Chapter XXXIII
Knowledge-Based Characterization of Test Questions

Javed Khan
Kent State University, USA

Manas Hardas
Kent State University, USA

ABSTRACT

The recent advances in knowledge engineering entail us to represent knowledge associated with a course in an expressive yet computable format as a hierarchical prerequisite relation-based weighted ontology. A schema called the course concept dependency schema written in Web ontology language (OWL) is designed to represent the prerequisite concept dependency. The knowledge associated with educational resources, like the knowledge required for answering a particular test question correctly, can be mapped to subgraphs in the course ontology. A novel approach for selectively extracting these subgraphs is given and some interesting inferences are made by observing the clustering of knowledge associated with test questions. We argue that the difficulty of a question is not only dependent on the knowledge it tests but also the structure of the knowledge it tests. Some assessment parameters are defined to quantify these properties of the knowledge associated with a test question. It is observed that the parameters are very good indicators of question difficulty.

INTRODUCTION AND BACKGROUND

Traditionally, concepts maps are used to represent the backend context for the course knowledge. Many efforts (Edmondson, 1993; Lee & Heyworth, 2000; Li & Sambasivam, 2003; Heinze-Fry & Novak, 1990) have gone into representing course knowledge using concept maps and using them to evaluate educational resources. In the recent past ontologies were being used to represent structured information in a hierarchical format. Concept maps offer a means to represent hierarchical knowledge; however, they are too expressive and consequently contain more information and semantic relationships than necessary for effective computation.
Ontologies provide a means to effectively map this knowledge into concept hierarchies. Course ontology, particularly, can be roughly defined as a hierarchical representation of the topics involved in the course, connected by relationships with specific semantic significance. Using ontologies for course concept hierarchies in the domain of education is only obvious. It seems such entailment can lead to several important pedagogical applications.

Testing is an integral part of any teaching and learning process. The main pedagogical focus of this research is to objectify the process of testing by estimating the difficulty of a test question based on the depth, breadth, and the amount of conceptual knowledge it tests. Consequently effective testing is possible by subjectively designing the test using these parameter values. A problem/question is one type of educational resource. The commonly observed properties of testware are difficulty or simplicity, breadth and depth of knowledge required to answer, relevance of the question to the root topic, the semantic distance between the concepts tested, ability of the question to test varying groups of students, applicability of the topics taught to a problem, and so forth. While designing a test an educator always tries to come up with questions which have maximum coverage of desired topics, diversity among the topics, good testing capabilities with respect to student knowledge, relevance to the material taught, and so forth. It is important to understand these properties for better design and reengineering of test problems. In this research, we attempt to visualize and understand these properties of test problems by qualitative knowledge-based evaluation. Currently the process of designing of test problems is completely manual, based on human experience and cognition. Design of test problems also follows the basic principles of any engineering design process. The primary elements of design in this case are the information objects. Much effort has been put in the creation and reusability of these information objects called the learning objects, for example, learning object metadata (LOM). Semantic representation standards like RDF and OWL allow the concept knowledge space symbolized by ontologies to be represented consistently.

The main hypothesis for this work is that “test questions” can be qualitatively analyzed for their perceived difficulty, using a purely knowledge-based approach given a background course knowledge base. The main contributing factor to the difficulty of a question is the knowledge associated with the question, that is, the knowledge required to answer the question. Furthermore, the observed difficulty of a question is in positive correlation with the structural properties of the knowledge associated. We propose an assessment system which attempts to evaluate an educational resource like test problem for its “difficulty” based on its knowledge content. We define some parameters which are able to quantify these structural properties associated with the knowledge and observe that they are indeed very good indicators of question difficulty. These parameters can give guidelines for setting up a standard for test problem assessment.

We also 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 a hierarchical order of learning. Another original method for specifically pointing out areas in ontologies of maximum relevance called CSG extraction is given.

Alternate educational approaches that exist do not have a concrete knowledge representation and depend too much on external psychological, cognitive, and syntactical parameters for calculating the perceived difficulty. We analyze test questions objectively from the point of view of the knowledge it tests rather than subjective external parameters. Some researches in cognition (e.g., Apted & Kay, 2002; Gruber, 1993; [18]) have identified other extrinsic parameters, like perceived number of difficult steps, steps required to finish the problem, number of operations in the problems expression, number of unknowns, and so forth which attribute difficulty to a question, but none of them are knowledge-based. The few knowledge-based approaches (Dean &
Schreiber, 2004; Lien, Chang, & Heh, 2003; Li & Sambasivam, 2003; Shrobe & Szolovits, 1993) are limited due to their lack of a solid representation technique which is often rigid, incomplete, and incomputable and disregarded for the structure of the knowledge.

The experimental set up was that of graduate level courses of operating systems and Internet engineering. Ontologies of around 1,500 and 1,000 concepts respectively were created referring to standardized text books and the help of the related instructor. A random set of 40-60 test questions was selected from the question bank relating to the course. The questions were answered by approximately 50 graduates and under graduate students and were graded by two graders. The scores from these were used for the experiments. It was observed that almost 60-70% of the time the performance parameters were in inverse correlation with the average score for a particular question, meaning the parameters were good indicators of the perceived difficulty of the question. From the experiments we were able to conclude: 1. Difficulty and knowledge associated with a question are closely correlated; 2. Knowledge associated with a test question tends to cluster in and around specific portions in the course ontology; and 3. Clustering of knowledge can provide information about which areas need to be developed, which areas are taught, and so forth, and can be used as a tool in guided tutoring. The chapter is structured as follows: in section 2 we present a method for representing the course knowledge more formally; in section 3 we quantify the semantics of the relationships and present a method for extraction; in section 4 we present the assessment parameters; and in sections 5 and 6 we observe the performance of the parameters and make inferences from the observed results.

**COURSE KNOWLEDGE REPRESENTATION**

Any design and evaluation system, like cognition-based models in humans, needs a back end knowledge base. 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, wherein 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.

**Granularity of Representation**

Davis, Shrobe, and Szolovits (1993) define knowledge representation as a “set of ontological commitments” and “a medium of pragmatically efficient computation” (Jaakkola & Nihamo, 2003). 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. The 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.

**Course Ontology**

In computer science, ontology is generally defined as “a specification of a conceptualization” (Heffernan...
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, and so forth.

Relationships are the way the concepts in the ontology are structured with respect to each other. 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 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 the 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 (see Figure 1). Process management is the prerequisite of OS.

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 the child concept to the parent concept. Child concepts imperative in the understanding of parent concepts will have 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.

## 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 (Khan, Hardas, & Ma, 2005). The mapping between the resource space and the concept space is called the concept mapping. All educational resources are based on a few selected concepts from the ontology. When an educator designs courseware, the educator 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 nontrivial problem which is addressed extensively in research of natural language processing, knowledge representation, and so forth. We limit our research to using the concept mapping idea.

## Course Concept Dependency Schema (CCDS)

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 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 subclass 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 CCDS defined course ontology should be able to infer that, since connectsTo links relation1 and MemoryManagement and hasPrerequisite link OS to relation1, MemoryManagement is prerequisite of OS.

**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 a semantic net 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 $l(i, j)$. Each vertex in $T$ is further labeled with self-weight value $s(i)$ 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.

**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 effects 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 $\{x_0, x_1, \ldots, x_{m-1}, x_m, \ldots, x_t\}$ then the node path weight between these two nodes is given by:

$$\eta(x_0, x_t) = W_s(x_t) \prod_{m=0}^{t} l(x_{m-1}, x_m)^{w_p(x_{m-1})}$$

(3)

The node path weight for a node to itself is its self-weight.

$$\eta(x_i, x_i) = W_s(x_i)$$

(4)

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...
Knowledge-Based Characterization of Test Questions

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 take 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.

**CSG Extraction**

A generalized CSG can be vast and processing a huge ontology becomes almost impossible and 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 subgraph called a *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 *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 does not 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. Moreover, 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), which is an English language ontology, LinKB (1 million in English, 3 million in other languages), which is a comprehensive medical/clinical ontology, Gene Ontology (now known as GO, with over 19,000 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 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 prerequisite weights for the leaf nodes too, although most times the prerequisite weight

![Diagram showing node path weights and calculations]

\[ \eta(B, E, L) = 0.3 \times 0.15 \times 0.6 \times 0.4 \times 0.8 = 0.0086 \]
\[ \eta(B, F, L) = 0.3 \times 0.5 \times 0.8 \times 0.55 \times 0.88 = 0.053 \]
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. The 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 local root concept \( x_r \) and projection threshold coefficient \( \lambda \), a projection graph \( P(x_r, \lambda) \) is defined as a subgraph of \( T \) with root \( x_r \) and all nodes \( x \) where there is at least one path from \( x_r \) to \( x \) in \( T \) such that node path weights \( h(x_r, x) \) satisfy the condition \( h(x_r, x) \geq \lambda \).

The projection set consisting of nodes \( [x_0, x_1, x_2, ... x_n] \) for a root concept \( x_0 \) is represented as \( P(x_0, \lambda) = P^{x_0} = [x_0, x_1, x_2, ... x_n] \), where \( x_i \) represents the \( i^{th} \) element of the projection set of node \( j \).

Consider an example CSG from Figure 2. We find the projection of the local root concepts B and D given the threshold coefficient of \( \lambda=0.001 \). The projections and calculations are shown in Figures 4 and 5 and Table 1 and 2. All nodes that satisfy the condition of node path weights greater than the 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 yields 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 resource, we can precisely extract parts from the course ontology which are relevant to the document and have a desired semantic significance.

ASSSESSMENT PARAMETERS

In the previous sections we saw that educational resources map to a set of concepts from the course ontology. For the purpose of demonstration, in this chapter, we consider a special type of educational resource, that is, a test question. A test question is any random question asked in a random test for a random course. Test questions were chosen as re-
Knowledge-Based Characterization of Test Questions

Figure 5. Projection, $P(D, 0.001)$

Table 1. $P(B)$

<table>
<thead>
<tr>
<th>Local root “n”</th>
<th>Node “n”</th>
<th>$\eta(r, n)$</th>
<th>$\eta(r, n) \geq \lambda$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>E</td>
<td>0.128</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>0.088</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0.012</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>J</td>
<td>0.027/0.035</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>K</td>
<td>0.00864</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0.0086/0.053</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>0.00884</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.00112</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>0.0024</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>0.003192</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>0.001512</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>0.003024</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 2. $P(D)$

<table>
<thead>
<tr>
<th>$P(D, \lambda = 0.001)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local root “n”</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

which are discussed are coverage of the question, that is, the amount of knowledge required to answer the question, the diversity of the question, that is, the amount “difference” between the knowledge, and the conceptual distance, that is, the amount of closeness or remoteness of the knowledge.

For the experimental setup we created a course ontology comprising of around 1,500 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. Although there are methods for automatic ontology construction (Kay & Holden, 2002; Lee & Heyworth, 2000; Li & Sambasivam, 2003), 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, and so forth, 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.

Coverage

The coverage of a question gives a cumulative prerequisite effect of the projection graph on the knowledge required to answer a particular question. Coverage of a concept is a direct indicator to the scope of the question in context of the concept space of the course. Formally, “coverage of a node \(x_o\) with respect to the root node \(r\) is defined as, the product of the sum of the node path weights of all nodes in the projection set \(P(x_o, \lambda)\) for the concept \(x_o\), and the incident path weight \(\gamma(r, x_o)\) from the root \(r\).”

If the projection set for concept node \(x_o\), \(P(x_o, \lambda)\) is given by \([x_0, x_1, x_2, ..., x_n]\) then the coverage for node \(x_o\) about the ontology root \(r\) is defined as:

\[
\alpha(x_o) = \gamma(r, x_o) \times \sum_{m=0}^{n} \eta(x_o, x_m)
\]

where \(\gamma(r, x_o)\) is called the incident path weight and

\[
\gamma(x_o, x_m) = \frac{\eta(x_o, x_m)}{W_r(x_o)} = \frac{\eta(x_o, x_m)}{\eta(x_o, x)}
\]

Total coverage of multiple concepts in a problem given by set \([C_1, C_2, ..., C_n]\) is

\[
\alpha(T) = \alpha(C_1) + \alpha(C_2) + ... + \alpha(C_n)
\]

The node path weight defines the prerequisite effect of a node to its designated root. Therefore the summation of the node path weights of all the nodes in the projection set gives the cumulative prerequisite effect of the nodes in the projection graph on their respective mapped concept roots. The concepts in the projection graph in turn are the concepts which are required to understand a particular concept, controlled by the threshold coefficient. The coverage is thus the amount of knowledge required to answer or rather understand a particular concept.

\[
\alpha(B) = \gamma(B) \times \sum_i \eta(B, P^B_i) = (0.5 \times 0.98) \times 0.335882 = 0.16458218
\]
\[
\alpha(D) = \gamma(D) \times \sum_i \eta(D, P^D_i) = (0.2 \times 0.98) \times 0.0560625 = 0.01098825
\]
\[
\alpha(\text{total}) = \alpha(B) + \alpha(D) = 0.16458218 + 0.01098825 = 0.17557043
\]

Diversity

Diversity is calculated by measuring the effect of common and uncommon prerequisite concepts from the projections of the mapped concepts. Diversity is formally defined as “the ratio of summation of node path weights of all nodes in the non-overlap-
ping set to their respective roots, and the sum of the summation of node path weights of all nodes in the overlap set and summation of node path weights of all nodes in the non-overlap set.”

Consider a question that asks a set of concepts, \( C = [c_0, c_1, c_2, \ldots c_n] \). The respective projection sets are given by

\[
P(C_0, \lambda) = [x^{c_0}_1, x^{c_0}_2, \ldots, x^{c_0}_r], \quad P(C_1, \lambda) = [x^{c_1}_1, x^{c_1}_2, \ldots, x^{c_1}_r]
\]

The nonoverlapping and overlapping sets are

\[
N = [x_i, N_1, N_2, \ldots N_r] \quad \text{and} \quad O = [O_i, O_1, O_2, \ldots O_r],
\]

where \( i \) and \( j \) are the local root parents of any element from \( N \) and \( O \), respectively, and \( \forall i, j \in C \).

Cardinality restriction:

\[
\forall O_i, O_1, O_2, \ldots O_r \left[ P(C_0, \lambda) \cap P(C_i, \lambda) \cap P(C_j, \lambda) \right]
\]

\[
\forall N_i, N_1, N_2, \ldots N_r \left[ \left[ P(C_0, \lambda) \cup P(C_i, \lambda) \cup P(C_j, \lambda) \right] - \left[ P(C_0, \lambda) \cap P(C_i, \lambda) \cap P(C_j, \lambda) \right] \right]
\]

Diversity is given by

\[
\Delta = \frac{\sum_{n=1}^{r} \eta(n, N_n)}{\sum_{n=1}^{r} \eta(n, O_n) + \sum_{n=1}^{r} \eta(n, N_n)} \quad \text{where} \quad \forall i, j \in C
\]

\[ (8) \]

The diversity calculated comes to 97%, which means that the concepts are very diverse in their concept space.

**Conceptual Distance**

Conceptual distance is a measure of distance between two concepts with respect to the ontology root. Alternatively, conceptual distance measures the similarity between two concepts by quantifying the distance of the concepts from the ontology root. Formally, it is defined as “the log of inverse of the minimum value of incident path weight (maximum value of threshold coefficient) which is required to encompass all the mapped concepts from the root concept.”

The conceptual distance parameter is designed in such a way that it should be sensitive to the depth of the concepts. Hence it is a function of maximum threshold coefficient required to cover all the nodes from the ontology root. Incident path weight \( \gamma \) of a concept to the root is equivalent to the threshold coefficient \( \lambda \) required to encompass the node. If question asks concept set \( C = [c_1, c_2, c_3, \ldots c_n] \), then the conceptual distance from the root concept \( r \) is

\[
\delta(C_0, C_r) = \log_{10} \left( \frac{1}{\min \{\gamma(r, C_i) \}} \right)
\]

\[ (9) \]
Calculation of conceptual distance for concept set \([E,M,F]\) is shown in Figure 7. Each different type of arrow represents the different paths for a particular concept to the root. In case of multiple paths, like \(M\), the path with the lowest value of incident path weight is considered.

**OBSERVING PARAMETER PERFORMANCE**

To observe the goodness of the parameters for measuring the properties of coverage, diversity, and conceptual distance for a test question, we plot the calculated values of the parameters for a question against the average score for that question. The values for coverage and diversity change according to varying threshold coefficients. Therefore, for these two parameters the values are calculated over a range of \(\lambda\). In the following analysis we assume that knowledge required to answer a particular question is the major factor attributing difficulty to the question without considering external psychological and physiological parameters. The familiarity of the student with the concepts (or how well the concepts have been taught) is a factor of the threshold coefficient. By varying the threshold coefficient the size of the projection graphs can be varied, thus varying the student’s familiarity with the concepts.

The coverage analysis for each question with varying threshold coefficient can be explained by the graph shown in Figure 8. It is observed that the coverage has an inverse relationship with the average score. As the average score increases the coverage for that particular question decreases and vice versa. For all values of \(\lambda\) the coverage has the same relationship; however, this relationship becomes more and more evident with decreasing values of \(\lambda\). As \(\lambda\) decreases, the projection graph increases, thus increasing the coverage values. Hence if the inverse correlation of the coverage graph with average score graph is more for decreasing values of \(\lambda\), we can infer that more concepts are required to answer that particular question. Coverage gives an approximation to the knowledge required to answer a particular question. From the graph it is seen that most of the times, coverage is inversely correlated to average score.

Diversity graph characteristics are similar to coverage graph (see Figure 9). In the case of diversity it is observed that as the threshold coefficient \(\varepsilon\) decreases, the diversity values for all the questions also go on decreasing. This is because as \(\lambda\) decreases the projection set for each concept in the concept set increases. As the projection set increases the probability of having more common concepts increases, thus increasing the coverage of overlap set and decreasing the diversity. In some cases, however, the diversity increases with decrease in \(\lambda\). This happens because sometimes when the threshold coefficient decreases, the projection obviously increases; however, instead of having more overlapping nodes, the nonoverlapping node set increases, consequently increasing the diversity.

\[
\delta(E,F,M) = \log_2 \left( \frac{1}{\min \left[ 0.1568, 0.2156, 0.0023275 \right]} \right) \\
\delta(E,F,M) = \log_2 \left( \frac{1}{\min \left[ 0.0023275 \right]} \right) \\
\delta(E,F,M) = \log_2 (429.65) = 2.63
\]
Knowledge-Based Characterization of Test Questions

Figure 8. Coverage vs. average score

The performance of conceptual distance vs. average score is observed in Figure 10. Although not directly dependent on a projection graph, distance is also inversely correlated to average score. This means that distance is a very good indicator of the similarity between concepts. As the distance between two nodes decreases, the similarity increases. As the similarity increases, the dissimilarity between knowledge required to answer the concepts decreases, consequently increasing the average points scored. As conceptual distance is not a factor of a projection graph, behavior in the graph is constant for all threshold coefficient values. Distance is a logarithmic function as \( \log \) gives the inverse behavior of an exponential function, which is observed here. Similar to coverage and diversity the conceptual distance is also inversely correlated to average score with good correlation. As seen from the behavior of all three assessment parameters, the average score has an inverse correlation with

Figure 9. Diversity vs. average score
Knowledge-Based Characterization of Test Questions

the parameters. This means that the parameters are pretty good indicators of the perceived difficulty of test problems.

OBSERVING CLUSTERING IN THE COURSE ONTOLOGY

In this section we try to make interesting intuitive inferences by observing the clustering of concepts in the ontology because of the projection graphs of several test questions spread out among several different tests.

Clustering of Concepts for Questions with Respect to 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 numbers 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 11 shows the question-concept distribution separated for the questions with high and low average scores.

Observations:

1. For concepts between 750-1,000, 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, and so forth. 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 educator’s perspective.

Figure 10. Conceptual distance vs. average score
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 clustering 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 750-1,000 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.

Figure 12 shows the distribution of concepts according to the concept mapping of the questions according to their test distribution. Questions 1-6 are in Test 1, and 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 750-1,000. 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 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 “in order” 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* observations and inferences about the clustering and behavior of knowledge in different domains.

**TEST DESIGN**

This research leads to an obvious interesting question: Can such characterization be used in learning activity design, such as a *test*? Creative design is nontrivial because a composite design is not only the simple sum of the characteristics of the individual elements used in the design; painting and music compositions are few examples. The characteristics of the elements (i.e., color, shape, and lines) in the design interplay in a very complex ways. Rigorous objective knowledge about the elements is absolutely necessary to be a master. Yet it further requires sophisticated additional senses such as aesthetics about the interplays and the ability to judge specific composition. In engineering design the detail characteristics of the elements are needed as essential design data to meet various design constraints (i.e., maximum allowable weight, size, etc.) and optimization objectives (i.e., weight/power ratio, efficiency, etc.). However, engineering design too has a functional objective which requires holistic metaknowledge about the overall interplay of the elements not in the elements.

Test is an important tool in all forms of learning activity. Here we have discussed a potential method for knowledge-based characterization of test problems. This is a critical step towards the design of tests. One of the main objectives of test-
Knowledge-Based Characterization of Test Questions

ing in education is to gauge the student level of understanding of a topic [19]. The unit of a test is a test problem. By estimating the amount, breadth, and depth of the knowledge required to answer a test question in some quantifiable way—as this research suggested—can provide intelligent design guidance to compose better tests, perhaps with specific coverage of topics, breadth of topics, and even the difficulty. Test design also has additional goals such that a test would gauge certain concepts and are generic enough to test a varied group of students. Certain aspects of a test design problem can probably be framed as a multiobjective constraint satisfactory/optimization problem in a knowledge space. A test also is designed with some formative constraints, such as time and space. Armed, with the knowledge-based assessment technique suggested in this research, it seems many of the content-oriented measurements involved, such as completeness of coverage of all the concepts in a test and if a test is within a certain range of a difficulty value, such a process formulation of test design can be assisted. The challenge of good test design is nevertheless a whole new interesting area ripe for future investigation.

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 the course concept dependency schema written in OWL is also given for the formal description of the course ontology. The schema is independent of the domain and can be used to represent other concept hierarchies 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 the clustering of knowledge associated with test questions in the course ontology and make some interesting inferences. We also define some parameters, namely coverage, diversity, and conceptual distance, which can be used to quantify the properties of knowledge associated with test questions and aid intelligent design of test questions. 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. 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


Knowledge-Based Characterization of Test Questions


APPENDIX

```xml
<?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</rdf: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:DatatypeProperty>
 <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>
```