Quality of Peer Assessment in CS1

John Hamer
J.Hamer@cs.auckland.ac.nz
Department of Computer Science
University of Auckland
Auckland, New Zealand

Helen Purchase
hcp@dcs.gla.ac.uk
Department of Computing Science
University of Glasgow
Glasgow, Scotland

Paul Denny and Andrew Luxton-Reilly
{paul,andrew}@cs.auckland.ac.nz
Department of Computer Science
University of Auckland
Auckland, New Zealand

ABSTRACT

While popularity of peer assessment in Computer Science has increased in recent years, the validity of peer assessed marks remain a significant concern to instructors and source of anxiety to students. We report here on a large-scale study (1,500 students and 10,000 reviews) involving three introductory programming classes which recorded grades and feedback comments for both student and tutor reviews of novice programs. Using a paired analysis, we compare the quantitative marks given by students with those given by tutors, for both functional and non-functional aspects of the program. We also report on an analysis of the lexical sophistication of feedback comments.

We find good correlations that improve with student ability and experience, and that marks for functional aspects correlate more closely than those for non-functional aspects. Our lexical sophistication analysis suggests student feedback can be as good as or better than tutor feedback. We also observe that a policy of selecting tutors based on their previous peer assessment performance leads to a large improvement in tutor feedback.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer Science education

General Terms

Human Factors

Keywords

CS1, Peer assessment

ICER'09, August 10--11, 2009, Berkeley, California, USA.

1. INTRODUCTION

The use of peer assessment in Computer Science courses is becoming increasingly popular (see, for example, [18, 7, 5, 10]). Proponents have identified a number of benefits, including: deepening understanding; highlighting the importance of presenting work in a clear and logical fashion; promoting social and professional skills; exposing students to a variety of styles, techniques, ideas and abilities; improving understanding and self-confidence; and encouraging reflection on course objectives [2].

Despite offering a range of learning benefits, mark reliability remains a common concern with instructors new to the process, and a source of anxiety for many students.

An opportunity to study the quality of student reviews in introductory Computer Science courses arose with the Aropä project. Aropä is a web-based system designed to support routine peer assessment activities in large classes [9], and has been used in a wide range of departments and courses at The University of Auckland since 2002. Of particular interest are two introductory programming courses, one for Engineering students (learning MATLAB and C) and the other for Computer Science students (learning Java). In 2007 and 2008, these courses ran assignments that were marked by both peer reviewers and tutors. From these, we collected 10,335 separate reviews from five assignments for which both student and tutor reviews were available.

The typical use of Aropä does not involve any tutor marking: the student's assignment grade is determined from a weighted average of the marks given by the reviewers (see [12] for information on how we compute final grades from a set of reviewers' grades). This averaging process has proved effective at mitigating the influence of a small number of "rogue" reviewers. However, our focus in this paper is on the quality of student reviews in their own right, before any post-processing is done.

In considering the quality of the reviews provided by the student reviewers, we look at two aspects of the review quality: the appropriateness of the quantitative marks awarded, and the lexical sophistication of the qualitative comments given. Our research questions are:

- How competent are the student reviewers at making appropriate assessments of their peers' assignments?
- How sophisticated are the reviewers' textual comments?

2. INSTRUCTIONAL CONTEXT

We use the following terminology. Students enrol in a *course*, and a combination of course and year is a *class*. In a small number of cases, a failing student will repeat a class the following year; we treat such cases as if they were distinct students. An *assignment* is a coursework activity given to a class. Typically, three or four assignments are set, some or all of which will be peer assessed. A *submission* is a piece of work written by an author (usually a student) and marked by at least one reviewer. An *allocation* is a combination of assignment, reviewer and author. Reviews are entered on a structured *rubric*, which contains a sections for both quantitative grades and text input fields for comments. A sample rubric is shown in the Appendix. For assignments that also marked by a tutor, the tutor provides a separate set of marks and feedback to the author. Where the context allows, we use *reviewers* to refer solely to student reviewers, rather than tutors.

Each submission is marked only once by a tutor, but typically reviewed up to four times by students as they are allocated multiple submissions. For some assignments, a submission written by the instructor may be allocated to all student reviewers.

The classes included in this study are:

- "Introduction to Engineering Computation and Software Development" 2007, assignments 1 and 2 (EG07a and EG07b)
- "Principles of Programming" 2007, assignment 1 (CS07)
- "Introduction to Engineering Computation and Software Development" 2008, assignments 1 and 2 (EG08a and EG08b)

For EG08a, three submissions were allocated to each student reviewer. EG07a and EG07b allocated four student submissions and also included one instructor submission. CS07 and EG08b allocated four student submissions to each reviewer.

The grading rubrics varied for each exercise, but each followed the same structure: a section on program correctness (typically comparing program output to sample output over several test cases); and a section on programming style (choice of identifiers, code indentation, etc.).

3. RELATED WORK

In [11] we noted markedly different attitudes and perceptions toward peer assessment from students in different departments using Aropä. We therefore expect any results in evaluating student reviewing performance to be contextual, and to vary (at least) with student experience, subject matter, and educational context.

This assumption is supported by the available literature on peer assessment reliability. For example, a tendency for students to under-mark was observed by Penny and Grover [13, p387]. Stefani [20] also found student grades to be slightly lower than tutors. On the other hand, Marcoulides and Simkin [14] found that their students graded accurately and consistently. Boud and Holmes [3] concluded that peer review was "as reliable as normal methods of marking" albeit with a slight bias to over-mark. Haaga [8] found student reviews of manuscripts to be more reliable than academic reviews.

A meta-analysis comparing peer marks with teacher assigned marks conducted by Falchikov and Goldfinch [6] concluded that peer marks tend to agree well with teacher marks overall, although they note that peer assessments that require marking of several dimensions appear to be less valid than those that require a single global judgement based on well understood criteria.

Chalk and Adeboye [4] found that a rubric requiring holistic judgements of quality elicited greater agreement between peers and teachers in a small introductory computing course than a more detailed rubric that focused on specific criteria. However, Miller [16] found that more specific, detailed rubrics provided better differentiation of performance. Sitthiworachart and Joy [19] found high correlations between tutors' and students' marks for objective criteria, but lower correlations between tutor and student marks for subjective criteria in a large CS1 course.

Miller [16] observed that more holistic rubrics that provided more opportunities to comment elicited a greater number of qualitative responses. Sitthiworachart and Joy [19] found that tutors tended to write more comments on program correctness while students tended to write more on program style.

We note that the data used in this paper is considerably more extensive than any of the studies cited above, as it includes more than 1,500 students and over 10,000 reviews.

The results in this paper apply to courses in two separate departments in the same institution, dealing with conventionally taught introductory programming material, taken over two years. Any extrapolation beyond these parameters should be done judiciously.

4. REVIEWERS' MARKS

The first question we address is: *How competent are the student reviewers at making appropriate assessments of their peers' assignments?* To do this, we compare the marks given by the reviewers with those given by tutors.

All the assignments required that students submitted a program: in MATLAB or C for the Engineering students, and in Java for Computer Science. The marking rubric for each assignment included both questions that required a numerical mark and between two and seven textual comment questions. Appendix A shows the rubric used for the Computer Science course. The other rubrics follow a similar form.

4.1 Comparing reviewers' total marks with tutors' total marks

Our first analysis compares the total marks given by the reviewers for each assignment with the total marks given by tutors to the same assignment, by calculating the correlation co-efficients over all reviewers (see Table 1). This allows us to see the extent to which the reviewers are making similar assessment judgements to the tutors.

Any assessments that are missing one or more mark for any of the questions have been ignored in these calculations. All p values are < 0.001

Figure 1 shows a scatterplot of the tutor and peer marks for all assignments, showing a strongly linear relationship ($R = 0.712$).

We also look at the difference between the actual percentage marks given by reviewers and tutors, and test for significant differences between them (Table 2 and Figure 2).

	EG07a	EG07b	CS07	EG08a	EG08b	All
Corr.	0.524	0.723	0.780	0.579	0.771	0.712
N	2543	2634	1627	1563	1968	10335

Table 1: Correlations of total marks between reviewers and tutors.

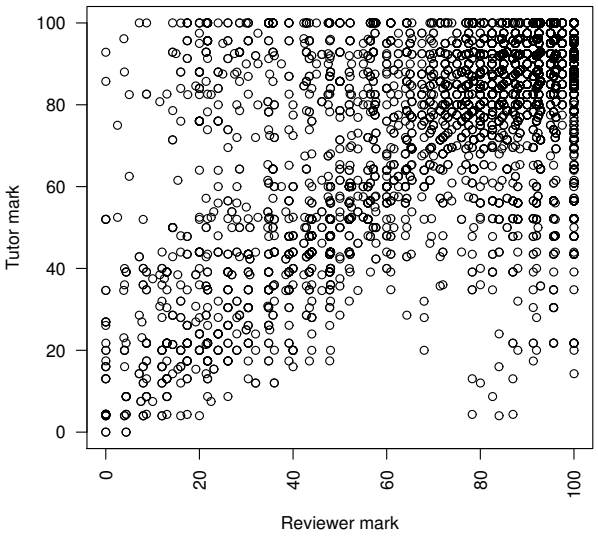

Figure 1: Scatterplot of tutor v. reviewer marks over all assignments.

	EG07a	EG07b	CS07	EG08a	EG08b	All
Peer	**12.5**	19.1	**22.9**	**34.2**	19.4	**84.7**
Tutor	**12.6**	19.1	**23.2**	**35.6**	19.6	**85.8**
out of	14	23	26	40	25	(100)
Δ	1.1%	0%	1.2%	3.5%	0%	1.1%
p	ϵ	0.50	ϵ	ϵ	0.10	ϵ

Table 2: Comparison of total marks between reviewers and tutors. The difference, Δ, is the tutor mean less the reviewer mean divided by the maximum marks (out of). Significance is calculated using one-sample t-tests of the pairwise differences, and shown in bold when not zero ($p < 0.05$).

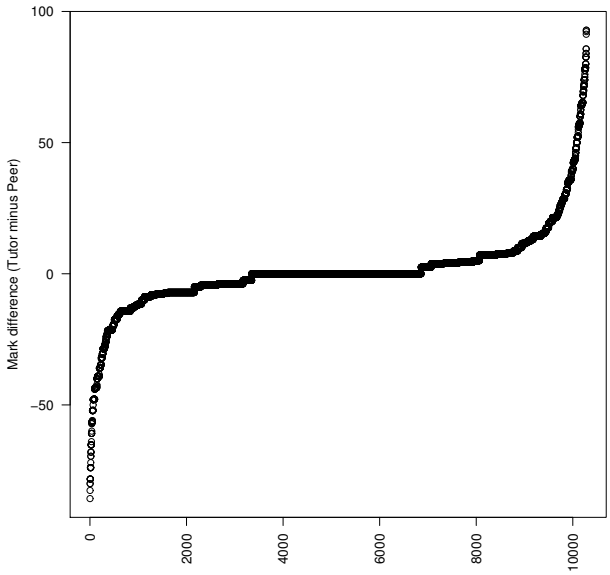

Figure 2: Mark difference over all reviews. The differences can be quite extreme (with cases of zero marks being given by reviewers for submissions to which the tutor awarded full marks), but such "rogue" reviews comprise a small proportion (89% of the reviews differ by less than 20%).

	EG07a	EG07b	CS07	EG08a	EG08b	All
Style						
Peer	**3.48**	**6.19**	**10.2**	4.49	**5.16**	**85.0**
Tutor	**3.55**	**6.32**	**10.4**	4.55	**5.23**	**86.5**
out of	4	7	12	6	6	(100)
Δ	1.8%	1.8%	1.4%	0%	1.0%	1.5%
p	ϵ	ϵ	ϵ	0.06	0.01	ϵ
Correctness						
Peer	**9.00**	**12.9**	**12.7**	**29.7**	14.3	**84.5**
Tutor	**9.10**	**12.7**	**12.8**	**31.1**	14.3	**85.3**
out of	10	16	14	34	19	(100)
Δ	0.8%	1.3%	1.1%	4.0%	0%	0.8%
p	0.02	0.03	ϵ	ϵ	0.30	0.03

Table 3: Comparison of style and correctness marks between reviewers and tutors. The difference, Δ, is the tutor mean less the reviewer mean divided by the maximum marks (out of). Significance is calculated using one-sample t-tests of the pairwise differences, and shown in bold when not zero ($p < 0.05$).

		Corr.	N
EG07a	Style	0.32	2575
	Correct	0.52	2557
EG07b	Style	0.66	2659
	Correct	0.69	2653
CS07	Style	0.63	1642
	Correct	0.81	1641
EG08a	Style	0.42	1608
	Correct	0.56	1568
EG08b	Style	0.53	2001
	Correct	0.76	1982
All	Style	0.53	10485
	Correct	0.71	10401

Table 4: Style and correctness correlations. All p-values are < 0.01.

	Q1 N	Q2 N	Q3 N	Q4 N
EG07a	0.641 619	0.710 706	0.751 684	0.780 654
EG07b	0.392 595	0.603 684	0.501 646	0.629 640
CS07	0.625 244	0.702 412	0.866 484	0.827 508
EG08a	0.445 361	0.602 417	0.594 412	0.684 423
EG08b	0.705 420	0.723 518	0.791 526	0.851 563
All	0.598 2239	0.693 2737	0.747 2752	0.793 2788

Table 5: Total mark correlations by reviewer quartile. Q1 includes students whose final exam mark fell in the lowest quartile, and Q4 includes students in the highest quartile. Note that the quartiles are not adjusted for participation in each assignment, resulting in some variation in quartile sizes.

4.2 Comparing reviewers' competence in assessing functional and non-functional aspects of a program

The questions in the rubrics are divided into two broad types: concerning style (assessing non-functional aspects of the program), and concerning correctness (assessing the functionality). Correlation co-efficients enable us to investigate whether the reviewers are better at making appropriate judgements on some types of questions than others (Table 4).

4.3 Comparing the reviewers' assessment competency, with reference to their own achievement

Using the final grades achieved by students, we divided each class into four quartiles and calculated the correlation co-efficient for each quartile (Table 5 and Figure 3). This allows us to see whether there is any difference between the good students and weak students in their ability to mark the assignments to a similar standard to the tutors.

4.4 Discussion

The correlation figures show a good level of agreement, all over 0.5. Competency rises between the two assessments within the two Engineering classes. The Computer Science correlations are, at first experience, as good as the Engineering students' on their second experience. This is despite the higher entrance standard required in Engineering. Whether

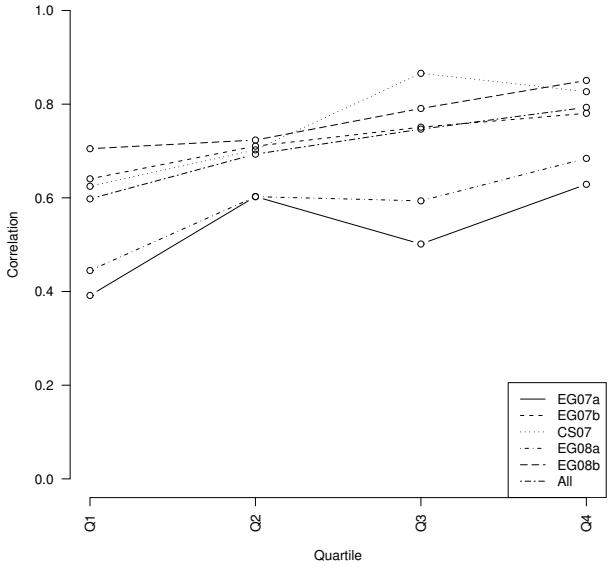

Figure 3: Correlations of reviewer v. tutor marks by student quartile.

it is due to better preparation by the Computer Science instructor, or some other external factor we cannot say.

The mean marks were closer for correctness than for style, as might be expected. While the mark differences are statistically significant in many cases, the actual mark differences are small — 4% in one case, but otherwise all less than 2%. The 4% case appears to be an anomaly, possibly due to a misunderstanding of a part of the rubric.

In each case where there is a statistically significant difference, it involves reviewers awarding lower marks than the tutors. We conclude there is a slight bias (between 1 and 2%) toward under-marking by student reviewers.

The ability to mark appropriately increases, as can be expected, with student ability: there are clear differences between Q1 and Q4 in all cases.

The middle two quartiles are somewhat volatile in CS07, which is the only one case in which the trend is not upward over all four quartiles.

We note that the rubrics for EG07b, EG08b and CS07 were prepared by the same instructor, who has considerable experience using Aropä for administering student peer reviews. The objectivity of the rubric naturally has an impact on the likelihood that independent reviewers will agree on a single item, and hence may explain the lower correlations seen in the EG07a and EG08a where in each course the rubric was prepared by a new instructor who was using Aropä for the first time.

In summary, our findings are:

- there is a high, significant correlation between marks given by tutors and peers;
- these tutor-peer correlations are higher for correctness marks than for style marks;

- there is a slight but consistent bias toward undermarking by student reviewers;

- higher performing students tend to mark more appropriately than lower performing students.

5. REVIEWERS' COMMENTS

The second question we consider is *How sophisticated are the reviewers' textual comments?*

While accurate, summative marks are important, peer assessment activities are most often conducted on a formative basis. The comments written by student reviewers provide a rich source of feedback, giving assignment authors multiple, possibly conflicting viewpoints to consider and reflect on. To a large extent, the actual content of comments is less important than their timeliness, variety and extent.

While it is, at least in principle, possible to analyse comments for their accuracy, relevance and depth of critical analysis, with over 10,000 reviews, this is infeasible. In this paper, we use a computational linguistics approach to focus on lexical measures of sophistication.

Despite the fact there is a large difference between measuring "lexical sophistication" and measuring "comment quality", various measures allow us to investigate the breadth and nature of lexical tokens used, as a broad indication of textual sophistication. Five measures of lexical sophistication were considered: comment length (LEN); number of distinct tokens (TOK); median word length (MED); a word frequency metric (FREQ); and the Average Token-Type Ratio (ATTR), a measure widely used in computational linguistics [15].

LEN and TOK turn out to be highly correlated (0.97), and we arbitrarily chose to use LEN. MED shows very little variability. ATTR requires written samples of at least 50 words, and only about half the reviewers' comments met this condition.

In both measures of lexical sophistication we chose, the higher the number, the higher the sophistication.

LEN Comment length (in words): the average number of words written in the comment sections in the rubric.

FREQ A measure of word frequency. Word frequencies are taken from the British National Corpus [1] and a large collection of television and movie scripts [21]. Words score 0 if they do not appear in the corpus, 1 if they are frequent (in the top 5%), 2 if common (next 20%), 3 if unusual (50%) and 4 if rare (for the 50% least frequent words). The metric is the sum of the score for each corpus, divided by 8 (to give a number between 0 and 1).

From [1]: "The British National Corpus (BNC) is a 100 million word collection of samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of current British English, both spoken and written."

LEN simply measures volume. It distinguishes students who have something to say from those who choose to say little or nothing. FREQ may distinguish between a limited and a rich, varied vocabulary.

	EG07a	EG07b	CS07	EG08a	EG08b	All
LEN						
Peer	13.0	10.5	8.5	8.9	10.0	10.5
Tutor	37.5	16.0	7.8	10.4	13.0	14.8
Δ	**28.3**	**7**	**-1**	**1**	**2.8**	**6.0**
p-value	ϵ	ϵ	0.003	0.005	ϵ	ϵ
FREQ						
Peer	0.07	0.07	0.06	0.08	0.08	0.07
Tutor	0.08	0.09	0.06	0.09	0.09	0.09
Δ	**0.015**	**0.015**	0	**0.01**	**0.01**	**0.01**
p-value	ϵ	ϵ	0.35	ϵ	ϵ	ϵ

Table 6: Comparison of reviewer and tutor LEN and FREQ metrics by assignment. Significance is calculated using a paired Wilcoxon signed rank test. The Δ value is the pseudomedian. Tutor and reviewer medians are also shown.

5.1 Comparing the lexical sophistication between different assignments

We calculated the sophistication metrics for each of the assignments. This enables several issues to be investigated. We can compare reviewers with tutors, and we can see if there are any changes in lexical sophistication as students become more experienced with peer assessment. For the latter comparison, we compare the "a" and "b" versions of each EG class; i.e. EG07a and EG07b, EG08a and EG08b. We are also able to compare the metrics for the Engineering students (all of whom take courses in writing) against those of the Computer Science students (for whom writing courses are optional).

The results of a comparison of the LEN and FREQ metrics between reviewers and tutors are given in Table 6. Note that in all cases the distributions are highly non-parametric, and our analysis uses a paired Wilcoxon signed rank test. The sample differences are estimated using the pseudomedian (i.e. the median of the difference between each reviewer-tutor pair). The distributions themselves are shown in Figures 4 and 5.

To compare lexical sophistication between classes, we concatenated the comments from all allocations by reviewer and computed the LEN and FREQ metrics for the aggregated data. A paired Wilcoxon signed rank test was then done for each reviewer who participated in both assignments (see Table 7). There was no significant change in FREQ in either year. However, the reviewers wrote more in the second assignment in 2007 but less in 2008. The distributions for the LEN metric for the aggregate data are shown in Figure 6.

5.2 Comparing reviewer lexical sophistication, with reference to their own achievement

Over all assignments, we can look at the lexical sophistication for the different quartiles, based on the final exam mark achieved by the students at the end of the course. This enables us to investigate whether the lexical sophistication of the comments varies with the ability of the students. Table 8 shows the median values for the LEN and FREQ metrics for each quartile group, together with the estimated difference between the first and fourth quartiles. The LEN metric increases steadily with each quartile, while the only significant

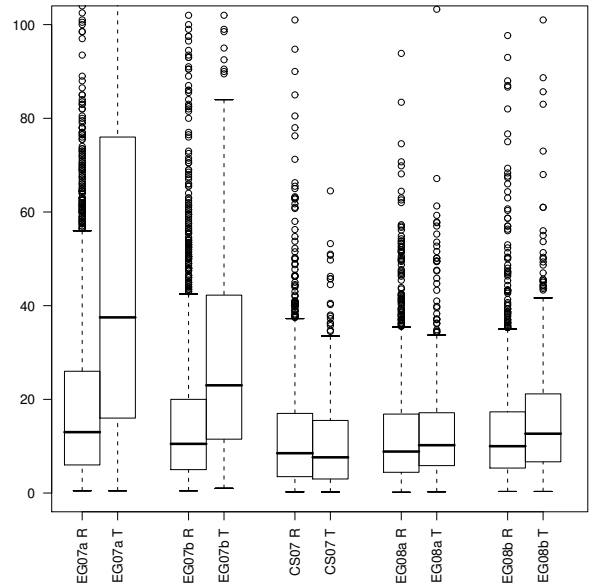

Figure 4: Distributions of LEN for reviewers and tutors. Tutors in EG write more than reviewers, while tutors in CS write slightly less.

	EG07	EG08
LEN		
median a	66.5	26.9
median b	54.0	37.0
Δ	**13.0**	**-7.6**
p-value	ϵ	ϵ
FREQ		
median a	0.08	0.08
median b	0.08	0.08
Δ	-0.002	0.003
p-value	0.22	0.08

Table 7: Comparison of LEN and FREQ in the two pairs of consecutive EG assignments. The Δ value is the pseudomedian. The median values for each assignment are also shown.

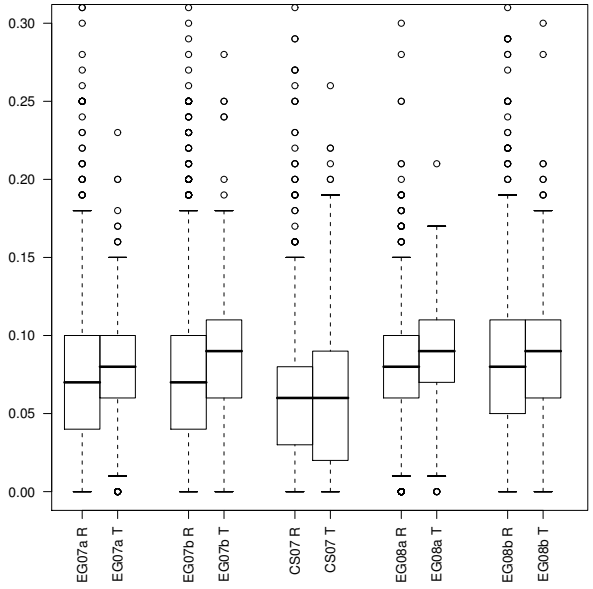

Figure 5: Distributions of FREQ for reviewers and tutors. Tutors in EG use a more sophisticated vocabulary than reviewers, while tutors in CS use a similar vocabulary range to reviewers.

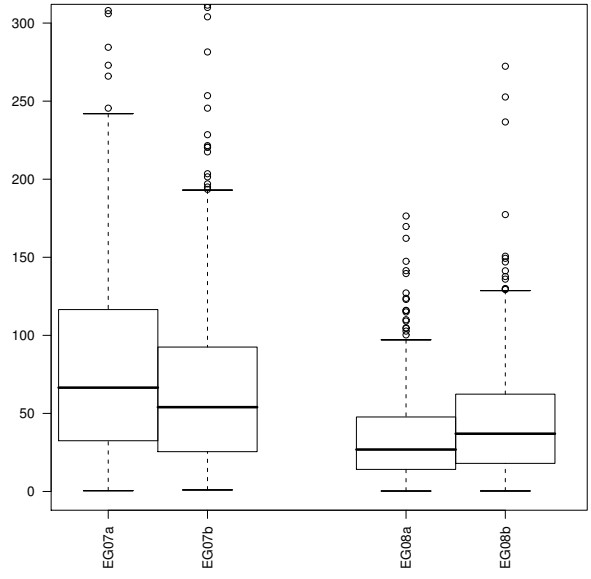

Figure 6: Distribution of LEN for aggregated comments for the first and second exercise in the EG classes. The volume of commenting decreased one year and increased in the next.

	Q1	Q2	Q3	Q4	Δ
LEN					
EG07a	10.0	12.0	14.3	17.5	6.5
EG07b	8.0	9.0	10.5	16.5	7.0
CS07	4.6	6.8	8.7	11.5	4.7
EG08a	7.4	8.6	9.1	10.8	3.3
EG08b	8.7	8.7	10.0	12.3	3.3
All	8.1	9.0	10.7	13.7	4.6
FREQ					
EG07a	0.07	0.06	0.07	0.08	0.01
EG07b	0.06	0.06	0.07	0.08	0.01
CS07	0.05	0.05	0.05	0.07	0.01
EG08a	0.07	0.08	0.08	0.09	0.01
EG08b	0.07	0.08	0.08	0.08	0.01
All	0.07	0.07	0.07	0.08	0.01

Table 8: Median values for LEN and FREQ by quartile for each assignment. Δ is the pseudomedian difference between the first and fourth quartiles. All p-values are < 0.01.

increase in FREQ occurs in the fourth quartile. Figures 7 and 8 show the distributions for the two metrics by quartile.

5.3 Discussion

Tutors outperform reviewers, in terms of both comment length and vocabulary frequency, in all the Engineering assignments, but have comparable performance to the Computer Science reviewers.

The tutor feedback in both the Engineering assignments from 2007 is noticeably greater in volume than in the other classes. This may be attributed to the fact that the tutors in 2007 were recruited based on a combination of GPA (grade point average) and the quality of reviews (all archived in the Aropä database) they had written when studying in a previous semester. Tutors in the CS07 course are not selected based on their archived Aropä reviews. Rather, they are typically senior level or graduate students with high GPAs. This recruitment process that considers the quality of an applicant's archived Aropä reviews appears to be an effective way of employing tutors capable of writing detailed feedback when grading student assignments.

There is a steady increase in LEN with attainment quartile. This is not surprising, as the strongest students may feel most confident providing feedback to their peers and are more likely to take the reviewing activity seriously.

FREQ only increases for the top quartile of students, and in the Engineering courses is consistently less than that of the tutors. We would expect that the tutors, having been drawn from a pool of successful students with a track record of producing good quality reviews, would tend to use a richer, more varied vocabulary than a typical student.

As Chalk and Adeboye [4] noted, differences in rubrics may account for the differences in lexical sophistication between assignments. One possible explanation for the difference between the quantity of comments written by Engineering students compared to those in Computer Science is the training/previous marking experiences in those subject contexts.

In summary, our findings are:

- in Engineering assignments, tutors out-perform review-

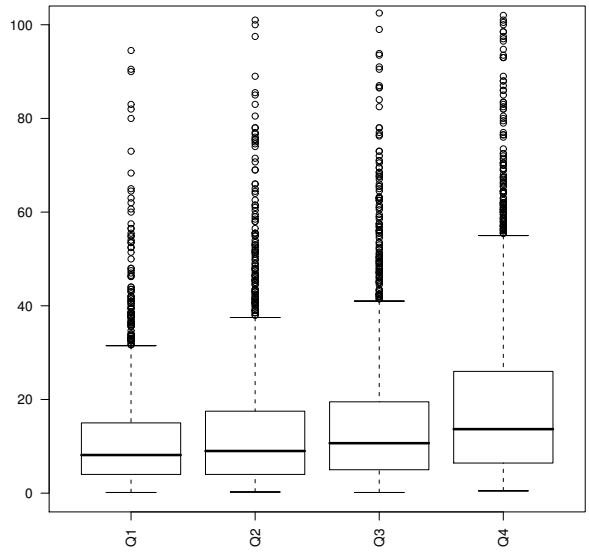

Figure 7: Distribution of LEN by reviewer quartile.

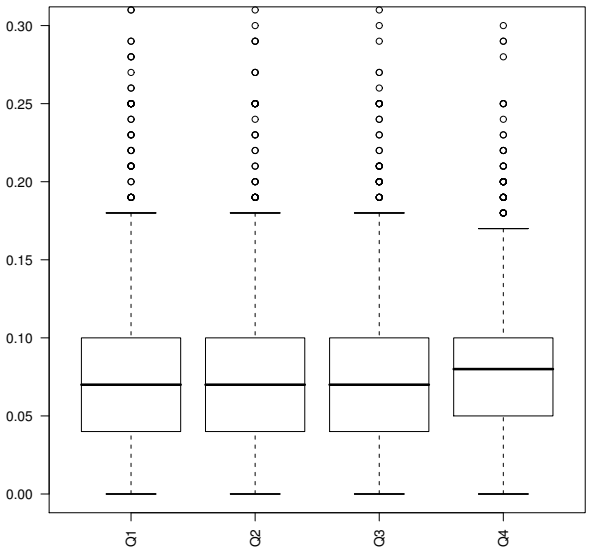

Figure 8: Distribution of FREQ by reviewer quartile.

ers in both lexical sophistication metrics. This is not the case in Computer Science assignments;

- high performing students write more than lower performing students;
- we only see a significant increase in vocabulary for the highest performing students.

6. CONCLUSIONS

We have conducted a large-scale study into the quality of student peer reviews in introductory programming courses where the artifact being reviewed is a novice program and both style and correctness aspects are considered.

Even in this constrained context, differences in performance are evident between different student groups. These differences are not always as might be expected. Engineering students have a considerably higher entrance standard than Computer Science, and yet their quantitative grading was weaker in some regards. The reasons for this difference deserve further investigation. On the other hand, the Engineers rated more highly than the Computer Science students in our metrics of comment feedback.

Overall, reviewer marks are highly correlated with tutors, even before any adjustments are made in computing weighted average grades. The need for tutors as a quality assurance measure is not strongly supported by our analysis.

There is a slight but consistent bias toward under-marking compared to tutors in questions that require greater subjective judgement. As we noted in our summary of related work, this result is likely to be highly contextual.

Our analysis of lexical sophistication, while crude, suggests that, as should be expected, student reviewers generally produce less sophisticated comments than tutors. The significance of this may be mitigated by peer assessment generating multiple reviews for each submission.

7. REFERENCES

[1] The British National Corpus. `http://www.natcorp.ox.ac.uk/`, Accessed 6 March 2009.

[2] S. J. Bostock. Computer assisted assessment — experiments in three courses. `www.keele.ac.uk/depts/cs/Stephen_Bostock/docs/caa-ktn.htm`, May 2000. Workshop at Keele University.

[3] D. Boud and H. Holmes. *Enhancing Learning through Self Assessment*, chapter Self and peer marking in a large technical subject, pages 63–78. Kogan Page, London, 1995.

[4] B. Chalk and K. Adeboye. Peer assessment of program code: a comparison of two feedback instruments. In *6th HEA-ICS Annual Conference*, pages 106–110, 2005.

[5] M. de Raadt, S. Dekeyser, and T. Y. Lee. A system employing peer review and enhanced computer assisted assessment of querying skills. *Informatics in Education*, 6(1):163–178, 2007.

[6] N. Falchikov and J. Goldfinch. Student peer assessment in higher education: A meta-analysis comparing peer and teacher marks. *Review of Educational Research*, 70(3):287–322, 2000.

[7] E. F. Gehringer, D. D. Chinn, M. A. Pérez-Quiñones, and M. A. Ardis. Using peer review in teaching computing. In *SIGCSE'05: Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, pages 321–322, New York, NY, USA, 2005. ACM.

[8] D. Haaga. Peer review of term papers in graduate psychology courses. *Teaching of Psychology*, 20(1):28–32, 1993.

[9] J. Hamer. The Aropä peer assessment system. `https://aropa.ec.auckland.ac.nz`, 2009.

[10] J. Hamer, Q. Cutts, J. Jackova, A. Luxton-Reilly, R. McCartney, H. Purchase, C. Riedesel, M. Saela, K. Sanders, and J. Sheard. Contributing student pedagogy. *SIGCSE Bulletin*, 40(4):196–214, Dec. 2008.

[11] J. Hamer, C. Kell, and F. Spence. Peer assessment using Aropä. In S. Mann and Simon, editors, *ACE'07: Ninth Australasian Computing Education Conference*, volume 66, pages 43–54, Ballarat, Victoria, Feb. 2007. Australian Computer Society.

[12] J. Hamer, K. T. Ma, and H. H. Kwong. A method of automatic grade calibration in peer assessment. In A. Young and D. Tolhurst, editors, *ACE'05 Australasian Computer Society Education Conference*, volume 42 of *Conferences in Research and Practice in Information Technology*, pages 67–72. Australian Computer Society, Jan. 2005.

[13] J. Heywood. *Assessment in Higher Education*. Jessica Kingsley Publishers, London, 2000.

[14] G. Marcoulides. and M. Simkin. The consistency of peer review in student writing projects. *Journal of Education for Business*, 70(4):220–223, 1995.

[15] P. M. McCarthy and S. Jarvis. vocd: A theoretical and empirical evaluation. *Language Testing*, 24(4):459–488, 2007.

[16] P. J. Miller. The effect of scoring criteria specificity on peer and self-assessment. *Assessment & Evaluation in Higher Education*, 28(4):383–394, 2003.

[17] D. Paré and S. Joordens. Peering into large lectures: examining peer and expert mark agreement using peerscholar, an online peer assessment tool. *Journal of Computer Assisted Learning*, 24(6):526–540, 2008.

[18] J. Sitthiworachart and M. Joy. Effective peer assessment for learning computer programming. *SIGCSE Bulletin*, 36(3):122–126, 2004.

[19] J. Sitthiworachart and M. Joy. Computer support of effective peer assessment in an undergraduate programming class. *Journal of Computer Assisted Learning*, 24(15):217–231, June 2008.

[20] L. Stefani. Peer, self and tutor assessment: relative reliabilities. *Studies in Higher Education*, 19(1):69–75, 1994.

[21] Most common words (TV and movie scripts). `http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists`, Accessed 6 March 2009.

[22] W. J. Wolfe. Online student peer reviews. In *CITC5'04: Proceedings of the 5th Conference on Information Technology Education*, pages 33–37, New York, NY, 2004. ACM.

APPENDIX

The grading rubric used in the Computer Science assignment is shown here. The assignments for Engineering have

a similar form.

A. SAMPLE MARKING RUBRIC

Question One (8 marks) Style

Examine the `Q1Program.java` file and review the following style categories:

Indentation

Inconsistent there is at least one place where the lines of code are not correctly indented according to the standard code conventions

Perfect all of the code is indented correctly according to the standard code conventions

Comments

(Note: the author information will be automatically filtered out of the comment at the top of the program by the peer review system)

None there is no comment at the top of the file clearly describing the purpose of the program (ie. a description of what it is that the program does)

Poor the comment appearing at the top of the file contains spelling mistakes or uses language which is unprofessional

Perfect a comment appears at the top of the file which clearly describes the purpose of the program

Comments for Question One Style

(A text box appears here).

Correctness

Compile the `Q1Application.java` and `Q1Program.java` files and run the program using the `Q1Application` class. You will have to examine the source code for the `Q1Program.java` file to check that an escape code has been used correctly in one of the print statements.

ASCII Art

(Note: the artistic quality of the picture is not worth any marks)

Output does not draw a picture of 10 lines or more the code does not compile, or the program does not print a picture which consists of at least 10 lines of output

No escape code a picture with at least 10 lines of output is printed, but there is no escape code in any of the print statements which produce the picture

Perfect a picture with at least 10 lines of output is printed

Title

No title appears following the picture there is no title, or the title appears above the picture

Perfect a title is printed below the picture

Comments for Question One Correctness

(A text box appears here).

Question Two (17 marks) Style

Examine the `Q2Program.java` file and review the following style categories:

Indentation

Inconsistent there is at least one place where the lines of code are not correctly indented according to the standard code conventions

Perfect all of the code is indented correctly according to the standard code conventions

Comments

(Note: the author information will be automatically filtered out of the comment at the top of the program by the peer review system)

None there is no comment at the top of the file clearly describing the purpose of the program (ie. a description of what it is that the program does)

Poor the comment appearing at the top of the file contains spelling mistakes or uses language which is unprofessional

Perfect a comment appears at the top of the file which clearly describes the purpose of the program

Descriptive variable names

Poor names chosen for at least two of the variables do not describe the information which is stored in the variables. If it is not possible to figure out what the variable is used to store simply by looking at the name, then the name is not adequate

Good the names chosen for most variables describe the information which is stored in the variables, however, there is one name which does not describe the information stored in the variable

Perfect the names chosen for all variables are excellent and describe what the variables are used for

Variable identifiers

Violate conventions there is at least one variable declared which has an identifier that violates the variable naming conventions (i.e. it violates one of the rules that all variable identifiers begin with a lower case letter, all subsequent words which make up the identifier should start with a capital letter, and all other letters should be lower case)

Perfect all variable identifiers adhere to variable naming conventions (i.e. all variable identifiers begin with a lower case letter, all subsequent words which make up the identifier should start with a capital letter, and all other letters should be lower case)

Use of symbolic constants

Poor there is at least one place in the source code (with the exception of String literals) where a literal value has been used where a symbolic constant could have been used in its place

Perfect symbolic constants have been used instead of literal values throughout the source code (with the exception of String literals)

Comments for Question Two Style

(A text box appears here).

Correctness

Incorrect change the number of coins of each type is not exactly the same as appears under "Change given:" in the example

Incorrect "extra profit" the number of coins of each type is correct, but the number of cents "xtra profit" is not correct

Perfect the number of coins of each type and the extra profit are correct

Test Case 2 (check values)

Enter the value 380. The output should be identical to that below:

```
Asst1Marking> java Q2Application
Total change (in cents): 380
Change given:
  $2:  1
  $1:  1
```

```
  50c: 1
  20c: 1
  10c: 1

Extra "profit": 0c
```

Values

Incorrect change the number of coins of each type is not exactly the same as appears under "Change given:" in the example

Incorrect "extra profit" the number of coins of each type is correct, but the number of cents "extra profit" is not correct

Perfect the number of coins of each type and the extra profit are correct

Comments for Question Two Correctness

(A text box appears here).