Chen, W. et al. (Eds.) (2016). Proceedings of the 24th International Conference on Computers in Education. India: Asia-Pacific Society for Computers in Education

Do Novices and Advanced Students benefit from Erroneous Examples differently?

Xingliang CHEN[a*], Tanja MITROVIC[b] & Moffat MATHEWS[c]
[a,b,c] *University of Canterbury, New Zealand*
*xingliang.chen@pg.canterbury.ac.nz

Abstract: Learning from problem solving, worked examples, and Erroneous Examples (ErrEx) have all proven to be effective learning strategies. However, what kind of learning material should be provided to students with different level of prior knowledge within Intelligent Tutoring Systems (ITSs) is still an open question. Recently, alternating worked examples and problem solving (AEP) has been shown to benefit students compared to problems only or worked examples only in SQL-Tutor (Najar & Mitrovic, 2013). However, how students with different prior knowledge learn from ErrEx in SQL-Tutor is unknown. In this paper, we compared AEP to a new instructional strategy (WPEP) which provides ErrEx in addition to worked examples and problem solving to students. The results show that that both novices and advanced students improved their post-test scores significantly in either condition. Our findings also show that novices acquired significantly more debugging knowledge when erroneous examples were presented (WPEP) in comparison to the AEP condition. Moreover, both novices and advanced students benefitted from ErrEx. In particular, advanced students who studied with erroneous examples showed better performance on problem solving as measured by the number of attempts per problem.

Keywords: Problem solving, Worked examples, Erroneous examples, Novices, Advanced students, Intelligent Tutoring System, SQL-Tutor

1. Introduction

Previous studies have compared studying from Worked Examples (WE) to unsupported problem solving (Atkinson, Derry, Renkl, & Wortham, 2000; Kalyuga, Chandler, Tuovinen, & Sweller, 2001; Van Gog & Rummel, 2010). A worked example consists of a problem statement, its solution and additional explanations, and therefore provides a high level of assistance to students. WEs reduce cognitive load on the students' working memory, thus allowing students to learn faster and deal with more complex problems (Sweller, Van Merrienboer, & Paas, 1998). The effectiveness of WEs has been investigated in various studies (Atkinson et al., 2000; Kirschner, Sweller, & Clark, 2006; Van Gog & Rummel, 2010). It has been shown that WEs are beneficial for novices (Kirschner et al., 2006), but problem solving was proven to be superior to WEs for students with high prior knowledge (Kalyuga et al., 2001). The effects of problem solving only (PS), worked-examples only (WE), worked-examples/problem-solving pairs (WE-PS) and problem-solving/worked-examples pairs (PS-WE) have been studied on novices by using electrical circuits troubleshooting tasks (Van Gog, Kester, & Paas, 2011). The results suggest that the performance of the WE and WE-PS conditions were significantly higher than that of the PS and PS-WE conditions. However, van Gog later claimed that the result of WE-PS and PS-WE conditions might be not sufficient because the examples and problems should be identical within and across pairs (van Gog, 2011). As a consequence, she employed an example-problem sequence (EP condition) and a problem-example sequence (PE) condition for learning. The result demonstrated that students showed better performance on EP condition than PE condition. For advanced students, they have sufficient knowledge to learn from practice without much feedback or support, therefore, worked examples are not as effective for them (Kalyuga et al., 2001), since worked examples provide redundant assistance for high prior knowledge students.

Many studies have also demonstrated the benefits of learning from WEs compared to learning from tutored problem solving (TPS) in ITSs (McLaren & Isotani, 2011; Schwonke et al., 2009). In comparison to unsupported problem solving, ITSs provide adaptive feedback, hints or other types of help to students. Kim, Weitz, Heffernan, and Krach (2009) compared ITS with pure worked examples in Statistics and both conceptual and procedural knowledge acquisition were measured. The results of the first experiment show that there was no significant difference between novices and advanced students. In the second experiment, Kim et al. (2009) found worked examples help improve both conceptual and procedural knowledge, and tutored problem solving significantly improve conceptual knowledge acquisition. McLaren and Isotani (2011) compared worked examples only, tutored problem solving only, and alternating worked examples / tutored problem in the domain of chemistry using Stoichiometry Tutor. The results show no difference in learning gain from the three conditions but worked examples only resulted in a shorter learning time. Contrary to that, in a study conducted in SQL-Tutor, a constraint-based tutor that teaches database querying in SQL, Najar and Mitrovic (2013) found that alternating worked examples with problem solving (AEP) significantly improved novices' conceptual knowledge in comparison with tutored problem solving only (TPS), but advanced students did not improve significantly from worked examples only (WE) condition. The paper concludes that the best condition for both novices and advanced students was AEP, which presented isomorphic pairs of WE and TPS to students.

Most recent studies have focused on erroneous examples, which provide incorrect solutions and require students to find and fix errors. Große and Renkl (2007) examined learning outcomes in the domain of mathematical probability when students explained both correct and incorrect examples. They found that erroneous examples were beneficial for advanced students on far transfer. Novices did significantly better when errors were highlighted, but advanced students did not show any benefit. Durkin and Rittle-Johnson (2012) studied whether learning with incorrect and correct decimals examples is more effective in comparison to learning with from correct examples only. They found that studying both worked examples and erroneous examples resulted in higher procedural and declarative knowledge compared to worked examples only condition.

While the studies on erroneous examples discussed above were paper based, there have not been many studies on the benefits of learning from erroneous examples with ITSs. Tsovaltzi, McLaren, Melis, and Meyer (2012) demonstrated the effect of learning from erroneous examples of fractions in an ITS. They found that erroneous examples with interactive help improved 6th graders' metacognitive skills compared to problem solving condition and erroneous examples condition with no help. Additionally, 9th and 10th grade students improved their problem solving skills and conceptual knowledge while studying erroneous examples with additional help. Another study by Booth, Lange, Koedinger, and Newton (2013) with the Algebra I Cognitive Tutor found that students who explained correct and incorrect examples significantly improved their post-test performance compared with those who received worked examples only. In addition, the erroneous examples condition and the combined WE / ErrEx condition were beneficial for conceptual understanding of algebra, but not for procedural knowledge.

We conducted a study that compared learning from alternating worked examples and tutored problems (AEP) to a sequence of worked example/ problem pairs (WPEP) in the content of SQL-Tutor. The results show that erroneous examples prepare students better for problem solving compared to worked examples (Chen, Mitrovic, & Mathews, 2016). In this paper, we present the results of additional analyses looking at how students with different levels of prior knowledge performed in that study. Our hypothesis is that the effect of the addition of erroneous examples to WEs and TPS would be more pronounced for students with high level of prior knowledge.

2. SQL-Tutor

We conducted a study with SQL-Tutor (Mitrovic, 2003), a constraint-based ITS for teaching the Structured Query Language (SQL). Three different modes of SQL-Tutor were used in the study, corresponding to WEs, ErrExs and tutored problem solving. Figure 1 illustrates the problem solving interface we used in this study. The left pane presents the structure of the database schema, providing information about tables, their attributes and the data stored in the database. The middle pane is the problem-solving environment. At the beginning of a problem, only the input areas for the *Select* and *From* clauses are shown; the student can click on other clauses to get the input boxes as necessary. The right pane presents feedback. SQL-Tutor supports six levels of feedback. *Simple (Positive/Negative)* feedback, which is the lowest level of assistance, simply specifies whether the solution is correct or reports the number of errors the student made. *Error Flag* feedback indicates the part of the solution that is incorrect. *Hint* discusses a mistake the student made, pointing out the domain principle which was violated. *Partial Solution* presents the correct version of the solution component where the student made an error. *List all errors* feedback identifies all errors the student made. *Complete solution* feedback provides the full solution. *Simple* feedback is the default feedback level for the first submission, unless overridden by the student. The feedback level is automatically increased up to the *Hint* level, but the student can ask for any feedback level at the time of submitting the solution.

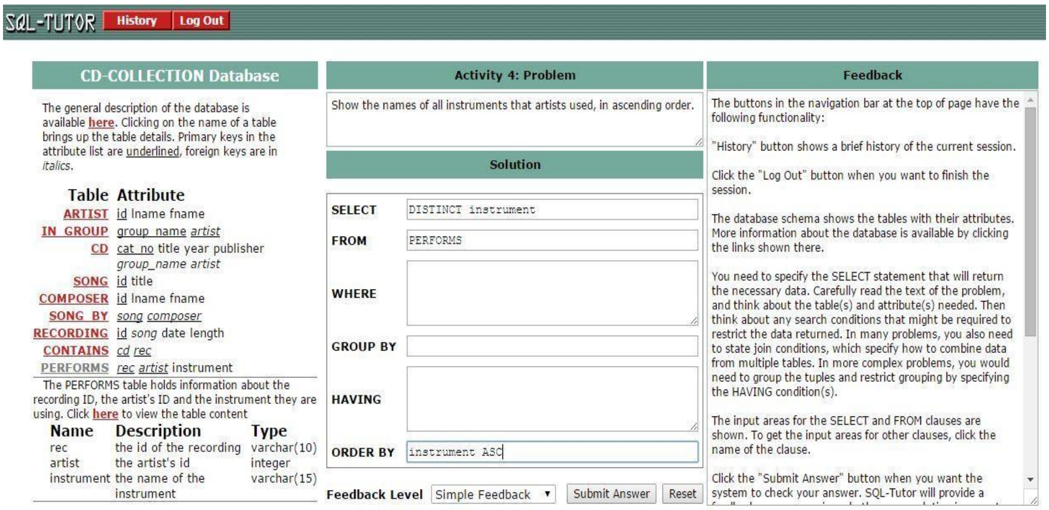

Figure 1. Screenshot of the problem-solving mode of SQL-Tutor

The interface of the worked example mode is illustrated in Figure 2. An example problem with its solution and explanation is presented in the center pane. A student can click the *Continue* button to confirm that they s/he has studied the example. Figure 3 shows the interface of erroneous example mode. The system provides an incorrect solution for a problem in the center pane. The student is required to analyze the solution, find errors and correct them. Similar to the problem-solving mode, the student can submit the solution to be checked by SQL-Tutor multiple times. In the situation shown in Figure 3, the student has identified the SELECT clause as being incorrect, and is defining the new version of it. The student has also added the *Group by* and *Order by* clauses.

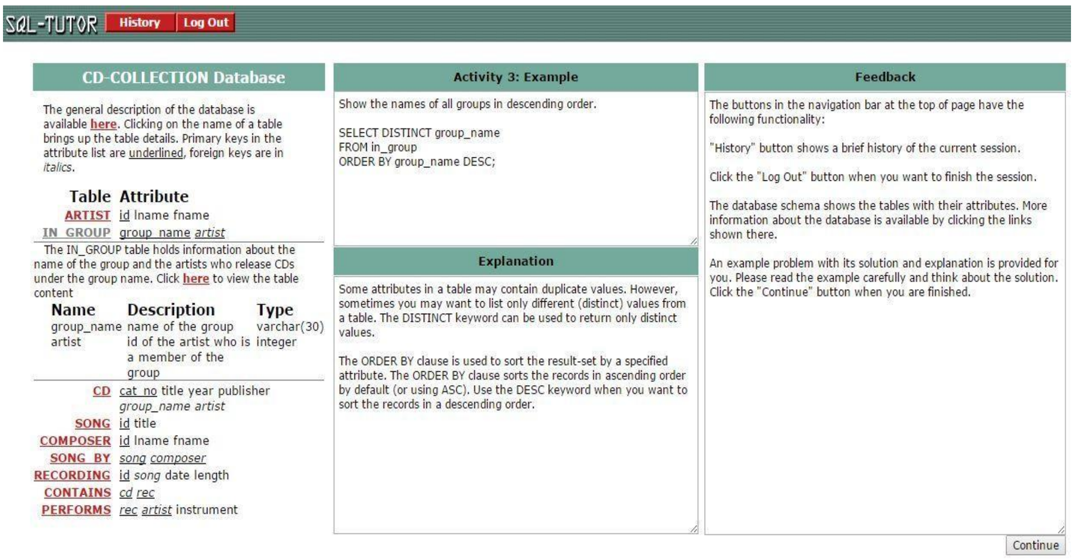

Figure 2. Screenshot of the worked example mode of SQL-Tutor

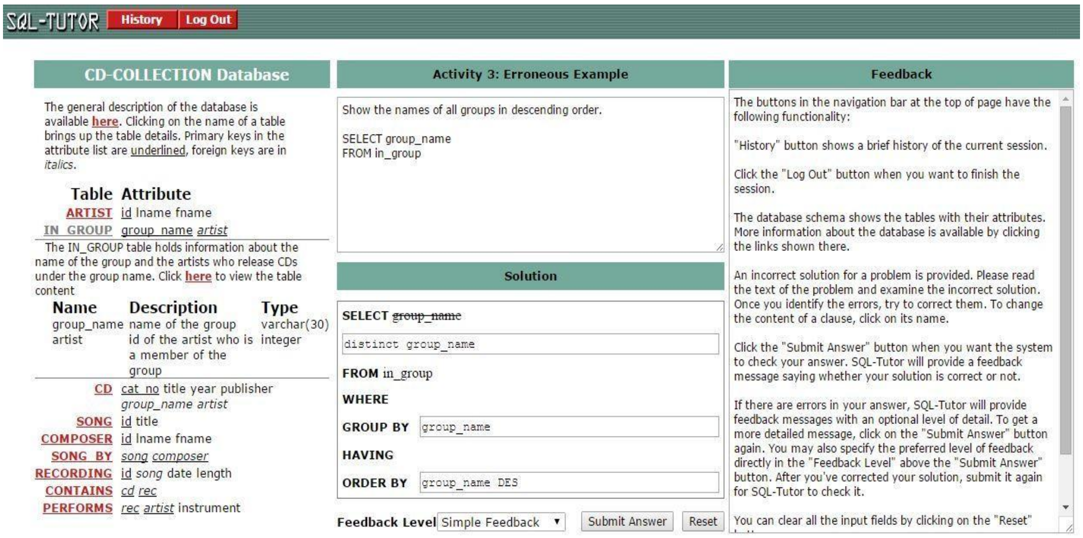

Figure 3. Screenshot of the erroneous example mode of SQL-Tutor

Previous studies have demonstrated that students learn more while they self-explain, but that many students do not self-explain spontaneously (Chi, Leeuw, Chiu, & LaVancher, 1994; Kim et al., 2009; Weerasinghe & Mitrovic, 2006). A common method to encourage students to self-explain is to provide Self-Explanation (SE) prompts. Previous work has found that problem solving help improve procedural knowledge more than conceptual knowledge, while WEs result in higher level of conceptual knowledge (Kim et al., 2009; Schwonke et al., 2009). Consequently, Najar and Mitrovic (2013) developed Conceptual-focused Self-Explanation (C-SE) prompts that support students to self-explain relevant domain concepts after problem solving, and Procedural-focused Self-Explanation (P-SE) prompts that aid students to self-explain solution steps after WEs. A C-SE prompt is provided after a problem is solved, in order to assist students to acquire conceptual knowledge corresponded to the problem they just completed (e.g. *What does DISTINCT in general do?*). On the other hand, a P-SE prompt is presented after WEs to aid learners to focus on problem-solving approaches (e.g. *How can you specify a string constant?*). In this study, we provided C-SE and P-SE prompts alternatively after ErrExs, since ErrExs contain both properties of problems and WEs.

3. Experiment Design

The study was conducted with volunteers enrolled in an introductory database course at the University of Canterbury, in regular labs scheduled for the course (100 minutes long). Prior to the study, the students have learnt about SQL in lectures, and had one lab session. There were two conditions: Alternating Examples and Problems (AEP), the most effective learning condition from (Najar & Mitrovic, 2013), and the experimental condition consisting of a fixed sequence of Worked example / Problem pairs and Erroneous example / Problem Pairs (WPEP). In both conditions, there were 20 tasks to be completed in a fixed order (of increasing difficulty), with the only difference being whether the tasks were presented as problems to be solved, WEs or ErrExs.

The students were randomly assigned to either AEP or WPEP condition after they logged on to SQL-Tutor, and then the pre-test was provided. The pre-test and post-test were administered online. After completing all 20 tasks, the participants received the post-test of similar complexity and length to the pre-test. Figure 4 shows the study design. The pre/post-tests consisted of 11 questions each. Questions 1-6 were multi-choice or true-false questions, which measured conceptual knowledge (with the maximum of 6 marks). Questions 7-9 focused on procedural knowledge; question 7 was a multi-choice question (one mark), followed by a true-false question (one mark), while question 9 required the student to write a query for a given problem (four marks). The last two questions presented incorrect solutions to two problems, and required the student to correct them, thus measuring debugging knowledge (six marks). Therefore, the maximum mark on the tests was 18.

AEP	WPEP
Pre-Test	
20 Problems / WEs (10 isomorphic pairs)	10 problems / WEs (5 isomorphic pairs), and 10 problems / ErrExs (5 isomorphic pairs), presented in alternation
Each problem followed by a C-SE prompt and each example followed by a P-SE prompt	
Post-Test	

Figure 4. Study design

4. Results

There were 64 participants in the study. Since the participation was voluntary, not all students completed all phases of the study, and we removed data about 38 students who have not finished the post-test. We present the results obtained by analyzing the data collected from the remaining 26 students (15 in the AEP and 11 in the WPEP condition). We classified students post-hoc based on their pre-test scores; the students whose pre-test scores are lower than 66% (the median of the pre-test scores for the whole group) were classified as novices, and the rest as advanced students (12 novices, 14 advanced students). Table 1 shows the overall scores, as well as scores for novices and advanced students.

Table 1: The pre-test scores (%)

	All students (26)	Novices (12)	Advanced students (14)
All questions	65.81 (13.14)	54.63 (6.3)	75.4 (9.17)
Conceptual questions	53.85 (17.2)	41.67 (13.3)	64.29 (12.84)
Procedural questions	85.26 (16.72)	81.91 (18.41)	88.1 (15.23)
Debugging questions	58.33 (24.15)	40.28 (16.6)	73.81 (18.16)

Table 2 shows the basic statistics for novices. The Mann-Whitney U-test revealed that there were no significant differences between the two conditions on the pre-test scores, post-test scores and the normalized learning gain. The Wilcoxon signed-test shows that novices in both conditions improved significantly between the pre- and post-test (the *Improvement* row of Table 2). The effect sizes (Cohen's d) are high for both conditions, with the WPEP condition having a higher effect size. On average, the students spent 63 minutes interacting with the learning tasks. There was no significant difference on the total interaction time between the two conditions. The students in both conditions solved the same number of problems (10). The AEP condition had 10 worked examples, while the WPEP condition had 5 worked examples and 5 erroneous examples. We expected erroneous examples to take more time compared to worked examples, but the difference was not significant.

Table 2: The basic statistics for Novices

	AEP (6)	WPEP (6)	p
Pre-test (%)	52.31 (7.94)	56.94 (3.4)	.28
Post-test (%)	80.09 (13.77)	91.2 (7.54)	.13
Improvement	$W = 21, p < .05, d = 1.54$	$W = 21, p < .05, d = 1.83$	
Normalized learning gain	0.57 (0.28)	0.8 (0.17)	.12
Interaction time (min)	67.71 (15.9)	58.78 (14.73)	.2

The basic statistics for advanced students are given in Table 3. The Mann-Whitney U-Test revealed no significant differences between the two groups on pre- and post-test scores, as well as on the normalized learning gain. The Wilcoxon signed-rank test identified significant improvements ($p < .05$)

between the pre- and post-test scores for both conditions (the *Improvement* row in Table 3). The effect sizes are also high for both groups, with the WPEP group having a higher effect size (d = 1.73).

Table 3: The basic statistics for Advanced Students.

	AEP (9)	WPEP (5)	p
Pre-test (%)	77.16 (9.8)	72.22 (7.86	.3
Post-test (%)	98.46 (3.7)	97.22 (3.93)	.5
Improvement	W = 45, p < .05, d = 1.62	W = 21, p < .05, d = 1.73	
Normalized learning gain	0.94 (0.13)	0.9 (0.14)	.5
Interaction time (min)	69.93 (15.7)	66.86 (8.52)	.84

We measured the improvement of conceptual knowledge, procedural knowledge and debugging knowledge in term of different pre-/post-test questions. Table 4 presents the scores on the three types of questions for novices and advanced students from the two conditions. The improvement on conceptual questions was significant for novices and advanced students in both AEP and WPEP conditions. In the WPEP condition, the score for debugging questions improved significantly for novices (W = 15, p = .043) and marginally significantly for advanced students (W = 10, p = .059), while only advanced students from the AEP condition improved their scores on debugging questions (W = 36, p = .01). The novices from the AEP condition did not improve their debugging knowledge. In the AEP condition, the score for procedural questions improved marginally significantly for novices (W = 10, p = .068) and advanced students (W = 10, p = .059), while there was no significant improvement on procedural questions for novices or advanced students in WPEP condition. The novices and advanced students from WPEP condition started with a very high level of procedural questions, as evidenced by the score of 93.06% and 90% respectively on the relevant pre-test questions. The normalized gain on debugging questions for the AEP group was 0.15 (sd = .71), while from the WPEP group it was 0.76 (sd = .39); the difference is marginally significant (U = 29.5, p = .063) and the effect size is large (d = .96). The fact reveals that both advanced and novice WPEP students improved on debugging knowledge.

Table 4: Detailed scores on pre-/post-tests.

		Questions	Pre-test (%)	Post-test (%)	p
AEP (15)	Novices (6)	Conceptual	44.44 (13.61)	88.89 (13.61)	.026**
		Procedural	70.83 (18.07)	94.44 (8.61)	.068*
		Debugging	41.67 (20.41)	56.94 (34.73)	ns
	Adv. (9)	Conceptual	66.64 (14.43)	98.15 (5.56)	0.007**
		Procedural	87.04 (16.2)	100 (0)	.059*
		Debugging	77.78 (14.43)	97.22 (5.89)	.01 **
WPEP (11)	Novices (6)	Conceptual	38.89 (13.61)	86.11 (6.8)	0.02**
		Procedural	93.06 (11.08)	100(0)	ns
		Debugging	38.89 (13.61)	87.5 (19.54)	.043**
	Adv. (5)	Conceptual	60 (9.13)	96.67 (7.45)	0.41**
		Procedural	90 (14.91)	95 (11.18)	ns
		Debugging	66.67 (23.57)	100 (0)	.059*

We also investigated whether correct and erroneous examples prepare novices and advanced students differently for problem solving. As explained previously, ten learning tasks given to learners were problems to be solved. Table 5 illustrates the average number of attempts (i.e. submissions) for ten problems. Overall, advanced students from the AEP condition made marginally significantly more attempts (U = 9, p = .072) on the ten problems, as evidenced from the results of the Mann-Whitney U Test. The table also presents the two measures for various subsets of problems, identified on the basis of the previous learning task. Problems 4, 8, 12, 16 and 20 were presented in the WPEP condition after ErrExs, and in the AEP condition after WEs. For those five problems, there was a marginally significant difference between the two conditions for advanced students (U = 8.5, p = .061), but there was no significant difference between the two conditions for novices. On the other hand, problems 2, 6, 10, 14 and 18 were presented to both conditions after WEs. For those problems, we found no significant differences between the two conditions on attempts for either novices or advanced students. These findings show that erroneous examples may prepare advanced students better for problem solving compared to worked examples. As the sample size is small, a larger study is necessary to confirm this result.

Table 5: Number of attempts on problems

		AEP	WPEP	p
All problems	Novice	4.17 (1.4)	3.17 (1.12)	ns
	Adv.	4.79 (1.91)	2.98 (1.1)	.072*
Problems 2,6,10,14,18	Novice	3.67 (1.27)	2.97 (1.59)	ns
	Adv.	3.24 (2.28)	2.32 (0.46)	ns
Problems 4,8,12,16,20	Novice	4.67 (1.61)	3.37 (1.17)	ns
	Adv.	6.33 (2.27)	3.64 (1.84)	.061*

5. Discussion and Conclusions

Previous studies show that worked examples are more beneficial for novices compared to problem solving (McLaren, van Gog, Ganoe, Yaron, & Karabinos, 2014; Najar & Mitrovic, 2013; van Gog, 2011). Najar and Mitrovic (2013) demonstrated that alternating WEs with problem solving was the best strategy in SQL-Tutor compared to learning from examples only, or tutored problem solving only. However, the inclusion of ErrEx has not been studied before in this instructional domain. In this study, we compared the performance of students with different levels of prior knowledge in two conditions: alternating worked example / problem (AEP), and worked example / problem pairs and erroneous example / problem pairs (WPEP). We found no significant differences between AEP and WPEP conditions on pre- and post-test performance, but the participants in both conditions improved significantly their scores on the post-test.

This paper presented additional analyses of performance of students who start with different levels of background knowledge. The findings show that both novices and advanced students in the WPEP condition improved their debugging knowledge marginally significantly than their peers of similar abilities from the AEP condition. A possible explanation is that extra learning during the correcting phase of erroneous examples contributes to this benefit. Therefore, the students with all knowledge levels benefitted from erroneous examples.

In particular, advanced students who learnt with erroneous examples showed higher performance on problem solving as measured by the number of attempts. This suggests that the erroneous examples aid advanced students more than worked examples. When asked to identify and self-explain errors in erroneous examples, advanced students may engage in deeper cognitive processing compared to when they engage with WEs. Therefore, they were better prepared for concepts required in the next isomorphic problem in comparison to the situation when they received WEs.

The small sample size is one of the limitations of our study. The timing of the study coincided with assignments in other courses the students were taking; so many students have not completed the study. We plan to conduct a larger study in order to confirm our conclusions.

Previous study reported that erroneous examples led to a delayed learning effect (McLaren, Adams, & Mayer, 2015). However, our experiment design did not include a delayed test. It would be interesting to see the results of the delayed learning effect.

This study suggests that the students with various levels of prior knowledge might perform differently with worked examples, erroneous examples, and problem-solving. Additionally, all participants in our study were familiar with SQL because they learnt SQL in the lectures prior to our study. In our future work, we plan to develop an adaptive strategy that decides what learning activities

(TPS, WE or ErrEx) to provide to the student based on his/her performance and prior level of knowledge would be an important issue.

References

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of educational research, 70*(2), 181-214.

Booth, J. L., Lange, K. E., Koedinger, K. R., & Newton, K. J. (2013). Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. *Learning and Instruction, 25,* 24-34.

Chen, X., Mitrovic, A., & Mathews, M. (2016). Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples? In Int. Conf. Intelligent Tutoring Systems (pp. 13-22). Springer International Publishing.

Chi, M. T., Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive science, 18*(3), 439-477.

Durkin, K., & Rittle-Johnson, B. (2012). The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learning and Instruction, 22*(3), 206-214.

Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and Instruction, 17*(6), 612-634.

Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of educational psychology, 93*(3), 579.

Kim, R. S., Weitz, R., Heffernan, N. T., & Krach, N. (2009). Tutored problem solving vs."pure" worked examples. Proc. 31st Annual Conf. of the Cognitive Science Society. Austin, TX: Cognitive Science Society.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist, 41*(2), 75-86.

McLaren, B. M., & Isotani, S. (2011). When is it best to learn with all worked examples? In Int. Conf. Artificial Intelligence in Education (pp. 222-229). Springer Berlin Heidelberg.

McLaren, B. M., van Gog, T., Ganoe, C., Yaron, D., & Karabinos, M. (2014). Exploring the assistance dilemma: comparing instructional support in examples and problems. In Int. Conf. Intelligent Tutoring Systems (pp. 354-361). Springer International Publishing.

McLaren, B. M., Adams, D. M., & Mayer, R. E. (2015). Delayed Learning Effects with Erroneous Examples: a Study of Learning Decimals with a Web-Based Tutor. *Artificial Intelligence in Education, 25*(4), 520-542.

Mitrovic, A. (2003). An intelligent SQL tutor on the web. *Artificial Intelligence in Education*, 13, 173-197.

Najar, A. S., & Mitrovic, A. (2013). Examples and tutored problems: How can self-explanation make a difference to learning?. In *Int. Conf. Artificial Intelligence in Education* (pp. 339-348). Springer Berlin Heidelberg.

Najar, A. S., & Mitrovic, A. (2013). Do novices and advanced students benefit differently from worked examples and ITS. In Proc. Int. Conf. Computers in Education (pp. 20-29).

Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior, 25*(2), 258-266.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review, 10*(3), 251-296.

Tsovaltzi, D., McLaren, B. M., Melis, E., & Meyer, A.-K. (2012). Erroneous examples: effects on learning fractions in a web-based setting. *International Journal of Technology Enhanced Learning, 4*(3-4), 191-230.

van Gog, T. (2011). Effects of identical example–problem and problem–example pairs on learning. *Computers & Education, 57*(2), 1775-1779.

van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology, 36*(3), 212-218.

van Gog, T., & Rummel, N. (2010). Example-based learning: Integrating cognitive and social-cognitive research perspectives. *Educational Psychology Review, 22*(2), 155-174.

Weerasinghe, A., & Mitrovic, A. (2006). *Studying human tutors to facilitate self-explanation.* Paper presented at the Intelligent Tutoring Systems.

Weerasinghe, A., & Mitrovic, A. (2006). Studying human tutors to facilitate self-explanation. In Int. Conf. Intelligent Tutoring Systems (pp. 713-715). Springer Berlin Heidelberg.