The Impact of Question Generation Activities on Performance

Andrew Luxton-Reilly, Daniel Bertinshaw, Paul Denny, Beryl Plimmer, Robert Sheehan
The University of Auckland
Auckland, New Zealand
andrew|daniel|paul|beryl|robert-s@cs.auckland.ac.nz

ABSTRACT

Recent interest in student-centric pedagogies have resulted in the development of numerous tools that support student generated questions. Previous evaluations of such tools have reported strong correlations between student participation and exam performance, yet the level of student engagement with other learning activities in the course is a potential confounding factor. We show such correlations may be explained by other factors, and we undertake a deeper analysis that reveals evidence of the positive impact question-generation activities have on student performance.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education—*computer science education*

General Terms

Design, Human Factors

Keywords

StudySieve, question-generation, student generated, user generated content, question posing, free-response, contributing student pedagogy, constructive evaluation

1. INTRODUCTION

Increasingly, educators are exploring the use of pedagogies that involve student generated content [3]. In such pedagogies, technology plays a central role in facilitating the interaction between students [12]. It is therefore important that evaluation of techonology used in education goes beyond simple correlational studies to investigate *impact*.

One promising pedagogy requires students to generate exam-style questions and corresponding sample solutions. The questions are peer-reviewed and may be used by other students for drill-and-practice [15]. A number of purpose-built tools have been designed to support this approach [2, 21, 20, 19, 6, 16, 8, 13].

SIGCSE'12, February 29–March 3, 2012, Raleigh, North Carolina, USA.

Evaluations of these tools have shown positive correlations between activity and performance [2, 7, 5, 13, 19]. However, this effect could be an epiphenomenon, occurring because "good" students who work hard during a course are more involved in all course-related activities (e.g. attending lectures, participating in tutorials, studying from textbooks) than weaker students.

In this paper we show that correlations between question-generation activity and performance may be explained by other factors, but deeper analysis of question-generation data reveals evidence that these activities do have a positive impact on performance.

.

2. RELATED WORK

Formal studies under controlled conditions have shown question generation to be an effective learning activity that results in improved test performance [1, 9]. In a review of studies in which students were taught to generate questions, Rosenshine et al. [18] describe how the act of generating questions does not directly improve understanding, but instead requires students to engage in tasks (such as reflecting on their understanding, searching relevant texts and combining information) that help improve comprehension.

Studies showing that student generated questions were not universally effective [4, 14] led researchers to focus on the specific content of the student generated questions compared to the questions used for assessment.

A study by Frase and Schwartz [11] found that question-generation activity resulted in improved recall in subsequent tests, but the recall effects were localized to content that was related to the generated questions. A later study by Foos et al. [10] confirmed these findings. A significant improvement in exam performance was observed for students who generated questions, but *only on questions with the same topic as the questions they generated*. The authors conclude that "generating potential test questions while preparing for an examination is a very effective technique leading to the highest performance on the material targeted by those test questions" [10, pg. 575].

A number of online tools have been developed by Computer Science educators to support pedagogies involving questions, answers and evaluations contributed by students. However, the numerous intrinsic and extrinsic influences on learning make it difficult to evaluate the impact of such interventions on student learning [17].

Studies of QPPA [21], Concerto [13], PeerWise [7, 5], QSIA [2] and ExamNet [19] have attempted to address the

impact of the intervention on student performance by focusing on the relationship between student activity and performance. In studies involving Concerto [13], QSIA [2] and PeerWise [5], an attempt is made to control for prior ability. However, none of the studies compare the level of student engagement in question generation activities with *other* activities in the same course. In other words, they don't measure the level of engagement and ability of students in the context in which the tools are used.

In this paper we investigate the relationship between activity and performance by focusing on two research questions:

- RQ1. Is the correlation between activity and performance a suitable indicator of the effectiveness of a tool?
- RQ2. Do students who use an online tool to author questions on a given topic perform better on similar topics in an exam?

3. METHODOLOGY

StudySieve [16] is an online tool designed to support student generated free-response questions. Students in two undergraduate courses (CS111 and CS105) taught at The University of Auckland were required to use StudySieve to create questions and answers for a small percentage of their final grade. Archival coursework data and the logs of StudySieve use were analysed after the courses were complete. The courses involved in this study, and the analysis of the data are described below.

CS111 — Mastering Cyberspace: An Introduction to Practical Computing is a large service course (404 students) that provides non-majors with an introduction to computing concepts and their practical application. The coursework involves weekly laboratory sessions and a mid-semester test held in week 6. Students were required to author 1 question and answer 5 questions prior to the mid-semester test to receive 0.5% of their final grade.

CS105 — Principles of Computer Science provides an introduction to data structures and algorithms using Java. This course is normally taken by Computer Science majors after completing an introduction to programming. The course was taught in the summer semester 2011 to a class of 50 students.

Course	Students	Activity Requirements	% final
CS111	404	1 question, 5 answers	0.5%
CS105	50	5 questions, 5 answers	2.5%

Table 1: Summary of courses and StudySieve activity requirements.

3.1 Relationship between activity and performance

Students using StudySieve were required to submit a minimum number of questions and answers for a portion of their final mark, but there was no maximum imposed on activity. Some students contributed more questions and answers than required for their assessment. Students also provided feedback to their peers by contributing comments (which are not required nor assessed). In this study we use the number of questions, number of answers and number of comments as three different measures of activity. Since we are investigating the relationship between activity and exam performance, any student that did not sit the final exam was excluded from the study. Section 4.1 reports on simple correlations between activity measures and exam performance. Section 4.2 reports on correlations between activity and exam performance when controlling for prior knowledge, and Section 4.3 reports on correlations between StudySieve activity and performance when controlling for student ability within the same course.

3.2 Relationship between question authoring and performance in related exam questions

To investigate the relationship between authoring a question on a specific topic and exam scores for that topic, we first categorized the student generated questions according to the topics described in the course outline for each course. The question repository contained some student generated questions that were related to more than one topic, so it was not possible to code questions into mutually exclusive categories. For example, the following question submitted by a student in the CS105 course relates to both **big-O** and **binary trees**

> *What are the best case and worst case big-O running times for retrieving an element in a binary tree? Give an example of a sequence of items that would produce each of these big-O running times.*

The process used to classify and analyse the student generated questions are described below.

3.2.1 CS111

Students in the CS111 course generated 350 questions. Two authors categorized each question in the CS111 repository, working together to classify the first ten questions. After some initial discussion about coding boundaries, the remainder of the questions were coded individually. Since a single question may be related to multiple topics, the categorization was performed as a sequence of independent coding tasks. For each topic, every question was examined and coded as either relating to that topic or not. The inter-rater reliability was calculated for each coding task using Cohen's Kappa and substantial agreement was found for most topics (kappa ranged from .630–.966). In cases of coding disagreement, the questions were discussed until agreement was reached. Table 2 lists the topics and the number of student generated questions related to each topic.

Subsequently, we turned to the questions used in the mid-semester test and attempted to identify the student generated questions that were related to the instructor questions. During this process, it became apparant that the coding scheme used to classify student generated questions in CS111 was too coarse. The content covered in CS111 is broad and shallow, resulting in a lack of cohesive related knowledge within topics. For example: questions about Moore's Law and questions asking "What does RAM stand for?" are both classified as Hardware, but are not related questions (e.g. knowledge of Moore's law has no bearing on knowledge of hardware acronyms such as RAM).

Each question in the test was classified according to one of the topics, and all student generated questions belonging to that topic were examined to determine if they were closely related. This resulted in a small set of student generated

Topic	N
Bits and Bytes	48
Standards	16
Hardware	39
Software	22
Internet infrastructure	17
World-Wide Web	19
Email and Forums	17
Blogs and Wikis	19
Word processing	38
Digital images	10
PowerPoint	15
Acronyms	18
HTML/CSS	69
Other	1
Irrelevant	1

Table 2: Topics used to classify student generated questions in CS111

questions that were related to the material assessed in 18 of the 20 test questions. In two cases (a question about web caches and a question about file formats), there were no student generated questions that covered similar material.

For each of the test questions, we divided the students into those who *had* authored a related question (AQ), and those who *had not* authored a related question (NQ). Within each group (AQ and NQ) we calculated the percentage of students correctly answering the question. If the act of generating a question improves subsequent performance on related questions, as claimed by Foos et al. [10], then we would expect to see a higher percentage of students in the AQ group answering a given test question correctly compared with the NQ group. Section 4.4.1 reports the results of this analysis.

3.2.2 CS105

The topics covered in the CS105 course are conceptually more cohesive than those in CS111. student generated questions contributed to the CS105 repository (166 in total) were classified into topics identified from the course outline. To classify questions that involved multiple topics, each topic was considered independently, and questions were coded as either relating to the topic or being unrelated. The inter-rater reliability was calculated for each coding task using Cohen's Kappa and an acceptable level of agreement was found (kappa ranged between .652–.957). In cases of coding disagreement the questions were discussed until agreement was reached. Table 3 lists the topics and the number of student generated questions related to each topic.

The questions used in the final exam were also classified using the topics extracted from the course outline. In the CS105 course, the questions generated by students in each topic were all deemed to be relevant to the corresponding exam questions. All topics were assessed in the exam, except file reading/writing, sorting and ADTs which were assessed during a mid-semester test.

For each question in the final exam, we compared the results of the students who had generated a related question (AQ) with those students who had not generated a related question (NQ). Section 4.4.2 reports the results of this analysis.

Topics	N
Multi-dimensional arrays	5
File reading/writing	8
Exceptions	12
Sorting	28
Recursion	22
Big O	41
ADT	8
Lists	15
Stacks	18
Queues	5
Binary Tree	16
Binary Search Tree	12
Hash table	13
Other	7

Table 3: Number of questions related to each topic in the CS105 curriculum.

4. RESULTS

4.1 Relationship between Activity and Performance

As described in Section 3.1, three measures of activity are used to investigate the relationship between student use of StudySieve and performance on final exams: number of questions created, number of questions answered, and number of comments contributed. Since the distribution of the data is not normal, Spearman's Rho is used to calculate the correlation between exam score and each of number of questions, number of answers and number of comments. For each correlation, a two-tailed significance test was conducted. In almost all cases, a significant positive weak correlation was observed as shown in Table 4.

Activity measure	CS111 (N=371)		CS105 (N=45)	
	ρ	p	ρ	p
No. of questions	.27**	.000	.40*	.007
No. of answers	.31**	.000	.28	.064
No. of comments	.10*	.049	.38*	.011

* 0.05 significance level (2-tailed)
** 0.01 significance level (2-tailed)

Table 4: Correlations between activity and exam scores for the CS111 course (N=371) and the CS105 course (N=45).

The significant and positive correlations between activity and performance present in both courses are consistent with previous studies involving similar tools [21, 13, 7, 5, 2, 19].

4.2 Relationship between activity and performance, controlling for prior knowledge

Almost all students enrolled in the summer semester CS105 course have completed a year of study at The University of Auckland, so their grade point average (GPA) can be used as a surrogate measure of prior knowledge and cognitive ability (although the GPA was not available for 6 students). A partial correlation was calculated between activity and exam score, using GPA as a control. The results show that the number of comments contributed by students is signif-

icantly correlated with exam performance, even when controlling for prior ability. Table 5 summarizes the results of the partial correlation. This is consistent with findings from studies involving QSIA [2], ExamNet [19] and PeerWise [5].

Activity measure	r	p
Number of questions	.07	.663
Number of answers	.19	.244
Number of comments	.36*	.025

* 0.05 significance level

Table 5: Results of the partial correlation between activity and exam score when controlling for grade point average in the CS105 course (N=39).

4.3 Relationship between activity and performance, controlling for student ability

In CS105, students also completed a number of programming assignments. The total marks for assignments are strongly correlated with exam scores (ρ=.73, p<.001). We believe that the assignment marks provide an excellent measure of how "good" the student is (where "good" is interpreted as the combination of ability and effort over a sustained time period).

A partial correlation was performed between activity and exam score, using the assignment marks as a control for student ability with material relevant to the course. No significant correlations were found between activity and exam score once assignment marks were taken into account. Although the StudySieve activity is significantly correlated with exam performance, this correlation can be entirely explained by how well students did in the assignments. Table 6 summarizes the results from the partial correlation.

Activity measure	r	p
Number of questions	-.18	.243
Number of answers	-.17	.261
Number of comments	.06	.700

Table 6: Partial correlations between StudySieve activity and exam score in the CS105 course, controlling for total assignment marks (N=45)

Similarly in the CS111 course, correlations between activity and performance can be explained by controlling for engagement with other course-related activities. Students in CS111 attend weekly laboratories that focus on the practical application of theory learned in lectures. The total marks obtained from laboratories (i.e. the sum of weekly laboratory marks) is strongly correlated with exam score (ρ=.62, p<.001), and each of the StudySieve activity metrics: number of questions (ρ=.56, p<.001), number of answers (ρ=.55, p<.001) and number of comments (ρ=.27, p<.001).

A partial correlation was performed between activity and exam score, using the laboratory marks as a control for student ability with material relevant to the course. No significant correlations were found between activity and exam score once laboratory marks were taken into account. Table 7 summarizes the results from the partial correlation.

As with the CS105 course, the correlations between StudySieve activity and exam score are explained by using marks for other coursework activity as a surrogate measure of student ability in the context of a specific course.

Activity measure	r	p
Number of questions	-.020	.701
Number of answers	-.025	.635
Number of comments	-.047	.368

Table 7: Partial correlations between StudySieve activity and exam score in the CS111 course, controlling for total assignment marks (N=371)

4.4 Relationship between question authoring and exam performance in related topics

Although correlations between activity and exam score are easily explained by other metrics of student ability within a course, the research from educational psychology shows that question-generation activities generally improve exam performance on related questions [10]. In this section we focus on the impact of student question-generation activities on *related* test and exam questions.

4.4.1 CS111 questions

Since the CS111 course covers a broad range of unrelated topics, the number of student generated questions related to a given test question is reasonably small, and is not enough to perform statistical tests on individual topics. Of the 20 questions used in the mid-semester test, 2 of the questions had no related student generated questions. Students who authored a question on a related topic performed better than students who did not author a related question in 14 of the 18 remaining questions.

If the question-generation activity had no impact on exam performance, then we would expect the same percentage of students who *had* authored a related question (AQ) to select the correct answer compared with students who *had not* authored (NQ) a related question. The binomial test is used to determine the likelihood of the AQ group exceeding the NQ group in correctly answering test questions. The results indicate that the chances of the AQ group exceeding the NQ group in 14 of 18 cases due to random variation is small (p=0.012). We can conclude that the question-generation activity is likely to have had a positive effect on the exam performance of related questions.

4.4.2 CS105 questions

All the questions used in the CS105 exam were short-answer questions, requiring students to write an explanation, perform a calculation, or write a solution in the form of a Java method. Student performance on each exam question was analysed using a Mann-Whitney U test to determine if any significant difference existed between the marks of students who authored a related question compared with the marks of students who did not author a related question. In three of the exam questions — Big-O, Stacks and Hash tables — a significant difference was observed, but no difference was observed for the other question topics. This suggests that authoring a question about Big-O, Stacks or Hash tables resulted in improved exam performance on those topics. Table 9 summarizes the results.

One possible threat to the validity of this finding is that students may simply be writing questions about topics in

MCQ Topic in test	N	%AQ	%NQ	AQ>NQ
Binary numbers	28	.93	.81	✓
Binary prefixes	2	.50	.71	
Hardware components	22	.91	.89	✓
Moore's law	3	1.0	.68	✓
Software licenses	4	.50	.47	✓
TCP	4	1.0	.86	✓
Internet infrastructure	4	.25	.11	✓
Email protocols	9	.56	.53	✓
Email	4	.50	.48	✓
Blogs	6	.50	.63	
Wikis	8	.38	.51	
WWW history	6	.83	.77	✓
WWW infrastructure	0	NA	.79	NA
ASCII	11	.18	.43	
Standards	0	NA	.60	NA
Image size	6	1.0	.62	✓
PowerPoint criticism	6	.67	.46	✓
PowerPoint advice	7	.71	.59	✓
CGI	1	1.0	.26	✓
CGI	1	1.0	.43	✓

Table 8: Number of student generated questions in CS111 related to each mid-semester test question, and percentage of students answering each question correctly, divided into those who authored a related question (AQ) and those who did not author a related question (NQ).

which they are already knowledgeable. In this case, the difference between author and non-author groups is due to previous knowledge and has no direct impact on performance. Although this threat is impossible to reject entirely, students authored questions *after* they had completed an assignment on that topic, so it is possible to compare the related assignment marks to determine whether students chose to author questions on topics in which they were more knowledgeable than other students.

Assignment 3 had five questions related to Big-O performance analysis. We extracted the marks for these five questions and used a Mann-Whitney U test to compare the assignment marks of students who had generated a question on Big-O (μ=3.45) with the assignment marks of students who did not author a question on Big-O (μ=3.25). The results indicate that there is no significant difference between the groups (p=.321).

This suggests that students in both groups performed equally on questions involving Big-O prior to the use of StudySieve. The students who used StudySieve to generate a question on Big-O subsequently performed better on that topic during a final exam. Although this does not prove a causal relationship between question generation and performance increase in related exam questions, it provides some evidence of such an effect.

A similar analysis was performed with the Stacks topic, however the data is less rich. Assignment 4 had a single question related to stacks which was graded on a coarse scale (between 0 and 2 with increments of 0.5), and fewer students authored questions related to stacks. We used the marks from the relevant question in assignment 4 and applied a Mann-Whitney U test to compare the assignment marks of students who had generated a question on stacks (μ=1.57) with the assignment marks of students who did not author a question on stacks (μ=1.05). The results indicate that there is no significant difference between the groups (p=.081), although the p-value is much smaller than in the previous analysis.

Unfortunately, none of the assignment questions related to Hash Tables, so we have no variable to use as a control for that topic.

	(AQ)		(NQ)		
Topic	N	Mean	N	Mean	p
Multi-dim. arrays	5	4.80	40	5.15	.625
Exceptions	11	6.82	34	6.27	.645
Recursion	18	6.83	27	6.52	.780
Big-O	27	8.26	18	6.67	.021*
Lists	12	7.58	33	7.70	.744
Stacks	13	7.38	32	5.06	.031*
Queues	4	3.75	41	5.20	.593
Binary Tree	13	5.42	32	5.31	.679
Binary Search Tree	9	10.00	36	8.78	.097
Hash Table	12	17.00	33	11.38	.001**

* — significant at 0.05 level
** — significant at 0.01 level

Table 9: Comparison of the students who authored a question related to a given exam question (AQ) with students who did not author a question related to a given exam question (NQ) for the CS105 course.

5. DISCUSSION AND CONCLUSIONS

Although positive correlations between activity and exam performance can be explained by the level of student ability in a given course, we have demonstrated that the act of question-generation has a positive impact on performance in related exam questions.

It is difficult to evaluate the impact of interventions in real teaching contexts due to the myriad of factors, both internal and external, that impact on student performance. The correlational findings in this paper mirror those of previous studies with a variety of related tools, providing strong evidence that activity with these tools is reliably correlated with exam performance. Although correlational studies are a good starting point in the evaluation of teaching interventions, the evaluation should not end there.

Intuitively, we might expect that the best students in the class would be actively involved in any learning opportunities that are offered. Conversely, the weaker students are less likely to take advantage of opportunies to engage in the kind of self-directed study that these question-generation tools provide. The findings of this study support such intuitions. Correlations between activity and exam performance can be explained by the level of student ability within the context of a course.

Integrating results from controlled studies in educational psychology has proved fertile in this study. We have shown that findings from studies of question-generation activity conducted under controlled conditions in psychology are transferrable to the domain of Computer Science, using online tool support. The act of question-generation has a positive impact on performance in related exam questions.

That some of the topics in the CS105 course show positive effects of question-generation activity while others do not warrants further investigation. While question-generation has been shown to improve performance on related exam questions, it is unclear exactly how similar the questions must be.

The vast majority of questions generated by students in the CS105 course tended to focus on concepts, or understanding code by tracing. Few questions asked students to write code for a solution. On the other hand, several of the actual exam questions required students to write fragments of code. This was the case for the exam questions that covered the topics of multi-dimensional arrays, lists, queues and binary trees. It may be that the questions authored by students on these topics, which represent 4 of the 7 topics listed in Table 9 for which a significant result was not found, were different enough in style that we do not see any measurable effects.

Both courses in this study placed fairly modest requirements on students in terms of their participation. Students were asked to author just 1 question in the CS111 course, and 5 in the CS105 course. As a result, most students did not produce many questions on any specific topic. Even in CS105, few students wrote more than one or two questions on the same topic. It may be that greater levels of authorship will produce more significant results, and this is something that we plan to investigate in the future.

We hope as a result of this study, education researchers involved in evaluations of technology push beyond correlational studies to focus on the ways in which interventions impact on students.

6. REFERENCES

[1] M. E. D. A. André and T. H. Anderson. The development and evaluation of a self-questioning study technique. *Reading Research Quarterly*, 14(4):605–623, 1979.

[2] M. Barak and S. Rafaeli. On-line question-posing and peer-assessment as means for web-based knowledge sharing in learning. *International Journal of Human-Computer Studies*, 61:84–103, 2004.

[3] B. Collis and J. Moonen. *An on-going journey: Technology as a learning workbench*. University of Twente, Enschede, The Netherlands., 2005.

[4] P. R. Denner and J. P. Rickards. A developmental comparison of the effects of provided and generated questions on text recall. *Contemporary Educational Psychology*, 12(2):135 – 146, 1987.

[5] P. Denny, B. Hanks, and B. Simon. Peerwise: replication study of a student-collaborative self-testing web service in a U.S. setting. In *SIGCSE '10: Proceedings of the 41st ACM technical symposium on Computer science education*, pages 421–425, New York, NY, USA, 2010. ACM.

[6] P. Denny, A. Luxton-Reilly, and J. Hamer. Student use of the PeerWise system. In *ITiCSE '08: Proceedings of the 13th annual SIGCSE conference on Innovation and Technology in Computer Science Education*, pages 73–77, Madrid, Spain, 2008. ACM.

[7] P. Denny, A. Luxton-Reilly, and B. Simon. Evaluating a new exam question: Parsons problems. In *ICER '08: Proceeding of the fourth international workshop on Computing education research*, pages 113–124, New York, NY, USA, 2008. ACM.

[8] P. Denny, A. Luxton-Reilly, E. Tempero, and J. Hendrickx. Codewrite: supporting student-driven practice of java. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, pages 471–476, New York, NY, USA, 2011. ACM.

[9] P. W. Foos. Effects of student-written questions on student test performance. *Teaching of Psychology*, 16(2):77–78, 1989.

[10] P. W. Foos, J. J. Mora, and S. Tkacz. Student study techniques and the generation effect. *Journal of Educational Psychology*, 86(4):567–576, Dec 1994.

[11] L. T. Frase and B. J. Schwartz. Effect of question production and answering on prose recall. *Journal of Educational Psychology*, 67(5):628–635, 1975.

[12] J. Hamer, H. C. Purchase, A. Luxton-Reilly, and J. Sheard. Tools for "contributing student learning". In *Proceedings of the 2010 ITiCSE working group reports on Working group reports*, ITiCSE-WGR '10, pages 1–14, New York, NY, USA, 2010. ACM.

[13] Y. Hirai and A. Hazeyama. A learning support system based on question-posing and its evaluation. *Creating, Connecting and Collaborating through Computing, International Conference on*, 0:178–184, 2007.

[14] J. R. Lehman and K. M. Lehman. The relative effects of experimenter and subject generated questions on learning from museum case exhibits. *Journal of Research in Science Teaching*, 21(9):931–935, Dec. 1984.

[15] A. Luxton-Reilly and P. Denny. Constructive evaluation: a pedagogy of student-contributed assessment. *Computer Science Education*, 20:145–167, 2010.

[16] A. Luxton-Reilly, P. Denny, B. Plimmer, and D. Bertinshaw. Supporting student-generated free-response questions. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, ITiCSE '11, pages 153–157, New York, NY, USA, 2011. ACM.

[17] T. C. Reeves. Design research from a technology perspective. In J. van den Akker, K. Gravemeijer, S. McKenney, and N. Nieveen, editors, *Education Design Research*, pages 52–66. Routledge, 2006.

[18] B. Rosenshine, C. Meister, and S. Chapman. Teaching students to generate questions: A review of the intervention studies. *Review of Educational Research*, 66(2):181–221, Summer 1996.

[19] E. V. Wilson. Examnet asynchronous learning network: augmenting face-to-face courses with student-developed exam questions. *Computers & Education*, 42(1):87 – 107, 2004.

[20] F.-Y. Yu. Scaffolding student-generated questions: Design and development of a customizable online learning system. *Computers in Human Behaviour*, 25:1129–1138, 2009.

[21] F.-Y. Yu, Y.-H. Liu, and T.-W. Chan. A web-based learning system for question-posing and peer assessment. *Innovations in Education and Teaching International*, 42(4):337–348, 2005.