CHAPTER VI

ONTOGRAPHY BASED AUTOMATIC GENERATION OF QUESTION BANK SYSTEM

6.1. Introduction

Online learning and subsequent evaluation have been important topics in information education. Because education is intended to change student’s behaviors, teachers must conduct tests to assess student’s achievements.

Creating and maintaining the question bank is a time-consuming. When the question bank system contains an insufficient number of items, the exposure frequencies of items may be too high and students may directly recall the answers. Therefore, how to prepare sufficient items in the bank and efficiently generate items have become important research issues.

In designing test items, teaching goals should be considered. According to education testing theory, educational goals can be classified into three different levels: cognition field, emotional field and movement ability.

MCQ (Multiple Choice Questions) are a very popular means of assessment and self-assessment in both traditional and electronic learning settings. They are appealing to the examinees, they can be automatically graded and they provide the capability of frequent testing. An Multiple Choice questionnaire comprises a number of questions named items. Each item consists of a short text describing a question or a sentence to be tested, called stem, and a number of alternative choices, typically four. In single-response Multiple Choice Questions, one of the choices is the correct answer and the wrong alternatives are called distractors.

The quality of multiple choice items is an important topic in education since poor quality items may give away the correct answer or frustrate the learner. One of the main challenges in constructing a multiple-choice test item is the selection of plausible distractors which will better distinguish confident test takers from unconfident ones.
The use of Ontologies in educational settings is ever-increasing. In these settings, domain specific ontologies can be provided as inputs for generating questionnaires in the following ways:

The manual summarization of a domain in the form of concept maps, generated either the teacher, or typically, collaboratively, by the students themselves. Concept maps can be converted into ontologies by using appropriate tools.

The reuse of ontologies created for specific educational technology such as educational content organization, searching, planning and knowledge representation for intelligent tutoring systems.

From the above it is presumed that domain Ontologies are a proper formalism for providing the basis for automatic assessment. Ontologies contain domain knowledge in the form of definitions of terms, individuals belonging to these terms and relationships between individuals. The above constitute the asserted knowledge, that is, explicitly defined facts within the ontology.

Ontologies also incorporate a reasoning mechanism in order to derive facts from explicitly defined knowledge. These facts, not explicitly defined in the initial ontology, constitute the inferred knowledge. In this approach, reasoning is applied before question generation and thus, generated questions are based on both asserted and inferred knowledge.

As a result, a student performing a test is assessed on recalling factual knowledge, but also is expected to apply some ‘lower level intellectual skills’, in the sense of simple domain specific rules, in order to answer questions based on inferred knowledge.

Automatic creation of Multiple Choice Questions tests can be considered as a specialized application of NLG (Natural Language Generation), which is based on the following:

(i) The existence of a knowledge base, expressed in a knowledge representation language, which contains a set of facts about a specific domain. From these facts, question items together with correct answers are extracted for the test.

(ii) The use of the semantic relationships between various elements in the knowledge base to assert “false” statements. These statements are used for generating the distractors in question items.
(iii) The application of Natural Language Generation techniques for actual sentence generation based on the semantic descriptions generated in the first two steps.

Papasalouros and colleagues [137] describe various ontology-based strategies for automatic generation of Multiple Choice Questions, from arbitrary knowledge domains. The generation is based on the basic meta-ontology relations between a ‘class’ and ‘individual’, as well as between two individuals (binary ‘role’). The strategy is further optimized and implemented as a plugin for the Protégé ontology editor, by Tosic and Cubric [138].

Creating fair and meaningful assessment strategy is one the most difficult areas for every educator. Objective testing has been extensively studied and evaluated as a method for formative and, to a lesser extent, summative assessment. While the objective tests need to be complemented by other assessment strategies in order to assess higher cognitive domains, they however provide important benefits that are directly addressing the student needs such as, providing prompt and frequent feedback.

Tosic and Cubric [138] describe ontology-based automatic generation of assessment of an arbitrary knowledge domain. The main novelty is in extending the existing body of research in two important directions: First, enriching the meta-ontology used for question generation with the new elements, such as annotations. Second, adding semantic interpretation to the mapping between the domain and the Multiple Choice Questions ontology in terms of ‘question templates’.

Fatih Ugurdag and colleagues [139] came up with a concept called smart question, where a static question, with fixed numbers and answer choices, is converted to a dynamic question with parameters. The parameters have a list of possible values, and the answer choices are unordered. A smart question is turned into specific static question instances through question generation software.

Li and Sambasivam [140] presented an approach for automatic question generation and difficulty assessment for intelligent tutoring. That approach was successfully applied to the computer architecture course using a quantitative hierarchy. A knowledge structure, the concept graph, based on semantic networks, was used to automatically generate verbal, descriptive questions about computer architecture. An algorithm generates the question in
multiple variables. The knowledge structure for question generation is a \text{DAG (Directed Acyclic Graph) concept graph}.

Lakshmi Palaniappan and Sambasiva Rao \[141\], presents a new method for \text{Ontology-based question answering with the use of query template for dining ontology as the domain} (\text{Service of the restaurant is called dining}). Hypothesis questions and query templates can be produced from domain ontology.

Onjira Sitthisak and Lester Gilbert \[142\] proposed a competency framework, named \text{COMpetence-Based learner knowledge for personalized Assessment (COMBA)}, is implemented with an ontological database that represents the intended learning outcome to be assessed across a number of dimensions such as levels of cognitive ability and subject matter content involved an experiment to test its outputs, and the results.

Xuchen Yao \[143\] proposes a novel approach based entirely on semantics. Question generation is the task of generating reasonable questions from a text. It is a relatively new research topic and has its potential usage in intelligent tutoring systems and closed-domain question answering systems.

\text{MRS (Minimal Recursion Semantics)} is a meta-level semantic representation with emphasis on scope under specification. Natural language sentence can be interpreted as an \text{Minimal Recursion Semantics structure} by parsing, and an \text{Minimal Recursion Semantics structure} can be realized as a natural language sentence through generation.

The methodology presented by Mitkov and colleagues \[144\] generates multiple choice questions based on text corpora in a specific domain. It utilizes several techniques such as shallow parsing, term extraction, sentence transformation and computation of semantic distance. It also employs ontologies such as WordNet. It describes a novel computer-aided procedure for generating multiple-choice test items from electronic documents. In addition to employing various \text{Natural Language Processing techniques}, including shallow parsing, automatic term extraction, sentence transformation and computing of semantic distance, the system makes use of language resources such as corpora and ontologies.
Figure 6.1. A Framework for Ontology Matching and Merging
6.2. A Framework for Ontology Matching and Merging

A framework for Ontology Matching and Merging is proposed as in Figure 6.1. Ontologies are converted into triples set and these triples set are used as input for the Graph Matcher and the Ontology Abstract Machine Matcher respectively. The Graph Matcher determines the matched triples set and generate the set of unaligned triples of the inputted ontologies. Using the Matched Ontology and the unaligned triples of the inputted ontologies, the Graph Merger determines the merged triples set which can be converted into a Merged Ontology.

The Ontology Abstract Machine Merger determines the triples of the Merged Ontology and the triples of the symmetric difference of all the inputted ontologies. Using the Merged ontology and the symmetric difference of all the inputted ontologies, the Ontology Abstract Machine Matcher determines the matched triples set which can be converted into a Matched Ontology.

6.3. Validation of Framework Using Ontology based Question Bank System

A framework for Ontology based Question Bank System is proposed in this section as in Figure 6.2. The teacher’s input is used for the creation of blueprint ontology. Jena API is used to extract the questions according to the blueprint generated by the teacher’s input. Eclipse IDE is used as the front end to display the questions as a web application in browser. With the integration of blueprint ontology and domain ontology, the question bank is created automatically along with answers.

The process begins by selecting a domain of knowledge for which multiple choice questions are to be generated. The domain of knowledge can be data structures or operating systems or artificial intelligence etc. Once, the knowledge domain is identified, an ontology expert converts the domain knowledge into domain ontology. This ontology is then fed to the new system where it is parsed for facts. These facts are eventually converted to multiple choice questions and accumulated in a temporary question bank. These questions could then be verified. Finally the the correct questions are added to the main question bank.
Domain ontology should adhere to certain conventions to generate syntactically correct sentences, with the Natural Language Generation techniques adopted. The names of classes, individuals and properties must contain words connected with underscores, hyphens, or multiple concatenated words. The Multiple Choice Questions presentation application is implemented in Java. JENA Semantic Web framework is used for Web Ontology Language ontologies management and storing and thus for the implementation of strategies. In Question Bank system, the action is initiated by entering the question and pushing the “Submit” button. After loading and reading the ontology for preparing the model, the process of parsing the sentence is started.

As a first step of this process, the query sentence is formed by an object derived from Domain Ontology and Blueprint Ontology. By using this object, the type of the sentence is investigated and the suitable question type is tried to found for the question sentence.

As a second step, the words of the sentence are kept for parsing (except "the", "a", "an", "of", "is", "was", "has", "who", "what", "when", "which", "how", "many ", "much", "why").

While parsing the sentence, the following strategies are taken into consideration:

(i) Matching of the words to the classes and attributes being controlled,

(ii) If the parsing with words cannot be done, it is tried to find classes and attributes by looking at the rules,

(iii) If the related classes and attributes cannot be found, it looks for the synonyms of the words in Simple Natural Language Generation.

After finishing the parsing, the query is built. While forming the query, question sentence types and auxiliary verbs are taken into consideration. Although individual names are not entered, they are taken into consideration with the closest individual names. To be able to find the closest individual name, fuzzy string matching method is used. Lastly, the query is sent to the reasoner. After getting and processing the query by the reasoner, it is sent to the web page over the Jsp. Finally, the answer is displayed to the users.
Figure 6.2. Ontology Based Question bank System
Figure 6.3. Matching Framework
The Matching Framework proposed in the Figure 6.3 consists of the following components:

(i) Key Ontology
(ii) Test Ontology
(iii) Matching Repository
(iv) Ontology Repository
(v) Rule Repository
(vi) Selection Engine
(vii) List (Matcher)
(viii) Execution Engine
(ix) Matching Result

The Key Ontology contains the answers for the questions that were created at the time the question pools are created. The Test Ontology contains the student’s responses for the questions answered by the students. The Matching Repository contains reusable matching components e.g. information concerning the ontological formats served by particular matches and information regarding the natural language that the algorithm can handle. The Ontology Repository manages the matching inputs defined by the ontology metadata.

The Rule Repository is used to determine which matching algorithm is to be used for which type of ontologies. The Selection Engine is responsible for the process to determine the algorithms applicable to a specific set of inputs and the engine responsible for the execution of the matching workflow. The engine is aware of the background information detailing the available matching services and the properties of the input ontologies. List Matcher uses the list of possible matching algorithms the instance matchers which deal with the representation language of the ontology. Execution Engine is responsible for completing the matching task. The final Test result of the student is displayed after performing match.
6.4. Summary

This chapter presents an approach for Ontology based question bank system using domain ontology. It involves Resource Description Framework, ontology model and web ontology language. In future this research work can be extended along several prospective directions: add other ontology components, such as rules, axioms, restrictions, events etc. to the domain ontology used for question generations; generalize the Multiple Choice Questions ontology to include other types of objective tests; apply integration of text similarity and ontological approaches to generation of distracters etc.