualization. Machines are liable to process, convert, and reason data in more useful and meaningful ways (Muthaiyah & Kerschberg, 2007). For example if a user had to look up the meaning of simple object access protocol (SOAP) on the search engine and needed to actually understand how SOAP services are set up, the Semantic Web search results would display the suite of related technologies such as extensible markup language (XML), universal description, discovery and integration (UDDI), and Web service definition language (WSDL), which are related to SOAP and even provide a step by step tutorial as to how a SOAP service can be setup. Unfortunately, the status of the Semantic Web today is far from what Berners-Lee and Gruber had envisioned and this is because it has not really begun including semantic processing in its search methods.

In a semantically Web-enabled environment, the Semantic Web agent would search the Web for SOAP where SOAP is defined as "a type of technology deployed for Web services to locate relevant results." The agent would also present to the user technologies related to SOAP. The Semantic Web agent would acquire, understand, match, reason, and interpret data from all over the Web and semantically as well as syntactically match them for achieving precise results. However, in today's situation the user who runs a search for SOAP would most likely see listings of bathing soap, detergent, or shampoo. This would unfortunately not be relevant to the user who is looking for SOAP technology. The user may then have to refine the search multiple times before finding relevant information. Search techniques used for current information retrieval processes are mostly based on word matching algorithms that use syntactic matching schemes and do not apply semantic correspondence for data labels. Since there are various semantically associated meanings with the word SOAP, purely syntactic matching will only produce unfavorable results. Considering the Semantic Web vision, we can understand that such a technology would rely on structured data, inferences, and rules to conjure information from various sources. This includes relational databases, Web files, XML documents as well as electronic data interchange (EDI) repositories. Machines would be enabled to access other machines and processes based on readily available semantic information. In order to achieve these goals, semantics must be included in the search algorithm.

BACKGROUND

Some theories indicate that ontologies provide the semantics necessary for semantically matching data. However, this is far from true because ontologies per se cannot include all the data definitions of the world (Maedche, Motik, Silva, & Volz, 2002). Ontologies are meant to allow machine processable metadata to be executed efficiently (Kurgan, Swiercz, & Cios, 2002) Different ontologies may include different sets of definitions and may not be developed in a unified fashion. This is because there are no standards to govern the creation on ontologies. Ontologists often do not agree on the same semantics or structure when developing their ontologies and as such two ontologies might have a similar concept but would have their data labels named differently (Muthaiyah & Kerschberg 2007) such as <price>$5</price> in one ontology and <quote>$5</quote> in another. Unless semantics are used, machines will not be able to comprehend that both data labels are equivalent and will not match them. Without semantics creation of rich data specification for a shared conceptualization will not be possible (Fowler, Nodine, Perry, & Bargmeyer, 1999).

Rules can be specified to enable machines to reason data in more useful ways and understand its semantic nuances (Missikoff, Schiappelli, & Taglino, 2003) but this would not be flexible and scalable, as thousands of rules would be needed. A more reasonable approach would be to mediate ontologies (Maedche et al., 2002) via their data labels. In the travel example earlier, </price> and </quote> data labels for travel bookings and reservations of disparate ontologies should be mediated to achieve data heterogeneity. Ontology mediation systems are mostly semi-automated at present (Muthaiyah & Kerschberg, 2007). They eliminate erroneous data by filtering out data labels that are not syntactically similar and present the results to an ontologist for manual input. The ontologist would manually select semantically related data labels and match them one by one to the with the target ontology (TO). Currently there are many techniques such as MAFRA (Maedche et al., 2002), IF-Map (Kalfoglou & Schorlemmer, 2003), SMART (Noy & Musen, 1999), and PROMPT (Noy & Musen, 2000) that use match algorithms which are based on string, prefix, and suffix matches. Some other techniques are only applicable to relational schemas in databases or XML type data (Muthaiyah & Kerschberg, 2006). Researchers have also attempted to use machine learning systems (Doan, Madhavan, Domingos, & Halevy, 2002) for the same purpose but as mentioned earlier they lack flexibility. Literature also shows that these works also do not incorporate linguistic matching. This is where our work fills the gap.

This article highlights a semi-automated hybrid approach that combines semantic and syntactic matching within a new matching algorithm to present more reliable and relevant results to the ontologist. The idea is to reduce the workload for the ontologist so that source ontology (SO) and TO data labels can be matched much faster. Our experiment shows that pure syntactic matching provided a weak correlation, relevance, precision, and reliability scores when matched with inputs from human domain experts but when syntactic scores were combined with semantic scores the results improved. In the next section we present our matching algorithm.

MATCHING ALGORITHM

Our matching algorithm runs matches and presents the results to the domain expert for final consideration. The domain expert's input is only required towards the final stage and this improves efficiency. The alternative method would be to manually configure matches, which would be voluminous for the human expert, especially in complex environments where large set of match candidates are found. The idea here is to reduce the workload of domain experts by eliminating extraneous data and this is how it works. Parameters are entered in each execution of the algorithm and the acceptance threshold is set for SRS scores.

Only classes that have SRS scores higher than the specified threshold are presented to the domain expert for scrutiny. We propose a semantic matching process where classes are matched using highly reliable algorithms based Lin, Gloss Vector, WordNet Vector and latent semantic analysis (LSA) measures. This is discussed further in the fourth section. The similarity function (s) has five components, that is, (E), inclusiveness (IC), consistency (CN), syntactic similarity (SYN), and semantic similarity (SEM). The five elements within the parenthesis are independent variables that determine the dependant variable (s); thus producing the following equation:

$$(s)\ f_x = \{ E, IC, CN, SYN,SEM \} \quad (1)$$

Multiple factors are considered for determining similarity including variables that have nothing in common in order to refine our results. The similarity function negates all disjoint (D) attributes between classes and the modified function is as follows:

$$(s)\ f_x = \{ E, IC, CN, D, SYN,SEM \} \quad (2)$$

The first three (i.e., E, IC, and CN) tests, are iterative. These tests are based on the definitions provided in the work ofLi Yang, and Yu (2005). We have expanded their definitions to include SYN and SEM which together makes up the SRS. The first three steps, specifically addresses all nodes in a hierarchical ontology structure. Then SYN and SEM tests are applied allowing full semantic matching to be computed among classes and instances. Tools such as OntoViz and RICE can be used to view the hierarchical graph structures. All the nodes in the graph can be tested for tests in step 1, step 2, and step 3 in our algorithm. The algorithm supports Web ontology language (OWL) and resource description framework (RDF) structures and SO and TO nodes can be matched and a similarity matrix is populated (see Appendix 2). We provide a matching algorithm based on similarity of classes that uses both syntactic and semantic matching in order to determine more reliable and precise similarity scores unlike other methods discussed earlier. We have also built a prototype agent-based system using Java agent development framework (JADE), which deploys a matching agent (MA) that computes similarity using SRS.

Those with lower scores are recorded into logs and the domain expert reviews them when it calls for his/her judgment especially in cases where the score is very close to the threshold. This is because the algorithm is set to discard input classes that do not attain the threshold level. In such cases the class is not recommended to the domain expert. If one or more of the ontology candidates have SRS scores higher than the acceptance threshold, the one with the highest value is chosen as equivalent (a synonym) to the input. Empirical evidence is provided to support this model which will be discussed in the following sections. The following are the steps involved for the matching algorithm (see appendix 1):

- **Step 1—Read loaded SO and TO taxonomies.** Semantic engine reads taxonomies of the SO and TO. Prepare for detailed matching tests of data labels, go to step 2.
- **Step 2—Equivalence test.** Test for the **equivalence** of source and target classes: Test 1) do they have semantically equivalent data labels, Test 2) are they synonyms, or Test 3) do they have the same slots or attribute names. Equivalence also implies adjacent neighbors are equal. If equivalent, proceed to step 3, 4, and 5. Else go to step 1.
- **Step 3—Inclusive test.** Source and target classes or concepts (C) are **inclusive** if, the attribute (c) of one is inclusive in the other. In other words *selling price* (c_i) is inclusive in *price* (c_j), this is applicable to *hyponyms*. If inclusive, proceed to step 6.
- **Step 4—Disjoint test.** Source and target classes or concepts (C) are **disjoint** if, the intersection of their two attribute sets (c), c_i and c_j results in an empty set {} or ø. If match test is not disjoint, proceed to step 6.
- **Step 5—Consistency test.** Source and target classes or concepts (C) are **consistent** if, all the attributes or slots (i.e., c_1 and c_2) in the class, have nothing in common s.t. $c_1 \cap c_2 = \{\}$. All slots must belong to class that is being tested. This can be configured with RacerPro. If consistent, proceed to step 6.
- **Step 6—Syntactic match.** Syntactic match similarity scores based on class prefix, suffix, substring matches are calculated. This calculation is performed for every class in the source and target ontology. Go to step 7.
- **Step 7—Semantic match:** Semantic match similarity scores based on cognitive measures such as LSA, Lin, Gloss Vector, and WordNet Vector are used. This calculation is done for every class in the source and target ontology. Go to step 8.
- **Step 8—Aggregate both similarity scores.** Similarity inputs from step 6 and 7 are aggregated, to produce SRS. Go to step 9.
- **Step 9—Populate similarity matrix.** The aggregated values (SRS) from step 8 of candidate labels are populated into the similarity matrix. Multiple matches are carried out. Values are to be verified against the threshold. Go to step 10.
- **Step 10—Set threshold.** Threshold value (t) is set based on scale used. For a scale between, 0 and 1 the threshold value is usually 0.5 ($t>0.5$). Those below threshold are logged in file in step 12. If greater than the threshold value, go to step 11.
- **Step 11—Domain expert selection.** At this stage, candidates from step 10 are presented to domain expert by the system. Input from step 12 is accepted at the discretion of the domain expert.
- **Step 12—Manual log.** Selection is made manually only for those values below threshold. The domain expert uses his own cognitive judgment. Go to step 13.

- **Step 13—Mapping/alignment/merge**: All the candidates for mapping, alignment or merge (i.e., integration) chosen from step 11 and 12 are processed. End.

The matching algorithm (see Appendix 1) shows detailed steps before the semantic matching engine produces mappings. The process begins when two ontologies are first loaded (i.e., O_1 and O_2) and they are identified as SO and TO.

The taxonomies are read and translated for beginning matching. An equivalent test (E) is carried out for data labels to test their similarity in terms of three parameters, (1) test for semantically equivalent data labels, (2) test for synonyms, and (3) test for similar slots or attribute names. *C* is used to refer to classes and *c* refers to attributes or slots. In the next section we discuss semantic and syntactic matching.

SEMANTIC AND SYNTACTIC MATCHING

Syntactic Matching

Syntax is a grammatical rule that refers to the structure of concepts and not their semantics (i.e., structure and meanings). The main distinction between syntax and semantics is that *syntax* always refers to form and structure. It uses approximate string, substring, prefix, and suffix matching for data labels (Chapulsky, Hovy, & Russ, 1997). The implicit reference made today is that syntactic matching always matches semantic matching results. This however, is not always true as mentioned in the SOAP example earlier.

Syntactic matching often results in less reliable results. They rely heavily on grammatical rules and do not support semantic similarity. Syntactic integration defines rules in terms of class and attributes names and does not take into account the structure of the ontology. Syntactic mapping also does not entail coordination of meanings or agreed definitions. As such, this kind of mapping could be conceptually blind although comparatively easier to implement.

Nevertheless it is useful and saves time in arriving at datasets that are likely to be matched. To demonstrate this, the Levenshtein's distance (LD) string match is used to measure the similarity of our data labels earlier, that is, </price> and </quote>. Here we say *distance (d)* to be the *inverse* of similarity *(s) and (d=1-s)* and on a scale of 0 to 1, if *(d=0),* then *(s=1).*

By applying LD, we first chose *price* as the source string and *quote* as the target string. Then they are syntactically matched on a scale of 1 to 10, LD = 4 denotes (*d*) = 4 and thus similarity (*s*) = 10-*d* = 6. In other words they are 60% similar (i.e., 6/10). When *price* and *cost* are matched, similarity is 50% (i.e., 5/10). Only *price* and *price* resulted in 100% similarity (i.e., 10/10).

Levenshtein defines distance as a number of deletions, insertions, or substitutions needed to transform a source string into the target string. This algorithm has been widely used however it does not apply semantics to match data labels. This also holds true for other syntactic matching systems alike such as suffix matching and prefix matching.

Semantic Matching

Semantics represents meaning. A given word has multiple meanings or commonly referred to as word senses. Our premise is that ignoring word senses in matching data labels causes inaccurate results. As such we propose to include semantics in ontology mediation efforts. Semantics rely on dictionaries to determine synonyms and evaluate concepts that share uncommon words. We use the combination of four measure such as Lin, Gloss Vector, WordNet Vector, and LSA because our experiments show that selected combination of these four measures out of 13 linguistic and nonlinguistic measures (e.g., PMI–pointwise mutual information, Resnik, Jiang-Conrath, Lin, NSS–Normalized Search Similarity) produce more accurate results. Precision and relevance also improved tremendously with HCR scores. All scores obtained from the selected four measures were then aggregated and normalized for producing SRS.

Experiments were conducted to compare human cognitive scores (i.e., HCR Rank) based on a 30 word-pair cognitive study carried out at Princeton (Miller & Charles, 1991). Table 1 shows semantic scores (Sem Rank) that were derived from cognitive measures mentioned previously, HCR, Syn, and SRS scores. We will now discuss the experiment carried out in the next section.

Experiment Design and Results

Fifty questionnaires were distributed to domain experts for this experiment. The respondents were carefully selected and only high school English teachers who taught English as a second language were picked. The idea was to choose only people who were highly skillful in the language to be able to rank the 30 word pairs. Obviously if we were to evaluate the domain of neurons then we would use neurologists as they would understand that domain better. Out of 50, 38 responses were processed and 12 survey responses were removed, as they were incomplete. The study had a 100% response rate. The questionnaire was focused on testing human judgment for similarity of 30 word-pairs (see Table 1).

Scores given by **respondents** is labeled as *HCR Rank*, *Syn Rank* denotes **syntactic** scores, *Sem Rank* denotes **semantic** score and lastly *SRS Rank* is the **hybrid** score that combines semantic and syntactic scores. Respondents were asked to rank on a scale of 0 to 10 for all 30 word-pairs. Rank 0 was for an unrelated word pair and 10 was for a highly related word pair according to their cognitive similarity judgment. They were instructed not to assign the same rank twice for the same word category. For instance if *lad* appeared twice for a word-pair then they should not give the same rank for another instance which included *lad* in it. This was to ensure that their previous answers did not have an effect on the new answers and also to prevent bias answers. The higher the score given by the respondents meant that the similarity of the word-pair was higher based on their cognitive reasoning.

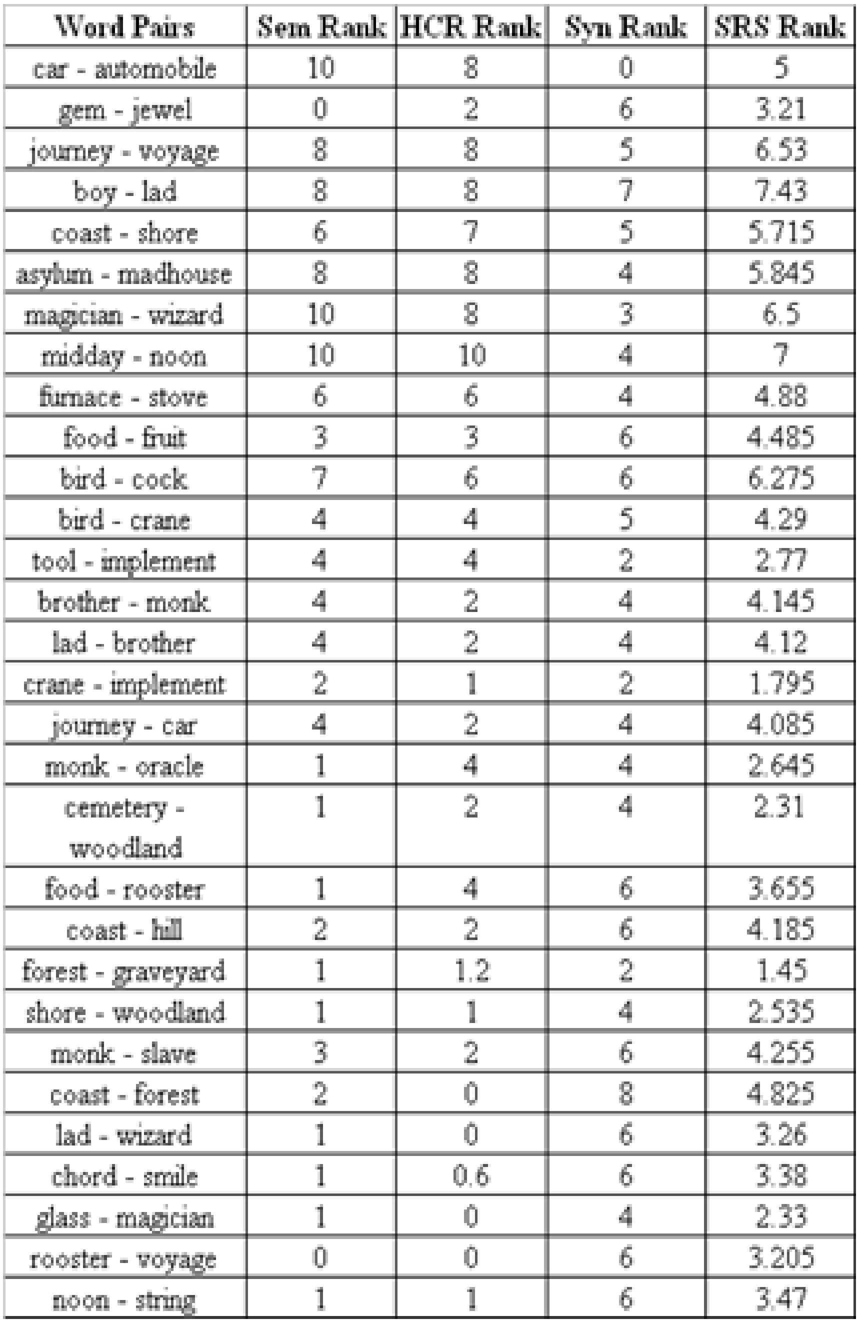
Figure 1 shows the results obtained for all four ranks, that is, *Sem Rank, HCR Rank*, *Syn Rank,* and *SRS Rank*. It illustrates the symbols that we used to represent the 30 word-pairs (i.e., a to ad). All the ranks have been scored from a scale of 1 to 10 and the higher the score indicated the higher the similarity between the word-pairs. Symbol "a" for "car-automobile" was scored 10 by semantic match (i.e., Sem Ranks) and 0 by syntactic match (i.e., Syn Ranks) and so on. Semantic scores (Sem Rank) has the closest match to human responses (HCR Rank), that is, 92% match and when combined with syntactic scores (Syn Rank), that is, the hybrid score, the correlation was 80%.

However the pure syntactic scores were clearly inaccurate, not only they had a weak but also a negative correlation with the HCR Rank. As such this supported our hypothesis that pure syntactic scores are not accurate for matching data labels as part of the ontology mediation process. We still use syntactic scores to eliminate erroneous data labels. After that we apply semantic match results to obtain the closest match to the human domain expert's ranking. Semantic agreement and semantic affinity measurement use cognitive measures derived from WordNet including synonymy, meronymy, antonymy, functions, and polysemy associations (Silva & Rocha, 2003). In the next section we run some hypothesis tests to validate our approach.

Hypothesis Tests and Results

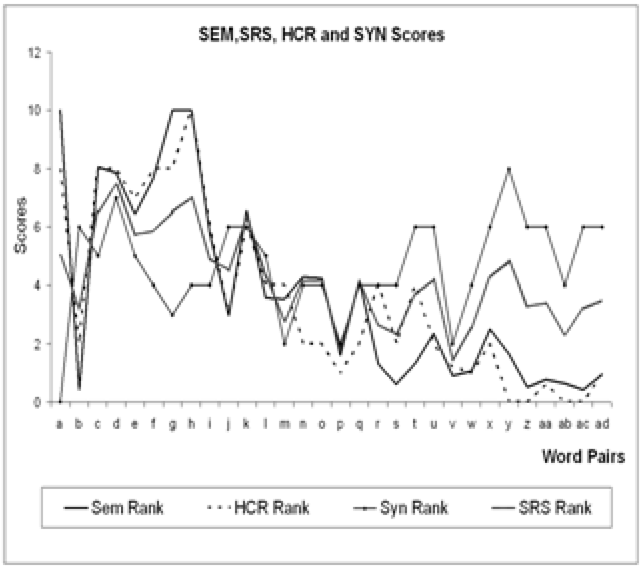
Thirty word-pairs were used in our experiment. Among them, 10 were highly related word-pairs (which should yield scores between 7 to 10), 10 were intermediately related word-pairs (which should yield scores between 3 and 6), and 10 were unrelated word-pairs (which should yield scores between 0 and 2). SRS scores were calculated based on Sem Rank scores and Syn Rank scores that were summed and averaged (see Table 1). HCR scores were obtained from domain experts who ranked the 30 word-pairs. The results were compared with the combined scores (i.e., syntactic and semantic). To prove the hypothesis that combined scores (i.e., SRS)

Table 1. Word-pair ranks



Word Pairs	Sem Rank	HCR Rank	Syn Rank	SRS Rank
car - automobile	10	8	0	5
gem - jewel	0	2	6	3.21
journey - voyage	8	8	5	6.53
boy - lad	8	8	7	7.43
coast - shore	6	7	5	5.715
asylum - madhouse	8	8	4	5.845
magician - wizard	10	8	3	6.5
midday - noon	10	10	4	7
furnace - stove	6	6	4	4.88
food - fruit	3	3	6	4.485
bird - cock	7	6	6	6.275
bird - crane	4	4	5	4.29
tool - implement	4	4	2	2.77
brother - monk	4	2	4	4.145
lad - brother	4	2	4	4.12
crane - implement	2	1	2	1.795
journey - car	4	2	4	4.085
monk - oracle	1	4	4	2.645
cemetery - woodland	1	2	4	2.31
food - rooster	1	4	6	3.655
coast - hill	2	2	6	4.185
forest - graveyard	1	1.2	2	1.45
shore - woodland	1	1	4	2.535
monk - slave	3	2	6	4.255
coast - forest	2	0	8	4.825
lad - wizard	1	0	6	3.26
chord - smile	1	0.6	6	3.38
glass - magician	1	0	4	2.33
rooster - voyage	0	0	6	3.205
noon - string	1	1	6	3.47

Figure 1. SEM, SRS, HCR, and SYN scores



provided a better match with HCR Ranks, the following hypothesis test was carried out:

(H0): *SRS scores do not match expert responses (HCR Rank)*

(H_1): *SRS scores match expert responses (HCR Rank)*

Table 2 illustrates the significant relationship between SRS and HCR ranks. This shows that there was a significant positive correlation between the two scores, that is, r = + 0.806 (i.e., 80.6%). The asterisks indicate significant correlation at 0.01, level (2-tailed). Significance value (p) for this 2-tailed test is <0.05 thus (r =0.806, p <0.05) rejects the null hypothesis (H_0) and accepts the alternate hypothesis (H_1). This proves that to achieve the Semantic Web dream, both syntactic and semantic matching must be given importance.

A t-statistic was also measured for Table 2 to test the hypothesis and with the r coefficient = +0.806. *t* resulted in 7.205 with the given degree of freedom of (n-2 = 28) and given the α = 0.01, the critical value of *t* was 2.7633. Since the t-statistic of 7.205>2.7633, this clearly rejected the null hypothesis (H_0) and the alternate hypothesis (H_1) was accepted.

Table 3 illustrates that there was a higher positive correlation between Sem Rank and HCR Rank. This shows that there is a significant positive correlation between the score, that is, r = + 0.919 (i.e., 91.9%). The asterisks indicate significant correlation at 0.01, level (2-tailed). Significance value (p) for this 2-tailed test is <0.05 thus (r =0.919, p <0.05) also accepts the alternate hypothesis (H_1) earlier as well. This

Table 2. Correlation SRS and HCR

		SRS Rank	HCR Rank
SRS Rank	Pearson Correlation	1	.806(**)
	Sig. (2-tailed)		.000
	N	30	30

Table 3. Correlation Sem and HCR

		Sem Rank	HCR Rank
Sem Rank	Pearson Correlation	1	.919(**)
	Sig. (2-tailed)		.000
	N	30	30

is because SRS Rank is made up of Sem Rank and Syn Rank. This proves that to achieve the Semantic Web dream, both syntactic and semantic matching must be given importance.

A t-statistic was also measured for Table 3 to test the hypothesis and with the r coefficient = +0.919. Since the t-statistic of 12.334 >2.7633, this clearly rejected the null hypothesis (H_0) and the alternate hypothesis (H_1) was accepted.

RELIABILITY TEST: SRS AND HCR

Precision, recall and the F-measures are currently standard test measures for IR systems; however, only the precision measure is appropriate for this study. A new test called the reliability test is introduced in this article for validating SRS and HCR scores earlier. The reliability test in this context is a function of precision and relevance, that is, Reliability (REL) = {precision and relevance}. Precision is denoted as (P_s) and relevance as (R_L), thus the function for reliability can be denoted as:

$$REL = \{P_s \text{ and } R_L\} \quad (3)$$

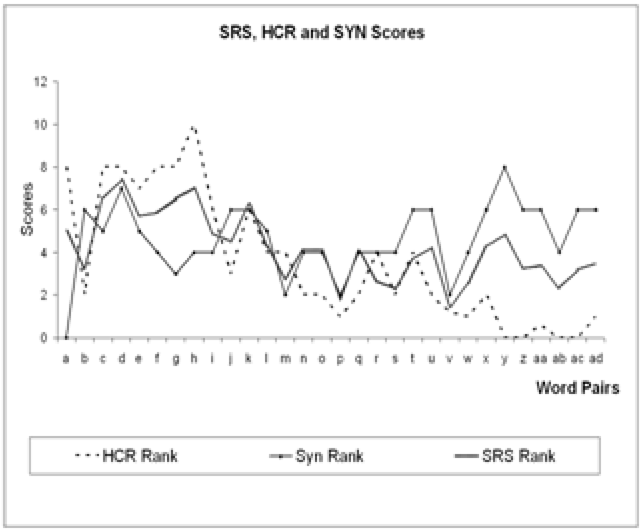
Precision (P_s) and relevance (R_L) is measured as:

$$P_s = \frac{\text{number of correct responses}}{\text{total number of responses}} \quad (4)$$

$$R_L = \frac{\text{number of relevant responses}}{\text{total number of responses}} \quad (5)$$

There are two parts to reliability, that is, precision and relevance. The semi-automated ontology mediation system is meant to reduce the workload of the ontologist, thus the ontologist must be served with reliable information before they decide to choose data labels to be matched. The hypothesis here is that SRS scores that include syntactic and semantic measures

Figure 2. SRS, HCR and SYN scores



are more reliable to an ontologist compared to using SYN scores. The null and alternate hypothesis is stated as:

(H0): *SRS scores are less reliable than SYN scores*

(H1): *SRS scores are more reliable than SYN scores*

Precision (P_s)

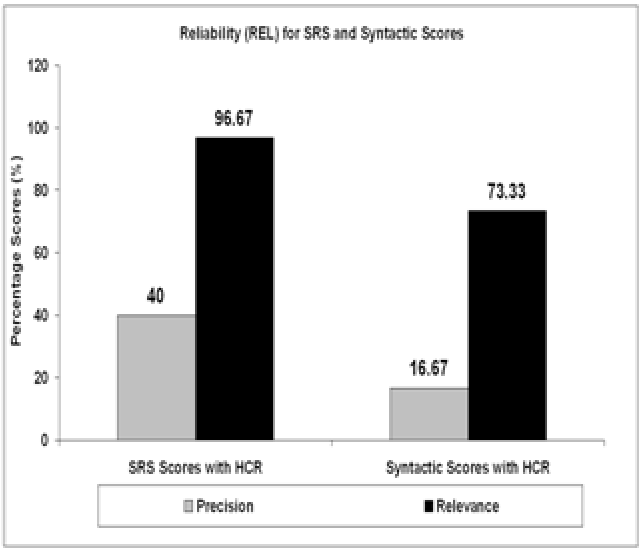
Considering 30 word-pairs, to calculate precision (P_s) all the SRS scores were first normalized. After which the HCR responses were matched against them. The idea was to compare exact matches only. Out of 30 pairs, there were 12 exact matches. Although there were ones that were really close but because they were not exact matches they were not considered for this test. Final precision score for the *SRS* score was 40% (P_s =12/30), that is given the equation above 12 correct responses were discovered out of 30 responses in total. However, the precision score for *only syntactic* match resulted in only 5 correct responses out of 30 responses in total. The precision score for syntactic match was 16.67% (P_s =5/30), which was lower than the SRS scores. In summary, SRS scores provided higher precision.

Relevance (R_L)

The same number of word pairs was tested for relevance (R_L). The SRS scores and HCR scores were matched. The relevance score (R_L) for the SRS scores was 96.67% (RL=29/30). This was for 29 relevant responses out of 30 responses. The relevance score (R_L) for *only syntactic* matches resulted in 22 relevant responses out of 30 responses in total. As such, pure syntactic match resulted in only 73.33% (R_L =22/30). In summary, SRS measures provided better relevance scores.

The hypothesis test indicates that SRS scores that include syntactic and semantic measures are more reliable to an ontologist compared to using purely SYN scores. Thus the null hypothesis (H_0) is rejected and the alternate hypothesis (H_1): is accepted. This due to higher precision and relevance of the SRS scores compared to pure syntactic match scores. This attributed to higher reliability as well.

Figure 3. Reliability of SRS and syntactic scores



CONCLUSION

This article stresses why semantic mediation should be included in ontology mediation systems. The hypothesis here was that syntactic matches alone would not suffice as they are usually based on prefix substring and suffix matching. The importance of coupling semantics and syntactic matching was empirically tested to support this theory. New measures were introduced such as precision, reliability, and relevance, which by themselves were a significant contribution. Empirical tests were conducted to validate our approach including hypothesis tests, t-statistics, reliability, and relevance measures. The main benefit of our approach is that erroneous data is filtered so now instead of going through 30,000 concepts, the ontologist only has to deal with one tenth of the data. We are not trying to prove that our matching algorithm is superior to others but we are introducing an element that is significant for concept matching in ontologies, which is the cognitive or linguistic element (Hirst & St-Onge, 1998). We believe that given the SRS measures, the workload of the ontologist would be significantly reduced. The ontologist will only select from fever concepts now and as such their productivity will improve drastically. In this article we have provided a detailed matching algorithm as well as a new evaluation measure. We recognize that modification of context does modify the semantic similarity of concepts. This aspect has not been included in this article and is a significant part of our future work.

ACKNOWLEDGMENT

Our matching algorithm expands the work carried out by researchers, Li Li, Baolin Wu, and Yun Yang of the Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, Australia. In particular, steps 2 to 5 of our matching algorithm incorporates the definitions provided in their article. We would like to thank Li Li, Baolin Wu, and Yun Yang for their prototyping work that allowed us to expand and build a new matching algorithm which includes the SRS. Their work also inspired us to build our own prototype using the JADE platform.

REFERENCES

Barbir, A. (2002). Web services security: An enabler of Semantic Web services. *Nortel Networks,* 1-5.

Chapulsky, H., Hovy, E., & Russ, T. (1997). *Progress on an automatic ontology alignment methodology.*

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2002). Ontology matching: A machine learning approach. *Proceedings of the 11th International Conference on the World Wide Web,* Honolulu, HI (pp. 662-673).

Fowler, J., Nodine, M., Perry, B., & Bargmeyer, B. (1999). Agent-based semantic interoperability in InfoSleuth. *SIGMOD Record, 28*(1), 60-67.

Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum (Ed.), *WordNet: An electronic lexical database* (pp. 305-332). Cambridge, MA: MIT Press.

Kalfoglou, Y., & Schorlemmer, M. (2003). IF-Map: An ontology-mapping method based on information-flow theory. In *Journal of Data Semantics* (LNCS 2800, pp. 98-127). Berlin/Heidelberg, Germany: Springer.

Kurgan, L., Swiercz, W., & Cios, K. (2002). Semantic mapping of XML tags using inductive machine learning. *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)* (pp. 99-109).

Li, L., Yang, Y., & Yu, B. (2005). Agent-based ontology mapping towards ontology interoperability. In *Australian Joint Conference on Artificial Intelligence (AI'05),* Sydney, Australia (LNAI 3809, pp. 843-846). Springer-Verlag.

Maedche, A., Motik, B., Silva, N., & Volz, R. (2002). MAFRA-A mapping framework for distributed ontologies. In *13th International Conference, Knowledge Engineering and Knowledge Management,* Siguenza, Spain (LNCS 2473, pp. 235-250). London: Springer-Verlag.

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*(1), 1-28.

Missikoff, M., Schiappelli, F., & Taglino, F. (2003). A controlled language for semantic annotation and interoperability in e-business applications. *Proceedings of the 2nd International Semantic Web Conference (ISWC 03),* Sanibel Island, FL (pp. 1-6).

Muthaiyah, S., & Kerschberg, L. (2006). Dynamic integration and semantic security policy ontology mapping for Semantic Web services (SWS). In *First IEEE International Conference on Digital Information Management (ICDIM),* Bangalore, India (pp. 116-120).

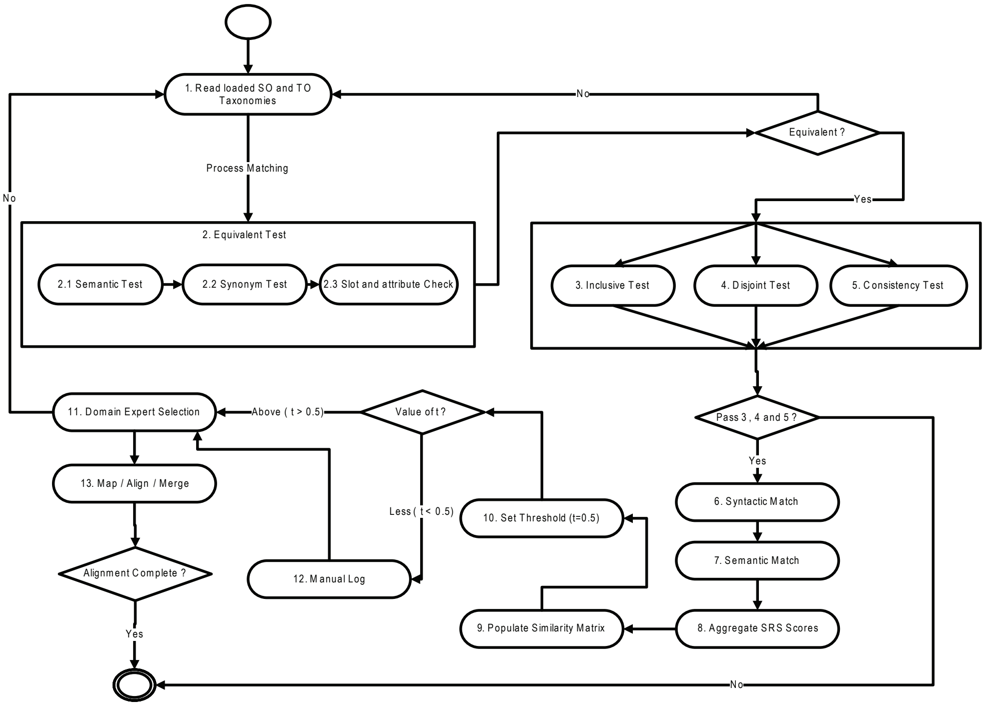
Muthaiyah, S., & Kerschberg, L. (2007). Virtual organization security policies: An ontology-based mapping and integration approach. *Information Systems Frontiers, 9*(5), 505-515.

Noy, N. F., & Musen, M. A. (1999). *SMART: Automated support for ontology merging and alignment.* Paper presented at the Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management, Banff, Canada.

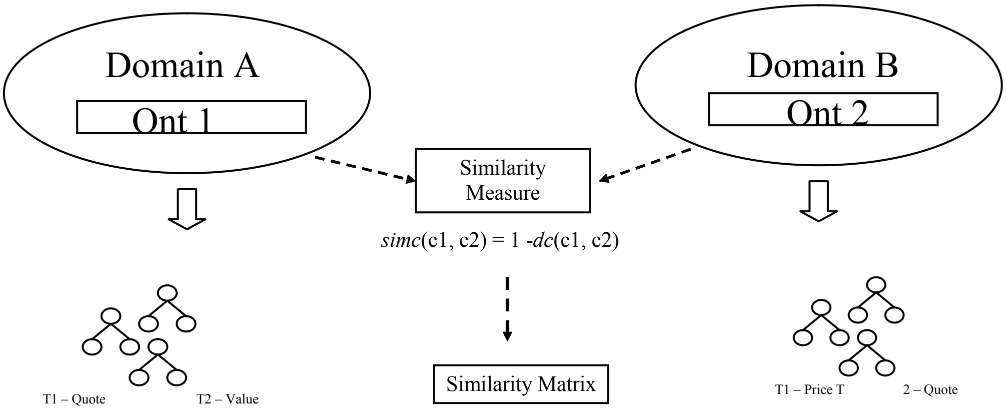
Noy, N. F., & Musen, M. A. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence,* Austin, TX: MIT Press.

Silva, N., & Rocha, J. (2003). Semantic Web complex ontology mapping. In *Proceedings of the IEEE/WIC International Conference* (pp. 82-88).

APPENDIX 1. MATCHING ALGORITHM



APPENDIX 2. SIMILARITY MATRIX



		Ont 2		Ont 4			
		T1	**T2**	**T3**	**T4**	…….	**Tn**
Ont 1	**T1**	0.7					
	T2						
Ont 3	**T3**	0.43					
	T4		0.89				
	…..						
	Tn						

Saravanan Muthaiyah is a senior lecturer at Multimedia University, Cyberjaya, Malaysia and currently a doctoral degree candidate in Information Technology at George Mason University, Fairfax, VA. He holds a master's degree in information technology and other degrees in the area of accounting and finance for his bachelors. He is also a Fulbright scholar under the auspicious graduate research exchange program sponsored by the US Department of State. His research interests include semantic web, ontology mapping, systems integration, systems engineering, topic maps, knowledge management and enterprise architectures. His recent papers have focused on ontology mapping and mainly solving heterogeneity issues for Semantic Web.

Larry Kerschberg is professor of computer science, at George Mason University, Fairfax, VA 22030 USA. He is director of E-Center for E-Business and directs the MS in E-Commerce Program. He is past chairman of the information and software engineering department at Mason. During 1998 he was a Fellow of the Japan Society for the Advancement of Science at Kyoto University. He holds a BS in engineering from Case Institute of Technology, and MS in Electrical Engineering from the University of Wisconsin—Madison, and a PhD in engineering from Case Western Reserve University. He is editor-in-chief of the Journal of Intelligent Information Systems*, published by Springer. He recently served as an editor and contributor of the book* The Functional Approach to Data Management: Modeling, Analyzing and Integrating Heterogeneous Data*, published at Heidelberg, Germany, Springer, 2004. His areas of expertise include expert database systems, intelligent integration of information, knowledge management, and agent-based semantic search. His recent papers have focused on ontology-driven semantic search in knowledge sifter, knowledge representation using topic maps, and methodologies for the creation and management of semantic Web services.*