Utilizing Eye Tracking to Improve Learning from Examples

Amir Shareghi Najar, Antonija Mitrovic and Kourosh Neshatian

Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand
amir.shareghinajar@pg.canterbury.ac.nz
{tanja.mitrovic,kourosh.neshatian}@canterbury.ac.nz

Abstract. In recent year, eye tracking has been used in many areas such as usability studies of interfaces, marketing, and psychology. Learning with computer-based educational systems relies heavily on students' interactions, and therefore eye tracking has been used to study and improve learning. We have recently conducted several studies on using worked examples in addition to tutored problem solving. In this paper we discuss how we used eye-tracking data to compare behaviors of novices and advanced students while studying examples. We propose a new technique to analyze eye-gaze patterns named EGPA. In order to comprehend SQL examples, students require information available in the database schema. We analyzed students' eye movement data from different perspectives, and found that advanced students paid more attention to database schema than novices. In future work, we will use the outcomes of this study to provide proactive feedback.

Keywords: eye tracking, learning from examples, intelligent tutoring

1 Introduction

Eye tracking involves determining the point of gaze of a person's eyes on a visual scene [9]. In recent years, eye tracking has been employed in many areas, ranging from usability studies of interfaces, to marketing and psychology [4, 13, 18]. Many Human Computer Interaction (HCI) projects utilize eye tracking data to investigate which interface design enables users to complete tasks and find the necessary information. Similarly, research on computer-based educational systems also relies heavily on students' interactions with systems. In order to have a more comprehensive and accurate picture of a user's interactions with a learning environment, we need to know which interface features he/she visually inspected, what strategies they used and what cognitive efforts they made to complete tasks [1, 21].

A category of educational systems we are interested in is Intelligent Tutoring Systems (ITS), which provide individualized instruction by observing the student's behaviour, modelling his/her knowledge and adapting to the student by providing adaptive guidance [24]. ITSs have been shown to increase learning by one standard deviation in comparison to traditional classroom learning [22]. Research on eye tracking in

adfa, p. 1, 2011.

ITSs ranges from predicting student errors and determining whether students read system feedback [8], over its use as a form of input [23], to the analysis of how students interpret open learning models [3, 14]. Eye tracking data can be used to improve student modelling by providing low-level information about the student's attention [5, 7, 11]. Results from such investigations can be used to further improve ITSs by providing adaptive hints to draw the student's attention to the important elements of the screen or to inform the student about suboptimal behaviour [6].

Although ITSs have proven their effectiveness in improving learning, they are still not close to the effectiveness of expert human tutors working with students one-on-one [2, 22]. A crucial difference between human tutoring and ITSs is in human tutors' versatility. Human tutors use multiple instructional strategies and switch between them seamlessly, while ITSs typically just support problem solving. One of the goals of our research is to expand the set of instructional strategies supported by ITSs. We have recently conducted a study on using worked examples in addition to supported problem solving [19, 20]. The study was conducted in the context of SQL-Tutor, a mature ITS that teaches SQL [16, 17]. SQL-Tutor complements traditional lectures; it assumes that the student has already acquired some knowledge via lectures and labs, and therefore provides numerous problem-solving opportunities to the student. We extended the system by adding the worked-example mode, which presents a problem, the solution, and the explanation to the student. The study had three conditions: learning from examples only, alternating examples and tutored problems, and tutored problems only. The results showed that students benefitted the most from alternating examples and problems.

Prior research, to the best of our knowledge, has never investigated productive and unproductive behavior in learning from examples. Knowing such information could improve ITSs by prompting students to avoid unproductive behavior and guide them towards successful behavior. Therefore, we conducted a study to find productive and unproductive behavior while students study SQL examples. Section 2 presents the experiment design, while the results are presented in Section 3. Section 4 concludes the paper.

2 The Study

Similar to our previous work, this study was conducted in the context of SQL-Tutor, but only using the worked-example condition, as we wanted to investigate how students learn from examples. We chose the Book database from the thirteen databases available in SQL-Tutor. The system presented a fixed sequence of six worked examples specified on the same database. A screenshot presenting one of these examples is presented in Figure 1. The schema of the database is shown at the bottom of the screen; primary keys are underlined, and foreign keys are in italics. The student can request additional information by clicking on the table or attribute names.

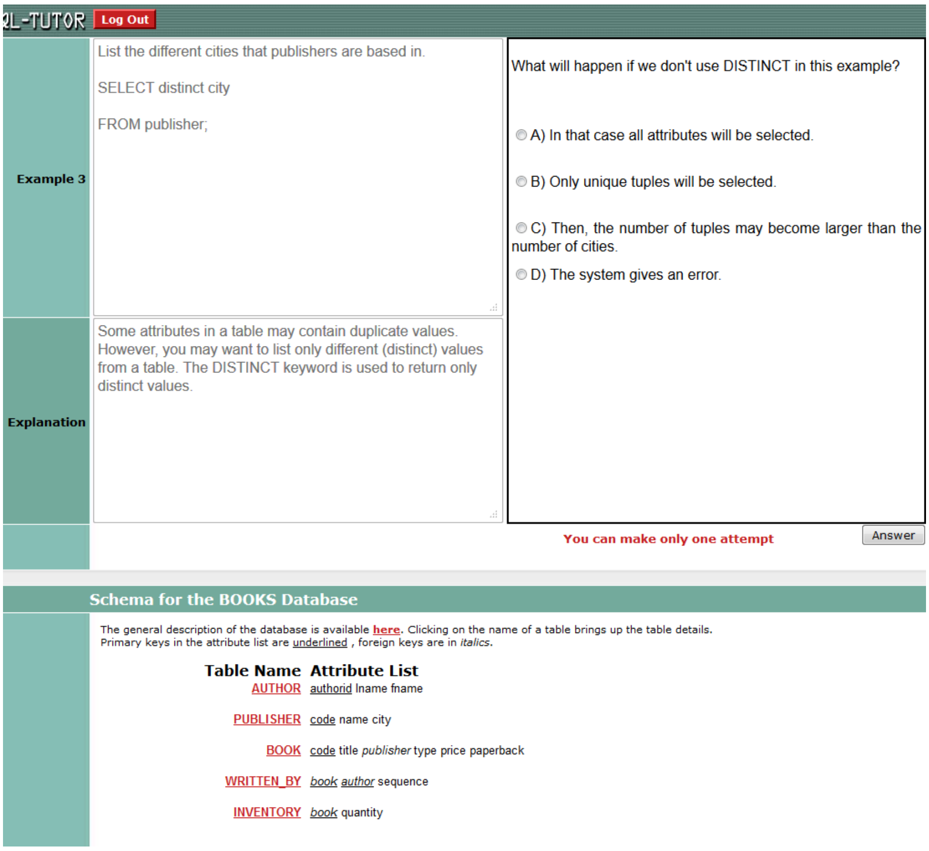

Fig. 1. Screenshot of SQL-Tutor

For each example, the system presents the problem text, the solution and an explanation (see Figure 1). Once a student confirms that s/he has finished studying the example (by clicking the *Continue* button), the system presents a self-explanation question. The goal of this question is to reinforce the knowledge presented in the example by making the student think about a particular construct used in the solution. The student has only one attempt at the question, after which the system informs him/her whether the answer is correct, and if it is not, reveals the correct answer.

For this study, we made minor changes to the interface used in the previous study [19]. We added fixed gaps (> 30 pixels for the 1920*1200 resolution) between the prompt text and each of the options, in order to make identification of eye gaze easier. Moreover, we disabled scrolling to fix the position of page elements on the screen.

3 Results

We collected data from 22 students recruited from an undergraduate course on relational databases. The participants had previously used SQL-Tutor in scheduled labs, but they had not seen the examples from the Book database before. The students participated in individual sessions which were one hour long. All the actions that the students performed through the user interface were recorded in the system log, and we used the Tobii TX300 eye tracker to capture students' eye movements.

Since the sessions were short, we have not used pre/post-tests in this study. However, we had pre-test scores for the same group of students from our previous study, held a week earlier [19]. Using those scores alone to classify students is not justified, since the students have learnt more about SQL in between the two studies. For that reason, we used the K-Medoids clustering algorithm [12] with the following data: pre- and post-test scores from the previous study, session length, and scores on self-explanation questions. The algorithm produced two clusters. The students in one cluster happen to have low scores on the pre-test scores, while the student in the other cluster have high scores, and therefore we refer to the clusters as novices and advanced students in the rest of the paper. There were 12 novices and 10 advanced students. The two groups differ significantly on the pre/post-test scores and the scores on self-explanation questions, while there is no significant difference on the time spent with SQL-Tutor (Table 1).

Table 1. Comparisons between the two clusters (standard deviations in brackets)

	Total (22)	**Novices (12)**	**Advanced (10)**	**p**
Pre-test (%)	40 (13)	33 (11)	48 (11)	<.01*
Post-test (%)	70 (16)	63 (16)	79 (12)	.02*
P-SE (%)	83 (13)	76 (11)	92 (9)	<.01*
Time (s)	129 (54)	120 (59)	139 (52)	.43

We then analyzed the quality of the eye tracking samples, and had to eliminate data from four participants as there were too few valid data samples recorded. That left 10 students in the advanced group and 8 novices.

In order to be able to explore how students studied examples, we defined important Areas Of Interest (AOIs) for the system's interface, such as worked example (W), explanation (E) and database schema (D), and analyzed the data in terms of fixations on these AOIs and transitions between them. In order to analyze the differences in student behaviors, we defined a coding scheme EGPA (Eye Gaze Pattern Analysis) which categorizes eye movements into patterns and combine patterns into behaviors. Patterns are the smallest elements that describe eye gaze movements over a short time interval (1.5 seconds). We define four types of patterns: pure reading, mixed reading, transferring, and scanning. A pure reading pattern indicates that a student is solely paying attention to one AOI. For example, if the student keeps reading the explanation, such pattern is labelled *E*. If the student has a quick look at an AOI while reading another area, then we call it a mixed pattern. An example pattern of this type is *EdE*, showing that the student started reading the explanation, quickly glanced at the database schema and then continued with the explanation. A transferring pattern shows that the student's eye gaze moved from one area of interest to another AOI. An example is *WE*, in which the student's eye gaze moved from the example to the explanation. Finally, the scanning pattern (*S*) describes the situation when the student scans the screen. Behaviours are sequences of patterns. An example is (*W WE EdE EW WdW*), in which the students started by reading the example, then moved to read the explanation, during which he/she had a quick glance at the schema. Then, the student's eye gaze moved from the explanation to the worked example, and had another quick look at the database schema.

Table 2. Average pattern frequencies

	Percentages of students using the pattern		Average pattern frequency		
	Advanced	Novices	Advanced	Novices	p
All patterns			18.60 (5.19)	18.75 (5.26)	0.97
W	100%	100%	4.8 (2.2)	5.25 (1.39)	0.83
E	90%	100%	2.2 (1.32)	2.37 (1.19)	0.83
D	90%	25%	1.1 (0.57)	0.37 (0.74)	0.03*
WeW	40%	63%	1.2 (1.81)	0.62 (0.52)	0.90
WdW	50%	25%	1.4 (1.84)	0.25 (0.46)	0.24
EwE	60%	75%	1.2 (1.32)	2.12 (2.1)	0.41
EdE	20%	38%	0.3 (0.67)	0.5 (0.76)	0.57
WE	90%	100%	3.5 (2.01)	4.62 (1.69)	0.24
WD	40%	25%	0.4 (0.52)	0.25 (0.46)	0.63
EW	50%	50%	0.7 (0.82)	0.87 (1.13)	0.90
ED	50%	0%	0.5 (0.53)	0	0.08*
DW	30%	0%	0.3 (0.48)	0	0.32
DE	0%	25%	0	0.25 (0.46)	0.41
S	70%	100%	1 (0.94)	1.25 (0.46)	0.41

We classified all the eye gaze data using EGPA, and analyzed the data in terms of frequencies of patterns and behaviors for novices and advanced students (Table 2). There was no significant difference between the total number of patters used by novices and advanced students. The advanced students used the D and ED patterns significantly and marginally significantly more often than the novices ($p = .03$ and $p = .08$ respectively). The D pattern was used by 90% of advanced students compared to only 25% of novices. The ED pattern was not used by novices at all, while 50% of advanced students have used it.

We identified 42 distinct behaviors, of which 10 were used by more than one student. Table 3 presents those behaviors and their average frequencies. One behavior was used only by advanced students: (W WE E ED D). As the advanced students had prior knowledge about the concepts covered in the examples, they looked at the explanation and database schema to find new information which they have not learnt before. Advanced students used B8, B9 and B10 more often than novices. These three behaviours contain only one pattern; therefore, advanced students used less complex behaviour than novices. Such simple behaviours may be explained by advanced students having more knowledge. On the other hand, novices used B2, B4 and B6 more than advanced students. In B2, students first studied the worked example followed by reading explanation, and finally they restudied the worked example. B4 is similar to B2, but instead of restudying worked example, students had a quick look at the worked example while reading the explanation. B6 represents that students first studied the worked example followed by reviewing explanation, but they did not pay attention to the database schema.

Table 3. Average behaviour frequencies

Name	Behaviour	Advanced	Novices
B1	W WE EwE EW W	0.1	0.1
B2	W WE E EW W	0.2	0.4
B3	W WE E ED D	0.2	0.0
B4	W WE EwE	0.4	1.3
B5	W WE EdE	0.1	0.4
B6	W WE E	1.5	1.8
B7	W WD D	0.2	0.1
B8	WeW	1.1	0.6
B9	WdW	1.0	0.3
B10	W	1.7	1.0

The eye tracking data was also used as input for several Machine Learning algorithms available in RapidMiner [15] with the Weka plug-in [10]. We were interested in classifiers that predict the class of the student (novice or advanced) based on patterns and behaviors exhibited while studying examples (over the whole session). The input vectors were specified in terms of 26 features (16 patterns and 10 behaviors), the values of which are frequencies of use of a particular pattern/behavior. Leave-one-out cross-validation was carried out on the normalized data. We generated classifiers using the following algorithms: W-J48, W-Ladtree, W-BFTree, Rule Induction, W-JRip, and Naiive Bayes. A ‘W’ prefix indicates that the Weka implementation of the algorithm has been used. Table 4 reports the accuracies of the generated classifiers.

Table 4. Accuracy of classifiers

Classifiers	Accuracy
W-J48	61.1%
Rule Induction	66.6%
W-BFTree	72.2%
W-JRip	77.8%
Naive Bayes	77.8%
W-Ladtree	94.4%

The best classifier was produced by the W-Ladtree algorithm (Figure 2), with 94.4% accuracy. The classifier predicted the advanced students with 100% accuracy, while novices were predicted with 90% accuracy. The W-Ladtree classifier indicates that advanced students study database schema more than novices.

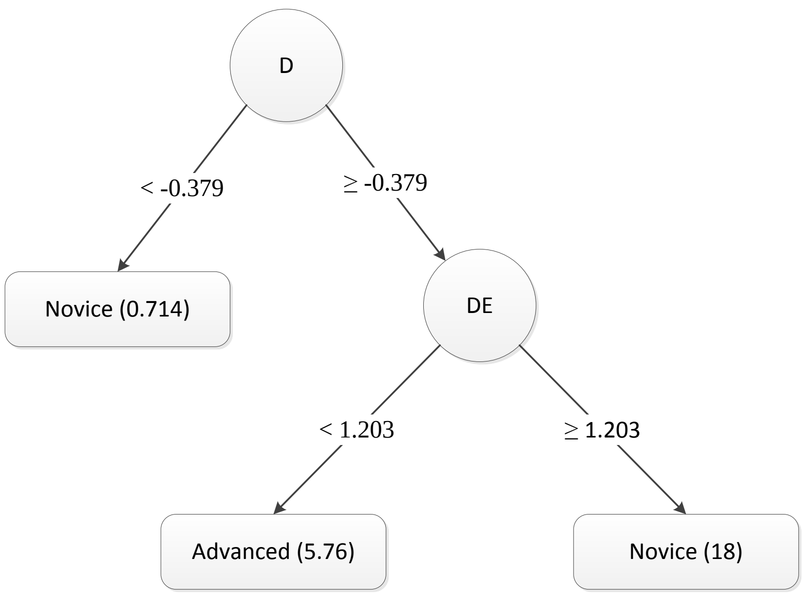

Fig. 2. The W-Ladtree classifier

4 Conclusions

Previous work has shown that learning from examples is beneficial but there has been no deep investigation of how students study examples. We conducted a study of the behaviors that novices and advanced students exhibit while studying examples in SQL-Tutor. Such information enables us to identify productive and unproductive approaches that students take to study examples. We collected information about all actions the participants took as well as the eye gaze data.

We found no significant difference in the time students spent studying the examples. The analyses of the students' eye gaze patterns show that advanced students studied the database schema significantly more than novices. Machine learning classifiers also corroborate this finding. Overall, the results emphasise the importance of the database schema for advanced students. Students need database information, such as names and semantics of tables and attributes, to understand examples. Therefore, looking at the database schema is a sign of learning from SQL examples. Now the question is why novices did not pay attention to this crucial area? Perhaps novices do not know how to study SQL examples. For instance, they may not know the basic concepts of primary keys and foreign keys in a database. Therefore, it is interesting to investigate whether or not prompting novices to study database schema while they study examples would improve students learning. All the analyses performed were based on data captured over the whole session; therefore, the results may change for longer or shorter sessions. It will be interesting to observe how patterns and behaviours change as students become more knowledgeable.

References

1. Bednarik, R. (2005). Potentials of eye-movement tracking in adaptive systems. In *Proc. 4th Workshop on the Evaluation of Adaptive Systems* (pp. 1–8).
2. Bloom, B.S. (1984). The 2-sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13, (pp. 4–16).
3. Bull, S., Cooke, N., & Mabbott, A. (2007). Visual attention in open learner model presentations: An eye-tracking investigation. In C Conati, K. McCoy, & G. Paliouras (Eds.), *Int. Conf. User Modeling* (pp. 187–196). Springer.
4. Charness, N., Reingold, E.M., Pomplun, M., Stampe, D.M. (2001) The perceptual aspect of skilled performance in chess: evidence from eye movements. Memory & Cognition 29(8), 1146-1152.
5. Conati, C., & Merten, C. (2007). Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge-Based Systems*, *20*(6), 557–574.
6. D'Mello, S., Olney, A., Williams, C., & Hays, P. (2012). Gaze tutor: A gaze-reactive intelligent tutoring system. *Int. Journal Human-Computer Studies*, 70(5), 370-389.
7. Elmadani, M., Mitrovic, A., & Weerasinghe, A. (2013). Understanding Student Interactions with Tutorial Dialogues in EER-Tutor. *Proc. 21st Int. Conf. Computers in Education* (pp. 30-40).
8. Gluck, K. A., Anderson, J. R., & Douglass, S. A. (2000). Broader Bandwidth in Student Modeling: What if ITS Were "Eye" TS? In *Intelligent Tutoring Systems* (pp. 504–513).
9. Goldberg, J. H., & Helfman, J. I. (2010). Comparing information graphics: a critical look at eye tracking. In *Proc.3rd BELIV'10 Workshop: Beyond time and errors: novel evaLuation methods for Information Visualization* (pp. 71–78).
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18.
11. Kardan, S., & Conati, C. (2012). Exploring gaze data for determining user learning with an interactive simulation. *User Modeling, Adaptation, and Personalization*, 126–138.

12. Kaufman, L. & Rousseeuw, P.J. (1987). Clustering by means of Medoids. Y. Dodge (Ed.) *Statistical Data Analysis Based on the L_1–Norm and Related Methods*, North-Holland, (pp. 405–416).
13. Liu, Y., Hsueh, P.Y., Lai, J., Sangin, M., Nussli, M.A., Dillenbourg, P. (2009) Who is the expert? analyzing gaze data to predict expertise level in collaborative applications. In IEEE Int. Conf. on Multimedia and Expo. (pp. 898-901).
14. Mathews, M., Mitrovic, A., Lin, B., Holland, J., & Churcher, N. (2012). Do your eyes give it away? Using eye tracking data to understand students' attitudes towards open student model representations. In *Proc. 11th Int. Conf. Intelligent Tutoring Systems* (pp. 422–427).
15. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. In L. Ungar, M. Craven, D. Gunopulos, & T. Eliassi-Rad (Eds.), *KDD '06: Proc.12th ACM SIGKDD Int. Conf. Knowledge discovery and data mining* (pp. 935–940).
16. Mitrovic, A. (1998) Experiences in Implementing Constraint-Based Modeling in SQL-Tutor. In: B. Goettl, H. Halff, C. Redfield, V. Shute (eds.), *Proc. Intelligent Tutoring Systems* (pp. 414-423).
17. Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. *Artificial Intelligence in Education*, *13*(2-4), 173–197.
18. Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, *124*(3), 372.
19. Shareghi Najar, A., & Mitrovic, A. (2013a). Examples and Tutored Problems: How can Self-Explanation Make a Difference to Learning? In H. C. Lane, K. Yacef, J. Mostow, P. Pavlik (Eds.), *Proc. 16th Int. Conf. Artificial Intelligence in Education* (pp. 339–348).
20. Shareghi Najar, A., & Mitrovic, A. (2013b). Do novices and advanced students benefit differently from worked examples and ITS? In L. H. Wong, C.-C. Liu, T. Hirashima, P. Sumedi, & M. Lukman (Eds.), *Int. Conf. Computers in Education* (pp. 20–29).
21. Sibert, J. L., Gokturk, M., & Lavine, R. A. (2000). The reading assistant: eye gaze triggered auditory prompting for reading remediation. In *Proc.13th Annual ACM Symposium on User interface software and technology* (pp. 101–107).
22. VanLehn, K. (2011) The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist*, 46(4), 197-221.
23. Wang, H., Chignell, M., & Ishizuka, M. (2006). Empathic tutoring software agents using real-time eye tracking. *Symposium on Eye tracking research & applications* (pp. 73–78).
24. Woolf, B. (2008) Building intelligent interactive tutors: student-centered strategies for revolutionizing e-learning. Morgan Kaufmann.