Class structures and Lexical similarities of class names for ontology matching

Sumit Sen[1, 2], Suman Somavarapu[1], N.L Sarda[1]

[1] Deptt of Computer Science, IIT Bombay, Mumbai-76 India
[2]IfGI, University of Münster, Robert-Koch Str. 26, 48149 Münster, Germany
sumitsen@uni-muenster.de, {suman, nls}@cse.iitb.ac.in

Abstract. Semantic Interoperability is a major issue for National Spatial data Infrastructures (NSDIs) and mapping across heterogeneous databases is essential for such interoperability. Mapping of schemas based on ontology mapping provides opportunities for semantic translation of schemas elements and hence for database queries across heterogeneous sources. Such semantics based mappings are usually human centered processes. This paper demonstrates semi-automatic mapping using semantic similarity values from an electronic lexicon. Lexical similarity of class names and class structures constitute knowledge base for mapping between two schemas. We employ semantic mapping based on synonym similarity matches from WordNet. We use heuristics based propagation of similarities using attribute mapping and superclass-subclass relations. The machine based similarity values are seen to be comparable to human generated values of mapping.

Keywords: ontology, semantic mapping, lexical similarity, similarity propagation, heterogeneous databases,

1 Introduction

Spatial databases usually store information relating to different themes but also spatial information of the records. The spatial information, serves as the common geospatial domain for such databases serves as a central point of integrated usage of such data. Geographic Information Systems and more recently, Web Mapping Services (WMS) as promulgated by the Open Geospatial Consortium (OGC) [1], display geospatial data from such spatial databases. With increased possibilities of sharing of databases across domains and user groups based on frameworks such as geospatial web services and Spatial Data Infrastructures (SDIs), the need for resolving the semantic interoperability of data has been identified as a major requirement. National Spatial Data Infrastructures (NSDIs) can be considered as a typical testbed for semantic interoperability experiments across heterogeneous database users

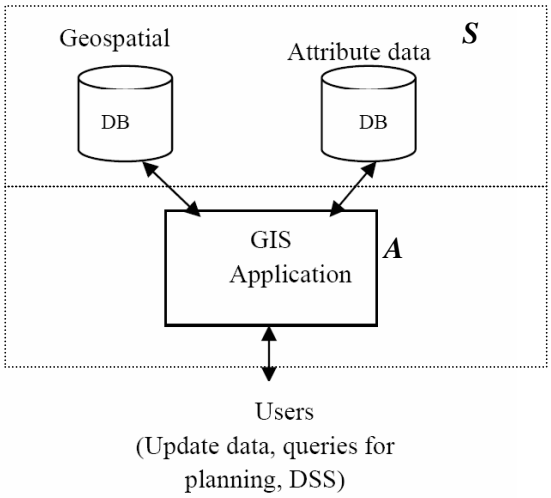


Figure 1 Geospatial data usage scenario in an NSDI. The two types of data sources include geospatial data sources and attribute data sources. The semantics of the data source region ***S*** need not be same as that of the application region ***A***

Semantic mapping across heterogeneous data sources is reported as a major requirement for National Spatial Data Infrastructures [2]. Such Infrastructures serve as a common interaction mechanism between multiple organizations which need to share geospatial data for their different applications. Figure 2 shows a typical scenario of data sharing in an NSDI with multiple (semantically heterogeneous) data sources being used. The traditional view of interoperability in an NSDI is based on mapping of information sources based on human based interaction and documentation. A strictly *Top-Down* approach advocating use of fixed class names can be seen as too rigid and impractical for actual use. On the other hand, schema mappings based on a bottom-up approach is difficult even if mappings can be achieved by organizations participating in the NSDI because

(1) Schemas are continuously evolving

(2) Human knowledge about semantics of the table names and attribute names are often not completely expressed in the names used. Therefore mapping should be seen as a probability based process.

(3) It is not necessary that mappings exists always. In a probability based model this situation is equated with zero values. On the other hand it is not possible or necessary to have values for every mapping. Such cases where the mapping is not done should be equated to null values of probability of mapping.

In addition to these observations about schema mappings of databases in an NSDI we also observe that organizations can join or leave the Infrastructure. Depending on this, new mappings need to be generated at times and older mappings need to evolve.

It is imperative that a semi-automatic process of mapping of databases need to evolve. Ontology based mapping has been increasingly viewed as an engineering solution to the problems. Based on specifications of the conceptualizations [3] as a more generic layer above the schema specifications, ontologies serve as an intermediate step to specify and resolve semantics of the contents of a database system. Ontology based mapping allows us to generate schema translation rules [4]. Two categories of semantics can be differentiated in regard to

(a) Classes or schema names and

(b) Individuals or instances of the classes.

While the later is by no means a trivial problem we state our approach based on semantics of the class or schema names. We aim to assist the generation of semantics based mapping for classes or schema names based on lexicon based similarity values. The approach is similar to the similarity flooding principle [5] but in our case, propagation of similarity values is somewhat restricted. It is based on heuristics such that class attributes and similarity values of superclasses and subclasses are reflected in the overall similarity values. The machine based values of similarity are compared to human generated values.

1.1 Paper outline

This section has provided the introduction and also explains the motivation of this work. Section 2 outlines the previous work in semantic mapping generation and describes the research problem at hand. Subsequently Section 3 describes the generation of lexical similarity values and their propagation based on attribute properties of classes and their superclass - subclass structures[1]. Finally in Section 4 we analyze the similarity values vis-à-vis human generated values. Some conclusions and areas for future work are identified in the end.

1.2 Motivation

The motivation of our research is derived from efforts to achieve schema translations from heterogeneous databases that participate in the NSDI. Since the objective of sharing of resources in the NSDI is to maximize the usage of data and applications, the requirement of allowing semantics based translations of queries and data is primary in nature. We restrict our problems based on logical steps as follows:

(i) To identify the translations (in the form of XQuery statements), which could be applied to interface semantically heterogeneous systems in the NSDI
(ii) To generate such translations based on mappings between the ontologies of the two systems
(iii) To semi-automate the process of mapping between the ontologies

[1] The term *Class Structure* in this paper refers to three different constituents - the attributes of the class, its superclasses and subclasses.

The last step is rather the focus of this paper. Such Mapping between ontologies is dependent on both the explicit semantics of the class names or attribute names and also the implicit semantics of subclasses and superclass relations. When we consider the objective of translations it is important to have a directional mapping such that all members of the target schema mapped to the source schema as shown in figure 2.

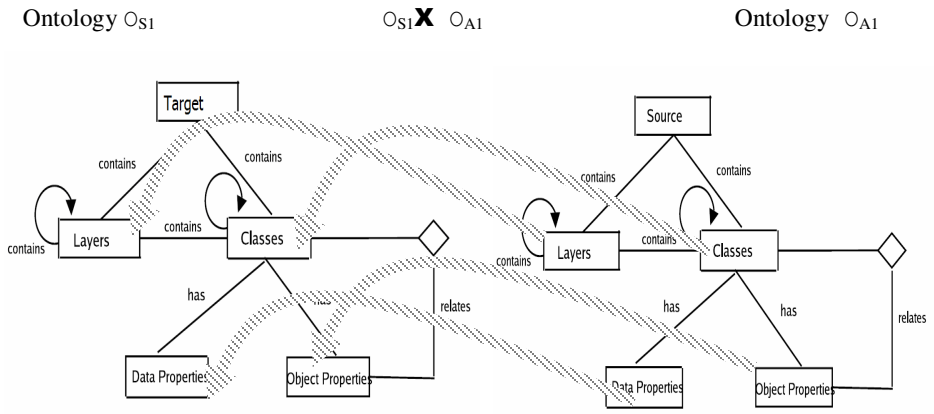


Figure 2 Ontology mapping between Target and Source. The different components of the Source ontology including layers, classes and their properties are mapped to each other. Layers can be considered as a group of classes. Classes can have inherited classes and so can layers. The relation between of layers and classes is not that of inheritance but rather that of aggregation.

2 Generating Semantic mapping

Semantic Mapping can be considered as process which generates rules for transformations between different data sources which do not necessarily have the same semantics for the same schema symbols. Schema symbols[2], for our case consists of layer names, class names and property names. We also need to be clear that having different semantics for the same schema symbols also entails that sometimes

1. Same symbols could have different meanings
2. Different symbols could have the same meaning
3. Some symbols in the first schema may not have corresponding symbols with the same meanings in second schema
4. Some symbols in the first schema could correspond to more than one symbol in the second schema such that the meaning is conveyed by simple aggregation (or further complex functions of aggregation) of the multiple symbols in the second schema

[2] We refer to schema elements as schema symbols to stress that these symbols have certain meaning and conceptualizations.

5. Some symbols in the first schema could correspond to part of a symbol in the second schema such that the meaning can be extracted fully from that corresponding symbol.
6. Also some symbols in the first schema could correspond to multiple symbols in the second schema but combining aspect 4 and 5 above.

Besides these we know that datatype heterogeneities (different datatype for the same schema component in different databases) are closely associated to the above contexts but we shall assume their absence for our case.

As discussed in the introduction, we use ontology based mapping to achieve schema translations. Now consider the situation described in figure 1 with attribute data source (***S***) and application (***A***). Here we have ontologies with elements corresponding to the different schema symbols – layers, classes, attributes as shown in figure 2.

We assume existing ontologies (O_{S1}, $O_{S2}\ldots O_{SM}$) of the data sources and the applications (O_{A1}, $O_{A2}\ldots O_{AN}$). The aim of establishing semantic interoperability is now reduced to provide mapping ($O_{S1} \times O_{A1}\ldots O_{SM} \times O_{SN}$). This higher level mapping is different from the XQuery-like physical level specification of mapping between schemas because it avoids datatype and other implementation constraints. The challenge here is to use an ontology of the database schemas and build up explicit mapping. Given two ontologies O_{S1} and O_{A1} (see figure 2) a mapping $O_{S1} \times O_{A1}$ is a set of pairs (s,a) where s and a are concept contained in O_{S1} and O_{A1} respectively. The mapping is complete and one-to-many. Any concept s maps to every concept in O_{A1w} but with different intensities which is dependent on how similar it is to the target concept. When such similarities are taken into consideration while determining the matching we can assume the highest mapping value as 1 and lowest as 0. Thus a mapping is defined as a matrix of similarity values as below

$$M[O_S \times O_A] = \{m_{S1A1}, m_{S1A2},.. m_{S1An} \quad m_{S2A1}, m_{S2A2,\ldots} m_{S2An} \quad \ldots\ldots \quad m_{Sm,A1}, m_{SmA2,\ldots} m_{SmAn}\} \qquad (1)$$

$$\text{such that} \quad 0 \le m_{XY} \le 1$$

The values of semantic similarity are dependent on the notion of semantics which is employed. The similarity matrix can be used across ontologies if the notion of semantics is consistent.

We discuss the previous work in the area of computing similarities for schema matching in the next section. Thereafter we explain the theoretical basis of our research problem.

2.1 Previous work

Similarity based approach for schema mapping has been studied using different approaches. Shvaiko [6] has classified schema matching approaches and has discussed the heuristics based approaches both at structure and element level. The Similarity Flooding approach [5] as implemented in Rondo [8] utilizes a hybrid matching algorithm based on the ideas of similarity propagation. Schemas are presented as directed labeled graphs; the algorithm manipulates them in an iterative fix-point computation to produce mapping between the nodes of the input graphs. The technique starts from string-based comparison (common prefixes, suffixes tests) of the vertices' labels to obtain an initial mapping which is refined within the fix-point computation. The basic concept behind the SF algorithm is the similarity spreading from similar nodes to the adjacent neighbors through propagation coefficients. From iteration to iteration the spreading depth and a similarity measure are increasing till the fix-point is reached. The result of this step is a refined mapping which is further filtered to finalize the matching process.

Cupid [9] implements a hybrid matching algorithm comprising linguistic and structural schema matching techniques, and computes similarity coefficients with the assistance of a precompiled thesaurus. Input schemas are encoded as graphs. Nodes represent schema elements and are traversed in a combined bottom-up and top-down manner. Matching algorithm consists of three phases and operates only with tree-structures to which no-tree cases are reduced. The first phase (linguistic matching) computes linguistic similarity coefficients between schema element names (labels) based on morphological normalization, categorization, string-based techniques (common prefixes, suffixes tests) and a thesaurus look-up. The second phase (structural matching) computes structural similarity coefficients weighted by leaves which measure the similarity between contexts in which individual schema elements occur. The third phase (mapping generation) computes weighted similarity coefficients and generates final mappings by choosing pairs of schema elements with weighted similarity coefficients which are higher then a threshold. Both Rondo [8] and Cupid [9] are important to our approach because they allow propagation of semantic similarity which is important to integrate the explicit and implicit semantic matching definitions stated previously. For a complete survey of other schema matching approaches see [6] and [10].

Lexical matching in ontologies has also been studied in detail in Semantic integration approaches using ontologies. A survey by Noy [10] separates matching approaches based on

(i) shared upper ontologies based approaches and
(ii) heuristics based machine learning approaches

While both of the above are said to have advantages in different objective settings, the later is significant in the absence of a commitment to a shared upper ontology. The mappings in this case need to be stored as GAV or LAV similar to the approach in schema matching based on directional mappings [11] and with an overall objective of allowing query answering across heterogeneous data. The Heuristics based

approaches are reported to employ automatic or semi-automatic techniques by looking at

- concept names
- class hierarchies
- property definitions
- instance definitions
- class descriptions (as Description logic statements)

While instance based approaches such as GLUE [12] can be seen as helpful to understand the ontology commitment of the instances, the luxury of availability of time and access to the data instances cannot be assumed. Giunchiglia and Shvaiko [13] on the other hand use WordNet as a common source for grounding. Subsequently mappings such as generalizations, specializations, and disjointness are determined using a SAT prover.

2.2 The ontology mapping problem

An assessment of the problems of semantic interoperability in spatial data infrastructures can be seen in [14] Semantic mapping is reported to work at two levels- (1) explicit semantics of the schema elements and (2) implicit semantics resulting from schema structure including class hierarchies and attribute properties. We divide these based on the following definitions

Definition 1. A mapping M is defined to be reflective of *explicit semantics of the schema* elements if and only if every schema element that maps to another schema element, can substitute the later in the absence of any schema structure.

In a lexicon such substitution entails that one is a synonym of the other

Example 1: For a mapping M [A, B] = {1, 0, 0, 1} where A={road, intersection} B = {street, crossing} we can say that it reflects explicit semantics of A and B if one could substitute 'road' by 'street' and 'crossing' by 'intersection'. In WordNet [7] this condition would be true. Also if this criterion can be proved, the mapping can be termed as reflective of explicit semantics of the schema elements.

Definition 2. A mapping M is defined to be reflective of *implicit semantics resulting from super-class structures* if and only if every element that maps to another element in the structure, has similar super classes and attributes (Also the related super classes have the same criteria with respect to its own super-classes and attributes)

Example 2: For a mapping M[A, B] = {1, 0, 0, 1} where A and B have two elements each, let us assume one element of both A and B are sub classes of the other and represented in figure 3. Here only if the explicit similarity of attributes of element1 of A and element1 of B are higher M is reflective of implicit semantics of the super class structure. In this case the explicit similarity of attributes of Element 1 of A and Element 2 of B should be 0 and so also that of attributes of element 2 of A and element1 of B. In regard to the implicit semantics of super-class we can say that

since element 2 of both A and B have similar super-classes, their own similarity value is higher than the original implicit value of similarity and explicit similarity of the attributes combined.

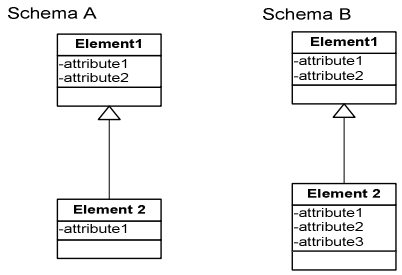


Figure 3 Implicit semantics of the super class structure

Definition 3. A mapping M is defined to be reflective of *implicit semantics resulting from sub-class structures* if and only if every element that maps to another element in the structure, has similar sub classes and attributes. Also the related sub classes have the same criteria with respect to its own sub-classes and attributes.

Example 3: For a mapping M[A, B] = {1, 0, 0, 1} where A and B have two elements each, let us assume one element of both A and B are sub classes of the other and represented in figure 4. The relation to similarity of attributes of Element1 and Element2 in both A and B is the same as explained in Example 2. In regard to the implicit semantics of sub-class we can say that since element 1 of both A and B have similar sub-classes, their own similarity value is higher than the original implicit value of similarity and explicit similarity of the attributes combined. (Note that here subclasses have same number of attributes although the significance of equal number of attributes cannot be considered as critical as is the case in Example 2)

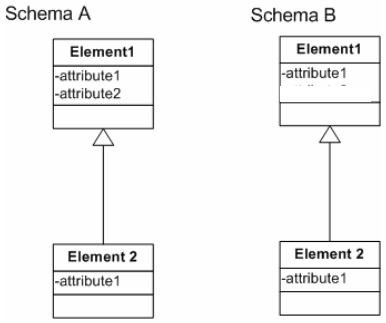


Figure 4 Implicit semantics of the sub class structure

Definition 4. A mapping M is defined to be reflective of *complete semantics resulting from both schema structure and semantics of elements* if and only if the mapping is reflective of implicit semantics of attributes, super-class and sub-class structures and explicit semantics of schema elements.

Let us be clear that definition 1 does not qualify as a syntactic match of the labels of the schema elements. The substitutability sense implied here involves semantics and implied meaning of the label. This may not be clear from the label name alone and usually requires a more verbose description. Secondly since definition 4 can be seen as a combination of the other three definitions, we define our problem stepwise: to obtain mappings which are reflective of

a. Explicit semantics of the schema elements
b. Implicit semantics of the super-class schema structure
c. Implicit semantics of the sub-class schema structure

3 Semantic mapping generation

We describe the approach of generating the semantic mapping as a three step process, namely (i) generating values of lexical similarity based on synonym relations (ii) propagating the similarity values for sub classes and similarly for superclasses (iii) combining the values of step (ii) to obtain the most similar classes and attributes (of the source ontology) for each class and attribute of the target ontology. We describe each step as below.

3.1 Generating Lexical similarity values

Definition 5 Lexical similarity S is a function defined between two element names x and y where

$S(x,y)= \beta$ (measure of the distance of the two words in a lexicon)

Such that $0 \leq S(x, y) \leq 1$

Remark 1 β is a weigthage function that we employ to sensitize our similarity function for optimality conditions. The measure of distance on the other hand is computed as the $(d)^{-4}$ **where** d is the number of nodes traversed in the graph of the lexicon (say WordNet). In case d is null or zero we assign a zero value to the measure of distance.

Lexical similarities are computed as binary values between two schemas components based on their corresponding entries in the lexicon. We assume a GAV approach by computing mappings for each target ontology. In the absence of a corresponding entry in the lexicon or in the case where there is no lexical relation we assume that d is null and zero respectively. Since there are two types of lexical relations in which we are interested (out of the 9 discussed by Evens and Smith [15]) we have lexical match algorithms for synonyms, hypernyms, and hyponym. For synonym relations the distance between two words is either 0 or 1 depending on their occurrence in a WordNet synset. For our case study the target ontology is that of Ordnance Survey UK [16] and source is OGC transportation schema (full version) [1].

We list lexical similarities of class names based on synonyms in column 3 of table 1 below.

3.2 Propagation of similarities of attributes and superclasses

If attributes of the target class have high similarity values with respect to certain attributes of the source class, such a mapping stands to be more attractive in comparison to any mapping where the attributes do not yield high similarity values. This is based on the definition of implicit semantics of superclass relations of definition 2 we can obtain a no penalty algorithm for computing the propagated similarity value as shown below.

For all attributes

Obtain lexical similarity matrix $M_a[O_S \times O_T]$ for all attributes

End For

For all classes **do**

Obtain lexical similarity matrix $M_c[O_S \times O_T]$ such that class c_{Tq} in O_T has similarity value cm_{SpTq} with respect to class c_{Sp} in source ontology O_S

For all attributes (a1, a2,…an) of c_{Tq} **do**

If $\frac{\alpha(cm_{SpTq}) + \beta(a_{n\,SpTq})}{\alpha+\beta} > (cm_{SpTq})$ **then**

$$(cm_{SpTq}) = \frac{\alpha(cm_{SpTq}) + \beta(a_{n\,SpTq})}{\alpha+\beta}$$

End If

End For

End For

Set {ParentClassBasket} = Null

While {ParentClassBasket} < O_T

For all Classes in O_T such that Parent Class p_{Tq} is in {ParentClassBasket}

If $\frac{\varphi(cm_{SpTq}) + \phi(pm_{SpTq})}{\varphi+\phi} > (cm_{SpTq})$ **then**

$$(cm_{SpTq}) = \frac{\varphi(cm_{SpTq}) + \phi(pm_{SpTq})}{\varphi+\phi}$$

End If

Include c_{Tq} as member of {ParentClassBasket}

End For

Figure 5 Algorithm for Propagation of similarity values of attributes and superclasses. $\alpha, \beta, \varphi, \phi$ represent weightages of propagation

In short this algorithm allows an increase of the similarity values if the combined value of similarity based on attribute similarity and thereafter, the superclass similarity has increased. The use of such weightages clearly shows the use of heuristics based measures. Table 1 below shows some values of improved similarity values using the propagation described above.

Target Class (c_{Tq})	**Source Class** (c_{Sp})	**Lexical Similarity S(x,y)**	**Propagated Similarity** (cm_{SpTq})
OS:RoadRouteInformation	OGC:RailRoadRoute	0,6666667	0,766666667
OS:InformationPoint	OGC:TransportationPoint	0,6052632	0,723684211
OS:InformationPoint	OGC:TransportationPoint	0,6052632	0,723684211
OS:RoadPartiaRouteInformation	OGC:RailRoadRoute	0,5714286	0,7
OS:road	OGC:RailRoadPoint	0,5	0,65
OS:road	OGC:RailRoadSegment	0,5	0,65
OS:road	OGC:RailRoadSwitch	0,5	0,65
OS:InformationPoint	OGC:TransportationPath	0,4166667	0,591666667
OS:InformationPoint	OGC:TransportationPath	0,4166667	0,591666667
OS:roadInformationMember	OGC:TransportationSegment	0,4047619	0,583333333
OS:roadLink	OGC:RailRoadStation	0,4	0,58
OS:roadLink	OGC:RailRoadPoint	0,4	0,58
OS:roadLink	OGC:RailRoadSegment	0,4	0,58
OS:roadLink	OGC:RailRoadRoute	0,4	0,58
OS:roadNode	OGC:RailRoadStation	0,4	0,58
OS:roadNode	OGC:RailRoadSegment	0,4	0,58
OS:roadNode	OGC:RailRoadRoute	0,4	0,58
OS:roadNode	OGC:RailRoadBridge	0,4	0,58

Table 1: Top class matches based on propagated values of similarity of supper classes and attributes

3.3 Propagation of similarities of attributes and subclasses

The propagation in this case is similar but uses subclass similarity values instead of the superclass similarity values. Results of the propagation are shown in the table below.

Target Class (c_{Tq})	**Source Class** (c_{Sp})	**Lexical Similarity S(x,y)**	**Propagated Similarity** (cm_{SpTq})
OS:RoadRouteInformation	OGC:RailRoadRoute	0,6666667	0,766666667
OS:InformationPoint	OGC:TransportationPoint	0,6052632	0,723684211

OS:RoadPartiaRouteInformation	OGC:RailRoadRoute	0,5714286	0,7
OS:road	OGC:RailRoadPoint	0,5	0,65
OS:road	OGC:RailRoadSegment	0,5	0,65
OS:road	OGC:RailRoadSwitch	0,5	0,65
OS:road	OGC:RailRoadStation	0,5	0,55000001
OS:road	OGC:RailRoadRoute	0,5	0,55000001
OS:road	OGC:RailRoadSignal	0,5	0,53
OS:road	OGC:RailRoadBridge	0,5	0,5
OS:InformationPoint	OGC:TransportationPath	0,4166667	0,591666667
OS:roadInformationMember	OGC:TransportationSegment	0,4047619	0,583333333
OS:roadLink	OGC:RailRoadStation	0,4	0,58
OS:roadLink	OGC:RailRoadPoint	0,4	0,58
OS:roadLink	OGC:RailRoadSegment	0,4	0,58

Table 2 Top class matches based on propagated values of similarity of subclasses and attributes

3.4 Most similar mappings

Generation of most similar mappings is based on a simple combination of the values generated from 3.2 and 3.3. We use weightages (50:50 and 70:30) to obtain two sets of most similar mappings. The basic lexical similarity values of both these mappings and also the attribute similarity propagation is same. The results are shown in the tables below.

Target Class	Source Class	Overall Similarity
(c_{Tq})	(c_{Sp})	(cm_{SpTq})
OS:InformationPoint	OGC:TransportationPath	0,591666667
OS:InformationPoint	OGC:TransportationPoint	0,723684211
OS:road	OGC:RailRoadPoint	0,65
OS:road	OGC:RailRoadSegment	0,65
OS:road	OGC:RailRoadSwitch	0,65
OS:roadInformationMember	OGC:TransportationSegment	0,583333333
OS:roadLink	OGC:RailRoadPoint	0,58
OS:roadLink	OGC:RailRoadSegment	0,58
OS:roadLink	OGC:RailRoadStation	0,58

Table 3 Top class matches based on overall similarity

4 Analysis of machine generated similarity values

Since the objective of generating similarity values is to assist in human based mapping and semi-automate the process of transformations, we need to analyze the generated values vis-à-vis human generated values of similarity in the absence of any assisting tool. The purpose here is to get an overview of how good the generated values are and also the presence of errors (which we shall group as false positives and false negatives)

4.1 Human generated similarity values

The human generated similarity values were obtained by a small experiment. A blank similarity matrix sheet, class-attribute list and the class diagrams of the ontologies ***A*** and ***T*** (Appendix) were made available to the subject. Three steps were followed

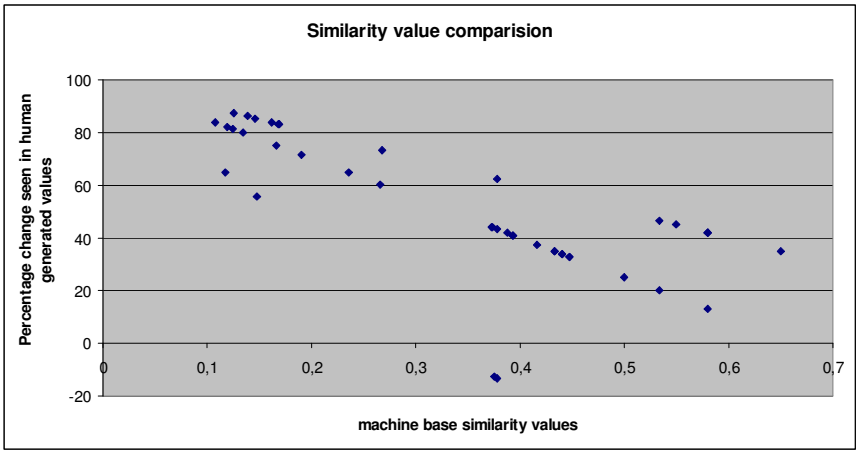
(i) A score of similarity (binary value) was recorded for every class name of the target with respect to each class name of the target based on English meaning of the words.

(ii) Two scores of similarity (binary values) were recorded for every class name of the target with respect to each class name of the target based on its position in the class structure. The first score is reflective of the subclass occurring in the class structure. Thus a class in the Target with same number of child classes and attributes as another class in the Source will have a higher score. The Second score is reflective of the superclass and hence if the target ontology superclass contains same number of attributes as the source ontology, it results in a higher score.

(iii) The three scores which are recorded in the similarity matrix sheet are combined to obtain the most similar class and attributes. The basis of combination is not fixed but left to the judgment of the human so that if he/she feels that the English meaning of the word is more important for matching, the values of subclass structure and superclass structure can be ignore. By default an average of the three is taken.

4.2 Performance parameters

We can now compare the performance of our machine generated similarity values. Graph 1 shows the difference in similarity values expressed as percentages. It should be remembered that the granularity of the human generated values is lower. Therefore it is more important to decide upon thresholds for the machine generated values in order to compare the two. Table 4, on the other hand, summarizes the top 10 class matches obtained from the human based similarity values. The numbers in red are machine generated values lower than the threshold limits discussed.



Graph 1 Percentage difference of human and machine based similarity values. We can see that there is higher percentage change among lower values of machine based similarity

False Positives: False Positives can be identified from the faulty values of the machine generated values. In our case this was 12.3% at t threshold of 0.50 and 36.9% at a threshold of 0.40. False positives were mainly seen in the cases where parts of the target class name existed as a part of the source class name.

False Negatives: Table 4 below shows the top ten class matches. The lower three cases have low machine generated values which indicate that such matches would not quality for mapping between the schemas. Overall Percentage of False negatives has been observed to be 4% at a threshold of 0.30 although the occurrence is higher(25%) in the top 20 class matches based on human generated similarity values.

Target Class	Source Class	Machine Similarity
(c_{Tq})	(c_{Sp})	(cm_{SpTq})
OS:roadLink	RailRoadRoute	0,58
OS:roadLink	RailRoadSegment	0,58
OS:roadMember	RailRoadStation	0,58
OS:road	RailRoadRoute	0,55
OS:roadMember	RoadLinearFeatureEvent	0,533333
OS:roadLink	TransportationPath	0,377778
OS:ferryTerminal	TransferCluster	0,267436
OS:roadLink	LinearFeatureEvent	0,169114
OS:ferryTerminal	RailRoadRoute	0,168297

Table 4 Top ten matches based on human generated similarity values

5 Conclusions and Future work

We have seen that lexical similarities of schema element labels and descriptions can help in ontology mapping. Along with similarity propagation based on heuristics allows integration of implicit semantics of the ontology structure and hence improves the mapping process. The propagation of similarity is directional in nature as opposed previous approaches [5,8,9]. However the experiments have also shown that there are problems with machine based similarity assessment.

(i) The semantic similarity of individual words does not always provide a good indicator of the semantic similarity of group words. Since class descriptions were used for similarity assessment this led to false positives in many cases.

(ii) Similarly although limited word senses were evaluated based on part of speech, word sense disambiguation would help to reduce number of false negatives. Such cases explain the occurrence of high percentage change of human generated similarity values among lower values machine generated values

It is also important to note that use of heuristics and threshold values is critical in order to use the semi-automatic mapping approach.

This is only the initial results from our efforts to allow transformations based on a semi-automated approach as discussed in the motivation. The whole exercise of ontology mapping can be seen in the context of ontology aware database management systems [17] and query answering across databases. Comparison of human generated values helps to see the utility of the approach. The main aspect of error prone and non-standard techniques followed in human based matching has not been out forth in this paper and is beyond the scope of this paper. We can assume that machine generated values provide an advantage. Future work in this area, therefore, has to involve a comparison of performance in human based mapping with and without the assistance of machine based values.

Acknowledgments

The work presented in this paper is supported by the NRDMS, Dept of Science and Technology, Government of India. We are also thankful to Ordnance Survey, UK for their help in this project. We are grateful to other members of the team at CSE and CSRE, IIT Bombay for their help and discussions.

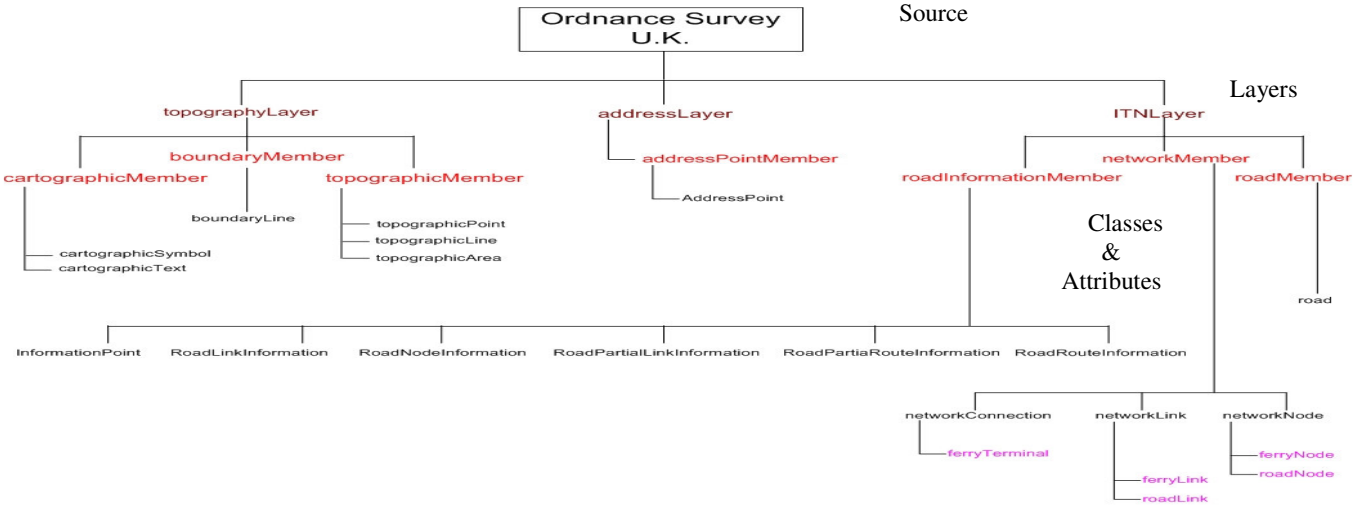
References

1. OGC.: Geography Markup Language (GML) Implementation Specification, Version 3.0 2003, available at http://www.opengeospatial.org/docs/02-023r4.pdf (last visited 22.04.2006)
2. Lutz, M & Klien, E.: Ontology-Based Retrieval of Geographic Information, International Journal of Geographical Information Science, 20(3): 233-260. 2006

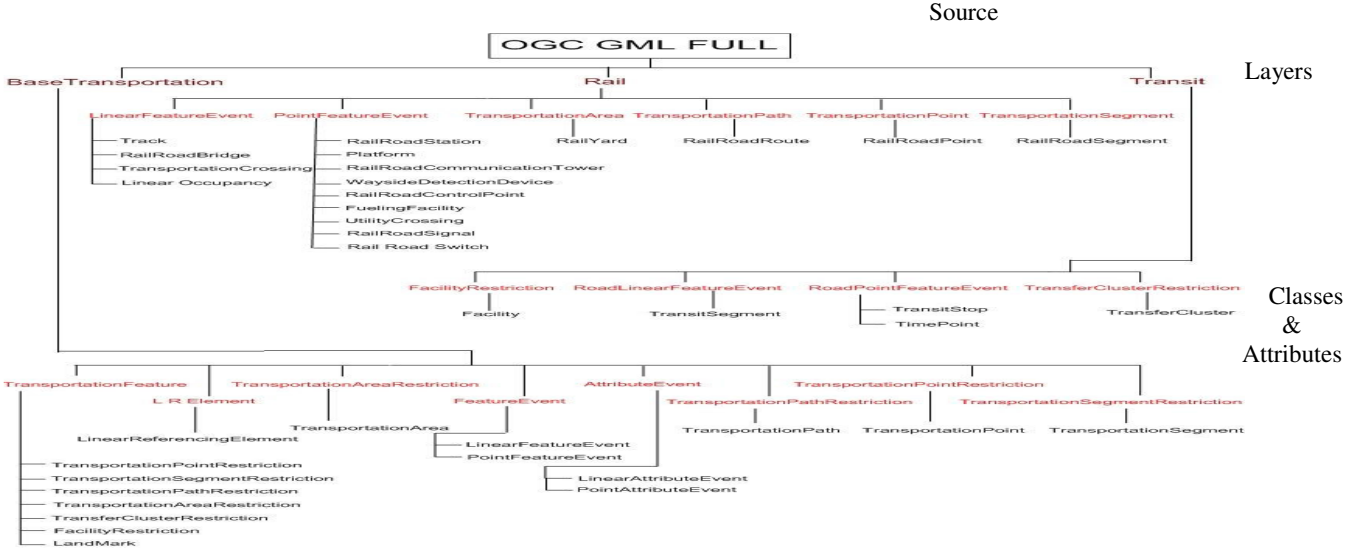
3. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Kluwer Academic Publishers (Formal Ontology in Conceptual Analysis and Knowledge Representation). 1993
4. Sen, S., Somavarapu, S. and Sarda, N.L.: Resolving Semantic Heterogeneity in the Indian NSDI: An Ontology Mapping Approach In Proc of MapIndia Conference New Delhi 2006
5. Melnik, S., Garcia-Molina, H. and Rahm, E.: Similarity flooding: A versatile graph matching algorithm. In Proceedings of the International Conference on Data Engineering (ICDE),pages 117–128, 2002
6. Shvaiko, P.: A Classification of Schema-based Matching Approaches. In Proceedings of Meaning Coordination and Negotiation Workshop at ISWC'04.
7. Fellbaum, C.: *(Ed)* WordNet - An Electronic Lexical Database. The MIT Press, 1999
8. Melnik,S., Rahm, E. and Bernstein, P. A.: Rondo: A Programming Platform for Model Management, In Proc. ACM SIGMOD 2003, San Diego, June 2003
9. Madhavan, J., Bernstein, P.A. and Rahm, E.: Generic Schema Matching Using Cupid Proc. VLDB 2001. (PDF, 140KB) Extended version: MSR-TR-2001-58.
10. Noy, N.F.: Semantic Integration: A Survey Of Ontology-Based Approaches SIGMOD Record, Special Issue on Semantic Integration, 33 (4), December, 2004
11. Halevy, A. Y. Ives, G. I., Mork, P., Tatarinov, I.:Data Management Infrastructure for Semantic Web Applications. IEEE Transactions on Knowledge and Data Engineering, Vol 16. No 7 pp 787-798 July 2004
12. Doan, A., Madhavan, J., Domingos, P. and Halevy, A.: Learning to map between ontologies on the semantic web. In The Eleventh International WWW Conference, Hawaii, US, 2002
13. Giunchiglia, F. and Shvaiko, P.: Semantic Matching. In The Knowledge Engineering Review Journal, vol. 18(3), pp. 265-280, 2003.
14. Bishr, Y.: Semantic aspects of interoperable GIS. Ph.D. Dissertation, International Institute for Aerospace Survey and Earth Sciences, Enschede, The Netherlands, ITC Publication No. 56, 154pp. 1997
15. Evens, M and Smith, R.: Properties of Lexical Semantic Relations, The Finite String, No. 4, 1978.
16. Ordnance Survey.: Ordnance Survey *OS MasterMap Integrated Transport Network (ITN) Layer* available at http://www.ordnancesurvey.co.uk/oswebsite/products/osmastermap/itn/ (last visited 22.04.2006)
17. Sarda, N.L.: Ontology-aware database management systems, Proceedings of IRMA International conference, Philadelphia. 2003

Appendix

T


Source
Ordnance Survey U.K.
Layers
topographyLayer
addressLayer
ITNLayer
boundaryMember
cartographicMember
topographicMember
addressPointMember
networkMember
roadInformationMember
roadMember
AddressPoint
boundaryLine
topographicPoint
topographicLine
topographicArea
cartographicSymbol
cartographicText
Classes & Attributes
road
InformationPoint
RoadLinkInformation
RoadNodeInformation
RoadPartialLinkInformation
RoadPartiaRouteInformation
RoadRouteInformation
networkConnection
networkLink
networkNode
ferryTerminal
ferryLink
roadLink
ferryNode
roadNode

A


Source
OGC GML FULL
Layers
BaseTransportation
Rail
Transit
LinearFeatureEvent
PointFeatureEvent
TransportationArea
TransportationPath
TransportationPoint
TransportationSegment
Track
RailRoadBridge
TransportationCrossing
Linear Occupancy
RailRoadStation
Platform
RailRoadCommunicationTower
WaysideDetectionDevice
RailRoadControlPoint
FuelingFacility
UtilityCrossing
RailRoadSignal
Rail Road Switch
RailYard
RailRoadRoute
RailRoadPoint
RailRoadSegment
FacilityRestriction
RoadLinearFeatureEvent
RoadPointFeatureEvent
TransferClusterRestriction
Facility
TransitSegment
TransitStop
TimePoint
TransferCluster
Classes & Attributes
TransportationFeature
TransportationAreaRestriction
AttributeEvent
TransportationPointRestriction
L R Element
FeatureEvent
TransportationPathRestriction
TransportationSegmentRestriction
TransportationArea
LinearReferencingElement
LinearFeatureEvent
PointFeatureEvent
TransportationPath
TransportationPoint
TransportationSegment
TransportationPointRestriction
TransportationSegmentRestriction
TransportationPathRestriction
TransportationAreaRestriction
TransferClusterRestriction
FacilityRestriction
LandMark
LinearAttributeEvent
PointAttributeEvent