Disburdening Tutors in E-Learning Environments via Web 2.0 Techniques

Frank LOLL and Niels PINKWART
Clausthal University of Technology, Germany

Abstract. Today, collaborative filtering techniques play a key role in many Web 2.0 applications. Currently, they are mainly used for business purposes such as product recommendation. Collaborative filtering also has potential for usage in "Social Semantic Web" e-learning applications in that the quality of a student provided solution can be heuristically determined by peers who review the solution, thus effectively disburdening the workload of teachers and tutors. This chapter presents a collaborative filtering algorithm which is specifically adapted for the requirements of e-learning applications. An empirical evaluation of the algorithm showed that the results of the collaborative filtering were more accurate than the self-assessment of the participants and that already four peer evaluations were generally enough to reach a satisfying accuracy. Based on these results, we developed a web based e-learning system (CITUC), which was successfully used in a university course in summer 2008. This chapter describes an evaluation of CITUC based on surveys, interviews and a detailed analysis of the system's usage by students. Our conclusion is that Social Semantic Web applications such as CITUC, which enable learners to review and comment on peer solutions, have high potential as a support for classic academic teaching in larger classes.

Introduction

The term "Social Semantic Web" describes an emerging design approach for building and using Semantic Web applications which employs Social Software and Web 2.0 approaches. In Social Semantic Web systems, groups of humans are collaboratively building domain knowledge, aided by socio-semantic systems [1]). The collaboration process of the users in Social Semantic Web systems can have multiple purposes – among them are the group based structuring of a domain (creation of domain ontologies) and the collaborative classification of content (determination of properties of ontology elements). Both of these are potentially valuable in educational settings. While the former can be a technique for collaborative knowledge building through jointly structuring an unknown knowledge domain, including the discussion of domain concepts and relations, the latter allows for jointly annotating or evaluating learning materials [2] and for heuristically determining the quality of task solutions through a collaborative effort.

This chapter presents an example for the latter approach. We present a system which is based on collaborative filtering (CF) algorithms [3]. This family of algorithms, which provide an essential base for the Web 2.0, is characterized by associations between users and system artifacts which are determined by explicit or implicit user actions and which are used to provide system services. Prominent examples for those associations are book orders at amazon.com, the input of user profiles in online dating sites as well as the tagging of pictures at flickr.com. All these applications have in common that the saved associations are used to recommend artifacts (books, potential partners, pictures, etc.). Although the calculation details vary between the systems, the underlying principle – the use of user information to assess or recommend artifacts in the system – is the same. In educational Social Semantic Web systems, CF algorithms have application potential: The quality of a student's task solution can be determined heuristically by assessments of other students (peer reviews) via CF techniques. In this case, the objective of the CF algorithm lies less in the calculation of a potential fit between users and artifacts (like in classical application areas) than rather in the estimation of a solution's quality. Such an approach is not unproblematic. Typical points of critique concerning a peer review approach in education are related to the students' lack of knowledge and experience in assessing task solutions and to the risks of intentional manipulation [4], [5]. Yet, this approach also has a lot of advantages in practice. It disburdens teachers and tutors from a lot of assessment tasks and at the same time it provides the possibility for students to train their evaluating and critiquing skills by assessing other students' solutions. If there are tasks which allow for more than one correct solution, then students have a chance to get to know different acceptable approaches and perceptions and have to compare them, which is beneficial for learning. Also, students can potentially empathize with other learners' problems easier and understand reasons for wrong task solutions sometimes better than experts, which can make their reviews sometimes more valuable than those of experts [6].

In summary, CF algorithms have potential as a tool in Social Semantic Web e-learning systems: they can allow learners to evaluate and annotate peer solutions and to build a semantic system heuristics based on multiple peer reviews. Literature shows that the resulting, collaboratively built, heuristics, could even lead to better annotations and evaluations of solutions than one single expert could do [7], [8]. In this chapter, we describe a CF heuristics which is especially designed for e-learning applications and the CITUC e-learning system which implements the heuristics in a practical context.

1. Collaborative Filtering in Existing Frameworks

In spite of their potential, CF mechanisms have only been rarely used in the e-learning sector until now, and there have only been a few empirical studies about the effectiveness of these methods.

One of the few existing systems is the web-based *PeerGrader* (PG) [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 [10] 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 [8], [11]. 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 Social Semantic Web techniques (peer reviews). An evaluation showed that the participants benefitted from multi-peers' feedback more than from single-peer's or single expert's feedback [8].

A different approach is used by the *LARGO* system [12], where students create graphs of US Supreme Court oral arguments. Within LARGO, collaborative scoring is employed for a group based assessment of the quality of "decision rules" student have included in their diagrams. Since this assessment involves interpretation of legal argument in textual form, it cannot be automated reasonably and is thus an ideal field for Social Semantic Web techniques. While the overall LARGO system has been tested in law schools and shown to help lower-aptitude learners [12], [13], 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) [2] 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 [2].

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).

2. Collaborative Filtering Heuristics

Based on the results of the existing e-learning systems reviewed above, we can state that the use of a combination of CF and peer reviews promises a benefit for classic environment. Yet, the existing systems have limitations in terms of generality and practical applicability. For this purpose we developed a heuristics which combines some of the features of PG, LARGO and SWoRD. Details of the CF heuristics will be described in the following - the main differences between our heuristics and the existing systems are:

- It is not constrained to a specified task area like the education of writing skills (SWoRD) or the education of argumentation in law (LARGO).
- There are no time-consuming re-writing phases and only short quality assessments on a Likert scale, but no detailed textual reviews.
- In our heuristics, peer assessments have an impact on the person who grades as well as on the solution that is graded.

2.1. Algorithm

The CF heuristics consists of two components – a *base rating* and an *evaluation rating*, which are finally merged into a *quality rating*. Figure 1 illustrates the workflow: In the first step a student works on a task and provides a solution. After that he assesses a couple of alternative solutions (in our lab study 3) for the same task. Based on his assessments, the heuristics calculates a first rating, the *base rating*, which is a result of the deviation between the solutions' *quality ratings* and the student's assessments. In the third step, other students assess the new solution, which results in the *evaluation rating*. Finally, the heuristics calculates the *quality rating* of the new solution based on the *base rating* and the *evaluation rating*. The underlying formulae for the *base rating*, the *evaluation rating* and the final *quality rating* will be described in more detail in the next subsections.

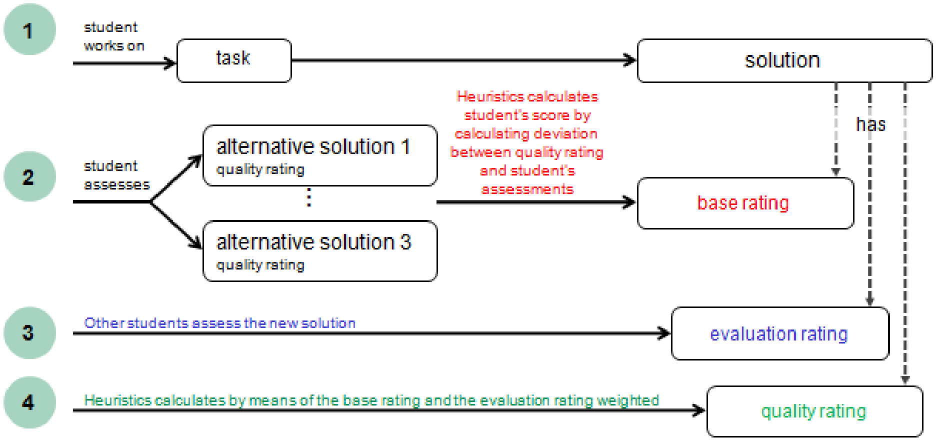

1
student works on
task
solution
has
2
student assesses
alternative solution 1
quality rating
alternative solution 3
quality rating
Heuristics calculates student's score by calculating deviation between quality rating and student's assessments
base rating
3
Other students assess the new solution
evaluation rating
4
Heuristics calculates by means of the base rating and the evaluation rating weighted
quality rating

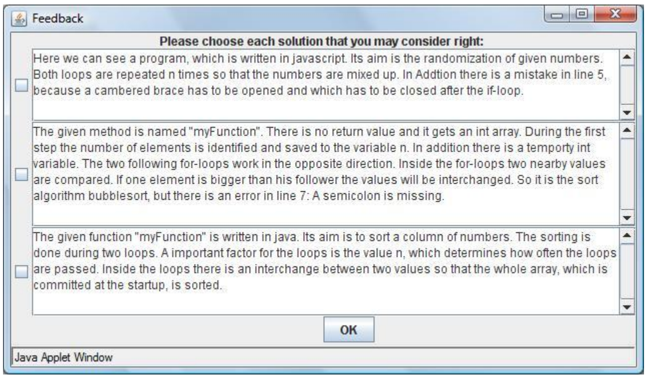
Figure 1. Heuristics' workflow

2.1.1. Base Rating

Based on the assumption that a student who can classify the quality of given alternative solutions correctly is also able to provide a high-quality solution himself, the heuristics first calculates a *base rating* for a student's solution. Once a student provided n assessments w_1, …, w_n for n other student's solutions (which have *quality ratings,* i.e. system's classification, of q_1, …, q_n themselves), the *base rating* is calculated by:

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

Here it is important to note that we tested two variants of the heuristics: variant N (normal) allowed only coarse-grain assessments of 0 (bad) and 1 (good) for all elements w_i, while system variant D (detailed) allowed a more fine-grain assessment in steps of 0.1. Figures 2 and 3 illustrate the different ways of assessment in the two algorithm variants.


Feedback

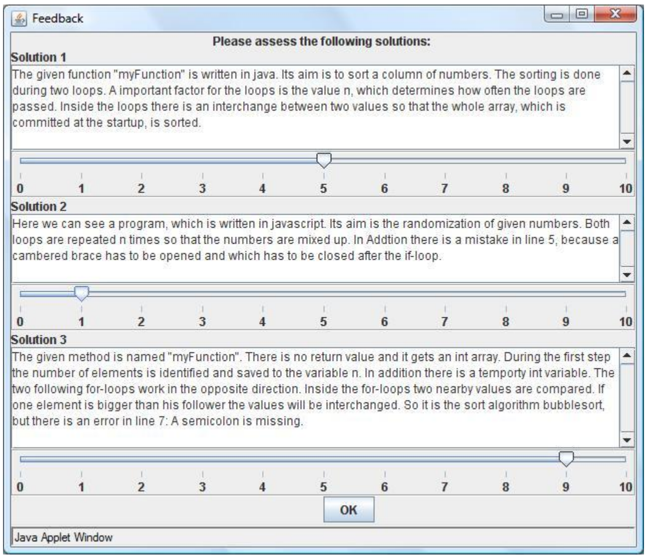
Please choose each solution that you may consider right:

□ Here we can see a program, which is written in javascript. Its aim is the randomization of given numbers. Both loops are repeated n times so that the numbers are mixed up. In Addtion there is a mistake in line 5, because a cambered brace has to be opened and which has to be closed after the if-loop.

□ The given method is named "myFunction". There is no return value and it gets an int array. During the first step the number of elements is identified and saved to the variable n. In addition there is a temporty int variable. The two following for-loops work in the opposite direction. Inside the for-loops two nearby values are compared. If one element is bigger than his follower the values will be interchanged. So it is the sort algorithm bubblesort, but there is an error in line 7: A semicolon is missing.

□ The given function "myFunction" is written in java. Its aim is to sort a column of numbers. The sorting is done during two loops. A important factor for the loops is the value n, which determines how often the loops are passed. Inside the loops there is an interchange between two values so that the whole array, which is committed at the startup, is sorted.

Figure 2. Solution assessment (variant N)[1]


Feedback

Please assess the following solutions:

Solution 1

The given function "myFunction" is written in java. Its aim is to sort a column of numbers. The sorting is done during two loops. A important factor for the loops is the value n, which determines how often the loops are passed. Inside the loops there is an interchange between two values so that the whole array, which is committed at the startup, is sorted.

Solution 2

Here we can see a program, which is written in javascript. Its aim is the randomization of given numbers. Both loops are repeated n times so that the numbers are mixed up. In Addtion there is a mistake in line 5, because a cambered brace has to be opened and which has to be closed after the if-loop.

Solution 3

The given method is named "myFunction". There is no return value and it gets an int array. During the first step the number of elements is identified and saved to the variable n. In addition there is a temporty int variable. The two following for-loops work in the opposite direction. Inside the for-loops two nearby values are compared. If one element is bigger than his follower the values will be interchanged. So it is the sort algorithm bubblesort, but there is an error in line 7: A semicolon is missing.

Figure 3. Solution assessment (variant D)

[1] The original user interface was in German language (for this figure and all others).

An example of how the algorithm works: Let's assume that, as shown in Figure 3, a student assesses the three given alternative solutions with scores w_1 = 0.5, w_2 = 0.1 and w_3 = 0.9. If the current system internal *quality ratings* of these three solutions (see section 2.1.3) are q_1 = 0.5, q_2 = 0.05 and q_3 = 0.95, then the *base rating b* for the student who makes the assessments is:

$$b = 1 - \frac{1}{3}\sum_{i=1}^{3}(|\, w_i - q_i) = 1 - \frac{1}{3}\left(|\, 0.5 - 0.5\,| + |\, 0.1 - 0.05\,| + |\, 0.9 - 0.95\,|\right) \approx 0.97 \tag{2}$$

This high *base rating* results from the fact that the student assessed the given solutions as correctly as possible (as compared to the *quality rating*). According to the assumption, he was thus probably able to provide a high-quality solution himself.

2.1.2. Evaluation Rating

The second component of the heuristics is the *evaluation rating*. Once a student has provided his solution (and has assessed some peer solutions), it is presented to other students to be assessed. All assessments get collected and averaged. Here, a weighting of assessments is made where assessments of better students get a higher weights. Thus, the *evaluation rating* is calculated as:

$$e = \frac{1}{\sum_{i=1}^{j} q_i}\left(\sum_{i=1}^{j} w_i q_i\right) \tag{3}$$

To illustrate the weighting, here is another example: Assume a solution gets four assessments w_1 *= 0.9*, w_2 *= 0.2*, w_3 *= 0.4* and w_4 *= 0.5* from students whose own solutions have internal system *quality ratings* of q_1 *= 0.8*, q_2 *= 0.1*, q_3 *= 0.3* and q_4 *= 0.7*. Then, the *evaluation rating e* for the assessed solution is:

$$e = \frac{1}{\sum_{i=1}^{4} q_i}\left(\sum_{i=1}^{4} w_i q_i\right) = \frac{1}{1.9}\left(0.9 \cdot 0.8 + 0.2 \cdot 0.1 + 0.4 \cdot 0.3 + 0.5 \cdot 0.7\right) \approx 0.63 \tag{4}$$

The first assessment gets a higher weight than the others because the student who provided it has a higher *quality rating* as compared to the others. His opinion is thus considered as more important than the other students' opinions by the system heuristics.

2.1.3. Quality Rating

Finally, the *base rating* and the *evaluation rating* are combined to a *quality rating*. The *evaluation rating* gets weighted dependent on the number of received assessments p for a solution. Its impact thus increases with an increasing number of assessments. The formula contains a constant *c* which corresponds to the number of presented alternative solutions in the dialogs (see Figures 2 and 3). In our example, we have *c=3* and *p=4*. Thus, the *quality rating* is calculated by:

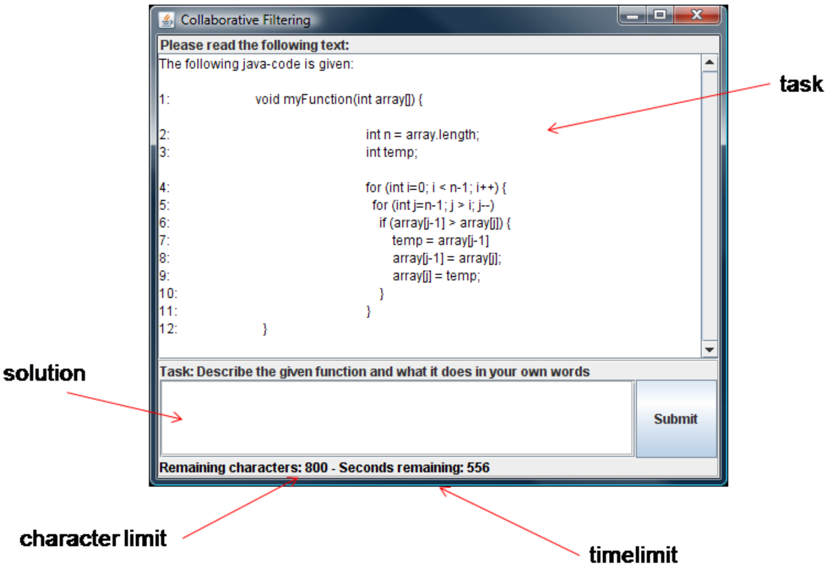
$$q = \frac{c}{p+c}b + \frac{p}{p+c}e = \frac{3}{4+3}0.97 + \frac{4}{4+3}0.63 \approx 0.73 \qquad (5)$$

2.2. Implementation

We developed a Java and XML based system to test the CF heuristics. After an initial login to the system, students go through the following phases as they use the system:

1. Work on task
2. Assess three alternative solutions for the just completed task (Figures 2 and 3)
3. Repeat steps 1 and 2 as long as there are tasks to complete
4. Self-assessment of their own solutions' qualities

Figure 4 shows the user interface with a sample task (in this case a task on Java programming) from the lab study we describe in section 2.4 in more detail. The users got a given text, a question as well as a time limit and a character limit for their solution. The limits were used as orientation guide for what kind of solution was expected.



Please read the following text:

The following java-code is given:

```
void myFunction(int array[]) {

        int n = array.length;
        int temp;

        for (int i=0; i < n-1; i++) {
          for (int j=n-1; j > i; j--)
            if (array[j-1] > array[j]) {
               temp = array[j-1]
               array[j-1] = array[j];
               array[j] = temp;
            }
        }
  }
```

Task: Describe the given function and what it does in your own words

Remaining characters: 800 - Seconds remaining: 556

Figure 4. User interface

2.3. Research Questions

To test whether our CF heuristics algorithm actually works, we investigated the following research questions:

1. *Does the heuristics correctly classify students' solutions in comparison to a manual grading of human experts?* This point is of course fundamental.

2. *Is the level of detail which is available in the peer assessments important?* This point is interesting since it is probably easier for students to perform a coarse-grain assessment (like right or wrong) than a fine-grain assessment (like an assessment on a 10pt Likert scale).
3. *Does the heuristics' performance dominate the participants' self-assessment in comparison to a manual expert's assessment?* This aspect is interesting, because a participant's self-assessment is usually much easier to get than multiple peer ratings.
4. *Will the estimation quality of the heuristics improve with a growing number of peer assessments?* This is typical for CF algorithms in other domains so that we hypothesize that it will improve. The critical part of the question is the number of assessments which is needed to achieve sufficient quality. If the number is low, then the algorithm will also be applicable in small learning groups where only few peer assessments are possible.
5. *Does the heuristics' estimation quality depend on the task type?* While for well-defined tasks, students only have to compare between the right solution and the solution to be assessed (if the student knows the right solution), more work is required for ill-defined tasks where students have to think about other students' viewpoints, since multiple solutions could be acceptable. Thus it is a priori unclear if the heuristics will be suitable also for those task types.

2.4. Study Description

To answer the research questions, we conducted a controlled lab study in May 2008 at Clausthal University of Technology with 45 participants, including 18 female and 27 male students. The participants were assigned randomly to the two system variants D and N, resulting in 7 female and 15 male participants in variant N and 11 female and 12 male participants in variant D. The participants were volunteers from different domains (e.g., computer sciences, physics or business) in different stages of their studies, i.e. there were first semester bachelor students as well as advanced diploma and PhD students. All participants were recruited via public announcements on the local newsgroups or e-mail lists and were paid for their time. The students had to work on 12 tasks from various knowledge areas. The tasks were of the following types:

1. Text summaries
2. Text interpretations
3. Knowledge tests without possibility to guess
4. Knowledge tests with possibility to guess

In the first task type (text summaries), the participants got articles dealing with different topics (e.g. a text about Second Life). These articles differed in their level of complexity and required, at least in parts, domain-specific knowledge to get the main points, which had to be summarized in a short text. The second task type (text interpretations) focused a fact-based news article about the take-over of DoubleClick by Google. The participants were asked to mention and discuss possible concerns towards privacy of customers based on the facts in the text. The third task type (knowledge tests without possibility to guess) consisted of five tasks where guessing was not possible (e.g. the calculation of a derivative of a function to calculate the slope at a given coordinate).

The fourth and final task type (knowledge tests with possibility to guess) consisted of problems which could at least be approximated by logical deduction even without specific domain-knowledge. An example here was the estimation of the population of Austria by means of a text about the size of the country.

The students had an overall time limit of 75 minutes. Furthermore, each task had a character limit as well as a time limit (see section 2.2). All participants were instructed to assess alternative solutions even if they did not know the correct solution for a task. To solve the cold-start problem [14] and offer alterative solutions also for the first participants who took part in the study, we provided 3 alternative solutions of different quality per task.

2.5. Results

To evaluate the results of the heuristics, all solutions were manually graded independently by two human experts (a professor of computer science and an advanced graduate student) on a scale from 0 to 10. To check whether the human grader's assessments were similar (if human graders disagree, then a realistic baseline for the heuristics is hard to define), we first calculated inter-rater reliability based on Cronbach's Alpha [15].

Table 1. Inter-rater reliability of human graders by means of Cronbach's alpha

	Task Group	α
1.	Text summaries	0.834
2.	Text interpretations	0.888
3.	Knowledge tests without possibility to guess	0.982
4.	Knowledge tests with possibility to guess	0.932

As Table 1 shows, of the agreement between the two graders was acceptable (and even excellent for the knowledge tests). Therefore, we averaged both human grader's scores and used the resulting "human grading" as a baseline for comparisons to the self-assessment of the students and to the results of the systems' *quality rating*.

As a next step, we needed to define what we consider as an acceptable level of deviation between the system's *quality rating* and the human grading. In this context, it is important to note that despite their overall agreement, the human graders still had slight differences between their grading, especially in the more ill-defined tasks. Thus it is not realistic to define the acceptable level of deviation as 0.0: This was achieved between the human graders only in 44.6% of the cases. Considering the fact that a random system assessment would have led to an expectation value for the difference of E[X] = 0.33 and a static default value of 0.5 would have led to an expectation value of 0.25 in theory and to E[X] = 0.305 (in variant D) and E[X] = 0.29 (in variant N) for our data set, we set the maximum acceptable deviation to 0.2. This choice is supported by the agreement between the human graders, who differed by more than 0.1 in 21.2% of the cases, but by more than 0.2 only in 11.5% of the cases. Using a limit of 0.2 thus, in our view, is an acceptable compromise between being unrealistically strict (so that even humans would not agree to this extent) and overly relaxed (so that even guessing would fulfill the criteria).

2.5.1. General Heuristics' Quality

To analyze the overall quality of the heuristics, we compared the average deviation between system score and human grading to the number of assessments. Figure 5 gives an overview about both systems variants.

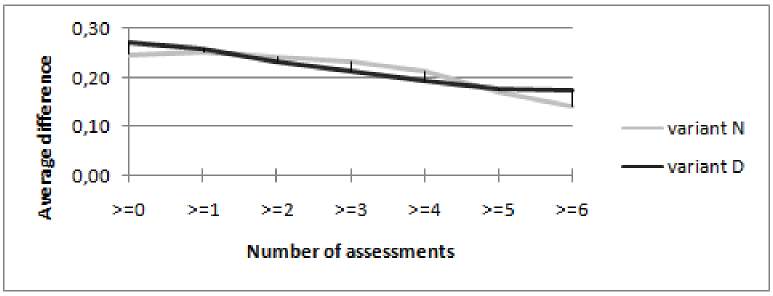


Figure 5. Heuristics quality measured by average difference between system score and human grading

The ordinate shows the average deviation between system's *quality rating* and the human grading, while the abscissa shows the minimal number of assessments a solution received. An example: The average difference between the system's *quality rating* and the human grading was 0.25 in variant N when only considering those solutions which had been assessed at least once.

Based on the quality threshold of 0.2 discussed above, Figure 5 shows that both variants of the heuristics provided acceptable results when a sufficient number of assessments were available. This was independent of the educational background (including their major topic and semester) of the participants. Thus we can answer our first research question: The heuristics is able to classify solutions correctly. As expected, Figure 5 also shows that the average deviation between heuristics and human grading decreases continuously with an increasing number of available peer assessments, resulting in an improved prediction quality of the heuristics (research question 4). Four (variant D) to five (variant N) assessments of participants were enough to achieve a prediction quality which differed from the human grade by not more than 0.2. Yet, Figure 5 also shows that the heuristics' quality is not sufficient if it is based only on the *base rating*, i.e. if there are no assessments from the participants'. We will discuss this later in section 2.6.

2.5.2. Influence of Assessment's Granularity

Next, we checked how the degree of granularity that was available for the peer assessments influenced the resulting heuristics' quality (research question 2). As Figure 5 suggests, there is no significant difference between both variants. This was confirmed by an ANOVA: The differences between the system's *quality rating* and the human grading were statistically not significant, i.e. F (1, 538) = 2.69, p > 0.1 for all solutions and F (1, 31) = 0.71, p > 0.4 for solutions with six or more assessments. Since both variants provided sufficient results and did not differ significantly, we used the combined results of both variants (D+N) in the following.

2.5.3. Heuristics Quality vs. Self-Assessment

To investigate whether the heuristics outperforms the participants' self-assessments (research question 3), we compared their average deviation to the experts' grades.

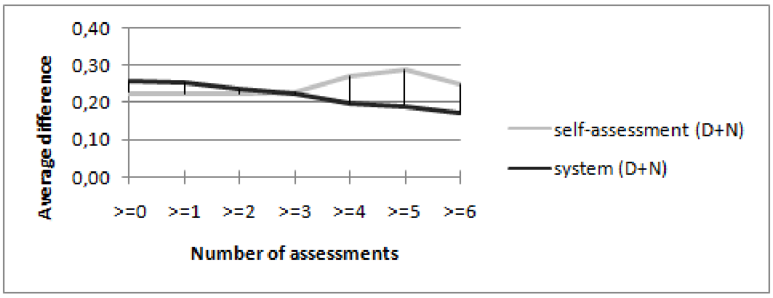


Figure 6. Average deviation between system's quality heuristics and participants' self-assessment

As shown in Figure 6, the heuristics outperformed the participants' self-assessments when three or more assessments were available for a solution. A t-test showed that this result is statistically significant (p < 0.05 for solutions with at least 4 assessments).

2.5.4. Task Group Dependency

Finally, we looked at the differences of the heuristics' quality between the four different task types (research question 5). As Figure 7 illustrates, the system provided satisfying results in all task groups, however it took more peer assessments for text summaries and for knowledge tests with possibility to guess. An ANOVA however showed that the differences between the task types were not statistically significant (p>0.5).

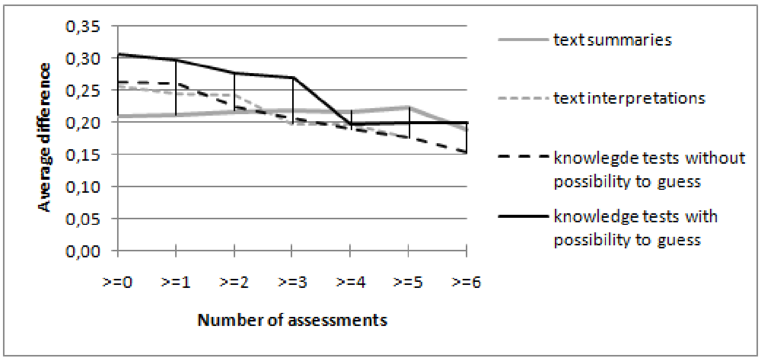


Figure 7. Results of system's quality heuristics depending on task group

2.6. Discussion

Overall, the pilot study confirmed our expectations. The collaborative filtering heuristics provided acceptable quality assessments for participants' solutions when enough, i.e. four to five, assessments were available. Confirming findings in literature [11], [16], the participants' self-assessments were qualitatively beaten by the peer assessments. The heuristics turned out to be adequate for different types of tasks, starting from well-structured knowledge tests, where solutions could be checked automatically, to ill-defined tasks like interpretations of rather complicated texts.

Contrary to our expectations, both variants (D and N) were on a similar quality level – so it did not make a difference whether peer grades were given on a coarse grain scale or on a fine grain scale. One possible explanation for this might be the fact that, while variant N did not allow for “medium” ratings, students in variant D tended to prefer less extreme scores (like 0.7 to 0.9 for good solutions and 0.3 to 0.1 for bad solutions). This finally led to a need for more assessments in variant D to achieve extreme scores of <0.2 or >0.8.

One aspect of the heuristics that was not confirmed by our study is related to the *base rating*. In section 2.1.1, the heuristic’s *base rating* was described: Based on the assumption that a student who can classify the quality of a given solution correctly is able to provide a high-quality solution himself, the *base rating* assigns a first quality score to a student’s solution even though it has not been reviewed by peers yet. Unfortunately, our analysis showed that this goal could not be fully achieved. Figure 8 shows the average deviation between the *base rating* and the human grading for both system variants. We compared it to a default initial value of 0.5 which results in an average difference to the human grading of 0.305 in variant D 0.29 in variant N. As the diagram shows, the base rating delivered comparable results to a default initial value in variant N and even worse results in variant D. Thus, theoretically, the *base rating* formula could have been replaced by a constant to improve system’s quality.

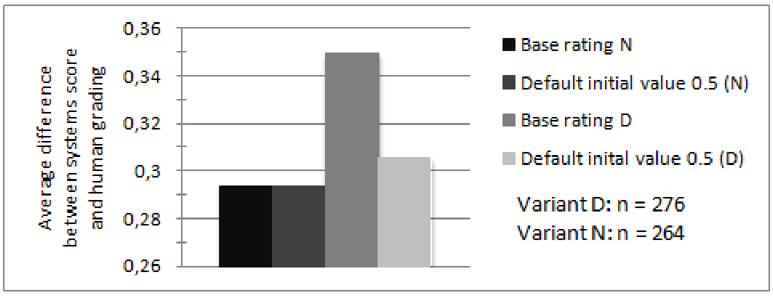


Figure 8. Comparison between base rating quality and default initial values, measured by average deviation to the human grading

But the *base rating* formula can be improved. In the study it became apparent that the major weakness of the *base rating* lies in its lack of achieving extreme (especially extremely low) scores. In variant D, there were 142 solutions which got a human grading of < 0.5. However, there were only 14 solutions which got a *base rating* of less than 0.5. In variant N, this effect was less extreme but still observable (137 to 86). The main reason for this effect can be found in the combination of alternative solutions. The problem lies is the following: Assume a participant got three solutions with *quality ratings* of 1.0, 0.67 and 0.0. Based on the worst imaginable assessment, i.e. 0.0, 0.0 and 1.0, the *base rating* results in:

$$b = 1 - \frac{1}{3}\sum_{i=1}^{3}(|w_i - q_i|) = 1 - \frac{1}{3}\left(|1.0 - 0.0| + |0.67 - 0.00| + |0.0 - 1.0|\right) \approx 0.11 \qquad (6)$$

Thus we know that it is not possible to achieve a lower score than 0.11 for the *base rating* in this constellation – this gets worse the more medium the quality ratings of the solutions to be graded are. In variant D, this problem is amplified by the participants' trend to avoid extreme assessments (as discussed before). Concretely, the lowest *base rating* achieved in our study in variant N was 0.14, and it was 0.31 in variant D. Hence here is potential for improvements.

3. The CITUC System

Based on the promising results of the lab study, our next step was to develop an e-learning system for practical use to test the heuristics in a more realistic setting. The resulting system called CITUC (Collective Intelligence @ Technical University of Clausthal) was intended to support students in their preparation for a final exam without increasing the workload of tutors.

3.1. System Modifications

To improve the *base rating* (cf. section 2.6), the algorithm was changed in a way that allows for achieving extreme scores independently of the *quality ratings* of the alternative solutions to be graded (even if they are near to 0.5). Thus we modified the *base rating* formula in the following way:

$$b_{new} = 1 - \frac{1}{n}\sum_{i=1}^{n} \frac{|\,w_i - q_i\,|}{\max(q_i, 1 - q_i)} \tag{7}$$

The advantage here is that it is possible to achieve extreme scores due to the linear scaling. Therefore, *base ratings* from 0.0 or 1.0 are always possible. To illustrate this: Assume there are solutions with *quality ratings* of q_1 = 0.35, q_2 = 0.6 and q_3 = 1.0. The worst ratings a user might make here are w_1 = 1.0, w_2 = 0.0 and w_3 = 0.0. In the old *base rating* this would have led to a *base rating* b_{old}:

$$b_{old} = 1 - \frac{1}{3}\sum_{i=1}^{3}(|\,w_i - q_i) = 1 - \frac{1}{3}\big(|\,0.35 - 1.0\,| + |\,0.6 - 0.0\,| + |\,1.0 - 0.0\,|\big) = 0.25 \tag{8}$$

This user would thus have got a far too high *base rating* score (0.25) with respect to his poor assessments. The new *base rating* b_{new} corrects for this:

$$b_{new} = 1 - \frac{1}{3}\sum_{i=1}^{3}\frac{(|\,w_i - q_i\,|)}{\max(q_i, 1 - q_i)} = 1 - \frac{1}{3}\left(\frac{|\,0.35 - 1.0\,|}{\max(0.35;0.65)} + \frac{|\,0.6 - 0.0\,|}{\max(0.6;0.4)} + \frac{|\,1.0 - 0.0\,|}{\max(1;0)}\right) = 0.00 \tag{9}$$

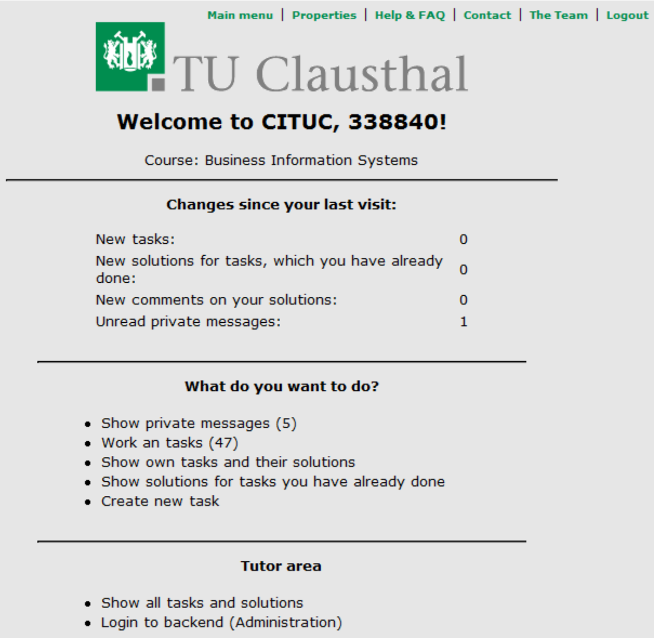
Another starting point for improvements is to offer the option to skip tasks if a student is not able to provide at least a basic solution. This appeared repeatedly in the lab study for the task type 3: knowledge tests without possibility to guess. Here, it was possible to get a high *base rating* by lucky guessing. This led to mistakes in the system's heuristic which lasted until enough peer assessments were available to filter this failure out. To exemplify: In some cases, solutions like "no idea" got a high *base* *rating* due to "good guessing". This then led to a low *base rating* for other participants who correctly assessed this solution as bad.

This propagation of mistakes could have been avoided by giving a possibility to skip tasks. In our concrete use case for CITUC (helping with the preparation for a final exam), a required sequential working through tasks would be misplaced. The problem described here was solved by allowing students a free choice among tasks to work on.

Based on the results presented in section 2.5, the number of solutions which had to be assessed by students was set to 5 to get a more reliable *quality rating*. We also opted for using the system variant D because this provided slightly better results in ill-defined tasks (text interpretations) and results of similar quality in the other categories.

3.2. CITUC System Description

CITUC was implemented as a web based system using PHP and a relational database for data storage. In addition to the "core functions" of entering and assessing solutions, the system offered facilities to comment solutions, to exchange private messages (for private call backs to comments) and e-mail notifications as awareness messages once new tasks, messages or comments were available or if there were new solutions for tasks that a student had already completed. After the login to the system by an anonymous identification number, the portal presented students some personalized awareness messages and a menu with options what to do next (see Figure 9).


Main menu | Properties | Help & FAQ | Contact | The Team | Logout
TU Clausthal

Welcome to CITUC, 338840!

Course: Business Information Systems

Changes since your last visit:

New tasks:	0
New solutions for tasks, which you have already done:	0
New comments on your solutions:	0
Unread private messages:	1

What do you want to do?

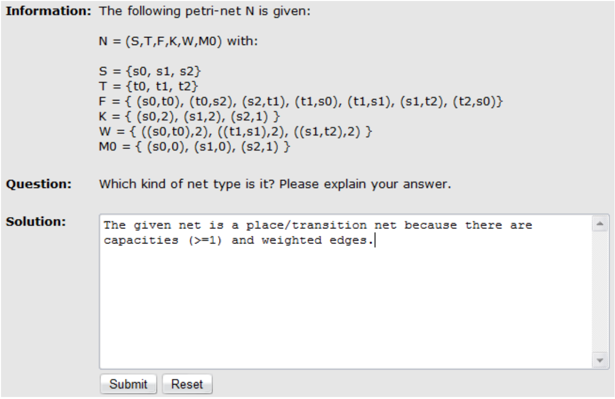
- Show private messages (5)
- Work an tasks (47)
- Show own tasks and their solutions
- Show solutions for tasks you have already done
- Create new task

Tutor area

- Show all tasks and solutions
- Login to backend (Administration)

Figure 9. CITUC: User-interface with awareness information

The most important point in the menu is the work on the tasks. After selecting it, the user will get a list of all tasks that the system offers (set up by a tutor or by other users) so that he can choose which task he wants to work on. After providing a solution for the task (see Figure 10), the user will see alternative solutions from other students, anonymously presented. Analog to the study’s variant D, he has to assess these these solutions on a scale from 0 (poor) to 10 (good). In addition, he has the possibility to add comments to each solution to help the respective author of the presented solution to recognize his possible mistakes (see Figure 11).



Information: The following petri-net N is given:

N = (S,T,F,K,W,M0) with:

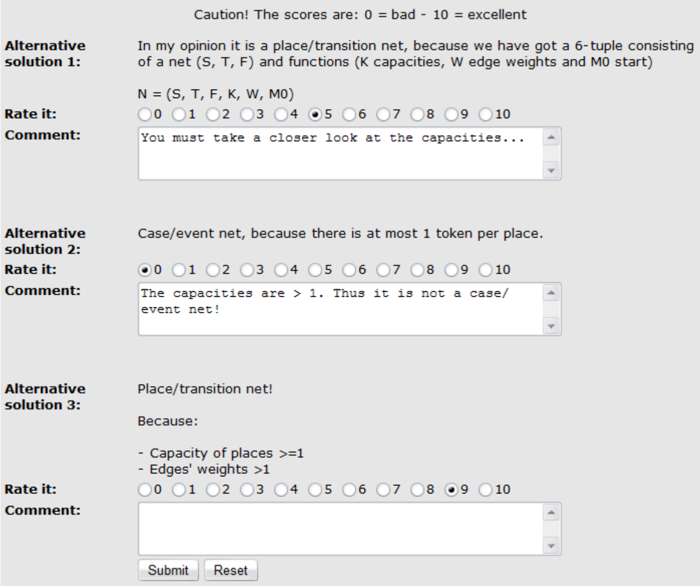
S = {s0, s1, s2}
T = {t0, t1, t2}
F = { (s0,t0), (t0,s2), (s2,t1), (t1,s0), (t1,s1), (s1,t2), (t2,s0)}
K = { (s0,2), (s1,2), (s2,1) }
W = { ((s0,t0),2), ((t1,s1),2), ((s1,t2),2) }
M0 = { (s0,0), (s1,0), (s2,1) }

Question: Which kind of net type is it? Please explain your answer.

Solution: The given net is a place/transition net because there are capacities (>=1) and weighted edges.

Figure 10. CITUC: Working on task

For each completed task, the user can take a look at all other solutions with their *quality ratings* and their comments for the respective task. Here, it is possible for the users to communicate via private messages or to add further comments.



Caution! The scores are: 0 = bad - 10 = excellent

Alternative solution 1: In my opinion it is a place/transition net, because we have got a 6-tuple consisting of a net (S, T, F) and functions (K capacities, W edge weights and M0 start)

N = (S, T, F, K, W, M0)

Rate it: ○0 ○1 ○2 ○3 ○4 ●5 ○6 ○7 ○8 ○9 ○10

Comment: You must take a closer look at the capacities...

Alternative solution 2: Case/event net, because there is at most 1 token per place.

Rate it: ●0 ○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9 ○10

Comment: The capacities are > 1. Thus it is not a case/event net!

Alternative solution 3: Place/transition net!

Because:

- Capacity of places >=1
- Edges' weights >1

Rate it: ○0 ○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ●9 ○10

Comment:

Figure 11. CITUC: Assessment of alternative solutions

As indicated before, the system offers students to enter tasks. This option was included to allow students to enter problems they may have had encountered during their exam preparation (to see how other students deal with these problems).

Nevertheless there is a roles management: the system differentiates between administrators, tutors and students. The first two groups have access to all tasks with their solutions and comments (see tutor area in Figure 9).

3.3. Research Questions

In our research, we focused on the investigation of the following questions:

1.) To what extent is the heuristics' *quality rating* ready for use in practice? Does a usage in a real context confirm the results from the previous lab study?

2.) Does the system have the potential to replace classical tutorials for exam preparation? Is the student's motivation to use the system on a voluntary basis sufficient, (usage frequency) is CITUC considered as helpful by the users (usage quality), and does it actually help students (effectiveness)?

3.4. Study Description

The CITUC system was used in the course "Business Information Systems II: Modeling of Information Systems" at Clausthal University of Technology in summer 2008. The course was attended by Business Information Systems students as well as Management and Economics students in the first semesters.

The system was made available after a short introduction in the last course lecture. It was available for approximately six weeks until the day of the course exam. The participation was voluntary. To motivate the students to use CITUC, e-mail reminders were sent at intervals of 2 weeks. 98 users were finally registered in the system, 85 students took part in the final exam. Overall, there were 50 tasks in the system: 22 of them were known to the students since they were taken from previous tutorials (they were put in to familiarize the students with the system) and 27 new tasks were explicitly marked as exam preparation tasks. One task was entered by a student. A few days before the final exam, the participants were asked to fulfill an online survey to assess the CITUC system. 29 of the 98 students participated in this.

3.5. Results

The following sections summarize the results of the system's evaluation parted according to the research questions, i.e. (1) performance of the heuristics in real settings, (2a) usage frequency of the system, and (2b) system's quality and effectiveness.

3.5.1. Performance of the Heuristics

To investigate the heuristics' classification performance, we looked at the 30 worst and 30 best solutions (according to the system's *quality rating*). Among the 30 worst solutions, there were 83% "spam", i.e. solutions like "foo". These "spam" answers were given by students who, apparently, wanted to look at other student's solutions (and had to provide their own one to do so). Therefore we can note that the heuristics is capable to filter out this kind of spam successfully. The remaining other solutions, in the "poorest 30" were classified correctly as being of low quality, too. Within this "poorest 30" set, the mean value of the quality ratings was m=0.087 (sd=0.034) and the mean value of the according *base rating* was m=0.238 (sd=0.179), which indicates that the *base rating* was improved as compared to the lab study. A similar picture was drawn when looking at the top 30 solutions. Among them, there was a single spam solution with a high *base rating*, which could be ascribed to excellent guessing on the student's side, but this solution did not receive any other assessments of other students until the end (it was one of the last ones entered), so that the *base rating* was the only available score. The other 29 solutions in the "top 30" set were classified correctly and received 5 assessments each. The mean *quality rating* value in this set was m=0.914 (sd=0.025). The mean value of the *base rating* was m=0.747 (sd=0.139). Thus, the heuristics confirmed the results of the lab study, but now even the *base rating* provided very good classifications.

3.5.2. Frequency of Use

As Figures 12 and 13 show, the system was used mainly during the last 1.5 weeks before the final exam. The last day before the exam had most logins (see Figure 12) and most provided solutions (see Figure 13) at the last day before the final exam. The small peaks in the system's use in the first days of use as well as after two weeks can be explained by the reminder emails. We conclude from this usage pattern that a pure voluntary use of the system was – at least within this course – a sufficient motivation for the students to use the tool during the exam preparation phase (yet not throughout a longer period).

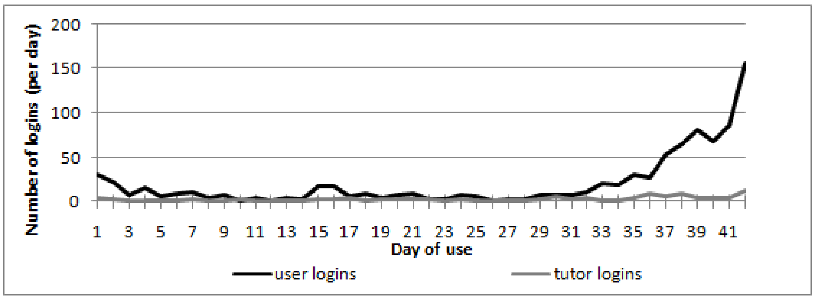


Figure 12. CITUC: Logins per day

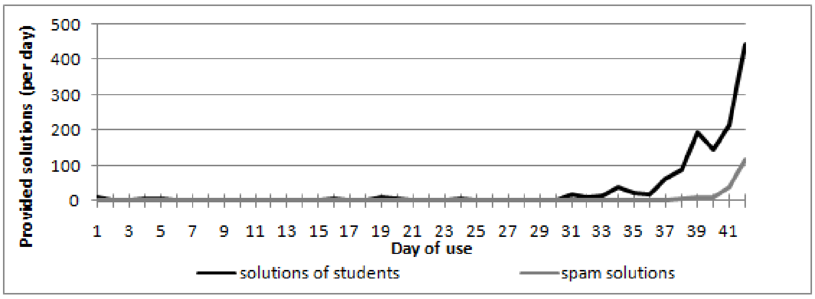


Figure 13. CITUC: Provided solutions per day

Figure 13 also shows the main advantage of the system as compared to the existing approaches (PG, SWoRD): Even solutions which were posted at the last day got feedback via comments and system ratings. Thus, nearly all students (but the last one) received feedback until the "last minute".

3.5.3. Students' Opinions

The results of the online survey show that the students found the system useful. They graded it with m=3.89 (sd=0.766, n=26) on a scale from 1 (very useless) to 5 (very useful). The question about the usefulness of the comment function drew a similar picture (m=3.556, sd=0.974, n=27). To the question if the CITUC system is a good preparation for the final exam 18 students voted for "yes", while 3 students voted for "no". It is important to note that the latter ones did not use the system at all, i.e. they registered to the system, but did not work on even one single task. A question about usability of the system resulted in an average value of m=3.704 (sd=0.993, n=27).

We noticed that not all students understood the sense of the system. A few of them thought in the "traditional" pattern where students work on a task and after that a tutor corrects their solutions or at least presents sample solutions. These students repeatedly asked for sample solutions, even if there were solutions in the system with an excellent score and content. Only after a written confirmation of a tutor that the online solution provided by another student was correct, they believed in it. So they wanted a clear sign that a solution is some kind of sample solution.

3.5.4. Tutor's Opinion

An interview with the course tutor showed that he believed that his workload was approximately equal to before (where he held classical tutorials instead of feeding tasks into CITUC), but the main advantage in the CITUC system was the possibility to handle more tasks than in a 90 minutes tutorial. In the tutor's opinion, the utility of CITUC was confirmed. Furthermore, he stated that the system allowed for addressing specific weaknesses "on the fly" during the course, which is not always possible in classical tutorial groups which have to be planned in advance. Concerns were mentioned by the tutor with respect to of solution assessment: He was not sure about whether students would also provide high-quality assessments if the tasks were more complicated and the solutions were longer. In our current setting we could not confirm or falsify this point, because most of the tasks were rather short.

3.5.5. System Effectiveness

Out of the 98 registered users in the CITUC system, 79 took part in the final exam. Overall there were 85 participants in the final exam, i.e. 6 participants did not register in the system. The achieved average score of all participants were 3.282[2], the average result of the CITUC users was 3.266 and there were no significant differences between students majoring in different topics. The correlation between the number of logins to the system and the exam's results was r=-0.1546, while it was r=-0.1504 between provided solutions in the system and exam's results. Both values suggest a trend in the desired direction (higher grade of use would lead to a better exam's result), but are clearly not statistically significant.

We investigated deeper and classified the users into active (more than average use) and passive users (no usage or less than average usage) dependent on their grade of

[2] 1 = A (very good), 2 = B (good), 3 = C (satisfying), 4 = D (sufficient), 5 = E (insufficient)

activity. Furthermore we divided the active users into three subgroups, i.e. low, medium and high, as shown in Table 2.

Table 2. CITUC user classification by means of their rate of use

Classification	Rate of Use	Characteristics	#
passive use	-	< 7 solutions, 4 logins	53
active use	low	≥ 7 solutions, 4 logins	11
	medium	≥ 14 solutions, 8 logins	24
	high	≥ 28 solutions, 16 logins	10

Out of the 45 active users, 44 took part at the exam. 41 passive users participated in the exam (35 of 53 with system logins, plus 6 who never logged in). The average result of the active users was 2.993 (sd = 1.344) compared to the passive users' result of 3.57 (sd = 1.42). Thus the latter clearly achieved a worse result. Again, this is not statistically significant, but still a noteworthy trend.

The failure rate was analogue: 20.4% of the active users failed in the exam, as compared to 45.71% of the passive users. Clearly, these findings are of correlational (not causal) nature, and the exam results depends on multiple factors beyond CITUC usage, but these results might be seen as indication that the system has some educational value.

4. Conclusion

The CITUC system, presented in this paper, is an example of a system which allows a student group to collaboratively build knowledge by classifying and annotating various (student provided) solutions to problems. CITUC uses CF algorithms in combination with peer reviews to address tutor workload issues in learning environments. In a controlled lab study, the CITUC heuristics provided ratings of sufficient quality (as compared to expert provided grades) and outperformed the participants' self assessments significantly when four or more assessments for each solution were available. The heuristics also proved its suitability for daily use beyond the limit of the study and provided persuasive classification results of student solutions in a field study. It thus has application potential for Social Semantic Web systems. Problems were identified in a lack of motivation to use the system among the students (apart from the last 2 weeks before the exam) as well as in the use of backdoors to get access to other students' solutions without providing content oneself. CITUC was assessed as helpful by the students and by the tutor, and an active usage of CITUC was correlated with better exam results.

References

[1] Morville, P. (2005). Ambient Findability. O'Reilly Media.

[2] Walker, A., Recker, M. M., Lawless, K., Wiley D. (2004). Collaborative Information Filtering: a review and an educational application. International Journal of AIED 14(1): 1-26. IOS Press.

[3] Goldberg, D., Nichols, D., Oki, B. M., Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM 35 (12): 61-70.

[4] Dancer, W. T., Dancer, J. (1992). Peer Rating in Higher Education. Journal of Education for Business 67(5): 306-309.

[5] Mathews, B. (1994). Assessing Individual Contributions: Experience of Peer Evaluation in Major Group Projects. British Journal of Educational Technology, 25(1): 19-28.

[6] 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(2): 205-221.

[7] Surowiecki, J. (2004). The Wisdom of the Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Doubleday.

[8] Cho, K., Schunn, C. D. (2007). Scaffolded Writing and Rewriting in the Discipline: A Web-Based Reciprocal Peer-Review System. Computers & Education, 48 (3): 409-426.

[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] Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006). Defining Ill-Defined Domains; A Literature Survey. In In V. Aleven, K. Ashley, C. Lynch, & N. Pinkwart (Eds.), Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems (pp. 1-10). Jhongli (Taiwan).

[11] 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, 98(4):891–901.

[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 (pp. 101-108). IOS Press.

[13] Pinkwart, N., Aleven, V., Ashley, K., Lynch, C. (2006). Schwachstellenermittlung und Rückmeldungsprinzipen in einem intelligenten Tutorensystem für juristische Argumentation. In: M. Mühlhäuser, G. Rößling, & R. Steinmetz (Eds.), GI Lecture Notes in Informatics - Tagungsband der 4. e-Learning Fachtagung Informatik, S. 75-86. Bonn (Deutschland), Gesellschaft für Informatik

[14] Maltz, D., Ehrlich, E. (1995). Pointing the Way: Active Collaborative Filtering. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

[15] Cronbach, L. J. (1951). Coefficient Alpha and the Internal Structure of Tests. Psychometrika, 16(3): 297-334.

[16] Stefani, L. A. J. (1994). Peer, Self and Tutor Assessment: Relative Reliabilities. Studies in Higher Education 19(1):69-75