How the Granularity of Evaluation Affects Reliability of Peer-Assessment Modelization in the OpenAnswer System

Maria De Marsico[1], Andrea Sterbini[1], and Marco Temperini[2]

[1] Dept. of Computer Science, Sapienza University of Rome, Italy
[2] Dept. of Computer, Control, and Management Engineering,
Sapienza University of Roma, Italy
{demarsico,sterbini}@di.uniroma1.it, marte@dis.uniroma1.it

Abstract. The OpenAnswer system has the goal of exploiting teacher mediated peer-assessment for the evaluation of answers to open ended questions. The system models both the learning state of each student and their choices during peer-assessment. In OpenAnswer, each student is represented as a Bayesian network made of a triple of finite-domain variables: K for student's Knowledge about a topic, J for the estimated ability to evaluate ("Judge") the answer of another peer, C for Correctness of the answer to a given question. The student's individual sub-networks are connected through further Bayesian variables which model each peer-assessment choice, depending on the type of peer-assessment performed: (G for grading, B for choosing the best, W for choosing the worst). During an assessment session, each student grades a fixed number of peers' answers. The final result for a given session is a full set of grades for all students' answers, although the teacher had actually graded only a part of them. The student's assessments are instantiated in the network as evidence, together with the teacher's (perhaps partially complete) grades, so that OpenAnswer deduces the remaining grades. In the former OpenAnswer implementation, all variables were represented through a probability distribution over three values (Good/Fair/Bad for K and J, correct/fair/wrong for C). We present experiments and simulations showing that, by increasing the domain granularity for all variables from 3 to 6 values (A to F), the information obtained from the Bayesian network achieves higher reliability.

Keywords: assessment, peer-assessment, social collaborative e-learning.

1 Introduction

The assessment of the actual knowledge achieved by learners is one of the hardest problems to address in education. In face-to-face oral examinations as well as in articulated written essays, the pedagogical experience as well as the deep knowledge of the learning domain can guide the teacher. Among the other possible forms of assessment, experts rank quite high the analysis of answers to appropriately open-ended questions. This holds in both classroom and e-learning settings, but may become a

D.D. Schmorrow and C.M. Fidopiastis (Eds.): AC 2014, LNAI 8534, pp. 212–223, 2014.

very demanding activity. Peer-assessment, especially if reliably mediated by the teacher's activity, may partially relieve her from the task. Moreover, it can provide rich information about the meta-cognitive ability of students to correctly evaluate their peers. Such ability can be deemed as important as the individual knowledge of the topic, so that related evaluation is a precious gain.

A pedagogically effective personalization of learning processes requires a reliable assessment of the learner's state of knowledge [1-5]. This is also a critical factor in systems providing social-collaborative e-learning [6-13].

Many researches (see for example [14]) identify the evaluation of answers to open-ended questions among the most powerful assessment tools. Closed answer tests (quizzes), either single (yes, no) or multiple-choice, require no further elaboration from students to explicitly demonstrate their knowledge and beliefs. As a consequence, they may also allow success by just choosing random answers. On the other hand, they are a very convenient tool to be frequently used, since they allow fully automatic grading. On the contrary, open-ended questions require open answers, i.e. true short essays in the form of free text produced by the student. They are more difficult to tackle for students, since they require the ability and knowledge to concisely show their proficiency with respect to a topic. Moreover, they are demanding for the teacher too, whose evaluation is required since they are formulated in free natural language. In our work, we are tackling the problem of semi-automatic grading of open answers by exploiting peer-assessment in a social collaborative e-learning setting. The overall process is mediated by the teacher. After the peer-assessment phase, she provides grades for a starting subset of answers, which are used by the system to deduce the final full set of grades through the peer-provided assessments. On their hand, students can assess their peers' answers by using the method selected by the teacher among the available ones, which are: grading each answer, choosing the best answer or choosing the worst one (or both best and worst). The system maintains a model of students achievements: in particular we define the student model as an evaluation of the learner's state of knowledge (K) and of the learner's ability to judge answers given by peers (J). For each question, we consider the correctness (C) of the learner's answer, and a variable for each peer assessment (e.g., a G variable if the assessment strategy for students is Grade, or else B and/or W variables for best and worst peer assessment methods); the teacher keeps grading students' answers until a termination condition flags that the remaining grades can be automatically computed from the current collected knowledge. In [15] we first introduced a simple Bayesian-network-based model to implement our assessment pattern. The model was further refined in [16] and [17]. Each students is represented by a Bayesian sub-network with K, C and J variables, and the individual students' networks are connected through the variables modeling peer-assessment. During a session, each student assesses a number of (e.g., three) answers from peers, according to the method selected by the teacher for the session. The values assumed by variables C and, e.g., G in the case of grading assessment method, are asserted as evidence and propagated in the whole network. Each student's model is updated accordingly. The aim is to finally produce reliable automated grading for those answers that were not directly graded by the teacher.

The present implementation of the OpenAnswer web-based system allows to deliver open ended questions, collect open-answers, collect peer (self-)assessments,

maintain the student model representing the achieved proficiency and the ability to (self) assess, and support the teacher's analysis and grading of the answers. The first help to the teacher comes as the suggestion of the next answer to assess; the selection is done according to a pre-selected strategy that either has the goal to retrieve the maximum information possible, by choosing the answer which is the most ambiguous (max_entropy strategy), or by choosing the currently most probably wrong answer (max_wrong strategy) or by just choosing a random one (random strategy). Finally, OpenAnswer can suggest to stop the correction because the information collected is sufficient. In all cases, the teacher is free to bypass these suggestions and correct a different answer or continue to grade.

In this paper we compare the results of experiments presented in [16] and [17], with the results obtained through a finer granularity of the Bayesian variables, moving from 3 discrete values to 6. These results show that a more discriminating set of values improves evidence propagation within the Bayesian network and increases the reliability of final results.

2 Related Work

Several methodologies have been proposed for the automatic analysis of open answers, stemming from different yet related disciplines such as data mining, natural language processing, concept mapping and semantic web techniques. We report here some relevant examples.

In the context of marketing, techniques for data mining and natural language processing aim at extracting customer opinions and synthesizing products reputation [18,19]. In [20] concept mapping is used with the same goal, by defining and applying "coding schemes", which allow to analyze and classify answers. The mentioned approaches can be also applied in educational contexts [21].

In [22] a (semi-)automatic assessment of open-answers is proposed, relying on ontologies and semantic web technologies. The ontology is used to formally define the knowledge domain to which the questions are related, and also aspects of the overall educational process. The ontological labels to be assigned to the answers are in the form of "Semantic annotations". After the answers have been labeled, they can be analyzed, by evaluating the similarity of their ontological labels against the ones assigned to the question. The mechanism of grade assignment is just based on the computed ontological correspondences. Teacher plays a crucial role especially at the beginning of the process, i.e. in the definition of course ontology and questions' semantic annotations, while such role is much less significant later.

In [23] open answers help determining the implicit conceptions of the students, and to treat the wrong ones that may hinder the cognitive process. The algebra defined in [24] allows practical manipulation of formulae through the proposed symbolic computation system. This system is effective when applied to answers on algebraic expressions, yet without added natural language, which would be often beneficial.

A detailed study of peer assessment in a prototype educational application is in [25]. Finally, in our previous work [26, 27], we showed an approach to the grading of

open answers, based on Constraint Logic Programming (CLP) and peer assessment. Students were defined through the same triple of finite-domain variables, with 3-valued domains. The modelization of both the student and of the peer-assessment was made by posting appropriate constraints. This approach has been temporarily set aside due to its high computational complexity, translating in huge processing times.

3 The OpenAnswer System

OpenAnswer is a module integrated in a preexisting PHP-based Leaning Management System (named sLMS). A teacher in OpenAnswer can:

a) define a questionnaire;
b) define a questionnaire session;
c) open/close the "answer time" for a session;
d) open/close the "peer-assessment time" for a session;
e) examine results from peer-evaluation of a session, and assess and grade (some of) the given answers;
f) publish the final session grading.

In a) the teacher specifies some questionnaire options: 1) number of questions, 2) number of answers to be peer-assessed by each peer, 3) number of components in each group of students (10-25), 4) probability for a student to assess her/his own answer: 100% (always), 0% (never), 33% or 66%; 5) possible answer anonymity. In b) the teacher defines a session, by specifying the questionnaire to be used and the composition of the groups of students. In c) the questionnaire is submitted to the students, and the answers are collected. After the answer period has expired the peer-assessment can start (d). In e) the teacher starts the analysis of a session. It is possible to compare different correction strategies by creating clones of the same session. Thus, two specific services are provided: 1) to "clone" a session that has been already peer-evaluated, and 2) to select the strategy used by the system to suggest the "best next answer to grade" during the grading phase for the current session (see below). Each manual assessment by the teacher propagates evidence in the Bayesian network. Appropriate ground truth for the simulation experiments presented in this paper can be provided by the "manual" strategy, which requires the teacher to assess all answers. Otherwise, the next answer suggested for grading is the one allowing fastest convergence towards a reliable automatic grading of the ungraded answers, according to the termination criterion. After such point we can accept the automatic grades obtained so far.

In the former implementation of OpenAnswer the student model (SM) included the above mentioned finite-domain variables ranging over a very limited set of values. For variables K and J the possible values were Good/Fair/Bad. For C and, say, G for peer assessment, the values were correct/fair/wrong. We present here the results of the comparison between this configuration of variable domains, and a finer-grained one with all variables ranging from A to F (6 values each). For each configured session, grading of answers starts using the present values in the SMs; values evolve, according to the propagation stemming from the teacher's progressive grading. After a

session grading has been completed, the new values in the student SMs are stored in temporary variables. They are not immediately substituted for the old ones, to allow having clones of a same session to be graded by different strategies yet starting from the same initial configuration. When the teacher activates the updating process the new K and J values are stored permanently.

4 The Bayesian Model

In a former implementation of OpenAnswer a Yap Prolog [28] module handled the management of the Bayesian network. The present system has been rewritten in Python, by using the open-source C++ libDAI library [29]. It supports the implementation of belief-propagation algorithms on Factor Graphs. These are a graph-based formalism which is well suited to represent networks of variables connected through probabilistic constraints [30]. The new implementation is at least an order of magnitude faster than the former one, with better memory usage.

The Bayesian network modeling peer-assessment is made of 4 finite-domain variables: K and J that make up the overall SM of each student, C which represents the current correctness of each answer, and a fourth variable depending on the kind of peer assessment defined for the session, e.g.. G for grade, with a behavior similar to C. In the 3-value implementation, we had:

- K: Knowledge (good, fair, bad)
- J: Judgment (good, fair, bad)
- C (G): Correctness (Grade) (right, fair, wrong)

We report now the value distributions with 3 and 6 variables.

The Knowledge variable is independent and based on the following default probability distribution over 3 (Table 1) or 6 values (Table 2):

Table 1. Probability distrib. of K over 3 values

K	good	fair	bad
P(K)	0.2	0.3	0.5

Table 2. Probability distrib. of K over 6 values

K	A	B	C	D	E	F
P(K)	0.1	0.2	0.3	0.2	0.1	0.1

Notice that all considered probability distributions are “synthetic”, i.e., they have not been (yet) learned from actual experimental data. We think this is acceptable for now, since our present aim is still to test if the methodology is sufficiently reliable for semi-automatic grading. As a matter of fact, as Bayesian networks allow learning such probability distributions from experimental data, we will derive these distributions from the student's interaction with the system.

The remaining variables of our model are related through two Conditional Probability Tables (CPTs) (the same considerations hold for the values).

We model the Judgment variable as probabilistically dependent on Knowledge. We can assume that judging the answers of peers is a higher (meta-)cognitive activity

with respect to both knowing and using ones' knowledge. This is supported by the Bloom's taxonomy of cognitive abilities [31]. A further conditional probability distribution relates the correctness of her/his answer to her/his knowledge about the questionnaire topic. We assume that since the answer is open, the student could not guess it. Without loss of generality, we assume the same starting CPT for J, C and G.

Table 3 reports the distribution over 3 values, whereas Table 4 is for 6 values.

Three selection strategies may suggest the next answer to be corrected:

- **max_entropy:** choose the answer with maximum entropy of the correctness probability distribution (i.e. the answer's correctness is the most ambiguous one);
- **max_wrong:** choose the answer with the highest probability to be a wrong answer (students would not deem acceptable to fail "just because the Bayesian model said so", without the teacher actually checking their answer, thus it's better to start with the wrong ones);
- **random:** choose an answer at random.

Table 3. Conditional Probability Table of P(C|K) (and of P(J|K)) over 3 values

	K		
P(C|K)	good	fair	bad
good	50%	20%	1%
fair	40%	50%	30%
bad	10%	30%	69%

Table 4. Conditional Probability Table of P(C|K) (and of P(J|K)) over 6 values

P(C|K)		K					
		A	B	C	D	E	F
C	A	20%	9%	1%	1%	4%	1%
	B	40%	20%	9%	7%	6%	1%
	C	20%	40%	20%	12%	10%	1%
	D	12%	20%	40%	20%	15%	7%
	E	7%	9%	20%	40%	25%	40%
	F	1%	2%	10%	20%	40%	50%

As an example of the need for using more domain values, we focus on the J domain. With three values, the peer-assessment variables modeled three types of judges: a "perfect" judge (which distinguishes among any pair of different correctness levels), a "barely sufficient" judge (which can just distinguish among wrong and not wrong), and a "bad" judge (which is unable to choose). There is a wide gap between the perfect judge ability and the ability of the "barely sufficient" judge, with plenty of intermediate levels of judging ability. In our new implementation we model with levels A to F the decreasing ability to distinguish among two correctness levels. If C1 and C2 are the correctness values of two answers and DeltaC=|C1–C2| then a judge is able to distinguish C1 and C2 if DeltaC is less than or equal to her J ability (with A=0, B=1, ..., F=5).

5 Simulating the Correction

To test the OpenAnswer model we have run simulated corrections on 3 data-sets.

The available parameters for a simulation are:

- **dataset:** the file containing the peer-assessment data together with the teacher's corrections (3 datasets are available)

- **question:** id of the question to be extracted from the dataset (24 are available)
- **domainsize:** size of the variables' domains (3 or 6),
- **min:** minimum grade required to participate to the simulation (default 0/10),
- **method:** peer-assessment method used by the students:
  - ✓ best: choose the best answer,
  - ✓ worst: choose the worst answer,
  - ✓ best-worst: choose both best and worst,
  - ✓ grade: grade each single answer
- **stats:** synthetic P(K) vs. Experimental P(K):
  - ✓ true: the P(K) distribution is taken from the teacher's grades in the dataset
  - ✓ false: use the above mentioned default hard-coded distribution
- **strategy:** the strategy used to select the next question to grade:
  - ✓ max_entropy: the answer with max entropy of its P(C) (i.e. the "most ambiguous" answer) is selected;
  - ✓ max_wrong: the answer with maximum probability P(C=Fail) is selected (no student would accept a "deduced Fail" grade, therefore all Fail grades should be manually graded, so it's reasonable to select first all max_wrong answers);
  - ✓ random: a random answer is selected.

  Moreover, in a simulation several termination criteria are handled:
  - ✓ no_flip(N): deduced grades are stable in the last N steps (with N=1,2,3)
  - ✓ no_wrong: no remaining answer has deduced grade Fail
  - ✓ no_wrong2: no remaining answer has P(C=Fail) > 50%

An OpenAnswer simulation is made of a number of phases. First, one chooses the dataset to load. It is then possible to filter grades below a minimum threshold (as a matter of fact, we observed that a high percentage of weak students can degrade the quality of propagated evidence). Afterward, the network of students is instantiated.

Depending on the peer-assessment method, from one to three Bayesian variables (connecting the student's J and her peers' C1, C2, C3) are added as Factors to the graph for each peer-assessment choice (e.g. if method=grade a "GradeX" variable for each assessed peer X is required, while if method=best a single "Best" variable is sufficient). For each student the peer-assessment variable's values are set to her specific peer-assessment choices and an initial belief propagation is performed through the Junction Tree belief propagation algorithm. The actual simulation loop starts from this point. The chosen selection strategy is used to find the (yet ungraded) student to be corrected next, and the teacher's correction of this student's answer (from the dataset) is asserted as evidence of her C variable. As a consequence, the junction tree is updated to get the new probabilities for all the network's variables. After each teacher's correction, a test for termination is performed: if some of the provided termination criteria is satisfied current probabilities and grades are printed, otherwise if some termination criteria remains not true a new simulation step is run, else the simulation is complete.

The simulation report contains both the Judgment probabilities for each student and the confusion matrix with deduced grades vs. the actual teacher's grades. The confusion matrix is used by the system to compute the following information: L: length of the correction (number of teacher's grades); OK: exact grades deduced; DP1: grades deduced one mark better than the teacher's; DN1: grades deduced one mark worse than the teacher's; DP2: two marks better than the teacher's; DN2: two marks worse than the teacher's. For six-valued variables, we also have DP: more than two marks better; DN: more than two marks worse. Once the simulations are done it's easy to collect from the generated log files the results for further analysis.

6 Experiments

Available data-sets come from different assessments:

- **M:** 1 question (on web-based languages), 12 students, each one assessing 3 peers
- **I:** 2 questions (high-school level Physics problem, given to two classes), 12 and 14 students, each one assessing 3 peers
- **G:** 3 exams (university-level Physics) with 6 questions each (4 mandatory, 2 optional), with 48, 29 and 21 students, each one assessing from 1 to 3 peers.

The grades' distribution over the union of the datasets is: A=5%, B=5%, C=8%, D=7%, E=11%, F=65%. The trend towards low values is mainly due to G dataset, where students often partially completed their assignments or chose to answer only to specific questions. The questions exhibit the grade distributions in Figure 1. Notice that the G dataset has almost 70% Fail grades while I and M show less than 20% of Fail grades.

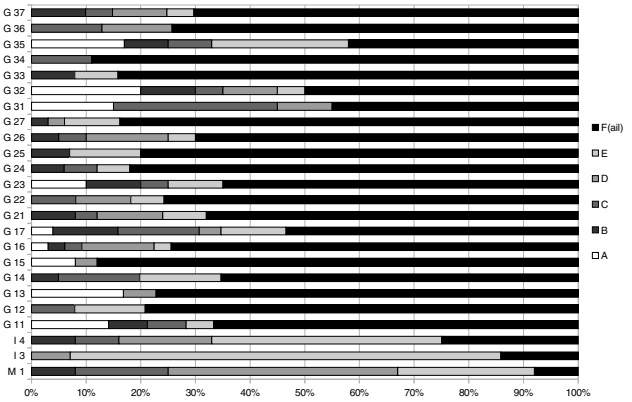

Fig. 1. Grade distribution over the single questions

In order to compare the use of 3 vs. 6 different values for Bayesian variables, we compute two relevant quantities: **OK/ASSESS**, which accounts for the ratio of correct grades derived by the system with respect to all the derived ones; **L/TOTAL**, which account for the ratio of manually graded answers with respect to the overall set.

Table 5 summarizes the results obtained using 3-valued variables with Grade student assessment method and max_entropy selection strategy, while termination criteria ranges over no_flip<N> with N=1,2,3. We performed tests with both a default distribution for K values, and with a more realistic one computed from the complete set of teachers' grades (STAT=YES/NO). Moreover, to show that the system works better with a better class (i.e. with better P(K)), we have run additional simulations by cutting off the lowest graded answers (MIN=0.4).

Table 5 and 6 (below) report the corresponding values for 3 and 6-valued variables.

Table 5. Results with 3-valued variables (greener=better, redder=worse)

DOMAIN=3	STATS=NO MIN=NO			STATS=YES MIN=NO			STATS=NO MIN=0.4			STATS=YES MIN=0.4		
	no_flip1	no_flip2	no_flip3	no_flip1	no_flip2	no_flip3	no_flip1	no_flip2	no_flip3	no_flip1	no_flip2	no_flip3
OK ---------- ASSESSED	67.05%	66.67%	67.23%	69.73%	68.83%	69.16%	33.47%	32.13%	33.67%	43.27%	41.01%	42.33%
L -------- TOTAL	4.40%	8.79%	13.37%	4.40%	9.52%	14.47%	8.92%	17.84%	27.14%	8.92%	19.33%	29.74%

Table 6. Results with 6-valued variables (greener=better, redder=worse)

DOMAIN=6	STATS=NO MIN=NO			STATS=YES MIN=NO			STATS=NO MIN=0.4			STATS=YES MIN=0.4		
	no_flip1	no_flip2	no_flip3	no_flip1	no_flip2	no_flip3	no_flip1	no_flip2	no_flip3	no_flip1	no_flip2	no_flip3
OK ---------- ASSESSED	23.95%	23.74%	24.77%	67.24%	67.27%	67.30%	47.35%	46.45%	40.70%	40.41%	37.50%	39.25%
L -------- TOTAL	4.40%	23.63%	40.11%	4.40%	8.79%	13.74%	8.92%	21.56%	36.06%	8.92%	19.70%	30.86%

A first observation resulting from the comparison of the two tables is that the lack of a reliable model of the class and the usage of all answers has a greater negative effect in the case of 6 values. This is expected, since we have a finer distinction among different levels of correctness, so that the former Good and Fair answers are now finely segmented in 5 groups, and thus it's more difficult to derive exactly the same grade as the teacher (corresponding to an OK outcome). Therefore we can assume this is not a bad result, but only one corresponding to a more accurate grading. Given this, we can easily justify why the influence of a correct starting model of the class is much more evident for the finer-grained range of values (OK/ASSESSED passes from about 67% to about 69% with 3 values, from about 24% to about 67% with 6 values). Again, the apparent slightly worse absolute result stems from the more accurate grading. As a matter of fact, the evidence propagation in the corresponding network is more sensible to a correct starting setting (STAT=YES). We notice that in both cases the use of a reliable knowledge distribution is more relevant than discarding the less accurate questions (i.e. to simulate a better class), and also than considering both factors. This can be explained with the consideration that eliminating wrong answers limits evidence propagation in the net, because it has the further effect of hiding the possible ability of peers of correctly evaluating even those answers. It is also interesting to notice the often longer manual correction required by the six-valued setting. However, the difference is not dramatic, except for the base case (STATS=NO, MIN=NO), where again it is the higher accuracy which requires a longer preparation of the system (the convergence towards a stable set of values is slower if the set of such values is larger).

In figures 2 and 3, below, we show the percentage (respect to the total number of answers) of discrepancy between deduced grades and teacher's grades, together with the correction length (L). Notice that, to be able to compare the case with 3-valued domain with the case of the 6-valued one, we have aggregated the 6-valued correctness classes in the 3 groups {A, B, C}, {D, E}, {F}, mimicking the 3-valued domain results and then computed the difference of grades (DG).

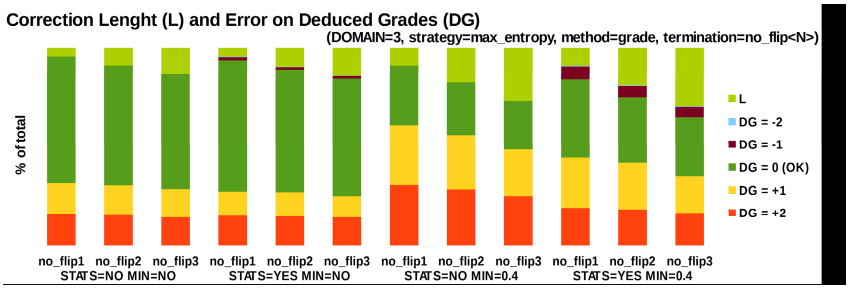

Fig. 2. Correction length and Grade distance - 3-valued domains

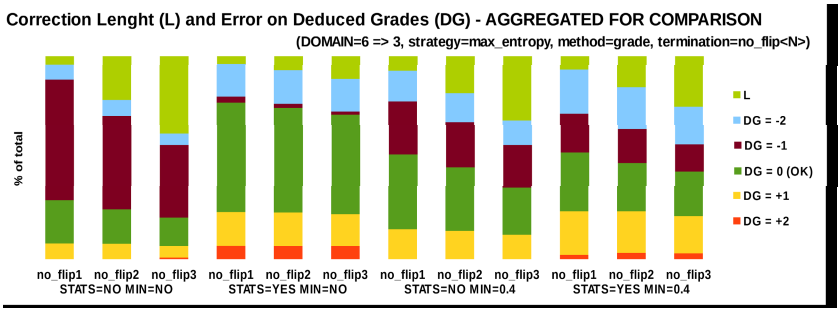

Fig. 3. Correction length and grade difference - 6-valued domain (aggregated)

As we can see from the graphs, with 6-valued domain as expected the length is slightly larger and the number of perfect deductions is slightly lower; the percentage of grades too far from the correct one (DG=±2) is slightly lower, which is an improvement. In practice we get similar performances with higher grade precision.

7 Conclusions

We have shown the new version of OpenAnswer, with an experimental comparison respect to the earlier version using 3-valued domains. The new version gives very good results, deducing correctly up to 60% of the grades by just correcting 10-15% of the answers. The performance of the 6-valued domain version are comparable to the 3-valued one, but we gain a lot more in precision, allowing us to obtain fine-grained assessment of the student's answers.

References

1. Limongelli, C., Sciarrone, F., Temperini, M., Vaste, G.: Lecomps5: A web-based learning system for course personalization and adaptation. In: Nunes, M.B., McPherson, M. (eds.) IADIS International Conference e-Learning 2008, Amsterdam, The Netherlands, July 22-25, vol. 1, pp. 325–332 (2008)
2. Popescu, E.: Adaptation Provisioning with respect to Learning Styles in a Web-Based Educational System: An Experimental Study. Journal of Computer Assisted Learning 26(4), 243–257 (2010)
3. Sterbini, A., Temperini, M.: Selection and sequencing constraints for personalized courses. In: Proc. 40th IEEE Frontiers in Education (FIE) Conference, Washington, DC, USA, October 27-30, pp. T2C1–T2C6 (2010), doi:10.1109/FIE.2010.5673146
4. Essalmi, F., Jemni Ben Ayed, L., Jemni, M., Kinshuk, Graf, S.: A fully Personalization Strategy of E-Learning Scenarios. Computers in Human Behavior 26(4), 581–591 (2010)
5. Limongelli, C., Sciarrone, F., Temperini, M., Vaste, G.: The Lecomps5 Framework for Personalized Web-Based Learning: a Teacher's Satisfaction Perspective. Computers in Human Behavior 27(4), 1310–1320 (2011) ISSN 0747-5632
6. Kirschner, P.A.: Using integrated electronic environments for collaborative teaching/learning. Learning and Instruction 10, 1–9 (2001)
7. Kreijns, K., Kirschner, P.A., Jochems, W.: Identifying the pitfalls for social interaction in computer supported collaborative learning environments: a review of the research. Computers in Human Behavior 19, 335–353 (2003)
8. Sterbini, A., Temperini, M.: Learning from Peers: Motivating Students through Reputation Systems. In: Proc. IEEE/IPSJ International Symposium on Applications and the Internet, July 28-August 01, pp. 305–308 (2008), doi:ieeecomputersociety.org/10.1109/SAINT.2008.107
9. Cheng, Y., Ku, H.: An investigation of the effects of reciprocal peer tutoring. Computers in Human Behavior 25, 40–49 (2009)
10. Popescu, E.: Providing Collaborative Learning Support with Social Media in an Integrated Environment. World Wide Web (2012), doi:10.1007/s11280-012-0172-6, ISSN: 1386-145X
11. De Marsico, M., Sterbini, A., Temperini, M.: The Definition of a Tunneling Strategy between Adaptive Learning and Reputation-based Group Activities. In: Proc. 11th IEEE International Conference on Advanced Learning Technologies, ICALT 2011, Athens, GA, USA, July 6-8, pp. 498–500 (2011), doi:10.1109/ICALT.2011.155
12. De Marsico, M., Sterbini, A., Temperini, M.: A strategy to join adaptive and reputation-based social-collaborative e-learning, through the Zone of Proximal Development. Int. Journal of Distance Education Technology, IJDET 11(3), 12–31 (2013)
13. De Marsico, M., Sterbini, A., Temperini, M.: A Framework to Support Social-Collaborative Personalized e-Learning. In: Kurosu, M. (ed.) Human-Computer Interaction, HCII 2013, Part II. LNCS, vol. 8005, pp. 351–360. Springer, Heidelberg (2013)
14. Palmer, R.: On-line assessment and free-response input – a pedagogic and technical model for squaring the circle. In: Proceedings of the 7th CAA Conference, Loughborough University (2003)
15. Sterbini, A., Temperini, M.: Correcting open-answer questionnaires through a Bayesian-network model of peer-based assessment. In: Proc. Int. Conference on Information Technology Based Higher Education and Training, ITHET 2012, pp. 1–6 (2012)
16. Sterbini, A., Temperini, M.: OpenAnswer, a framework to support teacher's management of open answers through peer assessment. In: Proc. 43th ASEE/IEEE Frontiers in Education, FIE, Oklahoma City, OK, October 23-26. IEEE Computer Society (2013)

17. Sterbini, A., Temperini, M.: Analysis of OpenAnswers via mediated peer-assessment. In: Proc. 17th IEEE Int Conf. on System Theory, Control and Computing, ICSTCC 2013 (2013)
18. Yamanishi, K., Li, H.: Mining Open Answers in Questionnaire Data. IEEE Intelligent Systems, 58–63 (2002)
19. Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T.: Mining product reputations on the Web. In: Proc. 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD 2002, pp. 341–349 (2002)
20. Jackson, K., Trochim, W.: Concept mapping as an alternative approach for the analysis of open-ended survey responses. Organizational Research Methods 5, 307–336 (2002)
21. Romero, C., Ventura, S.: Educational Data Mining: A Review of the State of the Art. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 40(6), 601–618 (2010)
22. Castellanos-Nieves, D., Fernández-Breis, J., Valencia-García, R., Martínez-Béjar, R., Iniesta-Moreno, M.: Semantic Web Technologies for supporting learning assessment. Information Sciences 181(9), 1517–1537 (2011)
23. El-Kechaï, N., Delozanne, É., Prévit, D., Grugeon, B., Chenevotot, F.: Evaluating the Performance of a Diagnosis System in School Algebra. In: Leung, H., Popescu, E., Cao, Y., Lau, R.W.H., Nejdl, W. (eds.) ICWL 2011. LNCS, vol. 7048, pp. 263–272. Springer, Heidelberg (2011)
24. Formisano, A., Omodeo, E.G., Temperini, M.: Layered map reasoning: An experimental approach put to trial on sets. Electronic Notes in Theoretical Computer Science 48, 1–28 (2001)
25. Chung, H., Graf, S., Robert Lai, K., Kinshuk: Enrichment of Peer Assessment with Agent Negotiation. IEEE Trans. on Learning Technologies (TLT) 4(1), 35–46 (2011)
26. Sterbini, A., Temperini, M.: Supporting Assessment of Open Answers in a Didactic Setting. In: Proc. 12th IEEE Int. Conference on Advanced Learning Technologies (ICALT 2012), Rome, Italy, July 4-6 (2012)
27. Sterbini, A., Temperini, M.: Dealing with open-answer questions in a peer-assessment environment. In: Popescu, E., Li, Q., Klamma, R., Leung, H., Specht, M. (eds.) ICWL 2012. LNCS, vol. 7558, pp. 240–248. Springer, Heidelberg (2012)
28. Costa Santos, V.: CLP (BN): Constraint logic programming for probabilistic knowledge. In: Proc. 19th Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufmann (2002)
29. Moij, J.M.: liDAI: a free and open source C++ library for Discrete Approximate Inference in graphical models. J. of Machine Learning Research 11, 2169–2173 (2010)
30. Kschischang, F.R., Frey, B.J., Loeliger, H.-A.: Factor graphs and the sum-product algorithm. IEEE Trans. on Information Theory 47(2), 498–519 (2001)
31. Bloom, B.S., Mesia, B.B., Krathwohl, D.R.: Taxonomy of Educational Objectives (two vols: The Affective Domain & The Cognitive Domain), NY. David McKay (1964)
32. Chatti, M.A., Sodhi, T., Specht, M., Klamma, R., Klemke, R.: u-Annotate: An application for user-driven freeform digital ink annotation of e-Learning content. In: Proc. Sixth International Conference on Advanced Learning Technologies, ICALT 2006, pp. 1039–1043 (2006)
33. Limongelli, C., Sciarrone, F., Starace, P., Temperini, M.: An Ontology-driven OLAP System to Help Teachers in the Analysis of Web Learning Object Repositories. Information Systems Management 27(3), 198–206, doi:10.1080/10580530.2010.493810
34. Dyckhoff, A.L., Zielke, D., Bültmann, M., Chatti, M.A., Schroeder, U.: Design and implementation of a learning analytics toolkit for teachers. Educational Technology and Society 15(3), 58–76 (2012)