Unsupervised concept tagging of mathematical questions from student explanations

Abstract. Assigning concept tags to questions enables Intelligent tutoring systems (ITS) to efficiently organize resources, help identify students' strengths and weaknesses, and recommend suitable learning materials accordingly. Manual tagging is time-consuming, and inefficient for large question banks, and could lead to consistency issues due to differences in the perspectives of individual taggers. Automatic tagging techniques can efficiently generate consistent tags at lower costs. Generating automatic tags for mathematical questions is challenging as the question text is usually short and concise, and the question as well as the answer text contains mathematical symbols and formulas. However, prior works have not studied this problem extensively. In this context, we conducted a study in a graduate-level linear algebra course to understand if student explanations to solving mathematical problems can be employed to generate concept tags associated with those questions. In this paper, we propose a method called Unsupervised Skill Tagging (UST) to extract concept tags associated with a given assessment item from explanation text. Using UST on the explanations generated, we show that the explanations indeed contain the expert-specified concept tags.

Keywords: question-tagging · text analysis · unsupervised concept-labeling

1 Introduction

Educational technology systems such as intelligent tutoring systems (ITS) consist of a domain model that stores the relations or dependencies between various exercises and topics. Here, the questions that are used to evaluate the student knowledge level, also referred to assessment items, are mapped with the concepts/skills that are required to solve them. This enables organization of content wherein questions associated with the same set of concepts can be grouped together. Assigning concept tags to questions also helps to track the learner's progress and build a profile of strengths and weaknesses. Concept tagging enables the ITS to assess the students' knowledge level which helps in the recommendation of suitable exercises/learning resources for the students. Therefore generating the correct tags for questions is important as incorrect tagging could adversely affect the performance of automatic tutoring and assessment systems.

Manual generation of question tags is tedious and costly in terms of the time involved in tagging and the labour costs incurred. Manual tagging could also lead to consistency issues due to the difference in perspectives of individual taggers with respect to a question. Automatic tagging procedures can generate consistent concept tags faster at a lower cost. Several techniques to generate automatic question tags for language related educational content have been proposed in literature [8]. Automatic tagging of mathematical questions has not been studied extensively when compared to automatic tagging of language questions. Generating tags for mathematical questions is demanding as the question text is mostly succinct and not rich enough to support tagging. The answers to such questions are mostly numerical values, which do not add any value to tag generation.

In this context, we study if student explanations on solving a problem could be utilized for unsupervised automatic tag generation. The key idea in this paper is that, when asked explicitly to explain their response to a given question, students who have the right approach to solving a problem would mention the right concepts related to the question in their explanation. We tested our idea in a graduate-level linear algebra course where we asked the students to explain their problem-solving strategy for a selected few questions in their weekly quizzes. It was observed that the most frequent unigrams and bigrams in the processed explanation text provided by students had good matching with the important concepts used in solving the problem. We then developed a technique called *Unsupervised Skill Tagging (UST)* to tag questions with labels from a master concept list based on the most frequent n-grams in the set of explanations for each problem. UST demonstrates good performance on our dataset comprising the student explanations which reinforces the idea that student explanations indeed describe the right concepts required to solve a problem, which is exploited by *UST* to generate the correct question tags.

The rest of the paper is organized as follows: Section II discusses some of the related work on existing question tagging methods for educational content. Section III describes the experiment setup used to collect student explanations. In Section IV, we present *UST*, our unsupervised approach to match questions with concept labels by processing student explanations on solving the item. Section V describes the results obtained when UST was applied on a test dataset collected during our experiment. In the final section, we summarize our work and discuss the scope of future work.

2 Related Work

In this section we describe the prior research work on automatic generation of question tags associated with educational content.

B. Sun et. al. [8] proposed a position-based attention model and keywords-based model to automatically associate questions with tags from a knowledge map. Their work has shown that employing words of answers with the question text boosts the performance of automatic tagging methods for multiple choice questions (MCQs) belonging to English course. Wankerl et. al [9] proposed a method to perform tagging of mathematical questions solely based on the textual and visual representations of mathematical formula provided along with the question. Both the proposed techniques are supervised learning methods that require a large amount of labeled data for training.

A semi-automatic method to perform tagging was proposed by Ramesh et. al. [6] to tag questions with metadata corresponding to cognitive level, question type, content and difficulty level. The tags were generated using techniques such as n-grams keyword matching, semantic dictionary and domain ontology.

Next we describe the studies that propose techniques to generate knowledge components (KC) for mathematical and English writing problems using crowdsourcing. Knowledge components represent the skills or knowledge needed to solve a problem. Moore et al.[4] studied whether knowledge components (KCs) identified by domain experts could be generated through crowd-sourcing. Participants in this study were given a problem in Math or English writing and were asked to concisely list three skills required to solve the problem at hand. For each problem, three KCs were identified by a domain expert. The participants' responses were coded, using a codebook developed manually from the participants' responses, and compared to the KCs identified by the expert. It was found that roughly 33% of the crowdsourced KCs directly matched those generated by domain experts for the same problem. While the participants for this study were selected through the Amazon Mechanical Turk (AMT) platform, a similar study was conducted with learners from two online courses in chemistry and programming [5]. In this case, it was found that half of the crowdsourced KCs matched expert-generated KCs in each problem. Both these works involved a substantial amount of human intervention to code the participant responses and perform the analysis.

The efficacy of using topic modeling approaches for identifying relevant KCs from student explanations has also been studied[3]. It was observed that interpreting the topics as KCs was difficult. Moreover, none of the topics were found to be directly related to any of the expert generated KCs.

Shen et al. [7] studied labelling educational content with proper KC labels using task adaptive pre-trained BERT model by considering three types of input, including KC descriptions, instructional video titles, and problem text. Recently Matsuda et al. [2] proposed a method to mine latent skills from the instructional text of assessment items and tag the discovered skills with human-friendly labels.

3 Experiment Setup

In this section, we describe the experimental setup used to collect data from the students. The study was conducted during the Linear Algebra for Engineers course in the Fall 2020 semester. The course was offered in online mode due to

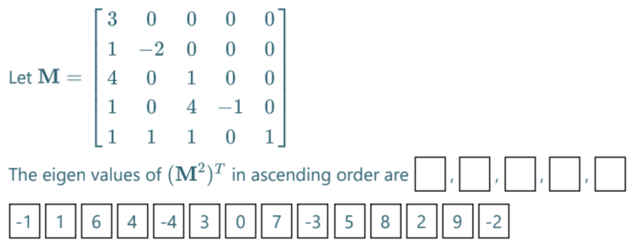


Fig. 1: A question asked in one of the weekly quizzes

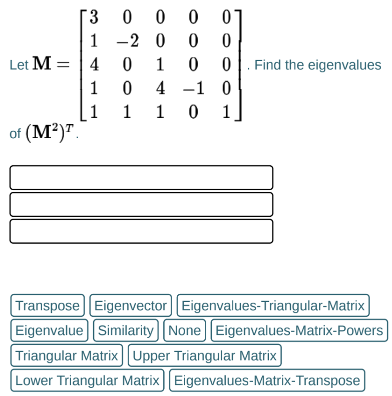


Fig. 2: A drag-and-drop concept label exercise used in one of the secondary quizzes

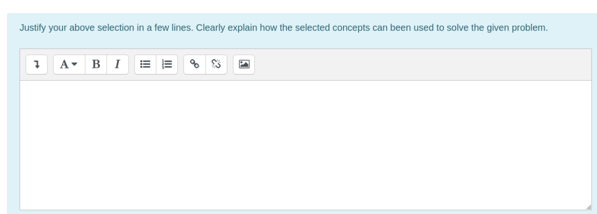


Fig. 3: A sample question for the student to explain how they solved the quiz question

the pandemic and was credited by 30 masters students. As a part of continuous evaluation, five weekly/regular quizzes $Q_1, Q_2, ..., Q_5$ were held through Moodle. Each quiz consisted of about 5-7 questions based on the topics covered in each week. The quizzes consisted of a mix of multiple-choice and numerical questions that were auto-evaluated. An example question has been provided in Figure 1.

orthonormal/orthogonal matrix, trace, transpose, frobenius norm, orthogonal projection, rank, range space, idempotent matrix, dimension, symmetry of inner product, linearity of inner product, positive definiteness of inner product, subspace addition, subspace intersection, linear independence, outer product, rank nullity theorem, null space, linear map composition, nullity, eigenvalues triangular matrix, eigenvalues matrix powers, eigenvalues matrix transpose, eigenvalue, eigenvector, invertibility, function space, derivative, inner product, linear transformation, nilpotent matrix, span

Table 1: Master list of concept labels

Each of the regular quiz $Q_1, Q_2, ..., Q_5$ were accompanied by a supplementary quiz $SQ_1, SQ_2, ..., SQ_5$. In each supplementary quiz SQ_i, the students were asked to perform additional activities for two selected questions included in Q_i.

For each question selected from the regular quiz, two activities were included in the supplementary quiz.

1. Manual Tagging using a drag-and-drop activity where the student selects the labels of the concepts which they think are required to solve the given problem from a given list of concept labels (Figure 2)

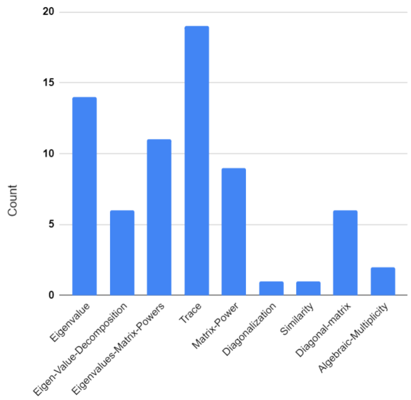
2. Justification of the choices they made in the earlier part by explaining how they solved the problem (Figure 3)

Separate time and marks were allocated for supplementary quiz to motivate the students to attempt the quiz earnestly. The responses to these questions were evaluated by the teaching assistant for the course.

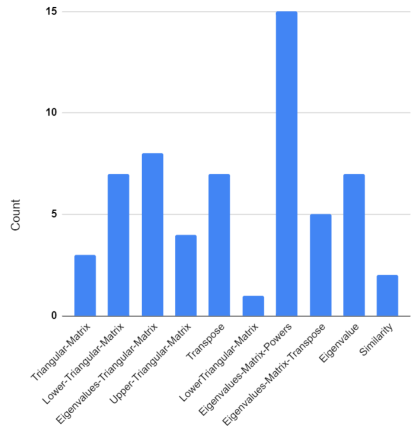
Student responses for a total of 10 questions were collected in this manner. The concepts required to solve each of the regular quiz questions included in the supplementary quiz were identified by the course instructor. These are referred to as the *expert-defined concept labels*, and these expert labels over all the questions is called the master list given in Table 1. The drag-and-drop activity was scored based on the level of agreement of the labels selected by the student with those decided by the instructor.

4 Unsupervised question tagging based on student explanations

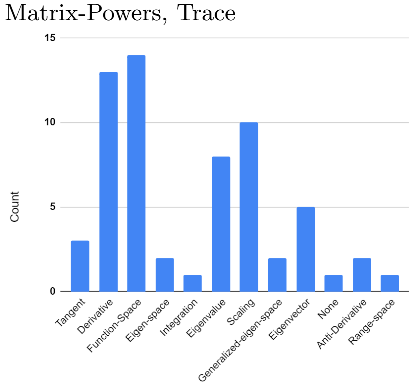
In this section, we first describe the performance of the manual tagging performed by students using the drag-and-drop activity and discuss why it doesn't always work well in practice. Next, we present *UST*, an unsupervised algorithm



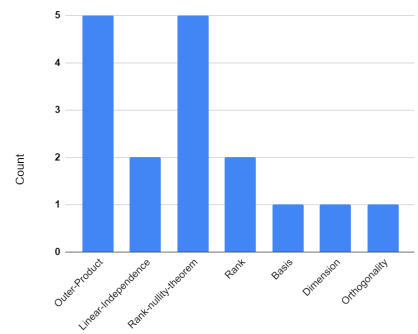
(a) Q1: Eigenvalue, Eigenvalues-Matrix-Powers, Trace



(b) Q2: Eigenvalues-Triangular-Matrix, Eigenvalues-Matrix-Powers, Eigenvalues-Matrix-Transpose



(c) Q3: Eigenvalue, Eigenvector, Function-space, Derivative



(d) Q4: Outer-Product, Rank, Dimension, Rank-nullity-theorem

Fig. 4: Bar-chart of concept labels selected by students who correctly solved the problem

to generate concept tags for questions based on student explanations of their problem-solving strategy.

4.1 Manual tagging based on drag-and-drop activity

Based on the student responses to the drag-and-drop activity, we attempted to do manual tagging by assigning the most frequently chosen concept labels to the question. Figure 4 shows four instances where this technique does not work well. Each bar chart displays the number of selections of each concept tag in the drag-and-drop activity for the questions in Table 2 by students who answered the question correctly in the regular quiz. The expert-defined concept labels for

Q1	If the eigen values of $\mathbf{M}$ are $-2, 0, 1, 1, 3, 4$, then the trace of $(3\mathbf{I} - 2\mathbf{M})^2$ is
Q2	Let $\mathbf{M} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & 0 \\ 4 & 0 & 1 & 0 & 0 \\ 1 & 0 & 4 & -1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$ The eigen values of $(\mathbf{M}^2)^T$ in ascending order are $\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}$. $\boxed{-1}\ \boxed{1}\ \boxed{6}\ \boxed{4}\ \boxed{-4}\ \boxed{3}\ \boxed{0}\ \boxed{7}\ \boxed{-3}\ \boxed{5}\ \boxed{8}\ \boxed{2}\ \boxed{9}\ \boxed{-2}$
Q3	The differentiation of $f(x)$ in function space w.r.t x is a linear transformation in x, which of the following is an eigen function of this linear transformation?
Q4	The nullity of the outer product between the normal of the hyperplanes x+y-z=0 and 2x-y-3z=0 in $\mathbb{C}^3$ is

Table 2: Questions associated with bar plots in Figure 4

each question are specified in the caption of each bar chart.

It could be observed from the plots that except for one or two expert-defined concepts none of the other tags received a strong mandate that distinguishes them from the non-relevant labels. Note that during the drag-and-drop activity, the students were presented with around 10 concept tags to choose from, which also contained the expert-defined labels. Therefore the activity was easier as compared to an actual tagging procedure where the tags are to be selected from the list of all available concept labels. This demonstrates the flaw of manual tagging where multiple taggers may not have a consensus regarding the concept-tags associated with a question. This surprisingly poor performance might be of interest of future study from a behavioral/cognitive perspective.

4.2 Our method: Unsupervised Skill Tagging (UST)

It has been observed that manual tagging doesn't always work well for mathematical questions. Automatic tagging methods that employ question text and words of answers to generate tags might not work well in the case of mathematical questions as: (i) the questions for mathematical problems are usually concise and contains various mathematical symbols and formula which makes tagging difficult. For instance, the concepts required to solve the question provided in Figure 1 depend on the properties of the matrix M, i.e., whether it is triangular, orthogonal, etc., which cannot be inferred from the question text. (ii) The answers to such questions are mostly numerical values, which do not add any value to tag generation.

Therefore, we study if student explanations on solving a problem could be used to generate tags for mathematical questions. Table 3 shows explanations provided by four students on solving a given problem. Unlike essays wherein words are linked by underlying syntactic/semantic structures, explanations to

Let $\mathbf{A}$ be the matrix for the orthogonal projection of vectors in $\mathbb{R}^3$ onto the line spanned by the vector [-1 0 1]. Then $rank(\mathbf{A}^7)$ =
It is projecting R^3 to a line, so rank (A)=1. And the projection is orthogonal. which means it is just a rotation, not changing magnitude of vector. so the vector in the line will be projected there itself. $rank(A^7)$=1
A will be a idempotent matrix and have full rank since A will be identity
The projection matrix is idempotent $A^2 = A$, Hence $A^7 = A$. The rank of the projection matrix is the same as that of input space, in our case, the line spanned by [-1 0 1] Projection does not change the span of the input space, and the matrix is idempotent.
Projection matrices are idempotent. Here the matrix is projecting onto a vector space basis having dimension 2. Hence rank of A will be 2 which is the dimension of its range space. Since projection matrices are idempotent $A^7 = A$ itself and hence $rank(A^7) = rank(A) = 2$.

Table 3: Student explanations on solving a problem

math problems are linked via the underlying procedural relationships between high-level concepts. Language models such as Bidirectional Encoder Representations from Transformers (BERT) are of little help in this task as such models are trained for language tasks and they learn the contextual relations between words in a text, which could be used for grading essays, short answers, etc.

In this section, we describe our proposed method called *Unsupervised Skill Tagging* (UST), an unsupervised technique for tagging mathematical questions by linking them with an existing list of concept labels, referred to as the *master tag-list*. It is based on the idea that explanations by students with the right problem-solving approach contain the correct concepts needed to solve the problem. The steps of the algorithm are detailed below:

1. Pre-processing: The student explanation text are processed by first performing the basic text preprocessing steps such as removing punctuation, extra spaces and digits, conversion to lower case and removal of stop words. Tokenization and lemmatization are then performed.

2. Extracting the most frequent unigrams and bigrams: For each question, the most frequent unigrams and bigrams from all the student explanations were extracted. The unigrams and bigrams with a frequency of occurrence less than a *threshold* were filtered out to form the *frequent n-gram list*. The best value of the *threshold* parameter is determined during the training phase.

3. Matching the frequent n-grams with concept labels in the *master tag-list*: A string $s1$ is said to have an *exact match* in another string $s2$ if the words in $s1$ are present as exact continuous words in $s2$. For instance, 'norm' doesn't have an exact match in 'orthonormal' whereas 'nullity' has an exact match in 'rank nullity theorem'.

$$contains(cLabel, fGram) = \begin{cases} True & \text{if there is an exact match of } fGram \text{ in } cLabel \\ False & \text{otherwise} \end{cases} \tag{1}$$

A concept-label *cLabel* in the master tag-list is matched with a frequent n-gram $fGram$ if $fGram$ has an exact match in *cLabel* or if the Levenshtein distance between $fGram$ and *cLabel*, as given by $levD(cLabel, fGram)$, is less than the threshold γ_{dist}. Levenshtein distance between two strings is defined as the minimum number of single-character edits (insertions, deletions or substitutions) required to change one string into the other. For instance, the Levenshtein distance between 'test' and 'tent' is 1, while that between 'test' and 'ten' is 2.

$$match(fGram, cLabel) = \begin{cases} True & \text{if } contains(cLabel, fGram) \text{ is True or} \\ & \quad levD(cLabel, fGram) < \gamma_{dist} \\ False & \text{otherwise} \end{cases} \tag{2}$$

A scoring function is then defined to perform the matching, as follows:

$$score(fGram, cLabel) = \begin{cases} \frac{\text{numWords(fGram)}}{\text{numWords(cLabel)}} & \text{if } match(cLabel, fGram) \text{ is True} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$numWords(term)$ is the number of words in *term*. For each question, a score is computed for every tag in the *master tag-list* against each of the frequent n-grams using the scoring function. The tags in the master tag-list with a score greater than or equal to the γ_{score} are determined to be the tags associated with that question. The values of γ_{dist} and γ_{score} are determined during training.

5 Results and Discussion

UST is an unsupervised tagging method. However, we make a train-test split so as to choose the values of the thresholds used for selecting the frequent n-grams and the scoring function. Hence the dataset was randomly split into a training-test set with the training set consisting of 6 questions and the test set with 4 questions.

Let $\mathbf{M}$ be a rank-3 matrix such that $\mathcal{R}(\mathbf{M}) = \mathcal{N}(\mathbf{M})$, then $rank(\mathbf{M}^7)$= ____, and the dimension of the input and output vectorspaces of this linear map are ____ and ____.
Expert-defined: subspace addition, subspace intersection, dimension, linear independence **Generated by our method:** dimension, subspace addition, subspace intersection, span
Let $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ be the basis of a vectorspace, $\mathcal{S}_1 = span(\mathbf{v}_1 + \mathbf{v}_3, \mathbf{v}_4)$ and $\mathcal{S}_2 = span(\mathbf{v}_3, \mathbf{v}_2 + \mathbf{v}_4)$. The dimension of $\mathcal{S}_1 + \mathcal{S}_2$ is
Expert-defined: range space, null space, linear map composition, rank nullity theorem, rank, nullity **Generated by our method:** rank, range space, dimension, rank nullity theorem, null space, linear map composition, nullity

Table 4: The expert-defined concept labels along with the labels generated by our method for two questions in the test set

The text preprocessing for all the questions was performed using the NLTK library [1]. The master tag-list was constructed by combining the relevant tags associated with all the questions along with some additional linear-algebra concepts (Table 1). A total of 32 concept-tags were included in the master tag-list.

UST was used to generate concept-tags for each question by including explanations given by:

Setting 1: all students
Setting 2: students who scored non-zero marks (partially correct answers) for the question in regular quiz
Setting 3: students who scored full marks for the question in regular quiz

For each of these setting, the values of frequency threshold for frequent n-gram extraction, γ_{dist} and γ_{score} were selected using parameter tuning on the training set. The best parameter values discovered under all three settings were the same: frequency threshold for n-gram discovery as one-tenth of the total number of student explanations for that question, $\gamma_{dist} = 1$ and $\gamma_{score} = 1$.

UST was applied on the test dataset with the chosen parameters. The concept labels generated for two questions in the test set have been provided in Table 4. As each question is tagged with multiple concept labels, this is an instance of multi-label classification and therefore the precision, recall and F_1-score of our method were computed for the test set as follows:

	Setting 1	Setting 2	Setting 3
Precision	0.713	0.742	0.775
Recall	0.937	0.875	0.792
F1-score	0.810	0.802	0.783

Table 5: Evaluation metrics computed for the 3 cases

$$precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i \bigcap y_i'|}{|y_i'|} \tag{4}$$

$$recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i \bigcap y_i'|}{|y_i|} \tag{5}$$

$$F_1 score = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

n**:** number of questions
y_i**:** labels chosen by instructor for question i
y_i'**:** labels generated by our method for question i

Precision is the fraction of expert-defined concept labels among the labels retrieved by our method. *Recall* is the fraction of expert-defined concept labels retrieved by our method. F_1 *score* combines the precision and recall into a single metric by taking their harmonic mean.

The precision, recall and F_1 score for our method in each of the three cases have been provided in Table 5. It could be observed that the highest F_1 score is obtained when the explanations of all students were considered. This could be attributed to students making a mistake in solving the problem, despite having some idea about the right problem-solving strategy. Note that while precision improves from Setting 1 to 2 and 3, a drop in recall is simultaneously observed. This might be due to the students who correctly solved a problem listing mostly the right concept labels while missing out on a few other relevant ones. The students who have incorrectly solved a problem but with the right problem-solving strategy might still list out all the required concepts required, while the ones with an incorrect strategy would list the irrelevant concepts in their explanations. Therefore including these explanations causes UST to lose on precision but gain on recall.

6 Conclusion and Future Work

Question tags are utilized by educational technology systems to organize content as well as perform ability analysis of students that helps with intelligent question recommendation. It has been shown that including the words of answers along with the question text boosts the performance of automatic taggers. However for mathematical questions, the answers are mostly numerical values, that reveal no extra information to taggers. As such we conducted a study in a masters level linear algebra course to examine if students with the correct problem solving approach mentions the right concepts required to solve the problem, when asked to explain their problem solving strategy. Based on our observations, we have proposed an unsupervised method called unsupervised skill tagging (UST), to generate concept-tags associated with a question item by processing the student explanations. UST extracts frequent unigrams and bigrams from the text and uses a scoring function based on exact word matching and Levenshtein distance to perform matching with a master list of concept tags.

In spite of its simplicity, the scoring function demonstrates good performance on the test set. It would be interesting to study if the performance of UST could be enhanced with the help of a concept graph that models the relationships between the various concepts. A concept graph could be employed to refine the concept tags generated by eliminating more general tags such as 'vectors', 'span', etc. that form a common ancestor to several other concept labels. It would also be interesting to study the performance of UST on questions from other domains.

References

1. Loper, E., Bird, S.: Nltk: The natural language toolkit. CoRR **cs.CL/0205028** (2002)
2. Matsuda, N., Wood, J., Shrivastava, R., Shimmei, M., Bier, N., et al.: Latent skill mining and labeling from courseware content. Journal of Educational Data Mining **14**(2) (2022)
3. Moore, S., Nguyen, H.A., Stamper, J.: Evaluating crowdsourcing and topic modeling in generating knowledge components from explanations. In: International Conference on Artificial Intelligence in Education. pp. 398–410. Springer (2020)
4. Moore, S., Nguyen, H.A., Stamper, J.: Towards crowdsourcing the identification of knowledge components. In: Proceedings of the Seventh ACM Conference on Learning@ Scale. pp. 245–248 (2020)
5. Moore, S., Nguyen, H.A., Stamper, J.: Leveraging Students to Generate Skill Tags that Inform Learning Analytics pp. 791–798 (2022)
6. Ramesh, R., Mishra, S., Sasikumar, M., Iyer, S.: Semi-automatic generation of metadata for items in a question repository. In: 2014 IEEE Sixth International Conference on Technology for Education. pp. 222–228. IEEE (2014)
7. Shen, J.T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., McGrew, S., Lee, D.: Classifying math knowledge components via task-adaptive pre-trained bert. In: International Conference on Artificial Intelligence in Education. pp. 408–419. Springer (2021)
8. Sun, B., Zhu, Y., Xiao, Y., Xiao, R., Wei, Y.: Automatic question tagging with deep neural networks. IEEE Transactions on Learning Technologies **12**(1), 29–43 (2018)
9. Wankerl, S., Götz, G., Hotho, A.: f2tag—can tags be predicted using formulas? In: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 565–571. IEEE (2020)