Semi-Automatic Generation of Metadata for Items in a Question Repository

Rekha Ramesh
Department of educational Technology,
Indian Institute of Technology Bombay,
Mumbai, India
e-mail: rekha.ramesh@iitb.ac.in

Shitanshu Mishra
Department of educational Technology,
Indian Institute of Technology Bombay,
Mumbai, India
e-mail: shitanshu@iitb.ac.in

Sasikumar M.
CDAC Bombay,
Mumbai, India
e-mail: the.little.sasi@gmail.com

Sridhar Iyer
Department of CSE,
Indian Institute of Technology Bombay,
Mumbai, India
e-mail: sri@iitb.ac.in

***Abstract*— Question repositories are organized collections of assessment questions (items) that serve many purposes. Questions required vary widely along many dimensions such as cognitive level, difficulty level content and question type depending upon the context in which it is used. This paper proposes a software system that semi-automatically generates metadata for items in a question repository. The metadata in this paper corresponds to "cognitive level", "question type", "content" and "difficulty level". The developed system was tested for its usability and accuracy and the results are promising. The accuracy with respect to automatic generation of cognitive level tags was 78% and other tags were generated with more than 90% accuracy. Usability testing has shown that the system is user friendly and useful in multiple ways.**

Keywords- Question Repositories; Semi-Automatic Tagging, Question Annotation, Question Metadata

I. INTRODUCTION

Question repositories (QR) are organised collections of assessment questions (items) that serve many purposes [1][2]. Teachers can utilize the questions and generate an assessment instrument. Questions required may vary widely along many dimensions such as cognitive level, difficulty level, content and question type depending upon the context in which it is used [3]. Apart from the summative assessment where we test the students' understanding of knowledge on completion of the course, questions are also required by teachers for formative and diagnostic type of assessment [4][5]. For formative assessment, teachers pose specific questions to individual or groups of students during the learning process to determine what specific concepts or skills they may be having trouble with [6]. Diagnostic assessments are taken at the beginning of a topic, where teachers can ask questions to students to determine prior knowledge of a particular subject [5].

Many instructional strategies irrespective of their mode of implementation require questions catering to varying specifications in different situations. For example, Problem Based Learning, Think-Pair-Share (TPS) [8], Peer Instruction (PI) [9], etc require questions with different attributes. Moreover, not only teachers, but students also require questions for self-learning and self-assessment.

So, teachers need different types of questions. Their usability in a particular context depends on parameters such as cognitive level, difficulty level, type of question, content / topic, etc. Hence it is important to tag questions in a QR with such a set of tags.

Many existing LMS like Moodle, Blackboard, Sumtotal, Sakai, etc contain assessment management system for the generation of tests, quizzes, etc. They also facilitate the creation of question repositories. These systems provide user defined custom tags which the teachers can use to manually tag each question at the time of creation of assessment instrument for tests. For example in Moodle, a question can be organised into categories and can also be associated with user defined tags [10][11]. Even though the question creation interface may support introduction of user-defined tags, the creators of question repositories generally did not seem to use this feature extensively. The 'higher-order' tags, such as cognitive level, and so on, are missing in these questions. Moreover, large number of questions in a repository have only 'basic-level' tags such as topic, subject and so on (section 2 contains detailed description) So, the teacher has to now verify the suitability of the question with respect to required attributes such as its Blooms level, type of question, difficulty level, the content or topic of the question, etc. for the desired assessment instrument. If these repositories contain questions that are tagged with such properties, then the process of selection becomes simply querying the Question Repository with required attributes.

Most of the questions created by teachers have insufficient tags. And without adequate tagging, they are difficult to use in practical scenarios. So, it is desirable to have enough tags for all questions in repository. These annotations can either be done manually or automatically or semi-automatically.

Manually tagging the question is an additional overhead for teachers. Moreover, it is essential that the people who tag questions should be expert in both subject knowledge and educational technology. Thus, it is highly desirable to have an automatic tagging system.

978-1-4799-6489-5/14 $31.00 © 2014 IEEE
DOI 10.1109/T4E.2014.38

Unfortunately, fully automatic tagging is a challenge for researchers. There is no standardized method to frame a question. The language used to frame the questions can drastically vary depending upon the perspectives of the individual examiners. Cognitive levels defined by the Blooms taxonomy are also not absolute. A question may belong to more than one category, particularly at adjacent levels [12]. There is no formula to calculate the difficulty level of question before giving it to the students. Teachers gain this expertise over a period of time. Hence, we opt for semi-automatic tagging of question.

We are building a semi-automatic system which facilitates human intervention to increase the accuracy of tagging. We discuss the design of a system which takes a question as input from the user and attempts to identify various tags such as Blooms level, type of question, difficulty level, and the content or topic of the question. If the teachers are not satisfied with some of the suggested tags, manual editing facility is provided to modify the tags. Teachers can also reformat the questions at the entry level itself so as to facilitate more accurate tagging. This is done by redrafting the question in line with one of the predefined question templates.

We have done only a preliminary evaluation of our tagging system with few users. Users perceived the system to be user friendly. They explored the various components of the system and felt that the system correctly annotates the questions, with less inconsistency and ambiguity. To investigate more into the accuracy of the tagging, we performed an accuracy-test, where two CS education researchers tested the tags generated for a set of 50 randomly picked Data Structure questions. The detailed explanation and results of testing is provided in section 5.

Section 2 discusses the related literature for our work in this area. The design approach is explained in section 3. The implementation aspects and the user interface are described in section 4. Results of usability testing are given in section 5. Section 6 consists of discussion, conclusions and future scope of our work.

II. Related Study

We did extensive literature survey on the need of QR, metadata that can be associated with the questions and the different possible values that can be associated with each tag.

The items in an item banks vary widely one from another in their characteristics and use [13]. The authors have described two procedures for describing an item's content In fixed category scheme, the content is divided into topics, subject matter areas, or instructional objectives. In keywords based methods the item can be associated with any number of user defined tags. In order to support efficient retrieval, questions must be described with appropriate metadata [1].

Most of the QR is part of the assessment management component of learning management system (LMS). It allows a teacher to create, preview, and edit questions in a database of question categories. For example, Moodle and Totara provides the facility of question bank creation and management and each question is annotated with the question type and category / topic to which it is associated [9][10]. Test and Surveys option in Blackboard LMS allows users to manually add metadata such as categories, topics, levels of difficulty, and keywords to each question [14]. In the Test and Quizzes Tool of Sakai LMS, user can tag each question with its question type that includes essay, multiple choice, fill in the blank, etc [15]. Sumtotal also has a randomized Quizzes feature [16]

Assessment should be aligned with the learning objectives intended for a course [7]. As the learning objectives can span across all the cognitive levels, the assessment also should consist of questions of varying cognitive levels to achieve intended outcomes. We have considered cognitive levels defined by Blooms taxonomy [12].

There are internationally accepted standards for question / item and repositories, e.g. IMS QTI. It provides commonly used question types such as multiple choice/response, true and false, image hot spot, fill the blank, select text, slide, drag object/target, order objects, match items and connect points [17]. Questions are broadly grouped into three classes namely short answer questions, long answer questions and others which are further tagged with their degree of difficulty (high, medium, low) and deep reasoning or knowledge deficit questions on the basis of Blooms level [18]. Teachers should ask wide variety of questions in terms of cognitive level and question type, which should be aligned with the education objectives defined for the curriculum. In order to distinguish between the different categories of questions, the need for classification scheme was emphasized and a semi-hierarchical classification scheme consisting of six categories spanning across question based on simple concepts to highest level of scientific research question was proposed [19]. In [23], the authors have used difficulty level as the metric to operationalize the quality of questions generated. They developed evaluation rubrics to rate the difficulty level of the question as high, medium, or low.

Based on the literature survey, varieties of questions are needed for different types of assessments and instructional strategies. Even though existing repositories provide the facility of associating user defined tags to a question, they are insufficient and are to be manually put by the user. This motivated to go for a semi-automated tagging system. From the literature surveyed and commonly used set of tags recommended by teachers, our present work focuses on four set of tags namely, cognitive level, difficulty level, question type and content / topic.

III. DESIGN

We have identified four tags for automated tagging namely, cognitive level, difficulty level, question type and content /

topic. We also restrict to questions from Data Structures course of engineering curriculum. For the various tags the value range is as follows:

TABLE I. QUESTION TAGS AND ITS VALUES

Tags	Values
Cognitive Level	Six levels of Blooms taxonomy: Recall, Understand, Apply, Analyze, Evaluate, Create
Question Type	Objective: Fill-in-the-blanks, Multiple-choice, Match-the-following, True-false, Answer-in-one-word
	Subjective: Short-note, Differentiate/Comparison, Program-implementation, Short-answer, Long-answer
	WH-type: Why, When, What, Who, Whom, How
Content	Topics and subtopics from the syllabus that forms the node names of the ontology.
Difficulty Level	Low, Medium, High

The key design element is how to extract relevant information from the question text to assign appropriate values for these tags. The following four subsections outline this. We do not use any other source of information other than the question text. Each question text is first parsed to separate into a set of words or tokens. All the punctuation marks are removed and the sentence is converted into lowercase. For example,

Que. *Perform preorder, inorder and postorder traversal on a binary tree”*

will be converted into following set of tokens

“perform preorder inorder postorder traversal on a binary tree”

The dictionary stores keywords that includes verbs associated with each cognitive level defined by Blooms taxonomy, different question types described earlier, and phrases that are normally present in the questions specific to a particular domain. The phrases can be “Write a program”, “Write an algorithm”, “Write a short note on”, “Provide an example”, etc. The keywords are stored along with its cognitive level. N-grams algorithm is used to identify token with multiple words. This dictionary is used to identify cognitive level and question type. For content identification, only ontology is used which is described in detail in subsection C.

A. Cognitive-Level identification

Blooms taxonomy forms the basis for cognitive level identification of a question. Every level of Blooms taxonomy namely, Recall, Understand, Apply, Analyze, Evaluate and Create is associated with an elaborate set of keywords [12]. These keywords are stored into a dictionary. The tokens are matched to the keywords in the dictionary and accordingly its cognitive level is identified. For example,

Que 1. Draw a binary tree for the given expression

*A*B – (C+D) * (P/Q)*

‘Draw’ is a keyword associated with blooms level Apply. So the cognitive level of question is identified as Apply.

If two tokens match with the keywords of two different Blooms level, then the higher level one is chosen as cognitive level of the complete question.

For example,

Que 2. State the difference between arrays and linked lists

The keyword ‘*State*’ is at Recall level and ‘*Difference between*’ is a phrase stored in the dictionary which is at Analyze level. So the cognitive level of the question is identified as Analyze.

B. Question-Type Identification

The dictionary stores many type of question types in it. Basically they are broadly categorized as subjective type, objective type and WH-Type. Objective type is further classified as Fill-in-the-blanks, Multiple-choice, Match-the-following, True-false, Answer-in-one-word, etc. Similarly, subjective questions are classified as short-note, differentiate/Comparison, Program-implementation, short-answer, long-answer, etc. WH-Type question are of type How, Why, When, What, Who and Whom. Question type is decided by matching the keywords extracted from the question with the keyword list in the dictionary. For example, consider a question

Que: Write a program to implement quick sort.

The keywords extracted will be *write a program* and *implement*. “*Write a program*” refers to Program Implementation. Question is of type “Program Implementation”, Question category is “Subjective” and since there is no WH-Type words, it is “*Not a WH-Type*”.

C. Content Identification

To identify the content/ topic of question, we have to map the concepts from a question to the contents of the syllabus. But there may not be direct matching of concepts in the syllabus. We have represented the syllabus using Ontology. Ontology describes a subject domain using notions of concepts, instances, attributes, relations and axioms [20]. The Data Structures subject from Semester IV of second year computer engineering of Mumbai University is chosen as an example domain in this paper. Fig. 1 represents syllabus ontology for the domain.

Every node in the ontology represents a concept/topic from the syllabus domain. It has the same name as the topic name from the syllabus. The name of the subject forms the root of the ontology tree. All the major topics form the level 1 nodes in the ontology. The major topics can be further narrowed down to subtopics that form the subclasses in the ontology. The syllabus ontology forms the semantically connected network of concepts (topics) from the domain.

Que 3. Write a program to implement queue using linked list.

Here the concepts *queue* and *linked list* will exactly match with the node names in the ontology. So these become the content tags associated with the question.

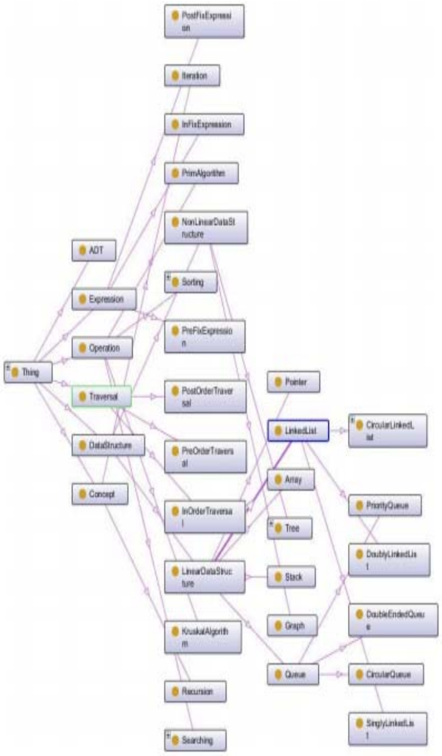


Figure1. Ontology for Data Structure subject

Teachers can frame the questions in many ways. Hence, concepts from the questions may not exactly match with the node names in the ontology. In such situations the earlier described process of exactly matching the node names may not work. This can be solved to some extent by annotating every node in the ontology with a set of synonyms. Synonyms form possible alternative names that the examiner may use in place of node names in the ontology while framing questions in the QP. For example,

Que 4. What are the advantages of using a LIFO structure as a linked list than array?

In this case, the concepts *linked list* and *array* will be directly identified, but *LIFO* does not directly map with the node names in the ontology. Then the annotations will be searched to get a suitable match as *stack*. N-Gram algorithm is used to find the concepts with multiple words.

D. Difficulty-Level Identification

From the literature, we found that the difficulty level of a question is based on using Cognitive Level, Concept Involved, Concept Difficulty and Question Type [21]. Questions pertaining to higher cognitive level of Blooms taxonomy are considered to be more difficult than the questions with lower cognitive level. The cognitive levels are coded to get numerical values which form the first parameter to calculate the difficulty level. Similarly questions with multiple concepts are considered to be more difficult than questions with less number of concepts [23]. So, number of concepts form the second parameter. Moreover concepts themselves are not of same difficulty levels. Some are more difficult than others. So, in our case, every concept is classified into one of the four levels of difficulty assigned by the domain expert. The lowest level of difficulty has the value 1 and the highest level has value 4. For example, the concepts such as *Arrays, Binary Tree, Minimum Spanning Tree and AVL Trees* are considered to be in increasing order of difficulty. If the question contains more than one concept, the highest value of difficulty level is taken among the difficulty level of all the concepts. The difficulty of the concepts forms the third parameter. So, the difficulty level of a question is the addition of the values of all these parameters. Higher the value, higher is the difficulty level. Based on these criteria, we have provided the estimate for the difficulty level. The values of the difficulty levels are stored as annotations for each node in the ontology.

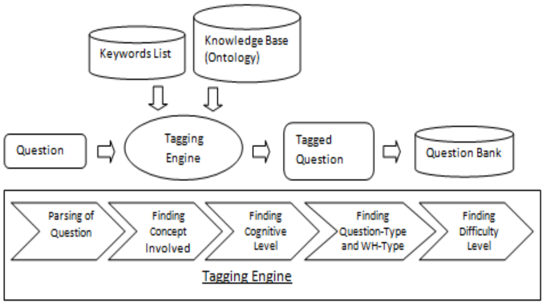


Figure 2. System Block Diagram

E. System Architecture

We have developed a system that facilitates semi automatic tagging of the questions based on the above approach. The system block diagram is shown in Fig. 2.

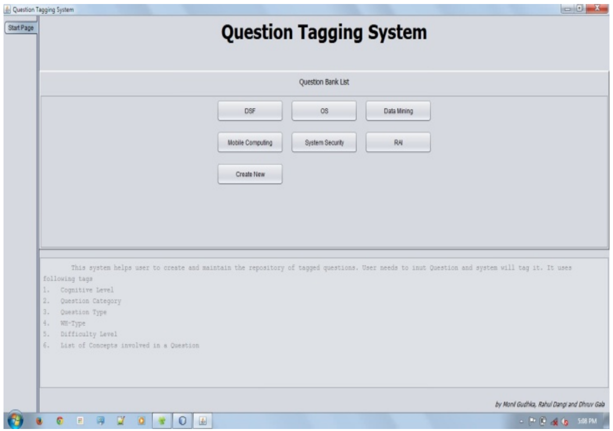
The tagging engine takes a question from a teacher, processes it to identify the tags and outputs a list of tags associated with it which is stored into a question bank or repository. Four main tags are considered namely cognitive level defined by the Blooms taxonomy, difficulty level, type of question and the name of content or topic from the syllabus as explained before.

Once the tags are identified, teacher can see them and manual editing facility is provided to change the tags if needed. When a teacher enters a question, the system asks whether he/she wants to reformulate the questions. If desired, teacher can modify the question using some predefined question templates provided. These question templates are extracted from the dictionary. This will help the system to tag accurately.

Actual implementation is described in next section.

IV. Implementation

The system is implemented using Java programming language. The ontology is created using protégé 4.3 application. The Protégé OWL file is parsed by the OwlParser class of Java. N-Grams algorithm is implemented to extract multi worded concepts from a question.



This system helps user to create and maintain the repository of tagged questions. User needs to inut Question and system will tag it. It uses following tags
1. Cognitive Level
2. Question Category
3. Question Type
4. WH-Type
5. Difficulty Level
6. List of Concepts involved in a Question

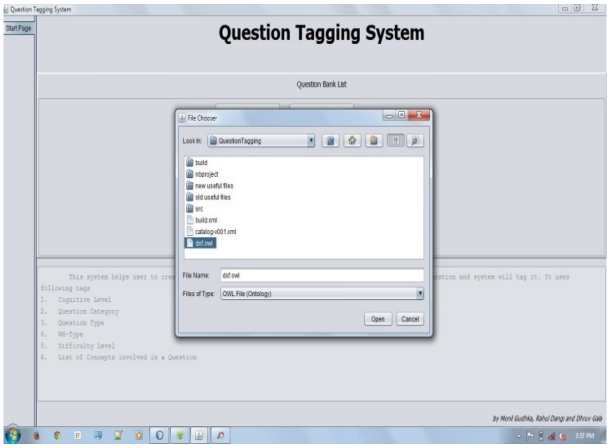
Figure 3. Home/ Question Bank Selection

User Interface: The beginning screen of the Question Tagging System is as shown in Fig. 3. The teacher can use any one of the available question banks which are categorized into different subjects or the teacher can create a new one for the subject of his choice.

Multiple question banks can be maintained in this system. If a teacher opts to create a new question bank for a particular domain, then the system will ask for an ontology file for that domain as shown in the Fig. 4. Currently we have developed ontology for data structure subject.

Once the teacher selects a particular question bank, the next interface will display the existing tagged questions in question bank as shown in the Fig. 5. This has options to enter a new question, delete existing question, edit tags of any question manually and save the changes in question bank.

If an option for entering a new question is selected by the teacher then the new window opens where he can enter a question as shown in Fig. 6.

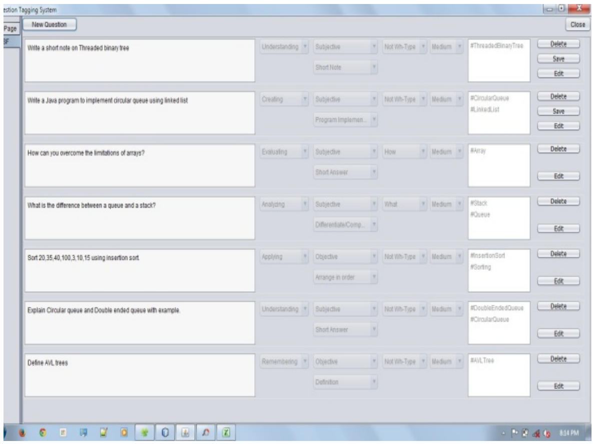


Look In: QuestionTagging

build
nbproject
new useful files
old useful files
src
build.xml
catalog-v001.xml
dsf.owl

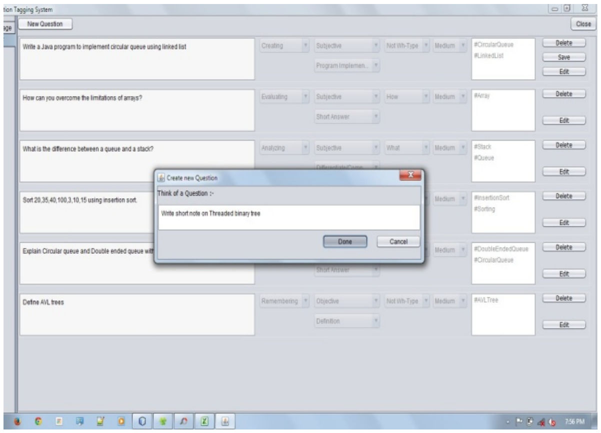
File Name: dsf.owl
Files of Type: OWL File (Ontology)

Figure 4. Ontology Selection



Write a short note on Threaded binary tree	Understanding	Subjective Short Note	Not Wh-Type	Medium	#ThreadedBinaryTree
Write a Java program to implement circular queue using linked list	Creating	Subjective Program Implemen...	Not Wh-Type	Medium	#CircularQueue #LinkedList
How can you overcome the limitations of arrays?	Evaluating	Subjective Short Answer	How	Medium	#Array
What is the difference between a queue and a stack?	Analyzing	Subjective Differentiate/Comp...	What	Medium	#Stack #Queue
Sort 20,35,40,100,3,10,15 using insertion sort.	Applying	Objective Arrange in order	Not Wh-Type	Medium	#InsertionSort #Sorting
Explain Circular queue and Double ended queue with example.	Understanding	Subjective Short Answer	Not Wh-Type	Medium	#DoubleEndedQueue #CircularQueue
Define AVL trees	Remembering	Objective Definition	Not Wh-Type	Medium	#AVLTree

Figure 5. Tageed Questions in Question bank

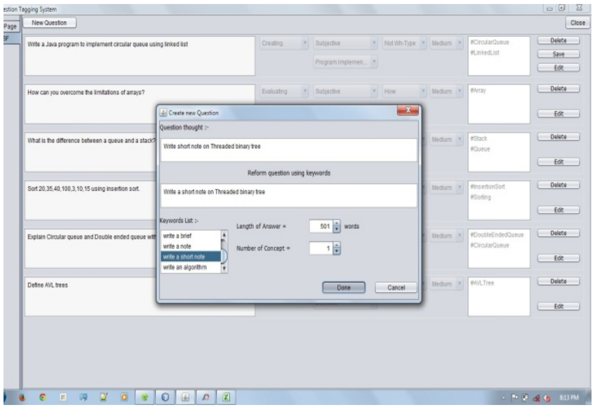


Think of a Question :-

Write short note on Threaded binary tree

Figure 6. Question Entering Window

After entering a question, the user will be asked whether he wants to reform the question. This is optional. If User selects “Yes” then Question reformulation window comes as shown in Fig. 7, where user can modify the entered Question. User can help system to tag questions more accurately. Teacher can change the wordings of the entered question by selecting from the available predefined question template.



Question thought :-

Write short note on Threaded binary tree

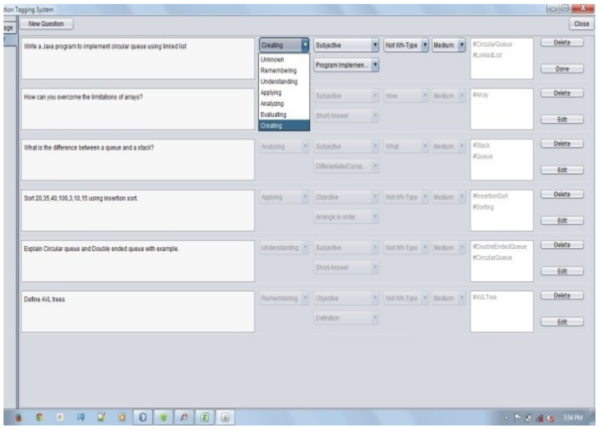
Reform question using keywords

Write a short note on Threaded binary tree

Keywords List :-
write a brief
write a note
write a short note
write an algorithm

Length of Answer = 501 words
Number of Concept = 1

Figure 7. Question Reformulation Window



Write a Java program to implement circular queue using linked list	Creating	Subjective Program Implemen...	Not Wh-Type	Medium	#CircularQueue #LinkedList
How can you overcome the limitations of arrays?		Subjective Short Answer	How	Medium	#Array
What is the difference between a queue and a stack?	Analyzing	Subjective Differentiate/Comp...	What	Medium	#Stack #Queue
Sort 20,35,40,100,3,10,15 using insertion sort.	Applying	Objective Arrange in order	Not Wh-Type	Medium	#InsertionSort #Sorting
Explain Circular queue and Double ended queue with example.	Understanding	Subjective Short Answer	Not Wh-Type	Medium	#DoubleEndedQueue #CircularQueue
Define AVL trees	Remembering	Objective Definition	Not Wh-Type	Medium	#AVLTree

Figure 8. Question Editing Window

After finalizing the question, question along with System generated tags will be displayed on screen as shown in Fig. 8. If the user is not satisfied, then manual editing facility is also provided.

V. USER TESTING

To investigate more into the accuracy of the tagging, we performed an accuracy-test, where two CS education researchers tested the tags generated for a set of 50 randomly picked Data Structure questions. It was found that the accuracy with respect to the cognitive level annotations was 78%; with respect to question type annotations was 90%. The difficulty annotations were 93% accurate, and the content identification was 87.5% accurate. The inter rater reliability was 100%.

In order to carry out the usability and user friendliness, we gave the system to 11 users to explore and use the system. Each user is a CS instructor with an experience of at least 10 years. The user testing involved two phases of activities: In the first phase, users executed the Semi-Automatic Question Tagger on their computer. They were given simple set of instructions, viz., (i) "explore each components of the system"; (ii) "Edit the existing Data Structures Questions"; (iii) “add your own Questions"; (iv) "cross check the auto-generated annotations”. In the Second phase, users were given a set of ten questions from the questionnaire prepared to test the System Usability (SUS) [22] score of the Question Tagger. We associated each question of the SUS questionnaire with an open-ended feedback question. For each of the SUS question, we asked the users to write an open-ended response to explain the justification of selecting a specific likert scale score for a question.

We have done only a preliminary evaluation with few users. While the N for the SUS data is not sufficient for statistical significance, we have attempted to triangulate the scores using open-ended responses and analyzed them to validate our inferences. All the open-ended responses from all the candidates were qualitatively analyzed and coded to test the usability of the Question Tagger. The test revealed that 8 out of 11 users found the system to be useful. Users perceived the system to be user friendly. Some of the reasons frequently cited were:

- the easy GUI
- nil or least requirement of technical knowledge to use the system
- properly structured components of the system

Most prominent benefits of the system as reported by the users are:

- Helpful in generating question papers
- Setting questions as per student's level
- Saves time
- Coverage of important metadata associated with a questions

Users also perceived that the system correctly annotates the questions, with less inconsistency and ambiguity.

VI. DISCUSSION AND CONCLUSION

To collect system requirements, we surveyed literatures and identified various attributes related to assessment questions. The identified attributes were Cognitive level, Difficulty level, Question type and content / topic. The Semi-Automatic Question Tagging system was built which tags each question with the value of these attributes. Usability testing has shown that the system is user friendly and useful in multiple ways. In the current system, the “cognitive level”, “content” and the “question type” tag generation used techniques like N-grams keyword matching, semantic dictionary and domain ontology.

Still there are challenges that needs to be addressed such as the inherent ambiguities present in the question framed, inability of the system to tag the question with cognitive level if question keywords do not match with any of the words associated with the Blooms taxonomy. Sometimes, the keywords present in the question itself may be misleading. For example, the question “List the differences between queues and stacks” will be classified into “Recall level” by our system because of the keyword “List” but it is at an “analyse level”. The tagging accuracy could be further enhanced by more sophisticated algorithms.

The current focus was actually building a semi-automatic tagging system with a reasonable accuracy which is confirmed by our preliminary testing. It would be highly desirable if we can minimize the need of manual intervention and maximize the accuracy of fully machine based tagging system. Our next research objective is to investigate further and do an extensive and rigorous usability test. In order to do that, we would integrate the system into an open source Moodle LMS.

The future scope includes extending this work to other subject domains of engineering curriculum and also strive towards improving the accuracy of tagging.

REFERENCES

[1] S. Currier, " Assessment item banks and repositories". JISC CETI, 2007.

[2] B. D. Wright and S.R. Bell, "Item banks: What, why, how. Journal of Educational Measurement," 21(4), pp. 331-345, 1984.

[3] J. Millman, and J. A. Arter, "Issues in item banking," Journal of Educational Measurement," 21(4), pp.315-330, 1984.

[4] W. Harlen and M. James, "Assessment and learning: differences and relationships between formative and summative assessment," Assessment in Education, 4(3), pp.365-379, 1997.

[5] Assessment Handbook, University of Ulster, 2012.

[6] D.R. Sadler, "Formative assessment and the design of instructional systems, " Instructional science, 18(2), pp. 119-144, 1989.

[7] J. Biggs, "Aligning teaching and assessing to course objectives," Teaching and Learning in Higher Education: New Trends and Innovations, 2, pp.13-17, 2003.

[8] A. Kothiyal, S. Murthy, and S. Iyer, " Think-pair-share in a large CS1 class: does learning really happen?, " In Proceedings of the 2014 conference on Innovation & technology in computer science education, pp. 51-56, ACM, 2014.

[9] Porter, L., Bailey Lee, C., Simon, B., & Zingaro, D. (2011, August). Peer instruction: do students really learn from peer discussion in computing?. InProceedings of the seventh international workshop on Computing education research (pp. 45-52). ACM.

[10] http://webmconf.cdacmumbai.in/design/corporate_site/override/pdf-doc/Question_Banking.pdf, July 2014.

[11] LMS: http://docs.moodle.org/24/en/Question_bank, July 2014

[12] D. R. Krathwohl, " A revision of Bloom's taxonomy: An overview. Theory into practice," Vol. 4 41, pp. 212-218, 2002.

[13] M. D. Gall, "The use of questions in teaching. Review of educational research," 707-721, 1970.

[14] https://help.blackboard.com/en-us/Learn/9.1_SP_10_and_SP_11/Instructor/070_Tests_Surveys_Pools/018_Adding_Question_Metadata, July 2014.

[15] https://sakai.rutgers.edu/helpdocs/tests.html, July 2014.

[16] www.sumtotalsystems.com/enterprise/learning-management-system/ , July 2014.

[17] http://www.imsglobal.org/question/qtiv1p2/imsqti_oviewv1p2.htm, July 2014.

[18] A. C. Graesser and N. K. Person, "Question asking during tutoring," American educational research journal, 31(1), pp.104-137, 1994.

[19] G. Marbach-Ad and P.G. Sokolove, "Can undergraduate biology students learn to ask higher level questions?," Journal of Research in Science Teaching, Vol. 837, pp. 854-870, 2000.

[20] Natalya F. Noy and Deborah L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford University, 2001.

[21] P. Denny, A. Luxton-Reilly and B. Simon, "Quality of student contributed questions using PeerWise," In Proceedings of the Eleventh Australasian Conference on Computing Education, Volume 95, pp. 55-63, Australian Computer Society, Inc, 2009.

[22] J. Brooke, "SUS-A quick and dirty usability scale. Usability evaluation in industry, " 189, 194.

[23] S. Mishra and S. Iyer, "Problem Posing Exercises (PPE): An instructional strategy for learning of complex material in introductory programming courses," In IEEE Fifth International Conference on Technology for Education (T4E), pp. 151-158, IEEE, 2013.