Artificial Intelligence in Education
H.U. Hoppe et al. (Eds.)
IOS Press, 2003

A Web-Based Tutoring Tool with Mining Facilities to Improve Learning and Teaching

Agathe Merceron and Kalina Yacef
ESILV - Pôle Universitaire Léonard de Vinci, France, merceron@devinci.fr
School of Information Technologies - University of Sydney, Australia, kalina@it.usyd.edu.au

Abstract. With the emergence of e-learning, flexible education, and the increasing number of students in some fields, online teaching tools are becoming more and more important. Such tutoring tools allow students to practice at their own pace, providing them with proper explanations and feedback when they make mistakes. They allow for storing complete student answers, including mistakes, in a database. It becomes possible to mine this database, extract pedagogically relevant information and provide feedback to the teacher. This paper illustrates our approach with the Logic-ITA, a teaching tool for formal proofs currently in use in the School of Information Technologies at the University of Sydney.

1 Introduction

With the emergence of e-learning, flexible education, and the increasing number of students in some fields, online teaching tools are becoming more and more important. Online teaching tools provide a more or less personalised environment where learners can practice at their own pace, have access to tutorial lessons, be given explanations and feedback on their performance and so on. These benefits to the learners are extremely valuable and we assist to a ”quiet revolution taking place in the classrooms” [4]. However, less attention has been given to the reflection and monitoring that can be made to improve the teaching. Moreover, since online teaching tools are computer-based, they allow for storing complete student answers, including mistakes made while solving exercises. The fact that they are online tools means that all this information, for all students using the tool, can be stored on a common server rather than stored locally. Having electronic access to complete student answers makes it possible to extract pedagogically relevant information and provide feedback to the teacher about how a class, a group of students or an individual student is going.

The Logic-ITA [1, 7] is a web-based Intelligent Teaching Assistant system that is currently used within the School of Information Technologies, University of Sydney. Its aim is to facilitate the whole teaching and learning process by helping the teacher as well as the learner [9]. It allows students to practice formal proofs in propositional logic whilst receiving feedback and also keeps the lecturer informed about the progress the class is making and problems encountered. The system embeds the Logic Tutor, a web-based intelligent tutoring system destined to the students, along with tools dedicated to the teacher for managing teaching configuration settings and material as well as for collecting and analysing data. A multimedia article on the Logic Tutor can be found at [1] and a description of the Logic-ITA at [6]. We are now extending the capabilities of the system to provide more intelligent help to the teacher. In this paper we describe the impact that such a tool can have on the whole process of teaching and learning and in particular how we extract information mining the database of students answers to improve teaching. In this preliminary work, mining is limited to queries and associations of mistakes. Some researchers have explored ways to provide assistance to the teachers [8, 5, 6, 9]. However in terms of help in diagnosis and assessment of learning, analysis and synthesis of results we found very little work. [5]'s PepiDiag and PepiProfil system, like the Logic-ITA, collects data from students' exercises, reports them to the teacher and provides tools to analyse these results. One main difference resides in the fact that it processes one student's data at a time, whereas the Logic-ITA combines data from all students.

The paper is organized as follows. Section 2 provides some background information about the Logic-ITA that is useful to understand the rest of the paper. Section 3 describes the impact of the Logic-ITA on learning. Ways to extract information from the database through SQL queries and association of mistakes are given in Section 4. The value of this information for teaching purposes is described in Section 5. Section 6 concludes the paper.

2 The Logic-ITA

The purpose of the Logic-ITA is to help students to master formal proofs in propositional logic. The student's side of the system, the Logic Tutor, proposes logic exercises to students and has the ability to apply logic rules to propositional formulas. It monitors the resolution of a particular exercise by a student, checks whether formulas entered by the student are correct, provides meaningful error messages in case they are not. If the student wants to, the Logic Tutor can retrieve an exercise that fits best his/her level and weaknesses (the student has also the choices of retrieving a specific exercise or creating his/her own). One task of the teacher's components of the Logic-ITA is to organize all exercises done or attempted by students, intermediary results and messages in a database that can be mined to find information.

Without going into too much detail here, it is useful to understand how exercises are monitored by the system. An exercise is a set of formulas from propositional logic. This set is composed of the premises plus one particular formula called the conclusion. Solving an exercise consists in deriving the conclusion from the premises, by using rules of logic in any number of steps. A step of the derivation works as follows. The student selects formulas, either premises or formulas already derived, applies a logic rule to them and obtains a new formula which is added to the set. If the student makes a mistake in a step, the Logic Tutor gives immediate feedback about the nature of the mistake and a tip to correct it. The derivation stops when either the last formula obtained by the student is equal to the conclusion, or the student gives up. As an example consider the exercise below with 4 premises and the conclusion introduced by $\vdash$.

$$1 : (A \vee (B \rightarrow D)) \quad 2 : (\neg C \rightarrow (D \rightarrow E)) \quad 3 : (A \rightarrow C) \quad 4 : \neg C \quad \vdash (B \rightarrow E)$$

Consider Figure 1. The column $Premises$ refers to the premises used, $Number$ to the line number of the derivation, the column $Formula$ is a formula, $Justification$ is the logic rule used and $References$ are the lines to which the logic rule was applied. The first three lines show three steps of a derivation. The special logic rule that allows to write line 1 and 2 is P indicating that these two formulas are premises. The logic rule that allows to write formula $\neg A$, line 3, is MT for *Modus Tollens* and it applies to lines 1 and 2, making use of premises 3 and 4.

$Premises$	$Number$	$Formula$	$Justification$	$References$
3	1	$(A \rightarrow C)$	P	
4	2	$\neg C$	P	
$3, 4$	3	$\neg A$	MT	$1, 2$
$3, 4$	4	$(\neg A \vee B)$	$Add.$	3
1	5	$(A \vee (B \rightarrow D))$	P	
$1, 3, 4$	6	$(B \rightarrow D)$	DS	$3, 5$
2	7	$(\neg C \rightarrow (D \rightarrow E))$	P	
$2, 4$	8	$(D \rightarrow E)$	MP	$2, 7$
$1, 2, 3, 4$	9	$(B \rightarrow E)$	HS	$6, 8$

Figure 1: A possible answer.

Suppose that the student makes a mistake, and uses the justification MP for *Modus Ponens* instead of MT. The Logic Tutor checks step by step the answer of the student. This step is checked as incorrect because *Modus Ponens* cannot be applied to the two lines entered by the student. The Logic Tutor looks for a reason to the mistake. The first search is to check whether the student made a mistake in the choice of the rule, i.e., whether another rule applies to the two lines indicated by the student. This is the case here. The Logic Tutor rejects the input of the student and produces the following error message *invalid justification Modus Ponens cannot be applied, try Modus Tollens instead*. The input of the student is rejected, so the student has to give another input, however the mistake is saved under the name *wrong rule used Modus Ponens*.

A possible complete answer to this exercise like it would be in the final window on the screen is shown in Figure 1. As already mentionned, P means *premise*, MT means *Modus Tollens*, Add stands for *Addition*, DS stands for *Disjunctive Syllogism*, MP means *Modus Ponens*, and HS stands for *Hypothetical Syllogism*. The mistake, though stored in the database, does not appear on this screen because the tool forces students to correct mistakes and produce correct inputs. However, for reflection purposes, students have the possibility to browse through their past mistakes in a separate window.

3 The Logic-ITA to Improve Learning

The logic-ITA has been used for 2 years, 2001 and 2002, in the course 'Languages and Logic' at the School of Information Technologies, University of Sydney. Respectively 442 and 279 students were enrolled. This undergraduate course is destined to provide 2nd year students with an overview of theoretical models underlying computer science: formal languages followed by propositional and predicate logic. The actual logic part is taught over 10 hours of lectures and 5 hours of small group work (5 weeks). Practice is very important to understand formal proofs. The short window frame makes the use of the Logic-ITA particularly useful to the students.

Conditions of use: The tool was made available and demonstrated to the students during the first lecture on logic. In 2001 we run a local version and in 2002 we introduced the web-based version, which brought geographical freedom to the users whilst storing data in one common place. As a result, the tool has been much more used by students in 2002. We enforced its use by asking students to submit their work from the Logic-ITA. Naturally,

	2000	2001	2002
Homework (out of 3)	1.9 (0.9)	2.3 (0.9)	2.9 (0.4)
Exam question (out of 7 for 2000-01, out of 6 in 2002)	3.3 (1.7)	4.3 (2.7)	4.1 (2.4)
Assignment (out of 5)	N/A	N/A	4.8 (0.5)

Figure 2: Marks: averages and standard deviations.

students were warned that information about their interactions would be stored on the server and would be used for personalization and teaching purposes. No one objected.

Results were very encouraging. We were able to compare the results of the 2001 and 2002 classes, who used the Logic Tutor, with those of the 2000 class, who did not use the tool. We gave similar homework in 2000, 2001 and 2002 as well as a similar exam question on formal proofs. The assignment in 2002 had to be made using the Logic Tutor, while the exam question was paper-pen, not using the Logic Tutor, like in 2000 and 2001. Marks obtained are shown in Figure 2. Students who did not sit the exam or did not hand in their homework are not counted, resulting in 431, 379 and 241 marks respectively at the final exam.

Whilst the increase in the homework mark and the high result in the assignment was partially due to the fact that the Logic Tutor would have picked up many mistakes before submission, there has been an increase in the exam question mark. This indicates to us that students were better prepared. If nothing else, the Logic-ITA gave students the opportunity to practice with feedback more often, which means they were more familiar with the process prior to sitting the exam.

We also conducted a student evaluation survey on a voluntary basis with students, teachers and tutors. In 2001 the feedback from the students was positive apart from the fact that they would have liked the freedom of using the tool from various locations. In 2002, this downside had disappeared and students thought the tool helped them understand better logic proofs. From a range of 1 (did not help at all) to 5 (helped a lot), their average rating was 3.9.

4 Mining Students Answers

Presently, the information that can be extracted from the database includes descriptive statistics given by SQL queries and association of mistakes given by the implementation of a fast algorithm for association rules [2].

4.1 SQL Queries

All the students' exercise history, mistakes, levels are stored in a Microsoft Access database that the teacher can query to retrieve certain information or visualize through Microsoft Excel graphs. In particular the following queries were made:

a- Analysis of the most common mistakes: most common mistakes made by student, number of mistakes per logic rule, percentage of incorrect usage of each rule.

b- Analysis of exercises: exercises that caused the most mistakes, how many students have attempted each question, average performance of all students in a specific question.

c- Status on students' progress: number of students per level (we set 5 levels), find the types of errors per level of exercise, students who stagnates at level 1 for more than 10 exercises.

	Item list
1	(1, 2)
2	(1, 2, 3)
3	(1, 3, 4)
4	(2, 4)
5	(2, 4, 5)
6	(2, 5)
7	(3, 4)
8	(3, 4, 5)
9	(3, 4, 5, 7)
10	(3, 5, 6)

Rule	Support	Confidence
$1 \rightarrow 2$	20%	66%
$1 \rightarrow 3$	20%	66%
$3 \rightarrow 4$	40%	66%
$4 \rightarrow 3$	40%	66%
$5 \rightarrow 3$	30%	60%
$5 \rightarrow 4$	30%	60%
$4, 5 \rightarrow 3$	20%	66%
$3, 5 \rightarrow 4$	20%	66%

Figure 3: 10 transactions and their list of items (left) and association rules obtained from these data (right).

4.2 Association of mistakes

The goal of association mining techniques is to find items, in our case mistakes, often occuring together.

a- General algorithm: we suppose that we have a population of N transactions and each individual is characterized by a list of items. Figure 3 gives an illustration. Transaction 1, for instance, is characterized by the list $(1, 2)$.
Items often occuring together are given by rules of the following form:
$3 \rightarrow 4$, *support* 0.4, *confidence* 0.66, or $1 \rightarrow 2$, *support* 0.2, *confidence* 0.66.
The first rule means that if item 3 is present, then item 4 is also present. This is supported by 40% of the transactions with a confidence of 60% .
The concepts *support* and *confidence* have a precise meaning that we introduce now. Let $t_{i,1 \leq i \leq N}$ be N transactions and I be the sets of items occurring in all $t_{i,1 \leq i \leq N}$. In our example, we have $I = \{1, 2, 3, 4, 5, 6, 7\}$. One is looking for rules of the form $X \rightarrow Y$, with $X, Y \subseteq I$ having *support* and *confidence* above a minimum threshold.

$$\text{Support: } sup(X \rightarrow Y) = \frac{|\{t_i | X \cup Y \subseteq t_i\}|}{N}. \qquad \text{Confidence: } conf(X \rightarrow Y) = \frac{|\{t_i | X \cup Y \subseteq t_i\}|}{|\{t_i | X \subseteq t_i\}|}.$$

The support is the number of transactions containing both X and Y divided by the total number of transactions N. The confidence is the number of transactions containing both X and Y divided by the number of transactions containing X. The concept of support is to make sure that only items occurring often enough in the data will be taken into account to establish the association rules. Confidence measures whether Y is really implied by X. Intuitively, if X occurs a lot anyway ($| \{t_i \mid X \subseteq t_i\} |$ is a big number), then almost any subset Y could be associated with it and there is no evidence that the presence of Y is caused by the one of X. A confidence which is high enough makes sure that X and Y have some causal link.

b- Associations of mistakes: to find associations of mistakes, we have implemented the algorithm described in [2] improving slightly its efficiency by adding a change in the generation of intermediary tables which make them smaller.
In the case of the Logic-ITA, transactions have to do with students and exercises and items have to do with mistakes. However, there are various ways to define them.
Consider the mistake *Wrong rule used Modus Ponens* described in Section 2. The mistake has 2 parts: *Wrong rule used* and *Modus Ponens*. If the student would have used *Disjunctive*

Syllogism instead of *Modus Ponens*, then the mistake would have been *Wrong rule used Disjunctive Syllogism*. If we take the first part only, we know that the student has not provided the right justification that matches the two lines. With the second part, we know that some mistake has been done while using the *Modus Ponens* rule. Looking for association on mistakes, we could take the full mistake name for an item, in the present case *Wrong rule used Modus Ponens*, or we can focus on the diagnosis, i.e. what went wrong, here *Wrong rule used*, and ignore the name of the rule presently used, or we can focus on the logic rule involved in the mistake, here *Modus Ponens* and ignore the diagnosis. All these 3 ways are allowed to mine the database.
When using the tool, a student may attempt one or several exercises. Not all students do the same exercises. Thus, a transaction could be the list of mistakes made by a student on a specific exercise, or the list of mistakes made by a student on all attempted exercises. Again, we allow both ways.

5 The Logic-ITA to Improve Teaching

5.1 SQL Queries

The teacher queries the student database with the purpose of identifying the most common mistakes and the logic rules causing the most problems and having a general feeling of how well the class is going.

a- In 2002, out of 2746 mistakes in total the most frequent mistakes was the use of *Premise set incorrect* with *Indirect Proof* and *Conditional Proof*. These were also the most frequently misused rules (71% and 59% respectively). This is due to the fact that they are the most difficult to grasp because they both require the assumption of an additional premise (for example the negation of the conclusion for *Indirect Proof*, aka proof by contradiction) and then the removal of this premise to reach the conclusion. The highest number of mistakes were with *Modus Ponens* (1146), however it was also the most frequently used rule. In the end it was used incorrectly 30% of the time.

b- Analysis of exercises: not surprisingly, the exercises that produced the most mistakes were the ones involving *Indirect* and *Conditional proof* but they were also attempted by a larger proportion of students.

c- Status on students' progress: more than half the class stayed on level 1 (this included students who only logged in once or twice), then 20% moved to level 2 and 10% reached level 3, 4 and 5. These last 3 levels mean that the student is able to complete successfully almost all the exercises attempted.

These findings were mostly useful for the revision lectures. Since time does not allow to review everything, the teacher was able to focus on the mistakes made the most frequently and the rules used the most incorrectly.

5.2 Association of mistakes

The purpose of looking for mistakes often occuring together is for the teacher to emphazise subtleties while explaining concepts to students. Thus, it makes sense to have a support which is not too low. We fixed it at 60%. We present the results obtained from the 2002 database. We have obtained associations when a transaction consists of mistakes of all exercises attempted

association	support	confidence
Rule can be applied, but deduction incorrect $\rightarrow$ *Premise set incorrect*	0.61	0.82
Premise set incorrect $\rightarrow$ *Wrong number of line references given*	0.67	0.87
Rule can be applied, but deduction incorrect $\rightarrow$ *Wrong number of line references given*	0.65	0.87

Figure 4: Associations found focusing on diagnosis.

by one student, and an item is interpreted as the diagnosis, i.e., the first part of the mistake. With other interpretations, support was to low.

a- Experimental results: associations found for diagnosis made used of the three following items. *Rule can be applied, but deduction incorrect*, *Premise set incorrect*, *Wrong number of line references given*. We explain what these messages mean, refering to the example shown in Figure 1. Consider line 4. If the student gives the formula $B \vee \neg A$ instead, the mistake *Rule can be applied, but deduction incorrect* is made. Indeed, addition can be applied, but the added proposition comes last, as shown in Figure 1, not first as written here. Consider now line 3. Suppose the student gives only 3 in the $Prem.$ field. Then the mistake *Premise set incorrect* is made. Suppose now the student gives only 1 in the $Refs.$ field. Then a *Wrong number of line references given* mistake is made, because 2 lines of reference are needed.
All possible permutations of these 3 items gave associations with a support above 61% and a confidence above 79%. The difference in confidence between $X \rightarrow Y$ and $Y \rightarrow X$ does not exceed 7% and is usually 3%. In Figure 4 we show the associations yielding the highest confidences.

b- Findings: the associations found show relations between mistakes involving line numbers in the premises (*Premise set incorrect*), line numbers in the reference lines a logic rule applies to (*Wrong number of line references given*) and incorrect use of logic rules (*Rule can be applied, but deduction incorrect*). This confirms observations made by human tutors. First, students often have difficulties at grasping all details required in a proof: one has to provide not only a logic rule, but also the lines it applies to, and these are different from the premises involved. Second, students do not realize at once that there are two kinds of logic rules: rules of equivalence that are applied to one formula only, and rules of inference that are mostly applied to two formulas. Most importantly, rules of equivalence can be applied to sub parts of a formula whereas rules of inference can only be applied to whole formulae. For example in the formula $((A \wedge B) - > C)$ we can validly replace $(A \wedge B)$ with $(B \wedge A)$ in the formula using the rule of equivalence *Commutation* but it is not valid to deduce B from $((A \rightarrow B) \rightarrow C)$ and A using the rule of inference *Modus Ponens*.

We believe that the details (line numbers, premises involved) are well taken care of by the Logic Tutor itself. However, these results indicate to the lecturers that, in the future, they should put more emphasis on the differences between rules of equivalence and rules of inference.

6 Conclusion

The Logic-ITA illustrates how online tutoring tools may be used to improve both learning and teaching. Students benefit from a tutor which can give them step by step feeback. They get a chance to understand their mistakes quickly, as they make them, not only after getting back their work from a human tutor. Teachers get a chance to know their students better.

Querying the database, they know who is doing well and who has difficulties, what are the most common mistakes made, and so on. They may extract hidden information, like mistakes often associated together and, that way, reflect on their way of explaining concepts to students.

More information can be extracted from the database, like classifying students according to their difficulties or abilities [3]. Future work includes looking at more data mining approaches and their relevance in our educational context.

Extracting information from students answers is very convenient when answers are provided via electronic tools, such as the Logic-ITA. However, developing such tools requires some effort. Another thought is to generalize our approach to students answers obtained more conventionally like through e-learning platforms. This generalization appears to be possible if exercises are structured in a number of well-defined intermediary steps. Multiple choice exercises constitute a particular case of this more general class. Work along these lines has been initiated [3]. Finally, some work is needed to automatise the process of answers mining, make it transparent to the teacher, and convey relevant information in a friendly way. We plan to tackle this issue in the near future.

Acknowledgement: We thank Romain Buquet, Vincent Chagny, Gregory Debord, Raphaël Di Cicco and Olivier Fayau for having implemented the association rule algorithm and extracted the association rules for the Logic-ITA.

References

[1] D. Abraham, L. Crawford, L. Lesta, A. Merceron and K. Yacef, The Logic Tutor: A Multimedia Presentation, Interactive Multimedia Electronic Journal of Computer-Enhanced learning, Vol. 3, Nb. 2, Nov. 2001.

[2] R. Agrawal, R. Srikant, Fast Algorithms for Mining Association Rules, Proceedings of the 20th VLDB Conference, Santiago, Chile 1994.

[3] B. Aguado, A. Merceron and A. Voisard, Extracting Information from Structured Exercises, proceedings of the 4th International Conference on Information Technology Based Higher Education and Training ITHET03, Marrakech, Morocco, (2003).

[4] A. Forster, Online Teaching and Learning, Discussion paper, SYNERGY, Issue 18, October, Sydney, Australia, ISSN 1325-9881, (2002).

[5] S. Jean, Ppite: un systme d'assistance au diagnostic de comptences, PhD thesis, University of Le Mans, France, (2000).

[6] O. Leroux, M. Vivet and P. Brezillon, Cooperation between a Pedagogical Assistant, a Group of Learners and a Teacher, In: proceedings of European Conference on AI in Education, Lisbon, Portugal (1996) 379-385.

[7] L. Lesta and K. Yacef, An Intelligent Teaching-Assistant System for Logic, Proceedings of Intelligent Tutoring Systems, Biarritz, France, Springer-Verlag, June 2002.

[8] M. Vivet, Uses of ITS: which role for the teacher?, In: New Directions for Intelligent Tutoring Systems, E. Costa, ed., Springer-Verlag, Berlin, Heidelberg, New York (1992).

[9] K. Yacef, Intelligent Teaching Assistant Systems, Proceedings of the International Conference on Computers in Education (ICCE'02), Auckland, New Zealand, December (2002).