3rd Indian International Conference on Artificial Intelligence (IICAI-07)

Hierarchical Course Knowledge Representation Using Ontologies

Javed Khan and Manas Hardas

Computer Science Department,
Kent State University,
Kent,OH 44240, USA.
{javed,mhardas}@kent.edu

Abstract. In this paper we present a method to represent knowledge associated with a course. Course knowledge can be represented in the form of hierarchical prerequisite relation based weighted ontology. We propose a schema using the Web Ontology Language (OWL) to represent course ontologies in a standard and sharable way. A novel approach for selectively processing relevant parts of the ontology is given. Design and analytical information extraction from educational resources is possible using this approach. The applicability of this method is not limited to the domain of education and can be extended to any domain in which knowledge can be represented as a structured hierarchy.

Key words: course ontology, knowledge representation, courseware, educational resources

1 Introduction

The web has greatly facilitated online sharing of course material. There have been many organized attempts to create large digital courseware libraries to promote sharing like NIST's Materials Digital Library Pathway, NSDL Digital Libraries, OhioLink, ACM Professional Development Center etc. MITs Open Course Ware (OCW) project has more than 1000 course materials freely available, Universia maintains translated versions of OCW courses in 11 languages, China Open Resources for Education (CORE) has a goal to include Chinese versions of the OCW. The amount of digital courseware content available on the web is huge. Surprisingly, the real sharing of the materials among the educators is still very low. In OCW it has been noted that only 16%of the users are educators out of which not more than 26%use it for planning their course or teach a class [1]. Most courseware today, on the web or otherwise is not accompanied with a conceptual design. There is no well formed encoding principle for capturing and sharing the schema associated with course materials. To make this digital content reusable, the associated meta data should be consistently represented. Traditionally, concept maps or knowledge maps have been used to represent the concept space for the course knowledge [2]. Ontologies provide a means to effectively map this knowledge into concept hierarchies. Standardization of semantic representation

Bhanu Prasad (editor): IICAI-07, pp. 1684-1698, 2007

standards like RDF [3] and OWL [4] offers great technical platform to represent the ontologies and greatly improve its machine usability. In this paper we present an approach to course knowledge representation using ontology in an expressible and computable format using *has-prerequisite* relationships where concepts involved in teaching a course are arranged in an hierarchical order of learning. Another original approach for specifically pointing out areas in ontologies of maximum relevance called as *CSG extraction* is given. Finally we try to observe some of the results in experimentation and make some interesting inferences about the clustering of knowledge associated with educational resources.

2 Course Knowledge Representation

Any design and evaluation system, like cognition based models in humans, needs back end knowledge base. A great deal of research has been done to make the corpora of knowledge available for machines. Knowledge representation techniques like semantic networks and ontologies make this possible. The corpus of course knowledge can be hypothetically divided into two tiered description framework namely, *concept space* and *resource space*. The course ontology is the graphical abstraction of the concept space, where in concepts are linked to each other using semantic relations. The resource space gives the description of actual resources for the corresponding concepts from the concept space. In this section we discuss the definition, specification, and constructs for course ontologies.

2.1 Granularity of Representation

It is used in AI, cognitive science, and other fields for problem solving, logical reasoning, data mining, question-answering, theorem proving, neural networks, expert systems etc. Davis et.al. define knowledge representation as a "set of ontological commitments" and "a medium of pragmatically efficient computation" [7]. It is important for the knowledge representation to be expressible and computable. This in turn brings us to the problem of granularity of information in course ontology. The granularity of the ontology is an important factor to consider while building the course ontology. The ontology can range from being fine grained to coarse grained. A finer grained ontology will contain more concepts in detail and more implicit relationships between concepts are also represented. Finer the ontology, the application will have more knowledge to work with giving better results. But defining a finely grained expressive ontology is costly in terms of computation. On the other hand, although coarse grained ontologies are computable, they do not have enough information. The depth of the knowledge to be represented is therefore an important question in representing any kind of knowledge. Most available finished materials today are coarse granular. Unfortunately, this is not suitable for machine processing.

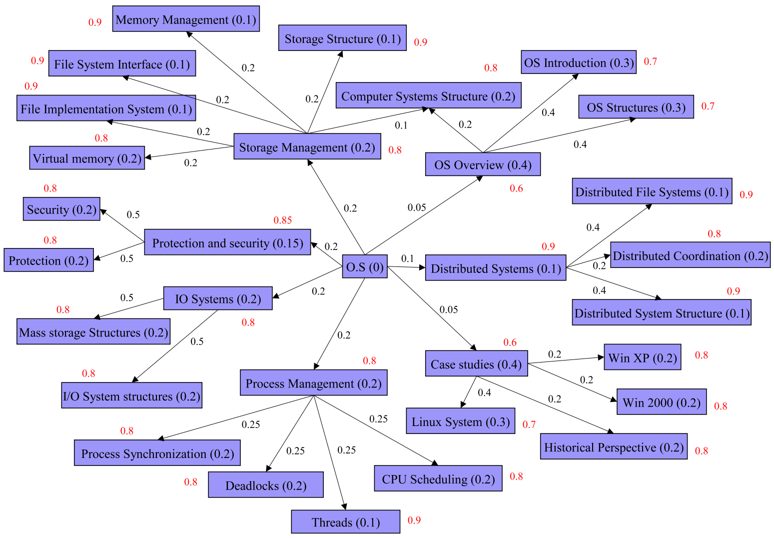


Fig. 1. Partial view of "Operating Systems" course ontology 2 levels deep

2.2 Course Ontology

In computer science, ontology is generally defined as "a specification of a conceptualization" [6]. Ontology is a data model that represents a domain and is used to reason about the objects in the domain and the relations between them. In the context of this research the domain is that of a *course*, the objects are *concepts* in the course and the relations between the concepts are that of *has-prerequisite*. Ontologies are increasingly being used to represent information in various domains like biological sciences, accounting and banking, intelligence and military information, geographical systems, language based corpus, cognitive sciences, common sense systems etc.

Relationships are the way the concepts in the ontology are structured with respect to each other. However, in the context of course ontology, the "part-of" semantics refers to the prerequisite understanding of the child node needed to understand the parent node. On the whole the course ontology is constructed in such a hierarchical fashion that the children of node represent the knowledge required to understand the parent node, and their children represent the knowledge required to understand them, so on and so forth. The ontology is created using the principle of "constructivism" borrowed from learning theory. The theory states that any new learning occurs in the context of and on the basis of already acquired knowledge. We use this theory to practically implement the has-prerequisite relationship based course ontology. Refer Figure 1. *Process Management* is the prerequisite of *OS*. However from the ontology it is obvious that *Storage Management-has prerequisite-Memory Management* is not the same as say, *Case Studies- has prerequisite-Linux System*. Nonetheless the relationship is designed to represent the necessary understanding of a child concept in understanding the parent concept. The semantics of *has-prerequisite* relation can be further expanded to include different types of relations but is not a part of this paper.

A node is characterized by two values namely, *self weight* and *prerequisite weight*. The self weight of a concept node is the value or the knowledge which is inherent to that node itself. It means that, the self-weight is the amount of knowledge required to understand the concept. To understand the concept entirely however, knowledge of the prerequisite concepts is also required, which is given by the prerequisite weight of the node. It gives the numerical realization of the importance of the understanding of the prerequisite concepts in the complete understanding of a parent concept. Another value which characterizes the course ontology is the *link weight*. The link weight is the numerical value for the semantic importance of child concept to the parent concept. Child concepts imperative in the understanding of parent concepts will have a greater link weights than the others. Thus the course ontology representation is a collection of concepts nodes with self weights and prerequisite weights and has-prerequisite relationships linking these nodes with a value attribute given by the link weight.

2.3 Concept Mapping

Most of the educational resources today are not accompanied with metadata which makes it very difficult for machine processing. For educational resources to be machine processable, they have to be presented in the proper context [8]. The mapping between the resource space and the concept space is called as the *concept mapping*. All educational resources are based on a few selected concepts from the ontology. When an educator designs courseware, she has a mental map of the concepts taught in the course. We define a rudimentary version of this mental map in the form of the course ontology. The research problem of automatically mapping a resource to concepts from ontology is an extremely non-trivial problem which is addressed extensively in natural language processing, knowledge representation, etc research. We limit our research to using the concept mapping idea.

2.4 Course Ontology Description Schema

The schema for the course ontology is mostly written in OWL Lite. OWL Lite supports basic classification hierarchy and simple constraint features. The schema is shown in the Appendix A. The elements of schema are header, class, property definitions and individuals. The language is designed to harness maximum computability at the cost of reduced expressive power.

The ontology header `owl:Ontology` is a collection of assertions about the course ontology. This section can contain comments, version information and imports for inclusion of other ontologies. Versioning can effectively be done to different levels of granularity of the ontology. All the individuals in the OWL representation are the instantiations of the class `Concept`. The object of the `sub class` axiom is a property restriction on `hasPrerequisiteWeight`, which describes an anonymous class, all of whose instances satisfy the restriction. The property restriction states that for all instances of class `Concept`, if they have a prerequisite then it must belong to extension of `Relation`. The extension of class means the set of all the members of that class. The class `Relation` is used to define all relations between concepts and give values to the `hasLinkWeight` property of the relation. It links two individuals of the class `Concept` with a data value. We first link instance of the class `Concept` to an instance of `Relation`, and then link that instance again to instance of `Concept`. The object property `connectsTo` is used to link instance of `Relation` to instance of `Concept`. Link weight is a characteristic of a relation therefore `hasLinkWeight` data type property applies to instances of class `Relation`. The range of the property is set by the resource `xsd:float`. For the purpose of computational convenience we set the values for all the concept and link properties between 0 and 1. The other two data type properties are `hasSelfWeight` and `hasPrerequisiteWeight` which are used to assign the self weight and prerequisite weights of a node respectively.

Individuals are facts about their class membership and their property values. In the example the concept instance `MemoryManagement` is a prerequisite for `OS`. Individual member `OS` is a member of class `Concept` and has the property values for `hasLinkWeight` as 0.2, `hasSelfWeight` as 0.39 and `hasPrerequisiteWeight` as 0.61. The most important part of the course ontology structure is the semantics between parent and child concepts. The tool which uses CODL defined course ontology should be able to infer that, since `connectsTo` links `relation1` and `MemoryManagement` and `hasPrerequisite` links `OS` to `relation1`, `MemoryManagement` is prerequisite of `OS`.

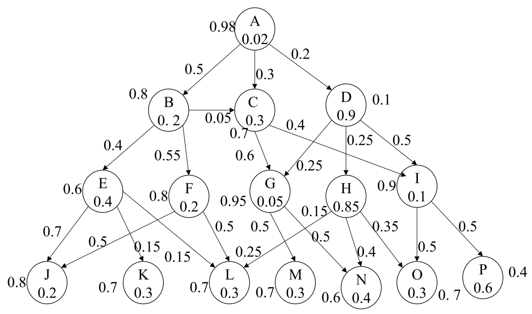


Fig. 2. Example of Concept Space Graph, T(A)

3 Symbolic Representation

The course ontology is mathematically defined in the form of a concept space graph (CSG). A CSG is a view of the concepts space distribution in the domain of a particular course.

A concept space graph $T(C, L)$ is a projection of the domain knowledge with vertices C and links L where each vertex represents a concept and each link with weight $l(i, j)$ represents the semantics that concept c_j is a prerequisite for learning c_i, where $(c_i, c_j) \in C$ and the relative importance of learning c_j for learning c_i is given by the weight. Each vertex i in T is further labeled with self-weight value $W_s(i)$ and cumulative prerequisite set weight $W_p(i)$.

A CSG with root A is represented as T (A) in Figure 2. For any node in the CSG, the sum of self weight and prerequisite weights and the sum of the link weights for all children is 1.

3.1 Prerequisite effect of a node

The notion of node path weight is introduced to compute the effect that a prerequisite node has on the understanding of a parent node through a specific path. A single node can have different prerequisite effect on a parent through different paths.

When two concepts x_0 and x_t are connected through a path consisting of nodes given by the set $[x_0, x_1, ..., x_t]$ then the node path weight between these two nodes is given by:

$$\eta(x_0, x_t) = W_s(x_t) \prod_{m=t}^{1} l(x_{m-1}, x_m) * W_p(x_{m-1}) \tag{1}$$

In the Figure 3, concept L is connected to B through E and F. Therefore the prerequisite effect it has on B is dependent on the prerequisite effect both E and F have on B respectively. From the node path weight calculations we can see that L has a stronger prerequisite effect on B through F rather than E. This is because, L is more important to F (0.5) than E (0.15), prerequisite importance of L is more to F (0.8) than E (0.6) and subsequently F (0.55) is more important to B than E (0.4). Thus node path weights takes into consideration not only the singular effect a node has on its immediate parent but also the combined prerequisite effect a node would have on a parent (B), along a specific path.

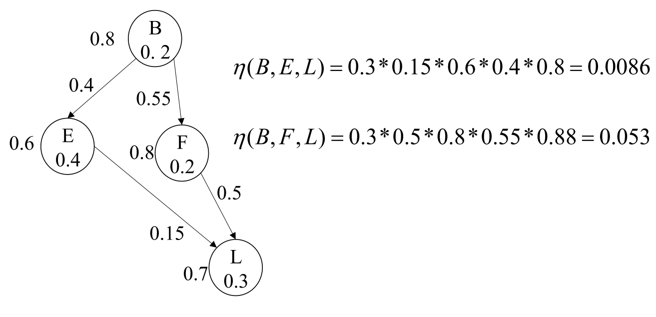


Fig. 3. Calculating prerequisite effect of a node along a path; *Node Path Weight*

3.2 CSG extraction

A generalized CSG can be vast and processing can be a gargantuan computation task. There needs to be a way to efficiently process the relevant information in these ontologies to give optimum results in minimum time and complexity of computation. Therefore we define a pruned sub-graph called as *projection graph* which cuts the computation based on a limit on propagated semantic significance. *The process of selecting projection graph nodes from the concept space graph is called as CSG extraction.* There are quite a few reasons to apply CSG extraction to ontology. It is computationally very expensive to work on big ontologies. Nowadays ontologies used range from thousands to millions of concepts. Therefore processing the whole ontology is very expensive and also doesnt logically make sense. The concepts which the question maps to are relatively very less as compared to the total number of concepts in the whole ontology. More over, say if the mapped concepts are very distant from each other in the ontology. This implies that the knowledge required to understand these concepts is very diverse in the concept space. Therefore it would be a squandering of computational resources to process the whole ontology instead of just the relevant portions. The concept space graph gives the layout of the course in the concept space with a view of course organization, involved concepts and the relations between the concepts. Examples of large CSGs include WordNet (150,000) an English language ontology, LinKBase (1 million in English, 3 million in other languages) a comprehensive medical/clinical ontology, Gene Ontology (now known as GO, over 19000 concepts) the genome mapping project and so on. Thus defining a workable area of ontology is of the utmost importance from the perspective of semantic relevance and computability and it is achieved by pruning the ontology by introducing a variable called as the threshold coefficient (λ).

Threshold coefficient (λ) By varying the threshold coefficient the size of the computable projection graph can be varied and thus the semantic significance. Since the projection graph is a subgraph of the concept space graph, it is necessary to have pre-requisite weights for the leaf nodes too, although most times the pre-requisite weight for the leaf nodes is zero. Flexibility for optional prerequisite weights for the leaf nodes allows the CSG to be extensible and easily extractable for the projection. Threshold coefficient is a kind of virtual limit by which the size of the projection can be controlled. Greater the coefficient more is the screening for the nodes to be added to the projection and thus smaller is the graph. Less coefficient value means more concepts will be included in the projection. In the context of education the threshold coefficient can be thought of as a parameter which can set the depth to which the topic has been taught. If a topic is not taught in detail, a greater coefficient is assigned so that the depth of the projection graph will be less. Conversely, if a topic is covered in great detail, the value assigned to the threshold coefficient is low, so that the projection graph for the concept is large, encompassing more prerequisite concepts. Threshold coefficient determines the limit to the quality of understanding of a particular concept.

Projection Graph *Given a CSG, $T(C, L)$, with parent concept x_0, and projection threshold coefficient λ, a projection graph $P(x_0, \lambda)$ is defined as a sub graph of T with root x_0 and all nodes x_t where there is at least one path from x_0 to x_t in T such that node path weights $\eta(x_0, x_t)$ satisfies the condition: $\eta(x_0, x_t) \geq \lambda$.*

The projection set for x_0 is given by $[x_0, x_1, ..., x_t]$ and is represented as $P(x_0, \lambda) = [x_0^{x_0}, x_1^{x_0}, ..., x_t^{x_0}]$ where x_i^j represents the i^{th} element of the projection set of node j.

Table 1. P(B,0.001) calculations

Parent 'r'	Child 'n'	$\eta(r,n)$	$\eta(r,n) \geq \lambda$?
B	E	0.128	✓
	F	0.088	✓
	C	0.012	✓
	J	0.027(E)	✓
		0.035(F)	✓
	K	0.008	✓
	L	0.008(E)	✓
		0.053(F)	✓
	G	0.0008	✗
	I	0.0011	✓
	M	0.0024	✓
	N	0.0032	✓
	O	0.0015	✓
	P	0.003	✓

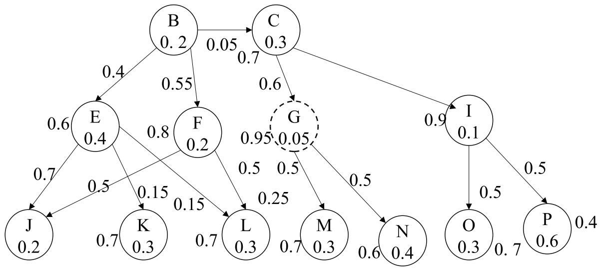
Table 2. P(D,0.001) calculations

Parent 'r'	Child 'n'	$\eta(r,n)$	$\eta(r,n) \geq \lambda$?
D	G	0.0012	✓
	H	0.0212	✓
	I	0.005	✓
	M	0.0035	✓
	N	0.0047	✓
	L	0.0003	✓
	O	0.0003(H)	✗
		0.0067(I)	✓
	P	0.0135	✓

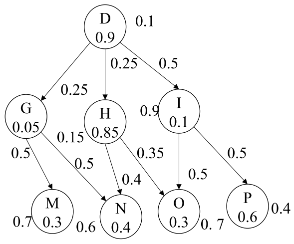
Consider an example CSG from Figure 2. We find the projection of the local root concepts B and D given the threshold coefficient of λ=0.001. The projections and calculations are shown in Figures 4(a) and 4(b) and Tables 1 and 2. All nodes that satisfy the condition of node path weights greater than threshold coefficient are included in the projection. Nodes can have multiple paths to the root (J, L, and O). For node J and L, both the paths (J-E-B, J-F-B and L-E-B, L-F-B respectively) satisfy the condition, whereas for O only one path satisfies the condition (O-I-D). Even then, O is considered in the projection of D, because it still wields some prerequisite effect on D through one of the paths. If the condition for the threshold coefficient is satisfied then the node is included in the projection. Thus by finding the projection graphs of the concepts which map to a resouce, we can precisely extract parts from the course ontology which are relevant to the document and have a desired semantic significance.

4 Observing Clustering in the Course Ontology

In this section we try to make interesting intuitive inferences from observing the clustering of concepts in the ontology because of the calculated projection



(a) P(B,0.001)



(b) P(D,0.001)

Fig. 4. Projections of concepts B and D for λ=0.001

graphs of the concepts mapping for several resources. For this purpose, we consider a specific type of courseware resource, a *test question.* A test question is nothing but a random question asked in a random test for a random course. Test questions were choosen as resources because their concept mapping can be easily identified by the resource creator. Also, answering the question correctly is nothing but identifying the concept mapping and therefore the concept mapping for answers can also be easily identified.

For the experimental setup we created a course ontology comprising of around 1500 concepts, for the graduate level course of "Operating Systems". The ontology was created for the course by consulting with the related instructor and referring to standardized textbooks. Althought there are methods for automatic ontology construction [9–11], we hand coded the ontology for accuracy and consistency. The node weights and link weights, which form an important constituent of the ontology, were assigned by guidance from the course instructor. Concepts with more intrinsic importance for understanding were assigned more self weight and those which depended on many other prerequisite concepts were assigned more prerequisite weights. Consequently it was observed that concepts higher up in the ontology had lower self weights, and self weight values went on increasing further down the ontology, reaching the maximum for leaf nodes. However, for the CSG to be extensible, the leaf nodes were also allowed to have prerequisite weights in case more prerequisite concepts were added later on. Keeping the ontology extensible allows for inclusion of newer concepts, results, researches, etc. adding to the inherent knowledge base, making the course ontology an ever changing and improving repository of course knowledge. The link weights were assigned based on the semantic importance and contribution of the child topic to the understanding of the parent topic. If the understanding of the child concept is detrimental to the understanding of parent concept, then it was assigned a greater link weight. Although by definition, the summation of the link weights for a node should add up to 1, it is generally not observed consistently. Most of the times, some space is left for the inclusion of newer links for prerequisite concepts which are newly added or already existing in the ontology. Again it is seen that higher up in the ontology there is no need to actually leave this space, as the probability of addition of newer links to higher level concepts (implying fundamental changes to the subject area) is less than that to the concepts lower in the ontology. The concept mapping for the "test question" resource was provided by the instructor. These test questions were administered by undergraduate and graduate students, the scores from which were used for the performance analysis. The answers were graded by a minimum of three graders per question, and the averages of the scores were considered to remove any anomalies.

4.1 Clustering of concepts for questions w.r.t. high and low average scores

In this analysis, we separate out the questions with high and low average scores and observe if their concept mapping results in clustering in the ontology. On observing the set of concepts to which these questions map to, it is seen that there are surprisingly high number of common concepts in their respective projection graphs. It is important to note here that, rather than just considering the mapped concepts, the projections of the mapped concepts were considered as they would give a better understanding of the whole set of prerequisite concepts required to answer the question. Figure 5 shows the question-concept distribution separated for the questions with high and low average score.
Observations:

1. For concepts between 750-1000 density of questions with high scores is more than questions with low scores. From this we can infer that students understand the concepts well, or the problems based on these concepts were fairly easy to answer, etc. For the same concepts though, problems 36 and 37 have low scores. This means that these problems were harder because of factors other than the understanding of the concepts. If similar clustering behavior is observed exclusively in questions with low scores, then it can conclusively be said that, those concepts or that part of the ontology needs more explanation from an educators perspective.
2. It is observed that in questions with low scores, concepts are more dispersed (not clustered) around the ontology as compared to those with high inverse correlation.
3. The small clustersing signifies a projection of a concept. It means that questions usually ask concepts near and around a primary concept. These small

clustered concepts mostly are those concepts which come in the primary concepts projection itself. Two small clusters near each other mean two primary concepts projections which are very near to each other.

4. Concepts around 200-400 and 750-1000 are frequently asked among the questions with high and low scores equally. This means that the tests were based on those concepts and the concepts which appear scattered around the plot are those which are needed to answer the specific question. The concepts which do not form the part of the cluster are most definitely concepts which are distant from the primary concept but necessary to answer the particular problem completely.

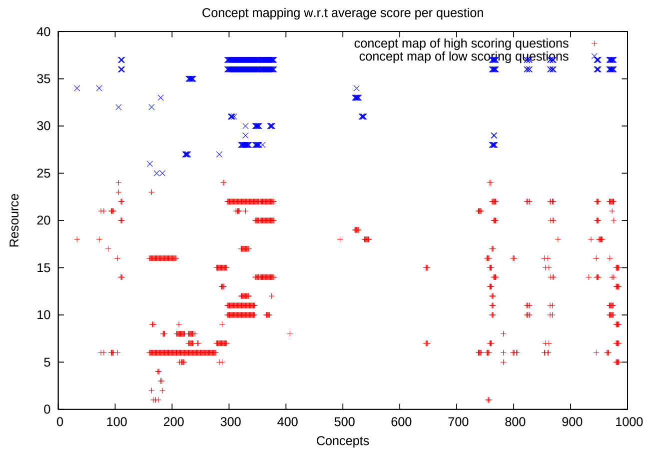


Fig. 5. Concept mapping scatter plot w.r.t. average score

4.2 Clustering of concepts for questions w.r.t. tests

Figure 6 shows the distribution of concepts according to the concept mapping of the questions according to their test distribtion. Questions 1-6 are in test 1, 7-12, 13-18, 19-37 are in tests 2, 3 and 4 respectively.
Observations:

1. Questions 13, 14, and 29, 30, 33, and 35 ask almost similar concepts. Out of these it was observed that 13, 14, 33 and 35 had high scores, and 29 and 30 had low scores. This implies that the correct answering of these questions needs some factors other than understanding the mapped concepts.
2. Most questions are based on or relate to concepts from 100-400 and 750-1000. That means that most of the tests were based on that part of the ontology. This inference has a very interesting implication. It means that the instructor chose to set the questions only on select topics from the course

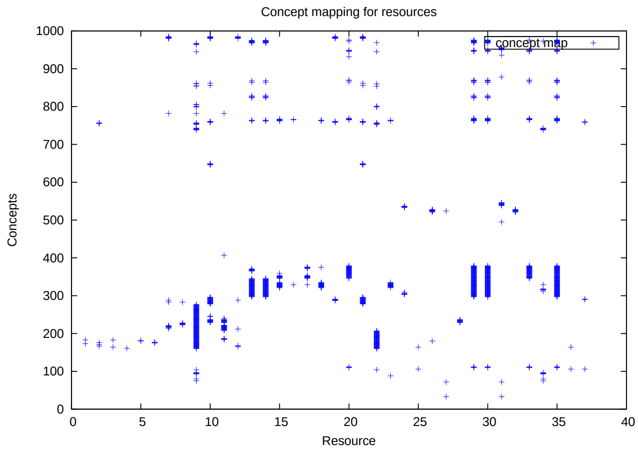


Fig. 6. Concept mapping scatter plot w.r.t. tests

ontology maybe because those were the only topics covered in the course from the ontology. The exact portions of the ontology which were taught and tested can be pointed out using this.

3. As more and more topics are taught from the ontology, tests are increasingly based on more concepts than the previous.
4. There are a lot of small clusters of concepts between concepts 50-400. Since the concepts were numbered "inorder" it means that the small clusters are the mapped concepts, while the bigger ones are the projections of the mapped concepts. Clustering following smaller clustering usually means projections of mapped concepts.

All the observations made are specific to the domain of *course knowledge* and experimental setup for *test questions*. The observations and inferences will change in case of different domains. We present an approach to enable making *interesting* onservations and inferences about the clustering and behaviour of knowledge in different domains.

5 Conclusion and Future work

We propose a technique for representing hierarchical structured knowledge using weighted ontologies and demonstrate it in the domain of courses (education). A representation schema called as the Course Ontology Description Schema is also given for the formal description of the course ontology and is defined in OWL. The schema is actually independent of the domain and can be used to represent other ontologies with similar properties. The relationships in the course ontology are kept to a minimum making them expressive and computable. Another novel approach called *CSG extraction* to extract relevant information from course ontology depending upon the desired semantic significance is given. Using this approach we observe clustering in the course ontlogy and mapping of concepts between resources and the course ontology and make interesting inferences. As future work we are trying to access the ways in which this method of representation and extraction can be applied to classical learning theories which require knowledge to be represented as prerequisite concept structures.

References

1. 2004 MIT OCW Program Evaluation Findings Report (June 2006). $http : //ocw.mit.edu/NR/rdonlyres/FA49E066 - B838 - 4985 - B548 - F85C40B538B8/0/05_{Prog}E_{val}R_{eport}F_{inal.pdf}$
2. Edmondson, K., Concept mapping for Development of Medical Curricula. Presented at the annual meeting of the American Educational Research Association (Atlanta, GA, April 12-16, 1993). 37p.
3. RDF Vocabulary Description Language 1.0: RDF Schema, Dan Brickley and R.V. Guha, Editors. W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/ . Latest version available at http://www.w3.org/TR/rdf-schema/.
4. OWL Web Ontology Language Reference, Mike Dean and Guus Schreiber, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owlref- 20040210/. Latest version available at http://www.w3.org/TR/owl-ref/
5. Javed I. Khan, Manas Hardas, Yongbin Ma, A Study of Problem Difficulty Evaluation for Semantic Network Ontology Based Intelligent Courseware Sharing. WI, pp. 426-429, the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), 2005.
6. T. R. Gruber. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993
7. R. Davis, H. Shrobe, and P. Szolovits., What is a Knowledge Representation?, AI Magazine, 14(1):17-33, 1993
8. Jaakkola, T., Nihamo, L., Digital Learning Materials do not Possess knowledge: Making critical distinction between information and knowledge when designing digital learning materials for education International Standards Organization, Versailles, 2003.
9. Fortuna, B., Grobelnik, M., and Mladenic, D. 2006. Background knowledge for ontology construction. In Proceedings of the 15th International Conference on World Wide Web Edinburgh, Scotland, May 23 - 26, 2006. WWW '06. ACM Press, New York, NY, 949-950.
10. Trent Apted, Judy Kay, "Automatic Construction of Learning Ontologies," icce, p. 1563, 2002 International Conference on Computers in Education (ICCE'02), 2002.
11. Judy Kay, Sam Holden, "Automatic Extraction of Ontologies from Teaching Document Metadata," icce, p. 1555, 2002 International Conference on Computers in Education (ICCE'02), 2002
12. Christos Faloutsos, Kevin S. McCurley, Andrew Tomkins: Fast discovery of connection subgraphs. KDD 2004: 118-127

Appendix:A

Course Ontology Description Schema

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs= "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema#">
<owl:Ontology rdf:about="###">
<rdfs:comment>A schema for describing Course Ontologies</rdfs:comment>
     <rdfs:label>Course Ontology</rdfs:label>
</owl:Ontology>
<owl:Class rdf:ID="Concept">
<rdf:comment>Course ontology concept</rdfs:comment>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#hasPrerequisite"/>
<owl:allValuesFrom rdf:resource="#Relation"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Relation>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#connectsTo">
<owl:allValuesFrom rdf:resource="#Concept">
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasPrerequisite">
<rdfs:range rdf:resource="#Relation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="connectsTo>
<rdfs:range rdf:resource="#Concept">
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasLinkWeight">
<rdfs:domain rdf:resource="#Relation"/>
<rdfs:range rdf:resource="xsd:float"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasSelfWeight">
<rdfs:domain rdf:resource="#Concept"/>
<rdfs:range rdf:resource="xsd:float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPrerequisiteWeight">
<rdfs:domain rdf:resource="#Concept"/>
<rdfs:range rdf:resource="xsd:float"/>
```

```
</owl:DatatypeProperty>

<Concept rdf:ID="MemoryManagement"/>
<Concept rdf:ID="OS">
<hasPrerequisite>
<Relation rdf:ID="relation1">
<connectsTo rdf:resource="#MemoryManagement"/>
<hasLinkWeight rdf:resource="#0.2"/>
</Relation>
</hasPrerequisite>
<hasSelfWeight rdf:resource="0.39"/>
<hasPrerequisiteWeight rdf:resource="0.61"/>
</Concept>
</rdf:RDF>
```