Using Collaborative Filtering Algorithms as eLearning Tools

Frank Loll
Clausthal University of Technology, Germany
frank.loll@tu-clausthal.de

Niels Pinkwart
Clausthal University of Technology, Germany
niels.pinkwart@tu-clausthal.de

Abstract

Collaborative information filtering techniques play a key role in many Web 2.0 applications. While they are currently mainly used for business purposes such as product recommendation, collaborative filtering also has potential for usage in eLearning applications. The quality of a student provided solution can be heuristically determined by peers who review the solution, thus effectively disburdening the workload of tutors. This paper presents a collaborative filtering approach which is specifically designed for eLearning applications. A controlled lab study with the system confirmed that the underlying algorithm is suitable as a diagnostic tool: The system-generated quality heuristic correlated highly with an expert-provided manual grading of the student solutions. This was true independent of whether the students provided fine-grained or coarse-grained evaluations of peer solutions, and independent of the task type that the students worked on. Further, the system required only few peer evaluations in order to achieve an acceptable prediction quality.

1. Introduction

The term "collaborative filtering", which was coined more than 15 years ago [1], stands for a family of today's most prominent and widely used algorithms in Social Software systems. In collaborative filtering based systems, users get associated to system artifacts through their (explicit or implicit) actions in the system. The specific type of artifacts and associations differ between applications. Examples include buying or looking at books at amazon.com, entering profiles in online dating services or tagging images on flickr. In any of these cases, the system then uses this information to recommend artifacts (i.e., products or users) to other users. While the details of the calculation clearly vary between systems, the general underlying principle of collaborative filtering is to build on the user base of a system and use their actions to generate the system knowledge. Today, collaborative filtering techniques play a key role in many Web 2.0 applications primarily for business purposes such as product recommendation in online stores (e.g., amazon.com).

Collaborative filtering also has potential for usage in "Social Software" eLearning applications in that the quality of a student provided solution could be determined by peers who review the solution. Based on these peer reviews, the system could then give feedback to learners – e.g., by recommending good answers to students who appear to have problems with a certain task. Such an approach has potential for effectively disburdening the workload of teachers and tutors, and can at the same time help students develop evaluation skills that they rarely learn in formal education: Students who conduct peer reviews read possible task solutions that were provided by other students, and they have to evaluate them. They thus have to deal with wrong or excellent solutions of others (and learn the skill of assessing), and – for more open tasks – they also have the opportunity to learn about different possible points of view.

A central prerequisite for any educational collaborative filtering tool that relies on peer reviews is that the system has to be able to accurately determine a quality measure for the student solutions based on the peer ratings – without such a base mechanism, advanced system features such as the recommendation of "good" or "poor" results are impossible. But are student's peer reviews a suitable base for such a quality measure at all? Here, main critiques are related to the issues that students are not experts in the field (and therefore their rating may be inadequate), that students are inexperienced in assessing the quality of a solution, and that peer ratings are prone to bias due to uniformity, race, friendship and purposeful manipulation [2, 3].

However, peer reviews and collaborative scoring can also have advantages as compared to a traditional "expert grading" approach in which a teacher assigns marks to the solutions of his students. Peers may be more adept in understanding problems of other students [4]. Especially in ill-defined domains [5] where solutions cannot simply be judged as being

978-0-7695-3450-3/09 $25.00 © 2009 IEEE

right or wrong and thus simple multiple-choice tests are inappropriate, assessment is a hard task for a single person (like the teacher or tutor) and usually not possible to automate. In a collaborative rating system, each peer is a critic so that each peer can provide comments or marks. Taken together, these multiple opinions may even be superior to a single expert opinion, as argued in scientific as well as in popular literature [6, 7].

In this paper, we propose a novel collaborative scoring algorithm that is able to assess the quality of student solutions based on relatively few peer ratings. The algorithm is tailored for educational purposes, suited for a variety of typical educational tasks, not restricted to specific domains, and can be used as a base for collaborative filtering based eLearning systems.

2. Educational Collaborative Filtering Applications

Despite their appeal and application potential as motivated above, collaborative filtering based tools for educational applications are still rare, as are empirical studies which investigate the effectiveness of these tools.

One of the few existing systems is the web-based *PeerGrader* (PG) [8, 9]. The purpose of this tool is to help students improve their skills by reviewing and grading solutions of their fellow students blindly. PG works in the following way: First, the students get a task list and each student chooses a task. Next, the students submit their solutions to the system, where they are read by another student who then provides feedback in form of textual comments. After that, the authors modify their solutions based on the comments they have received, and re-submit their modified solutions again to the system, where they will be reviewed by other students. Then, the solutions' authors grade each review with respect to whether it was helpful or not. Finally, the system calculates grades for all student solutions.

One of PG's strengths is to provide students with high-quality feedback also in ill-defined homework tasks that do not have clear-cut gold standard solutions (such as design problems). This kind of feedback could not be generated automatically. A disadvantage is the time required for the system to work effectively: due to the complexity of the reviewing process and the textual comments, the evaluation of a single student answer is very time consuming. This may cause student drop-outs and deadline problems [9]. Also, studies with PG revealed problems with getting feedback of high quality. An evaluation of subjective usefulness showed that the system was appreciated by its users [9], yet a systematic comparison of PG scores to expert grades has not been conducted.

A newer web-based collaborative filtering system is the Scaffolded Writing and Rewriting in the Discipline *(SWoRD)* system [6, 10]. SWoRD addresses the problem that in the writing discipline, homework solutions are often long texts, which cannot be reviewed in detail by a teacher for time reasons. Because of this, students do often not receive any detailed feedback on their solutions at all. Having such feedback would be beneficial for students though, since they could use it to improve their future work. To address this problem, SWoRD relies on peer reviews and implements an algorithm that follows the typical journal publication and reviewing process. An evaluation showed that the participants benefitted from multi-peers' feedback more than from single-peer's or single expert's feedback [6].

A different approach is used by the *LARGO* system [11], where students create graphs of US Supreme Court oral arguments. Within LARGO, collaborative scoring is employed to assess the quality of a "decision rule" that a student has included in his diagram. Since this assessment involves interpretation of legal argument in textual form, it cannot be automated reasonably. While the overall LARGO system has been tested in law schools and shown to help lower-aptitude learners [12], empirical studies to test the educational effectiveness of the specific collaborative scoring components have not been conducted.

Another area where collaborative filtering has been used in educational technology systems is the recommendation of learning resources. The system *Altered Vista* (AV) [13] provides a database in which user evaluations of web-based learning resources are stored. Users can browse the reviews of others and can get personalized learning resource recommendations from the system. In contrast to the other systems mentioned before, AV does not aim to support learners directly by giving them feedback on their work. Instead, AV provides an indirect learning support in which (presumably) suitable learning tools are recommended. A survey-based evaluation of AV showed a predominant positive feedback, but also identified issues with the system's incentive and with regard to privacy [13].

In summary, the relatively few educational technology systems with collaborative filtering components all have an underlying algorithm to determine solution quality based on collaborative scoring. Yet, existing systems are often specialized for a particular application area such as legal

argumentation (LARGO), writing skills training (SWoRD), or educational resource recommendation (AV), or they involve a rather complicated and long-term review process (SWoRD, PG).

In contrast, this paper presents an approach for a collaborative scoring algorithm which is a) *general* in that it works for different types of tasks in various disciplines, and b) at the same time *applicable quickly*, requiring neither a lot of time-consuming phases nor a large user base (which is an important practical requirement for many learning scenarios). Our algorithm is designed to build the base for a collaborative filtering educational technology system. As such, our main research hypothesis in this paper is that the proposed collaborative scoring algorithm indeed "works", i.e. that

[H1] the system's quality heuristic of student solutions accurately classifies the quality of student solutions, as measured by manual grades provided by a human expert.

Besides this main research hypothesis, the study described in section 4 of this paper also investigates the following questions:

[H2] Does the degree of detail in which a peer rating is possible (fine grain vs. coarse grain) have an impact on the prediction quality of the algorithm? Fewer options for grading peer solutions, such as only classifying them into "good" and "bad", may be easier and faster than a detailed grading scheme, but are they also less precise overall?

[H3] Is the system's quality heuristic of better quality than the self-assessment of the participants (again, measured by difference to the expert provided grades)? Since a self-assessment is usually easier to obtain than multiple peer reviews, this comparison is an important baseline to be outperformed by a collaborative filtering system in educational applications.

[H4] Do more assessments improve the system's quality heuristic, and how many assessments are required for a sufficient quality? Obviously, we hypothesize that more reviews improve the system's quality assessment. Yet, educational applications are often characterized through smaller user numbers. As such, an effective algorithm should also work when only few peer ratings are available.

[H5] Are there specific types of educational tasks for which the algorithm works best? The peer review tasks differ greatly between, for instance, simple knowledge questions and questions that ask for student opinions on a specific topic. While rating a peer solution for the former type only involves comparing it to the ideal solution (if the rater knows this), the latter type requires a more in-depth thought about possible different views and alternatives. This may result in different grading attitudes of the students, as compared to more simple questions. It is not a priori clear that one single algorithm for collaborative filtering can deal with these varying types of questions.

Taken together (and under the assumption that H1 is confirmed), answers to these questions help inform educational software developers make design decisions where (and in which form) collaborative filtering may be a useful technique for their applications. While H3 and H4 have been investigated before in a few studies within different contexts [10, 14], H2 and H5 are (to our knowledge) novel research questions.

3. System description

To investigate these research questions, we designed a Java and XML based tool which uses collaborative scoring to filter student solutions for quality. .This section briefly describes the design and algorithmic aspects of this software.

3.1 Usage Phases

After the login to the system, the usage of the tool involves three phases.

1. Work on a task
2. Assessment of three alternative solutions
3. Self-assessment of own solution

Figure 1 illustrates the user interface for phase 1[1] with an example task from the study reported on in section 4 (in this case, a Java programming question). The users can see the task and are given a time limit as well as a character limit for their solution. These limits serve to control for time on task, and also to give the users an orientation about the expected length of their answer. In order to investigate research question H2, we implemented two system variants which differ in the degree of detail that is available for the assessment in phase 2. The first variant (named N for "normal" later on) just asks the

[1] All screenshots have been translated to English for this paper. In the original study, they were in German language.

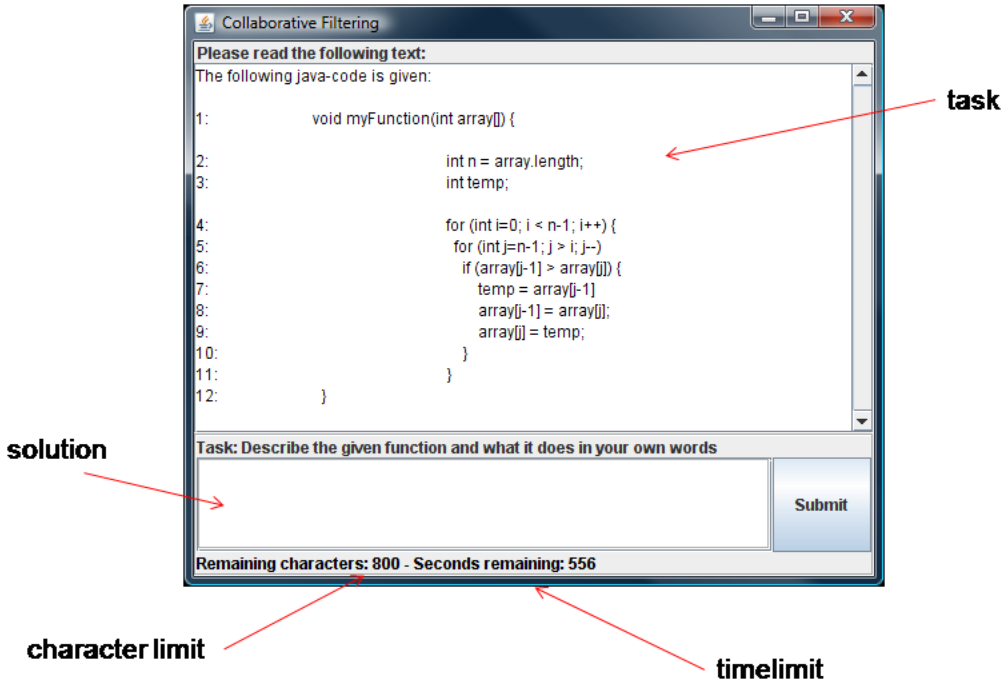

Figure 1: Task editing

students whether the shown solutions are of good or poor quality (see Figure 2). The second variant (named D for "detailed") allows a more fine granular rating on a scale from 0 (poor) to 10 (excellent).
The user interface of the D variant in the assessment phase is shown in Figure 3. Another difference between the D and N variants is in the self-assessment phase. In order to realistically compare the self assessments to the system results, the "N" condition also has fewer options here, whereas the "D" variant uses the same fine granular scale as in phase 2.

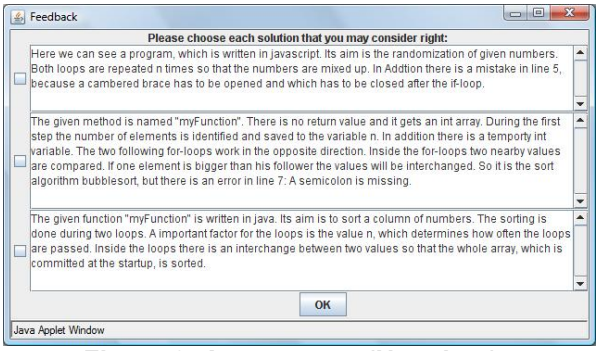

Figure 2. Assessment (N variant)

3.2. Collaborative Filtering Algorithm

The collaborative scoring (or: quality filtering) algorithm we employed to generate the heuristic for the quality of student-provided solutions is based on two components.

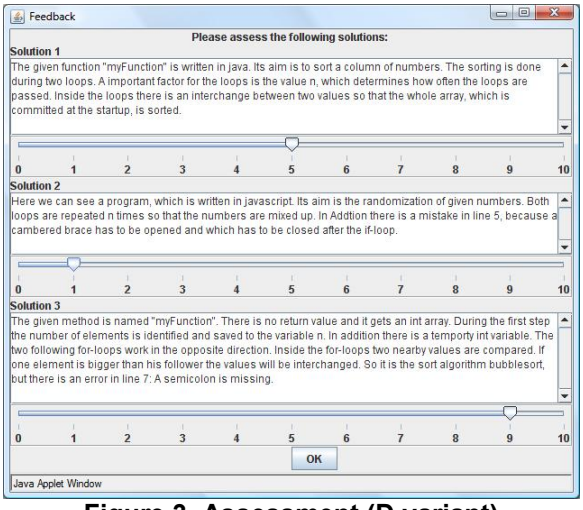

Figure 3. Assessment (D variant)

First, based on the evaluations a student makes, a heuristic of the quality of this student's *own* answer is calculated based on the hypothesis that recognizing good solutions is an indication for having understood the problem at hand, which in turn is a prerequisite for having created a high quality solution oneself. We call this first heuristic measure the *base rating*. If a student provides evaluations for n peer solutions which have a quality measure $q_1, \ldots, q_n$ (0 standing

for very bad, 1 for very good, see below for the calculation of quality measures for peer answers), and his assessments of these solutions are $w_1, \ldots, w_n$, then the base rating b is calculated as:

$$b = \frac{1}{n}\sum_{i=1}^{n}\left(1-\left|w_i - q_i\right|\right)$$

Note that the "N" variant of our system only allows the values 0 and 1 for the w_i elements in that equation (0 = rated as bad, 1 = rated as good), whereas the "D" variant allows values in steps of 0.1. In the example shown in Figure 3, the three answers that the student has to assess have the quality ratings 0.5, 0.05 and 0.95 (these ratings, of course, not shown to the user). As shown in the figure, the student rates them as 0.5, 0.1 and 0.9. Consequently, the base rating for this student's own answer is b = 1/3 * (1–|0.5–0.5|+1–|0.05–0.1|+1–|0.95–0.9|) ≈ 0.97. This is a very high rating which is due to the fact that the student has rated all solutions as exactly as possible (according to the system heuristics). Presumably, this student did correctly estimate the quality of the other solutions he had to evaluate, and has thus probably understood the task – otherwise, he would not have been able to do so well in the assessment of answers to the task.

The base rating of an answer is immediately available once a student has provided his evaluations. It measures to what extent a student can recognize good answers and thus serves as a heuristic of his own answer's quality, but it is not directly related to the answer the student has actually typed in. Following the collaborative information filtering paradigm, this can be measured over time through the positive and negative evaluations that a student's answer receives. We call this the *evaluation rating* e. If j students, whose own solutions for a specific task have the quality measures $q_1, \ldots, q_j$, evaluate an answer and assign them the ratings $w_1, \ldots, w_j$, (again, for the "N" condition the values of w_i are restricted to 0 and 1), then e is calculated as:

$$e = \frac{1}{\sum_{i=1}^{j} q_i}\left(\sum_{i=1}^{j} w_i q_i\right)$$

In this formula, the evaluations given by (presumably worse) peers who have a lower quality rating receive a lower weight. If we continue our example from above and assume that four other students had the chance to evaluate the solution, and one of them (with own quality measure of 0.4 on this question) gave a negative evaluation of 0.1, whereas three others with own quality measure scores of 0.7, 0.9 and 0.8 provided high grades of 0.9, 0.8, and 1.0, then the evaluation rating is e = (0.1*0.4 + 0.9*0.7 + 0.8*0.9 + 1.0*0.8) / (0.4+0.7+0.9+0.8) ≈ 0.78. This high-to-medium score results from the mostly positive evaluations by the peers, and at the same time assigns higher weights to the ratings of students who have a high quality measure score.

Finally, the overall *quality rating* q of a student answer is calculated as the weighted average of the base rating and the evaluation rating that were described above. It is reasonable to make the impact of the of the evaluation rating on the overall quality rating dependent on the amount of evaluations that a student answer received: If a student's answer was subject to evaluation by p peers, and c is the number of answers that was presented to the student for evaluation himself, q is calculated as:

$$q = \frac{c}{p+c}b + \frac{p}{p+c}e$$

The design of the formula ensures a normalization of the quality rating q in [0, 1] and takes into account the importance of the peer's opinions (with large p, the base rating gets less important) while at the same time eliminating the new-item problem of collaborative filtering systems (i.e., the fact ratings for new items are not available) through the inclusion of the base rating. In our example, we have p=4 and c=3, and thus an average q = 3/7*0.97 + 4/7*0.78 ≈ 0.86. This high rating takes into account both the base rating and the evaluation rating. Since the evaluation rating is based on statements of four peer students only, it determines 4/7 (~57%) of the overall score, the remaining 43% come from the base rating. This definition of the quality rating formula gradually fades out the impact of the base rating once a larger number of peer evaluations are available.

While the presented approach works fine for most of the students in a group, the first and last students who use the system need special attention.

For the *first* students that solve a task, peer answers are of course not available yet. Here, our approach relies on a few system provided answers of known quality (some bad, some good) in order to allow also these users a rating of other solutions. For the study described in the following, we included three solutions per task that were developed by domain experts. These solutions, together with their expert-provided grades, are designed to ensure a good initial quality heuristic in the system. This is important for our approach. If the system has a poor heuristic of the first answers initially shown to students in evaluation dialogs, this increases the

number of needed evaluations in order to stabilize the quality of the overall quality heuristic. For relatively small user groups in educational settings, the time available to the system might then not be sufficient to produce good quality ratings.

Finally, the answers of the *last* students that work on a specific part of the text will not be evaluated by peers at all. Therefore, the quality rating of their answers is equal to the base rating of their answers, which stresses the importance of the latter.

4. Study Description

In spring 2008, a controlled lab study at Clausthal University was conducted in order to investigate the research hypotheses. 45 participants, students and postgraduates of varying disciplines, took part in the study (27 male and 18 female). Participation was voluntary, and all participants received a small financial remuneration. We randomly assigned the subjects to the two study conditions N and D. Condition N consisted of 15 male and 7 female students, while the condition D consisted of 12 male and 11 female students.

In the study, the subjects had 75 minutes to work on 12 tasks in total. Each condition worked on the same 12 tasks, but the collaborative filtering algorithm did not mix the conditions (i.e., a separate server was used for each of the two conditions). The students were explicitly instructed to assess peer solutions even if they did not know the solution to a task themselves.

4.1 Tasks

In order to investigate research question H5, we included tasks of four different types in the study, all of them represent typical educational tasks: text summaries (3 tasks), text interpretations (1 task), knowledge tests with the possibility for guessing (3 tasks) and knowledge tests without such possibility (5 tasks).

In the first group "text summaries", the students received articles about different topics (such as a text about Second Life). The articles differed in terms of complexity and required know-how to understand the main points. The task of the subjects was to summarize the most important facts of the articles. The second group "text interpretations" consisted of a (fact-centered) news article about the takeover of DoubleClick by Google. The subjects had to interpret this article by providing possible critiques about the takeover with respect to privacy. The third group "knowledge tests without possibility for guessing" consisted of five simple knowledge questions tasks where the chances of "lucky guesses" are minimal. An example for such a task is "For the function $f(x)=3x^2$, provide the derivative at x=2". The fourth group "knowledge tasks with possibility for guessing" contained three tasks. One example here was the task to describe the "Bubble Sort" algorithm (cf. Figure 1), another example was to estimate the population of Austria on the base of a given text about the size of the country. In contrast to the third task group, the subjects here had a realistic chance for guessing solutions and, based on their guesses, hit the right answer while evaluating the alternative solutions, since they had some guidelines in the task description. We decided to split the "knowledge tests" questions into these two groups since for the collaborative filtering process, it may well make a difference whether a student has a realistic chance of guessing a solution quality or not.

5. Results

To answer our research questions as listed in section 2, we compared the system's estimation of solution quality that was provided by the collaborative filtering algorithm against the student's self-assessments of their own solutions and against the manual grades provided by human experts.

Table 1. Inter-rater reliability

Task group	Cronbach's Alpha
Text summaries	0.834
Text interpretations	0.888
Knowledge tests with possibility for guessing	0.982
Knowledge tests without possibility for guessing	0.932

As a preparatory step for our analysis, two human domain experts (one of them a computer science professor, the other an advanced MSc student) manually graded the student-provided answers independently of each other on a scale of 0 to 1. The aims of this procedure were twofold: first, we wanted to investigate inter-rater reliability (did the two experts generally agree with each other?). Secondly, we sought to determine the degree of deviation between the two human graders in order to use this as a baseline for defining an acceptable level of deviation between system grading and human grading in the next analysis steps. Table 1 shows that the inter-rater agreement, measured by Cronbach's Alpha Scores [15] was very high in both knowledge tests groups and still high in the other two groups, where

the "correctness" of a solution is much harder to assess. For all further analyses, the average of the two grader's scores was used as the "human grading".

A next decision to be made was to define what constitutes an acceptable level of deviation between the system score and the human grader's score. The extreme position - requiring that the system-generated score and the human grading do not differ at all for a "correct classification" – is much too strict: even the human expert graders had slight differences between their grades, and a precise quality measure is simply *not possible* for the more ill-defined tasks in the study. Even considering a maximum deviation of 0.1 as the limit for an "acceptable" system heuristics is too strict based on our data: despite the (overall) high inter-rater reliability, the grades assigned by the two human experts differed by more than 0.1 in almost 30 % of all cases. Only in 11.5 % of the cases however, the difference between the human grader's scores exceeded a level of 0.2. We count this as evidence that (at least for our study) a maximum deviation of 0.2 between system score and human score can reasonably be considered as acceptable. Using a greater deviation than 0.2 may render the whole analysis useless, since even a randomly chosen value in [0, 1] differs from a score with an expected value of 0.33. Regarding these facts, we will use 0.2 as an acceptable level of deviation between human grade and system score in the following, since this takes into account the typical level of deviation between the two human graders who, overall, agreed with each other's grading.

5.1 Overall Quality of System Heuristic

Figure 4 shows an overview of all tasks in both system variants. The y-axis denotes the average difference between the system heuristic and the expert grade, while the x-axis shows how many peer reviews a solution has minimally received. For example, the average difference between the system scores in the N variant and the expert grades is 0.25 for all the student solutions that have been reviewed by at least one peer.

As the Figure shows, both algorithm variants finally achieve an average deviation of less than 0.2 from the expert score when enough peer reviews are available. In the N condition, 5 peer reviews lead to such an acceptable prediction quality. For the D condition, 4 peer reviews were already sufficient for this. Overall, this confirms our main research hypothesis H1: the algorithm is suitable for estimating the quality of student solutions.

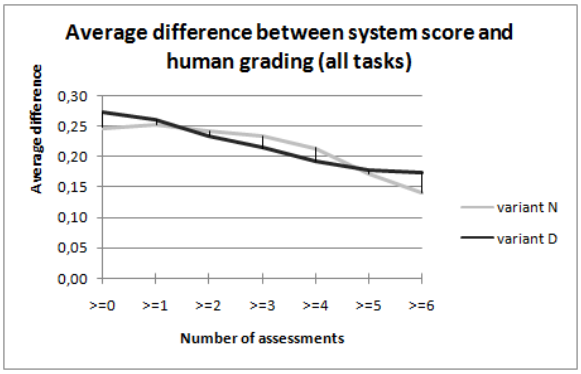

Figure 4. Quality of System Heuristic

Next, we analyzed research question H2. As Figure 4 suggests, the estimation quality between the D and the N conditions is similar. An ANOVA analysis confirmed this. The differences between system scores and expert grades did not differ significantly between N and D variant: $F(1,538)=2.69$, $p>.1$ over all student solutions, and $F(1,31)=.71$, $p>.4$ for those solutions with six or more peer reviews.

In conclusion, both variants worked well and delivered quality scores that did not differ from each other significantly. Therefore we use the combined results (D+N) for the investigation of the research questions H3 through H5 in the following sections.

5.2 System Heuristic vs. Self-Assessment

In order to determine whether the collaborative filtering algorithm yielded better results than the self-assessment of the students (question H3), we compared the differences between system scores and expert grades to the differences between self assessments and expert grades.

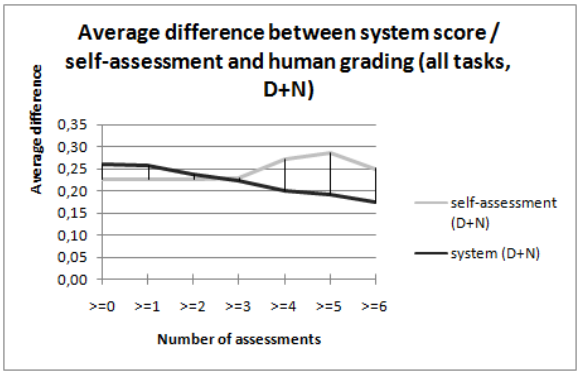

Figure 5. Self Assessment vs. System

Figure 5 shows the results (using the same scales as Figure 4). The Figure clearly shows that the system heuristic score is more accurate than the self-

assessment if more than 3 peer assessments have been given during the evaluation process. A Paired-Samples T-test confirmed that the difference is statistically significant for all the solutions that have received at least 4 ratings ($p<.05$).

5.3 Assessments Required for Sufficient Quality Heuristics

Figure 4 shows that, as expected, the average difference between the system score and the human grading continually decreases with a growing number of assessments. Thus, the prediction quality of the systems increases with the number of assessments. More specifically, Figure 4 shows that 4 to 5 ratings are sufficient to achieve an acceptable quality assessment. For those solutions that received 6 or more peer reviews, the mean difference between system score and expert grade is as low as m=.17 (sd=.09). Yet, Figure 4 also suggests that the system's quality measurement for a solution is not sufficient if no peer reviews for this solution are available. This issue is further analyzed in section 6.

5.4 Type-Specific Results

Finally, we investigated whether the system's quality heuristics differed between the task groups that were described in section 4.1 (research question H5). Figure 6 shows the results of this analysis. Based on the quality threshold of 0.2, the Figure illustrates that the system delivered sufficient results in all four task groups we tested. One may argue that this is a "borderline case" for the task types of text summary (where also the inter-rater reliability was lowest) and knowledge tests with guessing opportunity (where, by chance, few solutions received more than three reviews). Here, the heuristics differs from the expert grades by a little less than 0.2 after more than 3 and 5 reviews, respectively.

An ANOVA showed that the differences between task groups did not reach the level of statistical significance ($p>.5$). A further in-depth analysis yielded that the fine-granular system variant D tended to work better for the task group "text summaries", while the coarse-granular variant N worked better for the task group "knowledge tests without possibility for guessing". While these differences were not statistically significant either, they may still be used to inform the system improvement (cf. next section).

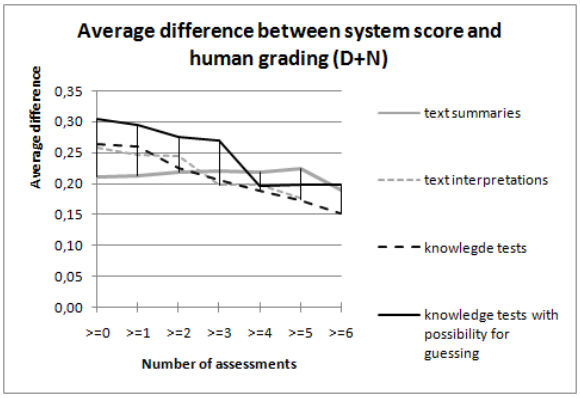

Figure 6. Results by task groups

6. Discussion

Overall, the results of the study confirmed our research hypotheses. In particular, the collaborative filtering algorithm delivered an acceptable prediction quality for student scores if enough peer reviews were available. Consistent with literature [10, 14], the quality of the peer reviews outperformed the student's self-assessments, and four to five peer reviews were sufficient for acceptable quality heuristics. We are also pleased to see that the collaborative filtering algorithm was suitable for different educational tasks, ranging from simple knowledge test questions (that could have been checked automatically as well) to ill-defined tasks dealing with the interpretation of a rather complicated text.

A surprise outcome was that the coarse grain variant N worked as good as variant D (and even better for some task groups). Apparently, the availability of more detailed options for grading peer solutions does not improve the system's overall quality heuristic. A simple good/bad scheme performed as well as a 10-pt Likert scale.

A possible explanation for this is the following: While a solution always got the extreme ratings 1 or 0 in variant N, students in the D condition frequently assigned values of 0.7 to 0.9 for solutions they considered good (and respectively 0.1 to 0.3 for presumably poor solutions), which should be investigated in detail in future work. Overall, this led to a need of more assessments in variant D to achieve extreme scores.

6.1 Base Rating Quality

In section 3.2, the base rating component of the collaborative filtering algorithm was described. The base rating assumes that students who identify the quality of a peer-provided solution correctly are able

to provide a high-quality solution themselves. As our analysis in section 5.3 shows, that assumption may not have been realistic – if it were, then one would have expected the algorithm to produce good predictions even with fewer peer reviews.

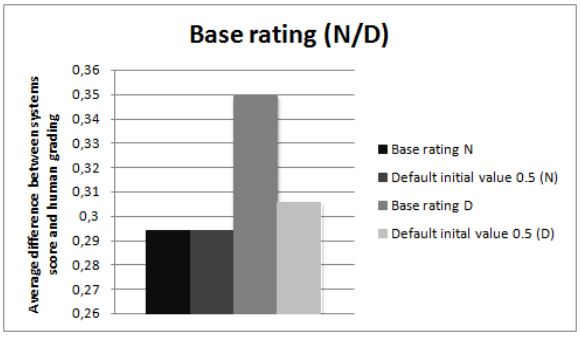

Figure 7. Base Ratings vs. Default Scores

Figure 7 shows a comparison between a default initial value of 0.5 and the base rating. In our study, a default initial value would have led to an average difference of 0.305 (in variant D) and 0.29 (in variant N). As the figure shows, the base rating in our algorithm performed equally to the default value (0.29 vs. 0.29) in variant N, but performed worse in variant D (0.35 vs. 0.305). This, in theory, means that the "base rating" part of the formula could be replaced with a default value without reducing the system quality. Practically, we argue that the base rating can indeed be improved. In our data analysis, we identified a lack of the algorithm in achieving extreme base rating scores, especially low scores. In variant D, where there were 142 solutions with expert grades less than 0.5, only 14 of these solutions had a base rating of below 0.5. In the N variant, this relation was 137 vs. 86.

The reason for this is the fact that the system provided a mixture of alternative solutions to assess. In the algorithm that selects the peer solutions that a student has to rate (which then determine his base rating), it was ensured that there were never three solutions of similar quality, and never more than one solution of middle-high quality. If, for example, a student receives solutions with scores 1.0, 0.67 and 0.0 for rating, then the worst choice he can make is to rate them as 0.0, 0.0, and 1.0. This will result in a base rating of $1 - ((1.0 + 0.67 + 1.0) / 3) = 0.11$. That is the theoretically lowest score for this example. If the scores of the items that are to be rated are not as extreme, then the range for the base rating gets even narrower. In variant D, this problem is amplified due to the fact that most students did not choose extreme scores during the evaluation process. In our data, 0.31 was the lowest base rating for the D variant, and in variant N it was 0.14.

6.2 Possible Improvements

Based on the results of section 6.1, an area for improvement of the algorithm is to allow for more extreme scores of the base rating, especially for the D variant. Another possibility to improve both algorithm variants is to offer students a point to pass a task if they have no idea about the solution, so that they will get a low score anyway. In the lab study, the participants had to work on each task even if they did not know a single part of the right solution. The tested algorithm is based on the assumption that anyone who knows the correct answer to a task will give a solution of good quality. However, if a student makes arbitrary guesses in his ratings, he may also hit the right answers, and then his score will start at a much higher value than "deserved".

Finally, the observed trend that the detailed rating options have advantages for the more ill-defined tasks (text summary), while the good/bad choices are preferable for simple knowledge tests and also for improving the base rating overall, can inform the design of collaborative filtering systems. While further studies are required, our results suggest that a simple clear-cut statement about what degree of detail for ratings is best for collaborative filtering systems in education cannot be made.

7. Conclusion

Collaborative filtering is a technology that has great potential for eLearning applications – it allows the construction of tools that enable learners to read and critique peer solutions (which not only disburdens the teacher, but also at the same time has positive effects on learning) and to get recommendations of good solutions for tasks that they may have problems with. As a prerequisite for building such collaborative filtering systems however, algorithms to assess the quality of a student solution based on peer ratings are required.

This paper presented such a collaborative rating algorithm which is similar to the reciprocal review system of SWoRD and PG, but differs in two respects: First, no textual reviews are required and only quick ratings are employed within the evaluation questions. While qualitative comments might be helpful for learners in order to help them improve their answer, our algorithm is geared towards delivering good prediction quality without much (time-consuming) input of learners. Another dif-

ference of our approach to other classical peer review systems is that a rating has immediate implications both for the *rated* solution (assuming that better solutions will receive more positive evaluations over time) and also the *rater's* own solution. The latter is based on the assumption that a student who recognizes good solutions is also more likely to have provided a good solution himself. Through active and passive evaluations (i.e., given and received ratings), our algorithm builds a quality heuristic for students' solutions without parsing the content of answers.

A controlled lab study confirmed that the system-generated quality heuristic correlated highly with an expert-provided manual grading of the solutions. This was true independent of whether the students provided fine-grained or coarse-grained evaluations of peer solutions, and independent of the task type that the students worked on (text summary, text interpretation or knowledge tests). The system heuristic outperformed the student's self assessment, and required only few peer evaluations (between 4 and 5) in order to achieve an acceptable prediction quality. This is an important result for the development of learning technology. In many domains, especially in ill-defined ones [5], tasks may not have clear-cut solutions and the automatic assessment of student-provided solutions through direct computer-based analysis may therefore be infeasible. As the study demonstrated, collaborative rating has the potential to solve this problem of assessing student solution quality, which is critical for educational technology development.

In our current work, we are integrating the described algorithm into a fully fledged web based eLearning system which uses collaborative filtering to provide students recommendations for task solutions. This system is available to students at Clausthal University as they prepare for a course examination in Information Systems. From this field study, we hope to gain further insights into how the system works in practical everyday usage.

8. References

[1] Goldberg, David; David Nichols, Brain M. Oki, Douglas Terry (1992). "Using collaborative filtering to weave an information tapestry". Communications of the ACM 35 (12): 61-70

[2] Dancer, W. T., & Dancer, J. (1992). Peer rating in higher education. Journal of Education for Business, 67, 306–309.

[3] Mathews, B. (1994). Assessing individual contributions: Experience of peer evaluation in major group projects. British Journal of Educational Technology, 25, 19–28.

[4] Hinds, P. J. (1999). The curse of expertise: The effects of expertise and debiasing methods on predictions of novice performance. Journal of Experimental Psychology: Applied, 5, 205–221.

[5] Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006). Defining Ill-Defined Domains; A literature survey. In Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems (p. 1-10). Jhongli (Taiwan), National Central University.

[6] Cho, K., Schunn, C. D. (2007). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. Computers & Education, Vol. 48 (3)

[7] J. Surowiecki (2004): The wisdom of crowds. Doubleday.

[8] Gehringer, E. F. (2000). Strategies and Mechanisms for Electronic Peer Review. In: 30th ASEE/IEEE Frontiers in Education Conference, p. F1B–2 – F1B–7.

[9] Gehringer, E. F. (2001). Electronic Peer Review and Peer Grading in Computer-Science Courses. In: Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education, February 2001, Charlotte, North Carolina, United States, pp. 139 – 143.

[10] Cho, K., Schunn, C. D., Wilson R. W. (2006). Validity and Reliability of Scaffolded Peer Assessment of Writing From Instructor and Student Perspectives. Journal of Educational Psychology, Vol. 98, No. 4, pp. 891–901

[11] Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006). Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques. In Proceedings of the 8th International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science 4053 (p. 227-236). Berlin (Germany), Springer.

[12] Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2007). Evaluating Legal Argument Instruction with Graphical Representations Using LARGO. In Proceedings of the 13th International Conference on Artificial Intelligence in Education (p. 101-108). IOS Press.

[13] Walker, A., Recker, M. M., Lawless, K., Wiley D. (2004). Collaborative Information Filtering: a review and an educational application. International Journal of AIED, Vol. 14, No. 1/2004 - pp. 3-28

[14] Stefani, L. A. J. (1994). Peer, self and tutor assessment: Relative reliabilities. Studies in Higher Education, 19(1), 69–75.

[15] Cronbach, L. J. (1951). Coefficient Alpha and the Internal Structure of Tests. Psychometrika, Vol. 16, No. 3.