SENSITIVITY ANALYSIS OF NON INSTANCE BASED LEARNING APPROACH FOR ONTOLOGY ALIGNMENT USING SSFPOA

S.Jayaprada[1] Sr Asst Professor
S.Vasavi[2] Professor
P.Bala Krishna Prasad[3] Professor
[1,2]Department of Computer Science & Engineering VRSiddhartha Engineering College (Autonomous)
Vijayawada , Andhra Pradesh, India
[3]Sri RamaChandra engineering College, Eluru
jayaprada79@yahoo.com[1], vasavi.movva@gmail.com[2]

Abstract— The invent of internet and Web have paved way to information sources belonging to same domain to be distributed that are structurally (to some extent) and semantically heterogeneous. In order to achieve semantic interoperability within these information sources heterogeneity has to be solved which exists at various levels such as at data, operating system or due to hardware heterogeneity. Many methods were proposed to solve data heterogeneity problem using ontologies. In this paper we considered ontology alignment as data mining problem and solved using machine learning based classification approaches using our compound semantic measure SSFPOA. Six different tests were made and performance measures such as precision ,recall, accuracy, f-measure and overall are calculated Sensitivity Analysis of each of the approach is calculated by varying the number of metrics and performance of each individual metric is analyzed in order to verify on, does propagation of similarity value after each matcher improving or not. Test results (Simple mappings) proved to be better when compared with existing approaches.

Keywords- Semantic interoperability, heterogeneity conflicts, ontology mapping, classification, machine learning, Sensitivity Analysis

Introduction

In order to achieve semantic interoperability within autonomous and distributed information sources, heterogeneity is a big hurdle. Many methods were proposed to solve data heterogeneity problem using ontology. Ontology is a logical system that refers to an abstract model of real world domain entities with an explicit specification of concepts, relationships and constraints on their use so that it is machine readable [33]. In short we can say that they are knowledge representations that represent shared conceptualization of a particular domain such as bibTex, Gene, etc. Ontology Alignment can be defined as deriving a set of correspondences (mappings) between two or more ontologies. Each correspondence gives semantic relationship between items of various entities such as equal, disjoint, less general, more general or in the range of 0- 1. Existing methods for ontology mapping can be classified as instance based and non instance methods. Instance based methods consider data instances during finding correspondences. Problem with these methods is that they also consider instance data properties for producing similarity value. For example **250km/h** equals to the word **fast** . So **speed** label is mapped to **characteristic** label. Semantic similarity can be found using edge counting methods[3,14,38] and information theoretic based methods[15,23,28]. Edge counting methods consists of calculating the distance between ontology concepts where similarity decreases with increase in distance. If there are several paths, minimum or average distances is used. Information content of a term decreases with its occurring probability. A new compound measure SSFPOA[29] has been proposed by us which uses 12 different matchers to find semantics. SSFPOA has been tested on product catalogue of B2B trade[30] that focused on interpreting the frequent patterns that are mined, especially extracting semantically similar items and clustering them. In this paper we reduced ontology alignment problem as data mining problem and solved the problem by using machine learning based classification approaches namely Decision tree, SVM, Multi Layer feed forward neural network without considering instances. The advantage of this reduction is that instead of evaluating the results of the method as does by exiting approaches we can evaluate the performance of combination of measures used within the method by appropriate training data. Our approach is similar to [2] in a way that we also used classification techniques but differs in implementing our compound semantic measure SSFPOA. SSFPOA is evaluated on the benchmark tests from OAEI ontology matching campaign 2011[9]- bibliography domain. Section II presents the related work to ontology alignment problem and exiting approaches to solve the problem. Section III describes the ontology test cases and presents experimental results of SSFPOA. Section IV gives conclusions and future work.

II Literature Survey

Ontology mapping approaches can be classified as Heuristic and Rule-based methods [5,18,23,36], Graph based methods[24,31], Machine-learning methods [1,6,17,20,26], Probabilistic approaches[19],Reasoning, theorem proving [25]. Ontology alignment as schema matching problem has been considered by [10,12,13,22,34,37] to find correspondences between pairs of elements of 2 or more ontologies. The same problem has been solved using data mining technique by [2,8,16]. [4,7,11] worked on finding semantics within fuzzy ontologies. We first find morphological root word with the help of Web feature such as wordnet[35] and use NLP techniques such as tokenization, lemmatization, elimination and string based techniques during preprocessing. Weighted sum of 12 matchers represents similarity for a given pair of items and is not just 0 or 1 like used by [2] and others, but in the range of 0-1 to represent different levels of similarity.

PROMPT[23]: It is used for Ontology Merging. In this user should manually enter two related terms (anchors) in two ontologies. Between a pair of similar terms in two ontologies, it first finds the set of all paths with equal length. Now PROMPT traverses the paths between the terms to find similar terms (exact labels). As it traverses, it increases the similarity score for the pair of terms at the same position For each different path between the anchors, it repeats the same process and finally aggregates the result. If for any pair of anchors, paths are not of same length, then output is empty (i.e) if one ontology is a deep one and the other shallow then it could not produce better results Finally, nodes with highest score are extracted.

Prior+ [22] :For linguistic similarity PRIOR+ takes only edit distance, which does not work when two concepts are semantically similar but lexically different (e.g. synonyms). Similarly for structural similarity it considers features such as the depth of the element, the number of its subclasses. If the depth of the items is same structural similarity is increased. Also it considers only those items for which number of subclasses is same. But sometimes few elements of one class may map to other class which is not considered by the approach. For the test case 303 Vs 304 it produces the result of **m(Proceeding, Proc, =, .36)** indicating more number of adjustments. For 301 Vs 304, even though the similarity score between *m*(Reference, Composite, =, .11) is .11, "=" will be the mapping output because it is the only mapping candidate left.

Chimaera[5]: It is used for Ontology Merging. Matching is based on the names of items supported by user feedback.

Cupid[13]: CUPID first performs linguistic matching (name, datatype, abbreviation) to form LSIM coefficient. Then it performs structural similarity to form SSIM coefficient. If two element names are equal then it increases the similarity coefficient of their corresponding children. otherwise decreases.

Similarity flooding[31] : Similarity Flooding (SF) constructs two graphs, for each node neighbors are name, datatype, instance data. In the first iteration it checks with name (suffix, prefix), if they are equal then it increases confidence measure for 2nd iteration (datatype checking) otherwise not. In the second iteration checks datatype similarity, if they are equal then it increases confidence measure for 3rd iteration (instance checking) otherwise not. In the third iteration compares instance data. Finally it produces similarity value between 0 to 1. It doesn't consider structural similarity

QOM[18]: Labels are very important for mapping. If labels are same, the algorithm infers that the entities are also same. For that it uses:

String equality as $sim_{streq}(c,d)=1$ if $\forall i$ $c.char(i) = d.char(i)$

$= 0$ otherwise

String similarity is calculated based on edit distance. It also considers instance data properties for producing similarity value. EX: 250km/h equals to the word **fast** . So **speed** label is mapped to **characteristic** label.

IF-Map[36]: It is used for Ontology Merging, It finds mappings based on channel theory, a mathematical theory of semantic information flow. It initially finds relation names from both ontologies that are syntactically equivalent, check if their argument types match. Also use these types to fix a partial map to start the infomorphism generation. If step 2 fails, then traverse the is-a hierarchy of types and finds syntactically common types that subsume or are subsumed by the common relations argument types Those that are found syntactically equivalent will be used as in step 3 for partially fixing the initial map of the two ontologies. If step 2 yields only one argument type match, use it and do step 4 for the other argument type.

GLUE[1]: For the given two ontologies, GLUE finds the most similar items between them instead of using a similarity measure. The output of Similarity Estimator is a similarity matrix between concepts of two ontologies. In this approach, the accuracy between portion of correct mappings changes in a large interval.

LSD[6] Name learner assigns label basing on its name EX: for **location** label **name** is given. Then Naive Bayes Learner assigns a label to an element based on its data value EX: for the value **250Rs** label **price** is given. A Base learner uses the training data to learn for each pair of mediated tag name. In LSD unmatched nodes cause decrease in efficiency. Other reasons for decrease in efficiency are: it does not use format learners, as such elements which are compatible could not be matched. Also if an element consists of more than one token , for such elements it could not find matching. Also if training data does not have the elements then t could not produce mappings.EX: **country** could not be matched to **area**

SemInt[17] Uses a neuralnetwork learning approaches,It matches schema elements based on attribute specifications and statistic of data content and Exploit both schema and data information .They do exploit previous matching efforts

APFEL [20] it is based on the general observation that alignment methods like QOM [18] or PROMPT [23] and extracts additional features by examining the ontologies for overlapping features, including domain-specific features. All features are combined in a combinatorial way with a generic set of predefined similarity assessments including similarity measures for, e.g., equality, string similarity, or set inclusion. Thus, APFEL derives similarity assessments for features.

Anchor Prompt [24]: It is used for Ontology Merging. In this user should manually enter two related terms (anchors) in two ontologies. Between a pair of similar terms in two ontologies, it first finds the set of all paths with equal length. Now PROMPT traverses the paths between the terms to find similar terms (exact labels). As it traverses, it increases the similarity score for the pair of terms at the same position. For each different path between the anchors, it repeats the same process and finally aggregates the result. If for any pair of anchors, paths are not of same length, then output is empty (i.e) if one ontology is a deep one and the other shallow then it could not produce better results Finally, nodes with highest score are extracted.

PSN[26] solves ontology mapping problem train multiple tasks simultaneously on a partially shared feed forward network. Each ontology has its own input bank and output bank, middle part of the network is shared by all ontologies. Only structure information is used to train the network.

OMEN[19](Ontology Mapping ENhancer) The Enhancer utilizes an electronic lexicon to adjust the similarity values that have been computed by the mapper, with the intention of re-ranking the mapping assertions in the result list. The Mapper performs a computation of a correspondence measure for the pairs of compared ontology elements, based on the similarity of their enriched structures. OMEN uses a Bayesian network to represent the influences between potential concept mappings across ontologies. the method we present herein also contains inference over networks, albeit with several improvements.

S-Match[25] S-Match uses wordnet during mapping. First, semantic similarity between words changes across domains. Even though a thesaurus may contain a sufficiently wide range of common words, sometimes it does not cover special domain vocabulary. For example, though *apple* is frequently associated with *computers* on the Web, this sense of *apple* is not listed in WordNet. Second, new words are continually created and new senses are assigned to existing words. Thesauri usually can not capture these new words and senses in time. Third, words which are not captured by wordnet are considered as noisy label by S-match

Falcon-AO[10]: The method has preprocessing steps such as tokenization, stemming, stop words, synonyms. It uses both linguistic and structural comparability of ontologies. Linguistic similarity is performed by using Editdistance matcher. Structural similarity (SS)= $1/(e^{ed}/(S_1.len+S_2.len-ed))$. It uses heuristic rules to integrate the result as follows:

- H1: Combine linguistic matcher value with some random generator number
- H2: Select structural matcher value only if H1 is greater than some threshold value (0.6)
- Basing on H1 value H2 may or may not be selected. also random number generator may generate different values at different times.

Rimom[37]: Name based decision using wordnet . It uses senses of wordnet. If multiple senses are available it takes maximum sense where as S-Match takes both senses and finds editdistance (difference is atmost 3 between the senses of label1 and label2)

$Sim(w_1,w_2)=(sim_d(w_1,w_2)+sim_s(w_1,w_2))/2$

where sim_d is the similarity according to wordnet and sim_s is according to dictionary

It does not use any other auxiliary information nor considers stuctural information for mapping. Uses data type constraints decision. Problem with RiMOM is due to Instance based decision. Even though source and target entities are different, but instances are same then they will be mapped.

Lily[34]: First a subgraph is identified manually. Then finds literal(exact labels) and structural information in the semantic subgraphs. It does not use any external knowledge like wordnet, synonyms nor uses constraint matchers . It can reuse the generated mappings to produce new mappings. EX: l1->l2, l2->l3, it can say about l1->l3. But can not reuse subgraph level results (a subgraph which may be common, its results can be reused)

Asmov[12]: It calculates the similarity by analyzing the following features: textual description (label), external structure (parents and children i.e exact structural similarity), internal structure (values) and these measures are combined into a single confidence value using a weighted sum. It does not consider datatype matcher and also can not find complex mappings.

Our approach: Overview of the Non-instance based Learning Approach: Perform the following steps 1 to 7 for each of the pair of ontologies. Part of the ontology test case is shown in figure 1.

Figure 1 : Part of ontology test case #301 and #302

1. Generate and normalize various domain dependent and domain independent features (12 matchers) varying from linguistic, syntactic, structural and web features for the given test ontologies.
2. Randomly generate training (80%) and testing (20%) set for a pair of ontologies
3. Train various classification models on the obtained training set.
4 Test these models on the testing set.
5. Cluster the mapping results in the range of 0-1
6. Evaluate the results basing on the parameters precision, recall, f-measure, overall
7. Also calculate the accuracy of each of the classification model

III Evaluating SSFPOA

Table 1 presents the ontologies (Academia) considered for evaluation. This dataset involves ontology pairs given specific alterations of the four real ontologies in the Benchmark series corpus of OAEI dealing with bibliographic references.

Table 1:

For the test case #301, #303,#304 structural similarity does not exist where as #302 poses flat hierarchy. Also linguistic information is adequate in case #301, #302,#304. Altogether these test cases have high similarity from linguistic point of view but low similarity from structural point of view. A total of 6 tests were made on these 4 test ontologies and expected results were shown in table 2 (only leaf nodes status is given) . Virtual root is added to reduce forest to tree and the shared elements are repeated in each of the class tree to convert graph to tree.

Table 2:

For each pair of ontologies .arff file is created with rows as pair of ontology entities and 12 columns representing 12 matchers. 13^{th} column gives the weighted sum of 12 matchers. Predicted cluster is given as 14^{th} column. This file is given as input to classification techniques such as Decision tree (Bayesnet), Support vector Machine (SMO Object) and Neural Network (Multilayer feed forward neural network). Weka tool 3.6.4 version on 2GB RAM, Intel core 2 with 1.86 GHz processor is used. Experimental set up for the three techniques is shown in figures 2,3,4. Multi layered feed forward neural network (5 layered) is used. Exact matcher output of each pair of items is given as input to the network. Hidden layers of the sequence <2,2,6,1> representing remaining 11 matchers are used. 0-1 range forms the output layer (5 units). Back propagation algorithm is used to predict the target value. Weights are normalized to -1 to +1.

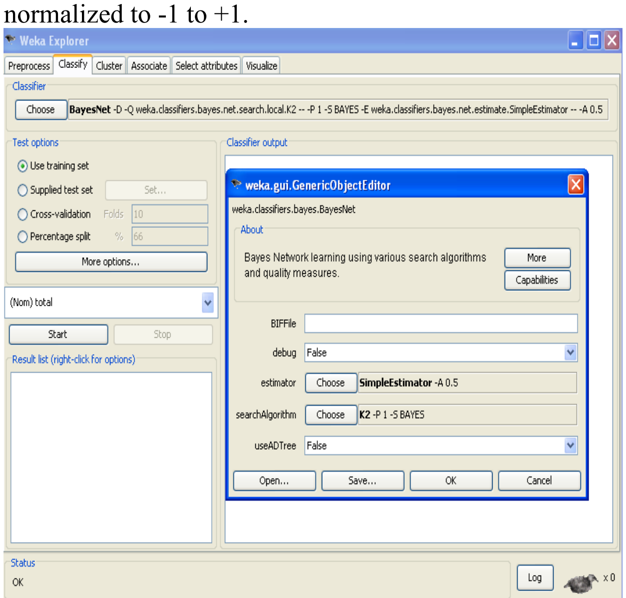


Figure 2: Experimental setup for Bayesnet

Sensitivity Analysis of three classification techniques is performed by supplying the same number of metrics. Performance of each individual matcher is calculated ie Linguistic Matcher (after 3^{rd} , 5^{th} , 6 to 11 matchers) structural matcher (all), exact matcher (3^{rd} matcher) as shown in figure 5 w.r.t the parameters precision, recall and accuracy in order to verify on does propagation of similarity value after each matcher improving or not. Accuracy of multilayer is high in all the 6 cases. Also accuracy of exact matcher is almost equal to using all matchers as the ontologies poses high similarity in linguistic point of view. Using only string matchers (6 to 11) does not improve accuracy of ontology alignment. Adding structural matcher

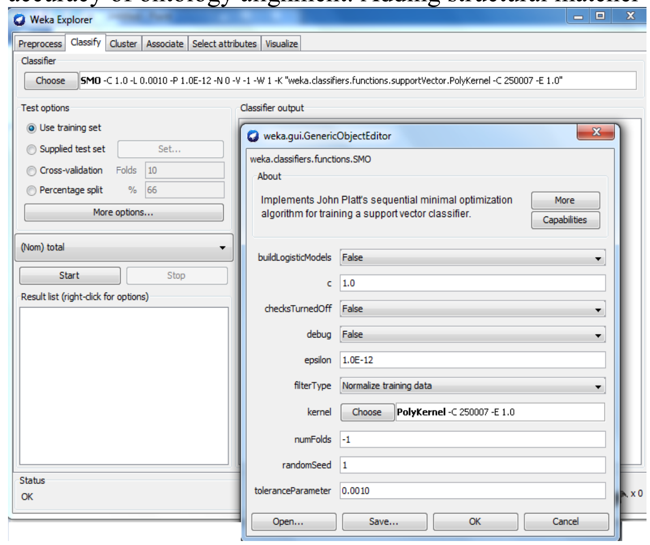


Figure 3: Experimental setup for SVM

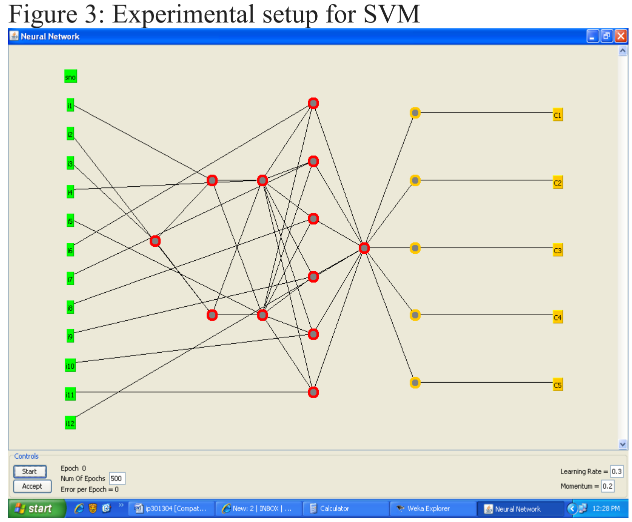
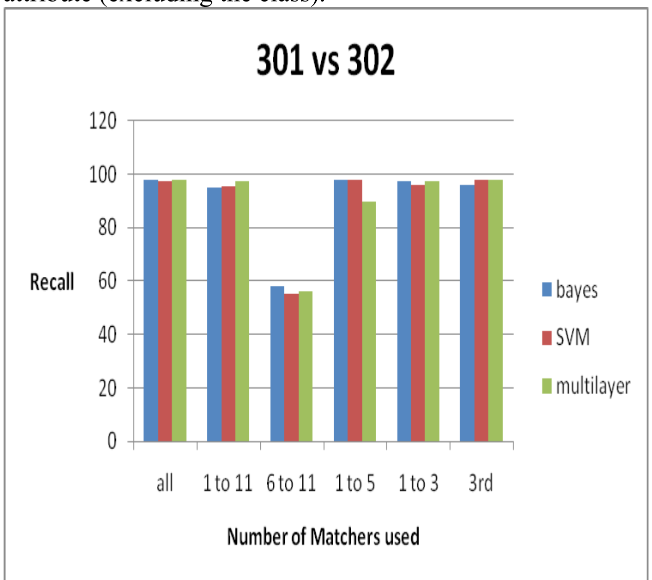
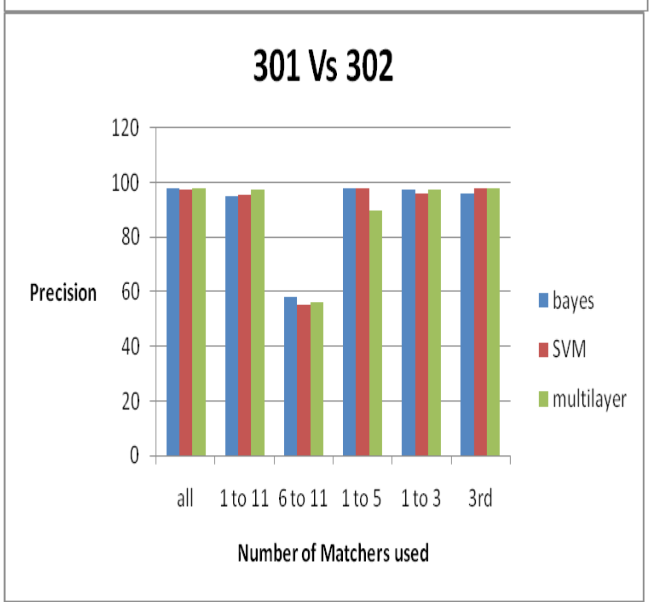
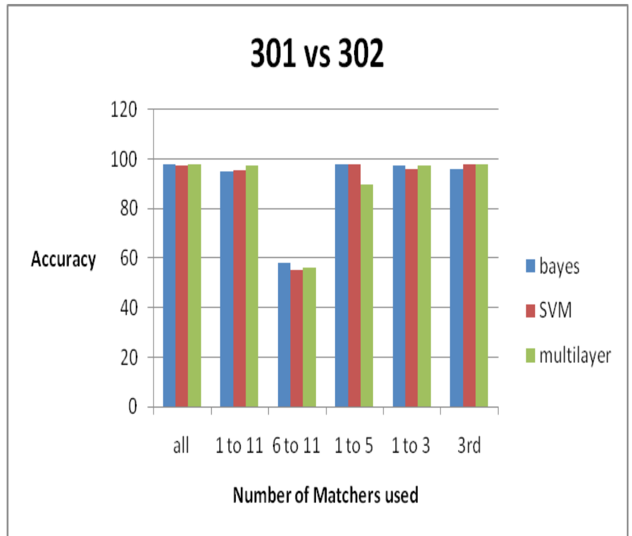


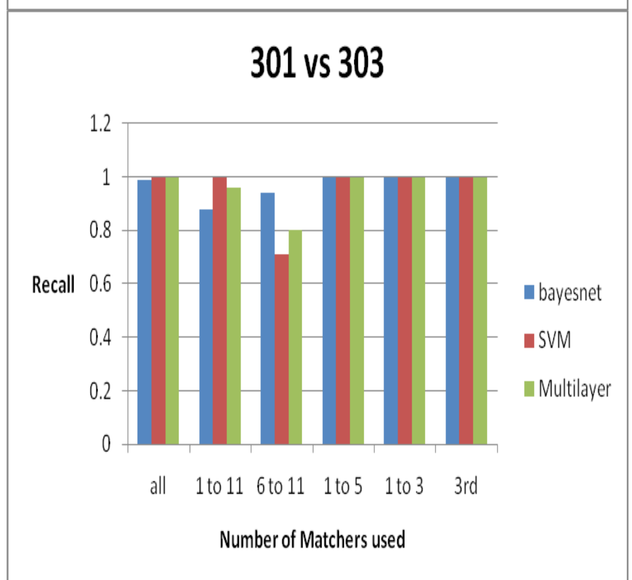
Figure 4: Experimental setup for Neural Network

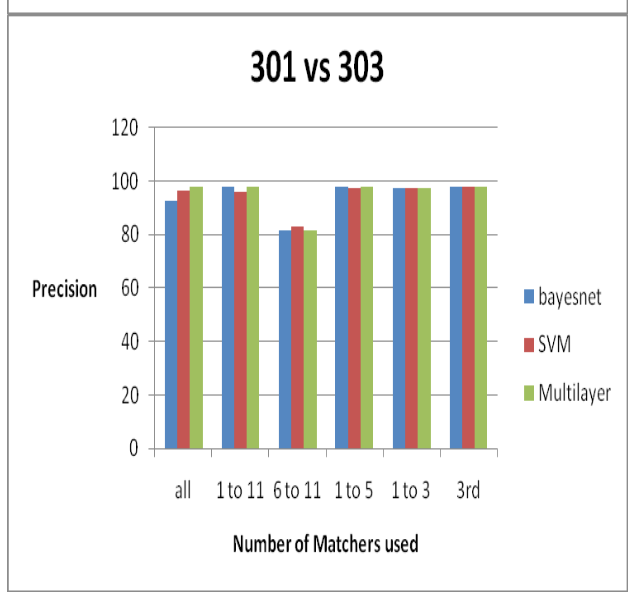
(12^{th}) helped in improving accuracy of test cases which had combination of #301, #303 and #304. Accuracy of Bayesnet (301vs302) has improved from 95% to 98% where as for the same combination SVM and neural network did not show any improvement. For #301vs #303 and #301 vs #304 there is no improvement in accuracy for all the three classification techniques. For #302 vs #303 SVM performed better than Bayesnet and multi layer neural network. For #302 vs #304 all the three classification techniques improved performance from 93% to 95%. For #303 vs #304 all the three classification techniques performed equally well. Precision and recall for test cases which had combination of #301, #302 and #304 are close to each other and is different when #303 is paired. This is due to linguistic similarity that exists within the ontologies. Altogether the performance is good when training data and testing data coincides. RBF kernel. K(x, y) = e^-(gamma * <x-y, x-y>^2) is choosenn in SVM technique that does not impose the condition that instances must have a single nominal attribute (excluding the class).

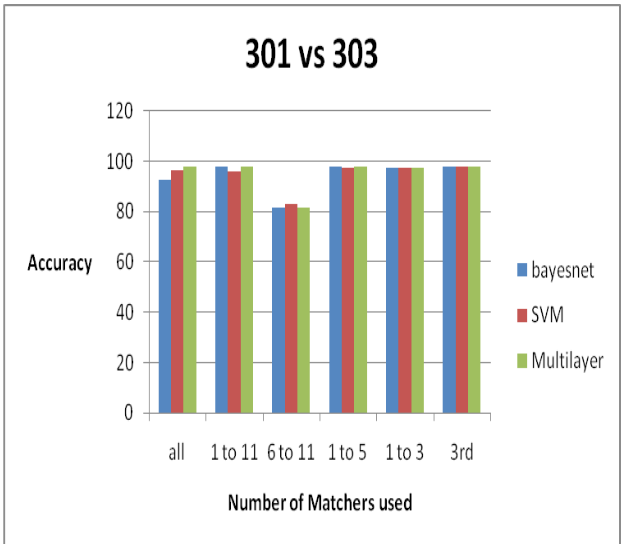


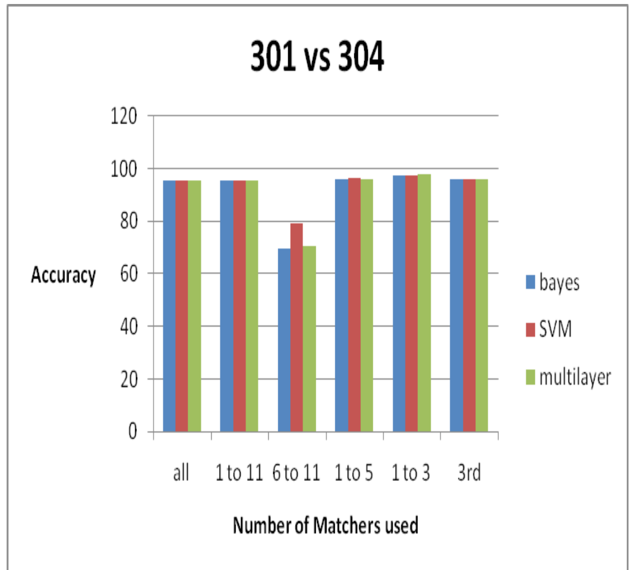


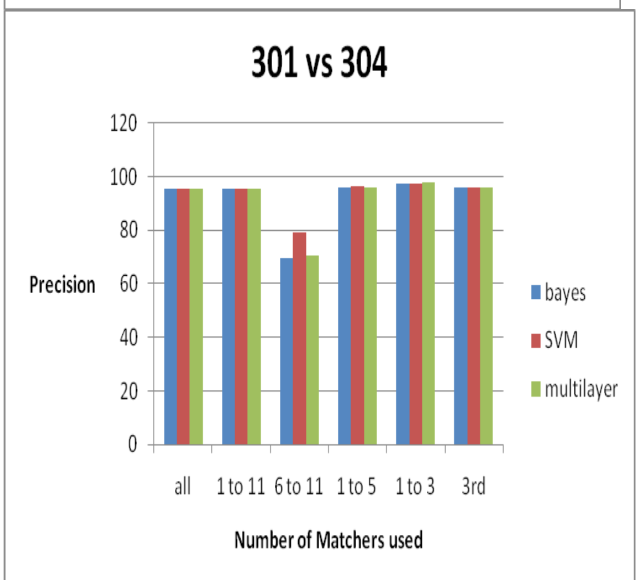


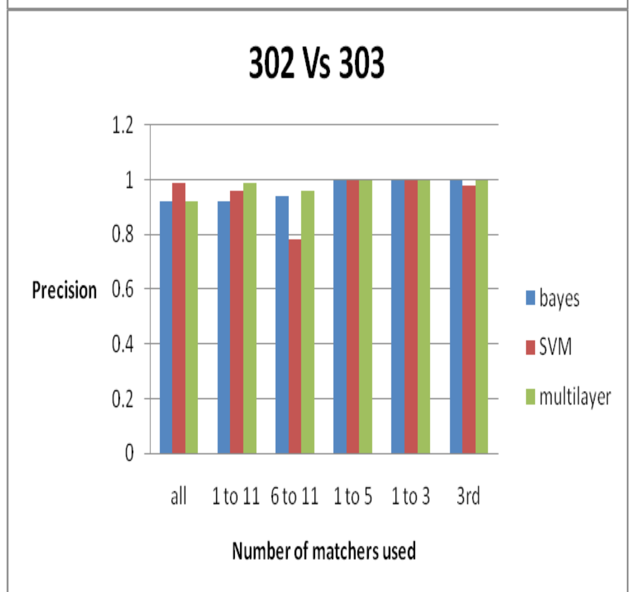


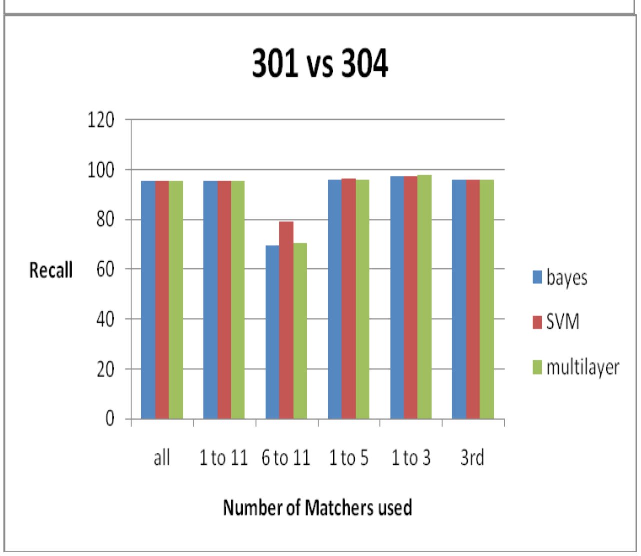


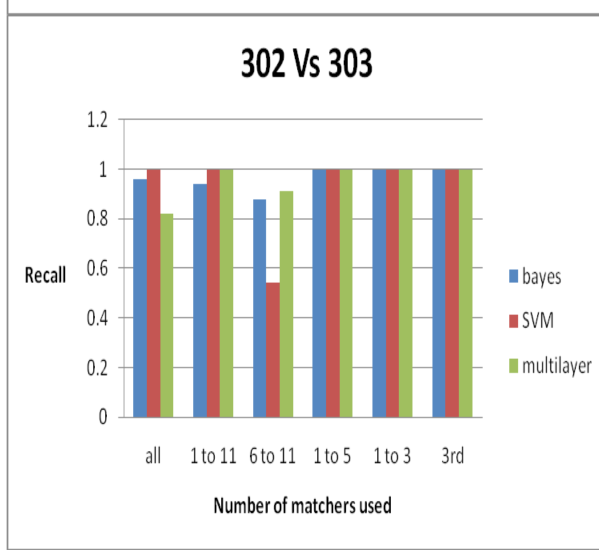

301 vs 303
Accuracy
Number of Matchers used
0
20
40
60
80
100
120
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
bayesnet
SVM
Multilayer

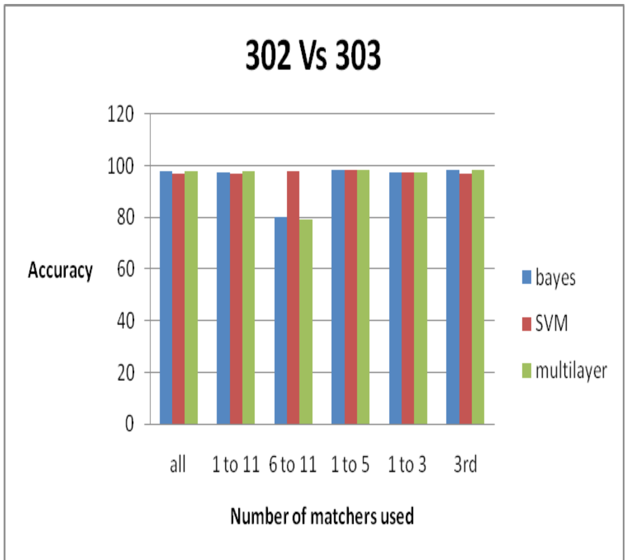

301 vs 304
Accuracy
Number of Matchers used
0
20
40
60
80
100
120
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
bayes
SVM
multilayer

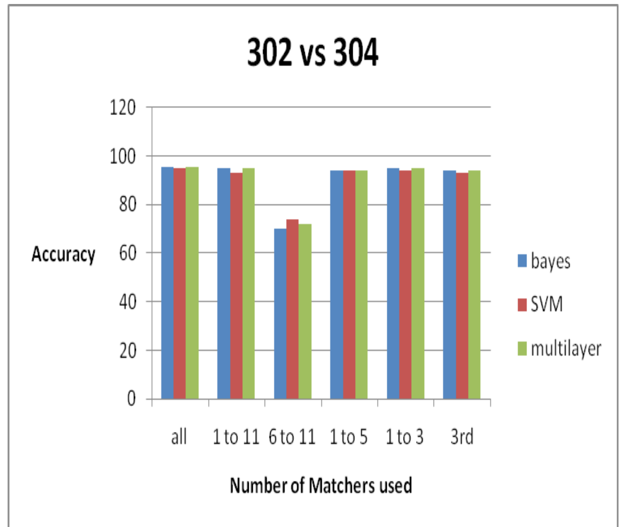

301 vs 304
Precision
Number of Matchers used
0
20
40
60
80
100
120
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
bayes
SVM
multilayer

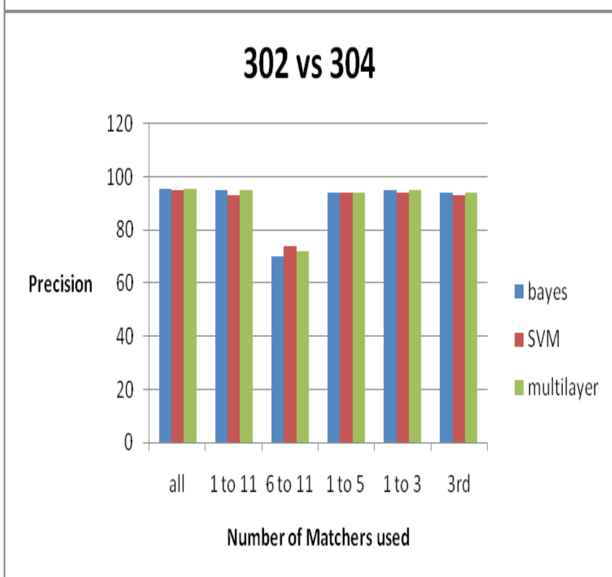

302 Vs 303
Precision
Number of matchers used
0
0.2
0.4
0.6
0.8
1
1.2
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
bayes
SVM
multilayer

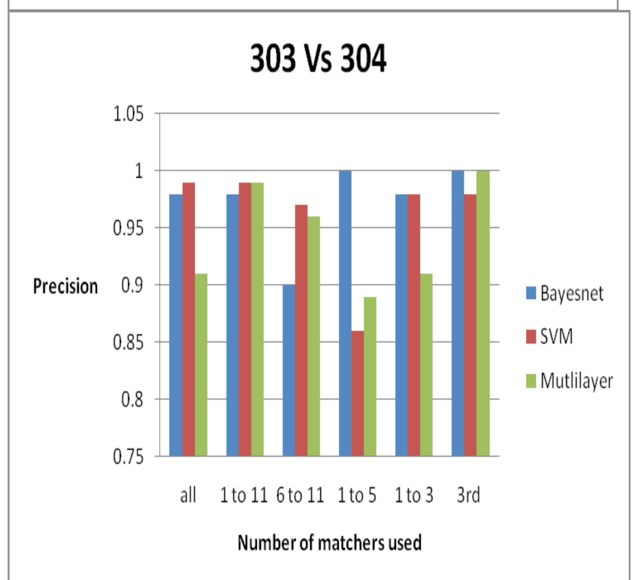

301 vs 304
Recall
Number of Matchers used
0
20
40
60
80
100
120
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
bayes
SVM
multilayer

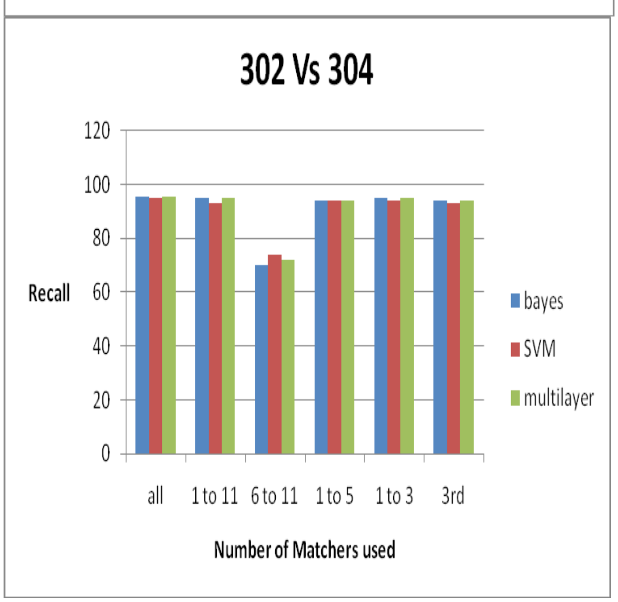

302 Vs 303
Recall
Number of matchers used
0
0.2
0.4
0.6
0.8
1
1.2
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
bayes
SVM
multilayer

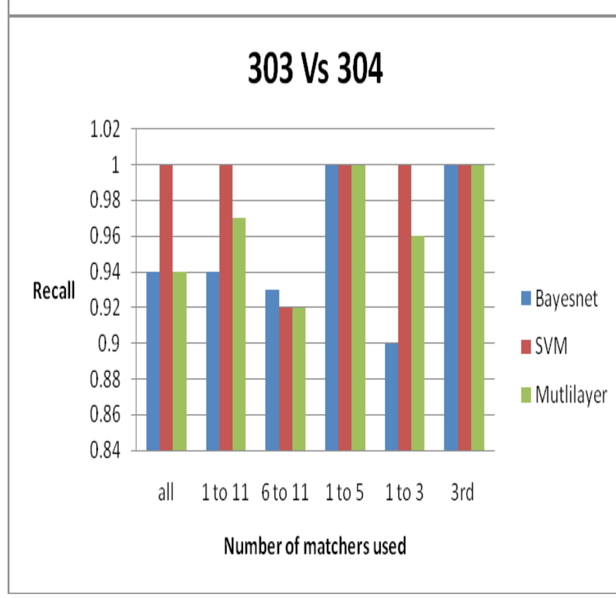

302 Vs 303
120
100
80
60
40
20
0
Accuracy
bayes
SVM
multilayer
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
Number of matchers used

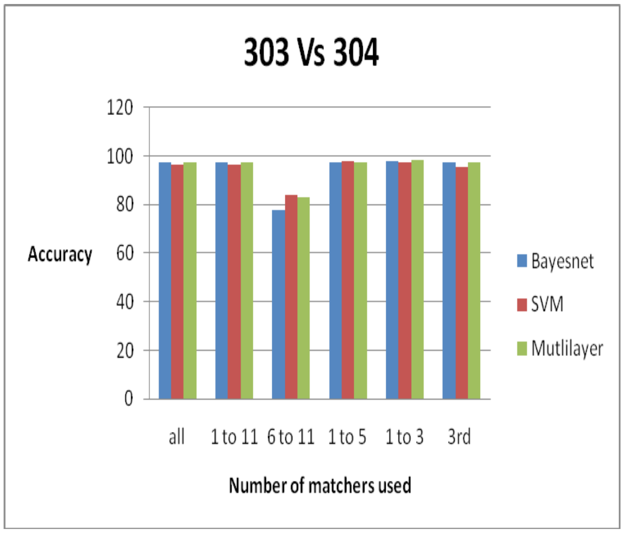

302 vs 304
120
100
80
60
40
20
0
Accuracy
bayes
SVM
multilayer
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
Number of Matchers used


302 vs 304
120
100
80
60
40
20
0
Precision
bayes
SVM
multilayer
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
Number of Matchers used


303 Vs 304
1.05
1
0.95
0.9
0.85
0.8
0.75
Precision
Bayesnet
SVM
Mutlilayer
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
Number of matchers used


302 Vs 304
120
100
80
60
40
20
0
Recall
bayes
SVM
multilayer
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
Number of Matchers used


303 Vs 304
1.02
1
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.84
Recall
Bayesnet
SVM
Mutlilayer
all
1 to 11
6 to 11
1 to 5
1 to 3
3rd
Number of matchers used



Time taken in seconds for each of the models is shown in table 4.

Sno	Testcase	Bayesnet	SVM	Multilayer
1	#301 vs #302	0.02	.38	67.44
2	#301 vs #303	0.09	84.72	11.2
3	#301 vs #304	0.11	15.61	93.95
4	#302 vs #303	0.06	1.33	81.49
5	#302 vs #304	0.03	2.36	149.05
6	#303 vs #304	0.02	0.73	71.69

Table 4: Time taken to build various models

It can be observed that if recall is maximized, precision is lowered and vice versa. Hence we cannot measure system performance basing on these two measures alone. F_measure from information retrieval field [32] and overall measure defined in [13] can be used for measuring accuracy.

F_Measure($\propto$) =(P*R)/(1-$\propto$) * P+ $\propto$*R), Where $\propto$ can range between 0 to 1.

Also when $\propto$ = 0 we can observe that F_Measure = Recall

and is equal to precision when $\propto$ =1. When $\propto$ is taken as 0.5

we have F_Measure = 2*(P*R)/(P+R) and Overall = Recall *(2-1/P). F-Measure and Overall for various test cases is shown in table 3.

Table 3:

On real world cases #301-304, the SVM-Class model performs much better than the PRIOR+[21] as it uses linguistic feature with the help of web feature (synonyms) where as PRIOR+ does not use this feature. PRIOR+ proved to be better compared to LILY, ASMOV, FALCON-AO and RiMOM . Comparision of SSFPOA w.r.t F-Measure is shown in figure 5.

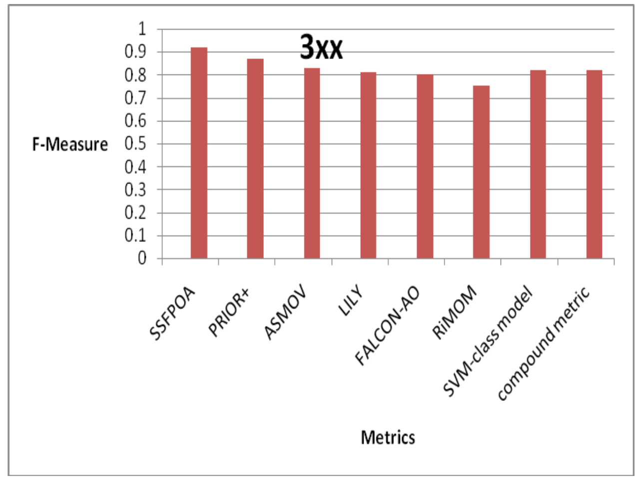


Figure5 : SFPOA vs [2,10,12,21,22,24,37]

LILY uses exact labeling and structural match but does not use synonymns nor wordnet. ASMOV uses labeling, structural match and instance match, aggregates all features into a single value, instead it could aggregate at each stage to improve efficiency. For linguistic similarity PRIOR+ takes only edit distance. Similarly structural similarity is performed only for the classes with same number of subclasses. Problem with RiMOM is it makes linguistic similarity based on wordnet and data type, but does not use structural mapping. As mentioned #301, #303, #304 has deep structure. Falcon-AO: Even though it uses both linguistic and structural comparability, but uses only edit distance to calculate linguistic similarity. Also basing on first heuristic value (which depends on random number generator) the second rule may or may not be selected. So it produces less efficiency even when compared to LILY, ASMOV, PRIOR+, RiMOM. One point that was observed during evaluation is that, if shared elements of ontologies were identified and clustered initially (calling it as sub_class_cluster) and mappings were computed and stored for this sub_class_cluster, this result can be reused at all the sub trees which uses the sub_class_cluster. But this may lead to two problems:

(1) Human intervention is required for identifying and storing sub_class_clusters.

(2) One extra proof step is required for searching in this sub_class_clusters. Also total time taken will be increased with extra time "t" that is taken for searching in the sub_class_cluster.

IV Conclusions and future work

Earlier works on ontology alignment problem have concentrated on finding mappings between items of two or more ontologies with or without considering instances. In this paper we reduced ontology alignment problem as data mining problem and solved using a compound semantic measure (SSFPOA) consisting of 12 matchers and weighted sum of these matchers. Our nonClassification results when compared with existing approaches proved to be encouraging. Disadvantage of these classification methods is the need for a suitable training set. If the training set is not constructed carefully with appropriate size and data, the results are not acceptable. Our work has concentrated on simple mappings (1-1) only. Our future work concentrates on applying SSFPOA within the domains such as text mining (where author and co-author are semantically same), taxonomies related to biological categories such as gene synonyms where similar genes can probably be replaced. Another direction to extend our work is Mapping Management i.e if input ontology is updated then match library should be updated automatically.

V References

1. A. Doan, J. Madhaven, et al., "Learning to match ontologies on the semantic web", VLDB Journal 12 (4) 303–319,2003

2. B. Bagheri Hariri, H. Abolhassani, H. Sayyadi, "A Neural-Networks-Based Approach for Ontology Alignment",2006

3. Choi, I., Kim, M."Topic distillation using hierarchy concept tree",In: ACM SIGIR conference, pp. 371–372 2003

4. Chen, G., Wei, Q., and Kerre, E. E.,"Fuzzy Data Mining: Discovery of Fuzzy Generalized Association Rules. In Bordogna, G. and Pasi, G., editors, Recent Issues on Fuzzy Databases, pages 45–66. Physica-Verlag.2000

5. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder,"The chimaera ontology environment",In Proceedings of AAAI, pages 1123–1124, 2000.

6. Doan,AHP Domingos, A Halvy,"Reconciling Schemas of Disparate data sources: A Machine Learning approach", SIGMOD 2001

7. Eduardo L. G. Escovar, Cristiane A. Yaguinuma, Mauro Biajiz,"Using Fuzzy Ontologies to Extend Semantically Similar Data Mining", XXI Simpósio Brasileiro de Banco de Dados 2006

8. Guangfei Yang, Kaoru Shimada, Shingo Mabu and Kotaro Hirasawa,"A Personalized Association Rule Ranking Method Based on Semantic Similarity And Evolutionary Computation",2008

9. http://oaei.ontologymatching.org/2011/oriented/index.html

10 Hu, W., Zhao, Y. et al, "Falcon-AO: results for OAEI 2007",. In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea.

11. Hong, T.-P., Lin, K.-Y., and Wang, S.-L.,"Fuzzy data mining for interesting generalized association rules" Fuzzy Sets Systems, 138(2):255–269. (2003).

12 Jean-Mary Y., Kabuka, M," ASMOV Results for OAEI 2007", In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea

13 Jayant Madhavan, Philip Bernstein, and Erhard Rahm," Generic schema matching using Cupid", In Proceedings of the International Conference on Very Large Data Bases (VLDB), pages 48–58, 2001.

14.Leacock, C., Chodorow, M. "Combining local context with wordnet similarity for word sense identification. In: Fellbaum, C. (ed.)WordNet: A Lexical Reference System and its Application." MIT Press, Cambridge , 1998

15. Lin, D.,"An information-theoretic definition of similarity", ICML conference 1998

16. Laurent Brisson, Martine Collard,"How to Semantically Enhance a Data Mining Process?",ICEIS 2008, LNBIP 19, pp. 103–116, Springer-Verlag Berlin Heidelberg, 2009

17. Li,W.S.C.Clifton,"Semint: A tool for identifying attribute correspondences in heterogeneous databases using neural network", Data nad knowledge engineering 33:1,49-84,2000.

18. M. Ehrig, S. Staab, QOM—quick ontology mapping, in: Proceedings of International
Semantic Web Conference 2004, 2004, pp. 683–697.

19. Mitra P., Noy N. F., Jaiswal A. R,"OMEN: A Probabilistic Ontology Mapping Tool", In: Workshop on Meaning coordination and negotiation at the Third International Semantic Web Conference (ISWC-2004), Hiroshima, Japan, 2004

20. Marc Ehrig, Karlsruhe, Steffen Staab,"Boot strapping Ontology Alignment Methods with APFEL ", WWW 2005, May 10–14, 2005, Chiba, Japan. ACM 1-59593-051-5/05/0005. Pgno:1148-1149

21. Ming Mao, Yefei Peng, Michael Spring,"Ontology Mapping: As a Binary Classification Problem", John Wiley & Sons, 2010

22 Ming Mao, Yefei Peng," PRIOR+ Results for OAEI 2007", In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea.

23. N. Noy, M. Musen, "The PROMPT Suite: interactive tools for ontology merging and mapping, International Journal of Human–Computer Studies" 983–1024,2003

24. N.F.Noy, M/A.Musen, Workshop on Ontologies and Information sharing at the seventeenth international joint conference on artificial intelligence (IJCAI-2001), Seattle, WA, 2001

25. . Pavel Shvaiko Jerome Euzenat, " Tutorial on Schema and Ontology Matching", ESWC'05 – 29.05.2005 – p. 1/71.

26. . Peng Y, Munro PW, Mao M. "Learning to map ontologies with neural network", OM, CEUR Workshop Proceedings, vol. 5

27. Resnik, P."Using information content to evaluate semantic similarity in a taxonom", IJCAI conference, pp. 448–453 1995

28. Resnik, P,"Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language", Journal of Artificial Intelligence Research 11, 95–130 1999

29. S.Vasavi, S.Jayaprada, "Extracting semantically similar frequent patterns using ontologies", Springer LNCS 7077 978-3-642-27241-7 pages: 157—165 ,2011

30. S.Vasavi, S.Jayaprada :Clustering Semantically similar frequent patterns using SSFPOA", Accepted for publication in International journal of Data Ware housing & Mining, Vol2 Issue 1 pg 45 to 51 , 2012 ISSN: 2249 –7161

31. S. Melnik, H. GarciaMolina,et al., Similarity flooding: a versatile graph matching algorithm and its application to schema matching, in: Proc. 18th International Conference on Data Engineering (ICDE), 2002.

32. Venkat Gudivada, Vijay V.Raghavan, William I Grosky, Rajesh Kasanagottu, "Information retrieval on the world wide web", IEEE Internet Computing Sep 1997 1089-7801/97.

33. www.wikipedia.com

34. Wang, P., Xu, B," LILY: The Results for the Ontology Alignment Contest OAEI 2007",. In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea.

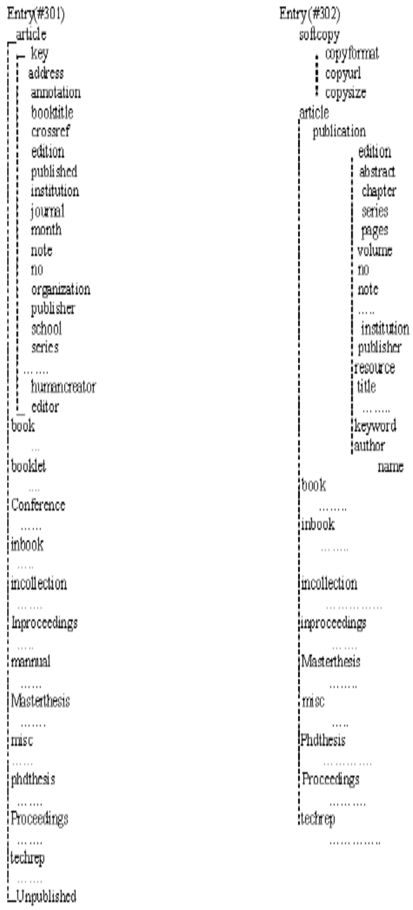
35. www.wordnet.princeton.edu .

36. Yannis Kalfoglou, Marco Schorlemmer, ",IF-Map: An Ontology-Mapping Method Based on Information-Flow Theory", Journal on Data Semantics, LNCS 2800, pp. 98–127, 2003.

37 Yi Li, Juanzi Li, Duo Zhang, and Jie Tang," RiMOM Results for OAEI 2007", In Proceedings of ISWC 2007 Ontology Matching Workshop. Busan, Korea.

38. Zhong, J., Zhu, H., Li, J., Yu, Y,"Conceptual graph matching for semantic search", ICCS conference, pp. 92–196 2002

Figure 1 : Part of ontology test case #301 and #302



S.no	Testcase#	Total Nodes	Institution	Total Classes	Shared Classes			Depth
					Total	Min	Max	
1	#301	575	Real: BibTeX/MIT	14	40	40	40	2
2	#302	315	Real: BibTeX/UMBC	11	25	25	25	4
3	#303	359	Real: Karlsruhe	50	17	2	16	5 (Virtual root added)
4	#304	375	Real: INRIA	16	33	2	16	5 (Virtual root added)

Table 1: Statistics of ontologies for testing SSFPOA.

S.No	Testcase#	Simple mappings	Complex Mappings	Disjoint Mappings	Wrong mappings	Total Mappings
1	#301 vs #302	180	50	228	200	181125
2	#301 vs #303	47	39	309	350	206425
3	#301 vs #304	156	57	201	250	215625
4	#302 vs #303	51	21	159	200	113085
5	#302 vs #304	48	21	201	245	118125
6	#303 vs #304	24	20	204	263	134625

Table 2: Statistics of output

	Bayesnet		SVM		Multilayer	
SNO	F-Measure	overall	F-Measure	overall	F-Measure	overall
301-302	0.843	0.67	.845	0.714	.85	.714
301-303	0.99	0.98	.974	0.95	.88	.74
301-304	0.81	0.58	0.854	.714	.85	.649
302-303	0.96	0.92	0.886	.773	.82	.657
302-304	0.96	0.92	0.881	.766	.92	.825
303-304	0.96	0.92	0.97	.941	.97	.902

Table 3 : F-measure,Overall for 3xx