Correcting open-answer questionnaires through a Bayesian-network model of peer-based assessment

Andrea Sterbini
Dept. Computer Science.
Sapienza University of Rome
Rome, Italy
sterbini@di.uniroma1.it

Marco Temperini
Dept. Computer, Control, and Management Engineering
Sapienza University of Rome
Rome, Italy
marte@dis.uniroma1.it

Abstract—**We have previously shown that, with the help of peer-assessment and of a finite-domain constraint-based model of the student's decisions, the teacher could have a complete assessment of the answers to open-ended questions, by grading just a subset of the answers (as low as half of the lot) and having the rest of the grading inferred by the supporting system. In this paper we present a probabilistic version of the earlier model, using Bayesian networks instead than constraints. Our aims are both defining the approach and prepare its validation: 1) modeling the peer-assessment activity of a student that evaluates others' answers, 2) using peer-assessment to help the teacher with a faster/shorter assessment process, 3) inferring the student's level of competence and ability to judge, from peer-assessment and from (partial) teacher-assessment, 4) learning the conditional probabilistic tables (CPTs) of the model from student data, and 5) comparing the probability distribution of competences in the class at different course phases. The model is under development and test with real data. We are developing a web-based interface to deliver open-answer and peer-assessment questionnaires and to assist the teacher-assessment.**

Bayesian networks; Student modeling; Peer assessment; Correcting Open-answers;

I. INTRODUCTION

Questions and answers, managed through web-based questionnaires, are an important part in the e-learning educational process. Usually questionnaires are made of various typologies of questions, with absolute prominence of those that are more available to a completely automatic correction and grading: questions that ask for a choice (multiple or single) among multiple options, explicit association of items, truth declarations, and some variations.

Open-ended questions, on the other hand, are a very significant resource for the assessment of students and of learning activities; they can support the teacher in the evaluation of students' knowledge and skills, and in the appreciation of the cognitive level [1] of such knowledge and skills. It is also noted that open-ended questions seem a better measure means in many cases, and in particular when it comes to the assessment of the highest cognitive levels [2, 3]

An automatic analysis of open answers (i.e. the textual, possibly freely shaped, answer to an open-ended question) is quite difficult. Research on this topic is varied and makes use of several techniques of Computer Science: information extraction, classification and grading, with different levels of human intervention in the process, are obtained through data mining with natural language processing, concept mapping and semantic web techniques. Applications from the former two fields can be found on the problem of summarizing questionnaires, extracting opinions and defining products reputation, with mainly marketing and commercial aims [4, 5].

An alternative approach, with similar aims as the previous, is in [6]: the idea is to use concept mapping to develop and evaluating/re-examining coding schemes through which the answers are examined. The process goes through five steps, and is good especially for applications on a "free list in context" type of answers. The codes mentioned above are basically semantic labels associated to parts of the answer, in a semi-automatic way. The human intervention of "coders" and "classifiers" is much needed. The labels allow to classify the chunk of answers and, consequently, to collect and classify the answers content.

Although the approaches described so far are not directly fitting into the assessment activity of an educational process, the techniques used are known to have application in e-learning at a wider range [7].

An approach to (a partially semi-) automatic assessment of open-answers, through ontologies and semantic web technologies is in [8]. The domain of knowledge related to the questions, with some other aspects of the educational process, is defined through ontologies. "Semantic annotations" are defined to label the questions by the ontological elements of the correct answer. The student's answer is also labeled in terms of the ontology; then the answer can be analyzed, by evaluating the similarity of its ontological elements against the question's ones; grading follows. Teacher's intervention in the process is high at the beginning, when course ontology and questions' semantic annotations have to be defined, while it seems fairer later on, when the answers' semantic annotations (stated by the system) have to be checked, corrected and integrated.

In [9] open answers are analyzed to determine the implicit conceptions of the students, and to treat those conceptions that may hinder the cognitive process. The system works as a cognitive diagnosing tool in an algebra-bound intelligent tutoring system. An evaluation of this system has proven worthy when applied to answers constituted by purely

978-1-4673-2334-5/12/$31.00 ©2012 IEEE

algebraic expressions, without intermixed text in natural language (that could make explicit the reasoning and the steps of development of the answer).

In this paper we show an approach to support the teacher's grading of open-answers, in a social collaborative e-learning setting. No automatic correction of questions is sought here: the answers are analyzed explicitly by the teacher, just they are actually only a (significant) subset of the whole corpus of students' answers. Peer-evaluation among students is exploited to obtain this reduction of, otherwise much heavier, work for the teacher.

By definition, the analysis and grading process managed by our system is independent of the shape that is given to the textual content of the open answers; it is also independent of the subject matter at hand.

II. EARLIER WORK

In a previous paper [10] we have shown that, with the help of peer-assessment and of a finite-domain-constraint-based model of the student's decisions, the teacher could deduce the correctness value of the answers by grading just some of answers instead than all. The model used is very simple, yet it's sufficient to show that the number of corrections required could be as low as half of the initial set. To capture the behavior of a student we used just 3 finite-domain variables:

- *Knowledge* in *{good, enough, bad}*, i.e. how much s/he knows the topic
- *Judgment* in *{good, educated_guess, bad}*, i.e. how much s/he is able to judge peer's answers,
- *Correctness* in *{correct, wrong}*, i.e. the correctness of his/her answer.

In this we used an (almost) minimal set of domain values to just capture the two actions of answering to the question and judging others' answers. The constraint model removed some combinations of values that were unreasonable, e.g. (by taking inspiration from the Bloom's hierarchy of cognitive levels [1]) it constrained a good enough judge (*J != bad*) to know the topic enough (*K != bad*).

The students assess each other by selecting the best answer among a small, randomly selected, set of others' answers. We modeled the student decision depending on his/her Judgment as follows:

- if *J = bad* the student can only guess randomly
- if *J = educated_guess* the student can distinguish (and remove) the wrong answers and then chooses randomly
- if *J = good* the student is able to choose among the best answers

Even if the above model is simple and (obviously) couldn't capture the real complexity of student's knowledge and ability to judge, simulated experiments have shown that the teacher can distinguish between right and wrong answers, by only knowing the peer evaluation chosen answers and correcting just 50-65% of the answers.

The system had been implemented and simulated in the constraint-logic system GNU-Prolog, and it is going to be integrated in an already available system (named SocialX) supporting social collaborative learning, in which mutual-peer and self assessment of students is allowed/required and motivated through a reputation system [11].

III. MODELING THE STUDENT'S CHOICE

In this paper we present a probabilistic version of the earlier model, which makes use of Bayesian networks.

Bayesian-networks allow the definition of network of probabilistic finite-domain variables, connected as a directed acyclic graph (DAG), where a connection among two variables depicts a probabilistic dependency of the "son" variable with respect to all its "parents". The probabilistic dependency of a variable from its parents is described by a *Conditional Probability Table* (CPT) that defines the probability distribution of the variable having each of its domain values, depending on the combinations of parent values. Bayesian-networks allow the efficient propagation of evidence in both directions (parents→sons and sons→parents) and thus are very useful to compute the probability for any variable to get a certain value, given the observation of some of the other variables in the system.

The probability distributions of each variable can be easily computed by applying the Bayes' theorem, by propagating the observed values through the DAG network and its CPTs [12].

A very useful property of Bayesian-networks is that the CPTs describing the probabilistic dependency of a variable from its parents can be learned from experimental data.

Our research aims, thus, could be described as:

- **modeling the network** of interactions and decisions
- **collecting evidence** from actual questionnaires
- **learning the CPTs** describing the model's details
- using the model as a correction tool to **help the teacher**
- **analyzing the CPTs** to get a better insight in the student's preparation

To model the student, we have transformed the earlier constraint-logic-based rules into a simple Bayesian network with its corresponding CPTs (see Fig. 1).

Every student is (still) represented as a triple <*K, J, C*> of finite-domain variables:

- ***K:*** *Knowledge = {good, enough, bad}*
- ***J:*** *Judgment = {good, educated_guess, bad}*
- ***C:*** *Correctness = {correct, enough, wrong}*

(where both *J* and *C* are depending on *K).*

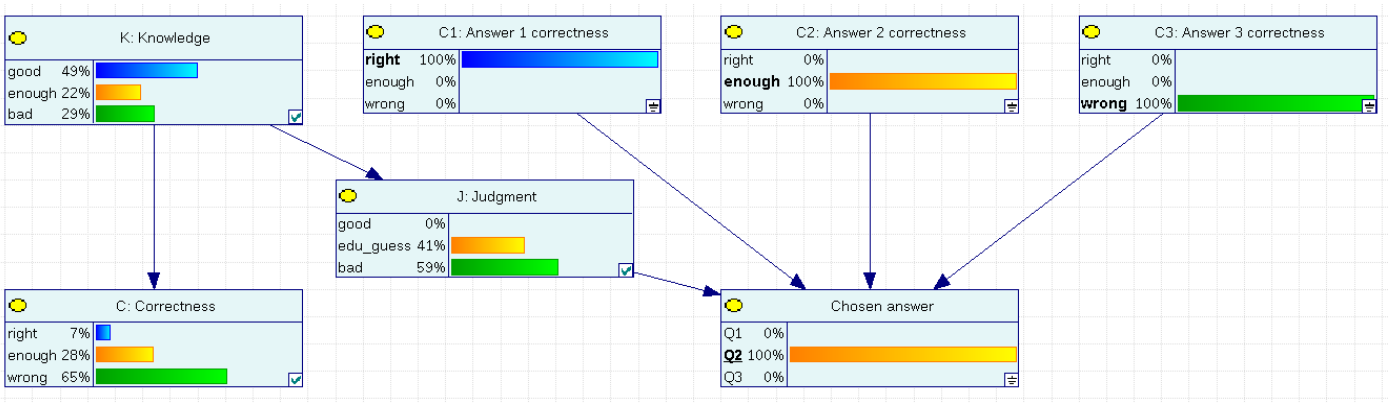

Figure 1: Example of the student's sub-network modeling a peer-evaluation choice

In our initial tests (made just to check the model functionality) we have used the "synthetic" CPTs shown in Table 1 to represent the constraint rules:

- good judges knows the topic (à la Bloom)
- the answer is open, thus bad students cannot write a correct answer

(actually, these CPTs will be learned from student data).

Each student is given a choice among three answers from his/hers peers, from which (s)he has to select the best.

To model the student's decision we define the variable *Chosen = {Q1, Q2, Q3}*, representing the answer selected. *Chosen* depends both from the actual correctness of the answers proposed to the student and from the student's judgment skills (*J*).

Each student is, therefore, described by the Bayesian sub-network shown in Fig. 1. In the figure, *C* is the correctness of the answer given by the student (which is transferred to other sub-networks), while *C1, C2, C3* represent the peers' answers proposed to that student for assessment (and in particular their value of correctness), while *Chosen answer* is the answer the student assesses as the better one among those proposed.

The figure also shows what happens when some evidence is added to the Bayesian network. In this case the *Chosen* and *C1, C2, C3* values have come to be assigned as follows: *C1=right, C2=enough, C3=wrong* and *Chosen=Q2*. From the *J* value in figure we also see that the student's judgment could not be *J=good*, otherwise the student should have chosen C1 instead that C2 (Q2 in Chosen Answer).

To compute the CPT table describing the *Chosen* probabilistic dependence from *J, C1, C2, C3*, we have used the following model:

	P(J | K)		
K	good	educated guess	bad
good	50%	30%	20%
enough	20%	40%	40%
bad	5%	20%	75%

P(C | K)		
right	enough	wrong
60%	30%	10%
20%	50%	30%
1%	9%	90%

Table 1: Example of conditional dependency tables P(J | K) and P(C | K)

- a **good** judge can properly compare the correctness of the answers, and selects any one of the best
- an **educated_guess** judge can spot **wrong** answers but cannot distinguish among **right** and **enough**
- a **bad** judge cannot distinguish at all

The *P(Chosen | J, C1, C2, C3)* CPT, shown in Table 2, are not learned from the student's data but are kept fixed to model the decision.

This is represented by assigning a probability P of choosing *Q1, Q2, Q3* depending on *J* as follows:

- if *J=good* *(first table)*
  - if some **right** answer exists: equal P among the **right** ones only (the others are ignored)
  - if no **right** answer exist but there is some **enough** answer: equal P among the **enough** ones only (wrong are ignored)
  - if all answers are **wrong**: equal P among them
- if *J=educated_guess* *(second table)*
  - if some answer is **right** or **enough:** equal P among them (and wrong are ignored)
  - if all answers are **wrong**: equal P among them
- if *J=bad* *(last table)*
  - there is no distinction among **right**, **enough** or **wrong**, thus equal P on all values

Notice that using a different type of peer-assessment choice corresponds to designing a different *Chosen* variable CPT.

E.g., to distinguish the case where a student recognizes and marks that all answers are wrong, we add the value "none" to the Chosen domain, with probability P=1 iff *J=good* or *J=enough* and all answers are wrong, else P=0 in all other cases.

J	good																								
C1	right									enough									wrong						
C2	right			enough			wrong			right			enough			wrong			right			enough			wrong		
C3	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w
Q1	1/3	1/2	1/2	1/2	1	1	1/2	1	1	0	0	0	0	1/3	1/2	0	1/2	1	0	0	0	0	0	0	0	0	1/3
Q2	1/3	1/2	1/2	0	0	0	0	0	0	1/2	1	1	0	1/3	1/2	0	0	0	1/2	1	1	0	1/2	1	0	0	1/3
Q3	1/3	0	0	1/2	0	0	1/2	0	0	1/2	0	0	1	1/3	0	1	1/2	0	1/2	0	0	1	1/2	0	1	1	1/3

J	educated_guess																								
C1	right									enough									wrong						
C2	right			enough			wrong			right			enough			wrong			right			enough			wrong		
C3	r	e	w	r	e	w	r	e	w	r	e	w	R	e	w	r	e	w	r	e	w	r	e	w	r	e	w
Q1	1/3	1/3	1/2	1/3	1/3	1/2	1/2	1/2	1	1/3	1/3	1/2	1/3	1/3	1/2	1/2	1/2	1	0	0	0	0	0	0	0	0	1/3
Q2	1/3	1/3	1/2	1/3	1/3	1/2	0	0	0	1/3	1/3	1/2	1/3	1/3	1/2	0	0	0	1/2	1/2	1	1/2	1/2	1	0	0	1/3
Q3	1/3	1/3	0	1/3	1/3	0	1/2	1/2	0	1/3	1/3	0	1/3	1/3	0	1/2	1/2	0	1/2	1/2	0	1/2	1/2	0	1	1	1/3

J	bad																								
C1	right									enough									wrong						
C2	right			enough			wrong			right			enough			wrong			right			enough			wrong		
C3	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w	r	e	w
Q1	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
Q2	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
Q3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3

Table 2: Modeling the student's possible choices when J is good, educated_guess, bad

To design and run the above Bayesian-network models we use the Genie/SMILE (Graphical Network Interface/Structural Modeling, Inference, and Learning Engine) Bayesian network system [13], which includes a graphic network editor and algorithms both to propagate evidence in the network and to learn the network parameters/CPTs from experimental data.

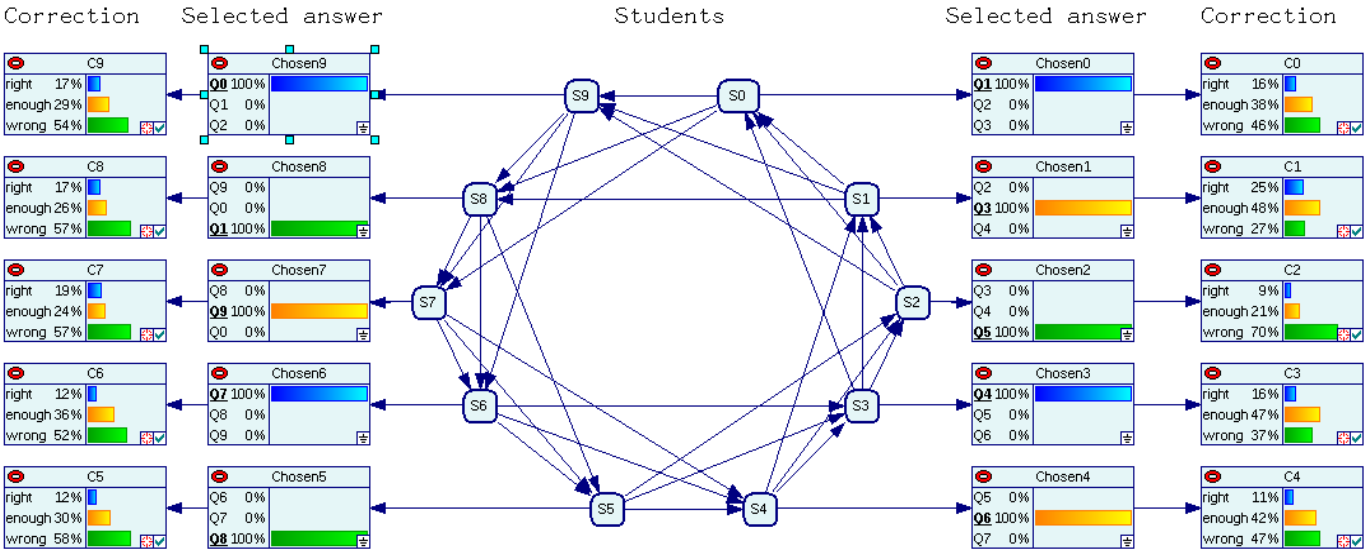

Figure 2: The Bayesian network of 10 students

IV. RUNNING THE MODEL

To model the group of students (they were 10 in our initial experiments) we made 10 copies of the above sub-network and connected them with the topology corresponding to the assessment questionnaires. E.g. if a regular topology is produced during the questionnaire-generation step (for instance by selecting for each n-th student the answers of the next three students (n+1, n+2, n+3, mod 10), the Bayesian network produced is the one shown in Figure 2, with the 10 student's sub-networks in the center and, on both sides, a copy of the *Chosen Answer* and *C* nodes for each one, allowing the teacher to enter the peer-assessment answers and his/her corrections. (notice that, even if the picture seems to be cyclic, the actual edges build an acyclic graph).

In particular, for each node, the figure shows the probabilities calculated for the given set of student's choices, inserted as evidence in the network by clicking the corresponding "Chosen" nodes, before starting the correction.

The teacher, now, can choose which answers are to be corrected and enter the corresponding evidence in the network, repeating until s/he feels the evidence collected is enough to spot all wrong answers. Strategies to choose which answer is to be corrected next include:

- **max_ambiguity**: choose the answer with minimum distance among the min and max computed probabilities for its values
- **max_information**: similar to the above one, but choosing the answer that would add more total information to the network depending on the possible outcomes of the correction
- **max_wrong**: choose the answer that has max probability to be wrong (as the student wouldn't accept a "deduced wrong" correction)

E.g. for the Figure 2 example: in the first case one would choose answer *C1* because *max(P)-min(P)=23%*, while in the third one would choose answer *C5* because *P(wrong) = 58%.*

V. LEARNING THE MODEL'S PARAMETERS

As seen above, the Bayesian network runs nicely, provided that the CPT tables defining the student's model are already known. That network instead, works poorly when used to learn the CPTs values from experimental data. This because, in Genie/Smile, a complete independent copy of the student's sub-network is inserted for each student, with a new copy of all its CPTs, instead than sharing a single definition and CPTs among all the student instances. This means that the number of CPTs and parameters which are to be learned (independently) by the system is 10 times the required ones.

Notice, yet, that in our particular case the data available to learn the CPT tables are the values of the *Chosen* and of the *C, C1, C2, C3* variables. These are exactly the only variables that are shared among the different student's instances. Therefore any evidence containing all the values of the tuple <*Chosen, C, C1, C2, C3*> would completely decouple a student from all the others. Thus, learning the parameters can be done within the Bayesian network model of a single student (Figure 1) instead than the within the whole peers Bayesian network (Figure 2).

(this suggests, also, the idea to learn separate personal CPT tables for each student, if enough data is available)

VI. CONCLUSIONS

We have presented a Bayesian model for the peer-assessment of open-answer questionnaires, aiming to:

- using peer-assessment to **help the teacher** with a faster/shorter correction
- **modeling the peer-assessment** choice of a student evaluating others' answers
- **inferring** the student's **level of competence** *K* and **ability to judge** *J*, from peer-assessment and from a partial correction
- **learning** the conditional probabilistic tables (CPTs) of **the model from student data**
- **comparing** the probability distribution of **competences** in the class **at different course phases**

The model is under test and validation (executed with real data) and its implementation in a social collaborative e-learning framework is under development. In particular we are developing a web-based interface to deliver open-answer and peer-assessment questionnaires and to assist the teacher with the corrections.

Other ideas spur from the methodology we proposed in the paper:

1) Dealing with the effectiveness of the network topology of peer-assessments

The network shown above (Figure 2) and used in our initial tests is very regular. This, paired to some irregular distribution of good students in the network, could affect the effectiveness of exploiting peer-assessment as a source of information. The following research question is then raised:

- how the distribution of (good) students in the network topology effects the information propagation in the network and its effectiveness?

2) Dealing with the network complexity

The network used is big, mainly because of the copies of students and because of the size of the *Chosen* variable CPT. This suggests researching other network structures:

- e.g. how to transform the *Chosen* variable into a *network of simpler variables* with a lower number of dependencies (e.g. a composition of binary comparisons).

3) Dealing with plagiarism and preserving anonymity

The methodology does not (yet) address some issues arising when open-ended questionnaires are intertwining with peer-assessment:

- *plagiarism***:** answers could be copied from others or from external resources,

- *non-blind assessment*: students could mark their answers to help their fellow's peer-assessments

To address these issues we are planning to use both traditional text-based comparison methods and to enlarge the student's model with variables representing the chance of plagiarism or the presence of identity information embedded in the answers.

4) Dealing with "dimensions" of the CPT analysis

Separate CPT could be learned depending on selected subsets of evidence collected. This raises the possibility to compare the learned network parameters among different dimensions, e.g.:

- the student dimension (paving so the way to the management of personalization also in the definition and maintenance of the student model)
- the topic dimension (making it possible to comparing how learning and judgment depend on the topic)
- the position in the course (meaning the moment of the course when the peer-assessment takes place: how do learning and judgment depend on when the peer-assessment did take place?)

ACKNOWLEDGMENT

Work partially funded by project UnderstandIT, 2010-1-NO1-LEO05-01839, Leonardo Da Vinci Programme, EU.

REFERENCES

[1] B.S. Bloom (Ed), "Taxonomy of Educational Objectives", David McKay Company Inc., New York, 1964.

[2] M. Birenbaum, K. Tatsuoka, and Y. Gutvirtz, "Effects of Response Format on Diagnostic Assessment of Scholastic Achievement", Applied Psychological Measurement, 16:4 pp. 353-363, 1992.

[3] K. Palmer, and P. Richardson, "On-line assessment and free-response input – a pedagogic and technical model for squaring the circle", Proceedings of the 7th CAA Conference, Loughborough: Loughborough University, 2003.

[4] K. Yamanishi, and H. Li, "Mining Open Answers in Questionnaire Data", IEEE Intelligent Systems, Sept-Oct 2002, pp 58-63, 2002.

[5] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining product reputations on the Web", Proceedings eighth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'02, pp 341-349, ACM New York, NY, USA, 2002.

[6] K. Jackson, and W. Trochim, "Concept mapping as an alternative approach for the analysis of open-ended survey responses", Organizational Research Methods, 5, Sage, 2002.

[7] C. Romero, S. Ventura, "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 pp. 601-618, 2010.

[8] D. Castellanos-Nieves, J. Fernández-Breis, R. Valencia-García, R. Martínez-Béjar, and M. Iniesta-Moreno, "Semantic Web Technologies for supporting learning assessment", Information Sciences, 181:9, pp. 1517–1537, Elsevier, 2011.

[9] N. El-Kechaï, É. Delozanne, D. Prévit, B. Grugeon, and F. Chenevotot, "Evaluating the Performance of a Diagnosis System in School Algebra", Proceedings Advances in Web-Based Learning - ICWL 2011, LNCS 7048, pp. 263-272, Springer, 2011.

[10] A. Sterbini and M. Temperini. "Supporting Assessment of Open Answers in a Didactic Setting", Social and Personal Computing for Web-Supported Learning Communities (SPEL 2012), ICALT 2012, Rome, Italy (2012).

[11] A. Sterbini, M. Temperini. Learning from peers: motivating students through reputation systems. Int. Symp. on Applications and the Internet, Social and Personal Computing for Web-Supported Learning Communities (SPeL). Turku, Finland, (2008).

[12] J. Pearl. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

[13] The Genie/SMILE system (Graphical Network Interface/Structural Modeling, Inference, and Learning Engine) http://genie.sis.pitt.edu