International Journal of Social Science and Economic Research

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

COMPARING THE EFFECTIVENESS OF AI-POWERED EDUCATIONAL SOFTWARE TO HUMAN TEACHERS

John Leddo[1] and Khushi Garg[2]

[1]director of research at MyEdmaster
[2]researcher at MyEdMaster.

MyEdMaster, LLC, 13750 Sunrise Valley Drive, Herndon, VA, United States of America

DOI: 10.46609/IJSSER.2021.v06i03.015 URL: https://doi.org/10.46609/IJSSER.2021.v06i03.015

ABSTRACT

Current educational assessments report that, nationally, the majority of United States students are performing below grade level (National Assessment of Educational Progress, 2019). Because this trend has been ongoing and there is a current teacher shortage, it seems unlikely that this problem will be rectified in the foreseeable future unless a paradigm shift occurs. The present paper explores the question of whether using artificial intelligence (AI)-powered educational software can effectively stand in for human teachers. High school students were taught the difficult Algebra II topic of dividing complex numbers. Half were taught by active high school math teachers while half were taught by AI-powered software that our team created. Results showed that, on a post-test, students using the AI-powered software scored, on average, 37% higher than those taught by teachers. A closer examination of the data reveals that the distribution of scores of students taught by teachers is rectangular with some performing highly, some performing in the average range and some failing, while most students who were taught by the AI software scored in the A range and no one scored below 70%. Results suggest that our AI software could potentially supplement or stand in for teacher-led instruction with an overall improvement in student performance.

Introduction

Over the years, there has been a proliferation of educational software available to individual students and classrooms. Since the introduction intelligent tutoring systems (ITSs- cf., Brna, Ohlsson and Pain, 1993; Greer, 1995), there has been an interest in seeing whether educational software that uses artificial intelligence (AI) to emulate human instructors can lead to improvements in educational achievement. We have been involved in this field of endeavor

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

ourselves. Previously, we have reported research in which students using our AI-based ITS software outperformed, on average, those using Khan Academy's software by 80% (Leddo et al., 2016) and those using Pearson Education's electronic textbooks by 300% (Leddo et al., 2019).

While comparisons among educational products is useful in helping schools or consumers pick which product is best for them, there is a growing trend that raises new questions about the potential role of educational software. As has been the case for years, the US Department of Education's National Assessment of Educational Progress has been finding that the majority of students perform below grade level in reading and mathematics. For example, in the most recent assessment (2019), 75% of 12th graders performed below grade level in math (with 40% performing below a basic level) and 63% of 12th graders performed below grade level in reading. The scores for math are down slightly from those in 2015 and the scores for reading are unchanged from those in 2015. In other words, education in America is not improving, and if anything, it is getting slightly worse.

Ordinarily, society would look to teachers to fix the problem of declining educational performance. Educational software, when used, would serve as a supplement to teacher-led classroom instruction. However, there are two trends that suggest that the time may be ripe for a different way of thinking. The first major trend is that there is a critical shortage of teachers in the United States, both in terms of sheer numbers of teachers and those who are credentialed (Garcia and Weiss, 2020). This suggests that turning to teachers alone may not be sufficient to address low student performance.

Second, with the increase in Internet resources, more and more people are engaging in self-directed learning. Self-directed learning involves learning without the aid of a teacher, and the question here becomes whether a student can learn just as well without a teacher as s/he can with a teacher. In our previous research (Leddo et al., 2017), we examined the question whether students learning basic webpage design on their own using publicly-available videos on the web could perform comparably to those who learned from a human teacher. In that study, we examined students who were in either their school's gifted and talented or regular academic programs. We found that gifted and talented students performed equally well when self-taught as they did when taught with a teacher, but students in standard programs performed better when taught by a teacher.

The present study investigates the question as to whether students taught with AI-based instructional software can learn academic material equally well as those taught with a human teacher. If so, this would suggest that one way to boost academic performance across the board in the face of a teacher shortage is to have some students learn using AI-based software instead

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

of attending traditional teacher-led classrooms. We investigate this research question by having students learn a difficult topic Algebra II math, division of complex numbers, either using AI-based educational software or in a teacher-led classroom.

Methods

Participants

Participants were 40students who were recruited from middle school and high schools in Fairfax and Loudoun counties in Virginia. Each one was enrolled in a geometry math class or in Algebra I and had learned the process of FOILing binomials, a pre-requisite for dividing complex numbers. FOIL stands for First, Outer, Inner, Last and is a process based on the distributive property. It was necessary that each student had not yet taken Algebra II since the subject matter of the present study, division of complex numbers, is a topic that is covered in the Algebra II curriculum. We wanted to make sure that participants in the present study had no prior knowledge of this topic. All participants met this criterion.

For the teacher-led condition, there were two high school math teachers, each of whom was currently teaching Algebra II. Both teachers were paid for their participation.

Topic Taught and Technology Used

The topic used in the present study was division of complex numbers of the form a + bi, where i is the square root of -1. This topic is typically part of the Algebra II curriculum.

The experimental condition technology used the educational software that combined AI and voice/natural language processing technologies. The software presented an initial lesson that was available in both video and text formats. It was supplemented with the following features.

Querying the Software

While a student goes through the division of complex numbers lesson, he or she has the ability to verbally query the system and receive an answer to his or her question. This is done by clicking on a microphone icon located on the top right part of the screen and then speaking the question.The student's question is translated into text and shown to the student, who can edit it if there is a mistake in the speech to text process. Alternatively, the student can type in the question directly instead of speaking it.

Since there are multiple ways to ask a question, the first step of the process is to match the student's question with a question the e-textbook can answer. Semantic similarity calculation is

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

the key process used here. The basics of the semantic similarity calculation is word embeddings. Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct the linguistic contexts of words. With Word2vec technology, we use multiple methods to calculate the semantic similarity between two sentences. The easiest method is the baselines method. This method takes the average of the word embeddings of all words in each sentence and calculates the cosine between the resulting embeddings. The result of the calculation is between 0 to 1. If two sentences are highly similar, the score is closer to 1. For example, the sentence "a man is cutting up a cucumber" has the same meaning as the sentence "a man is slicing a cucumber." The score of these two sentences is 0.97. A similar process is used when a student asks a question. Semantic similarity technology outputs the highest score between the student's question and all the questions in our database. If the score of a question is extremely low (for example, less than 0.15), the student's question may not be a normal sentence. It may be no logical sentence or just repeating words. In this case, our system requests that the student input a more well-structured sentence. If the score of this question is high enough (for example, higher than 0.8), we claim that this question has the same meaning with student's question. If the score of this question is intermediate (for example, more than 0.5 and less than 0.8), the system is not quite sure of the meaning of the student's question. In this case, we match the student's question to the question with the highest similarity score that the system can answer. The student is then asked if the matched question is what the student is asking. If the student says yes, the system answers the matched question. If the student says no, the system looks for the next highest match above .5 and asks again. A maximum of three cycles are possible. If the system goes through three matching cycles and finds no match or it cannot find a match with a similarity rating above .5, it assumes it cannot answer the student's question and informs the student of that fact.

Assessing the Student

Following the instruction, the software assesses the student to see if he or she understood the material. The assessment uses the same voice interface as the querying feature. Students are verbally presented with questions and respond in kind. The system then processes the students' responses and matches them to the underlying knowledge model. If there is a match as described in the previous section, the correct answer is deemed to have been given. If the match is between .5 and .8, the system clarifies what the student stated as described above. If the match is low or the student states that the system's interpretation of the student's response is wrong, the system assumes the student gave the wrong answer and corrective feedback is provided.

The assessment queries themselves are derived from an underlying knowledge model of the subject matter to be mastered. The knowledge model is based on the Integrated Knowledge

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

Structure (INKS) framework developed by John Leddo (Leddo, 1994), which combines several knowledge types described in the cognitive psychology literature. These include factual or semantic knowledge (Quillian, 1966), general problem solv3ing plans or scripts (Schank and Abelson, 1977), problem solving procedures or production rules (Anderson, 1982), and causal principles or mental models (de Kleer and Brown, 1981) that explain why procedures are down. Queries are presented to students that are derived from each of these types of knowledge (e.g., "What is i?", "How do you divide two complex numbers?") to assess whether they have a thorough understanding of the subject matter being taught. Our previous research (Leddo and Sak, 1994) found that assessments of student knowledge produced using the INKS framework as their bases correlated .88 with how well students could solve practical problems that use this knowledge.

The Electronic Worksheet

When students finish learning the subject matter and are assessed and remediated, if necessary, they proceed to an electronic worksheet where they can do practice problems. Students type in their work, step by step. When this happens, the software processes the students' step-by-step work and offer hints when requested or corrective feedback to mistakes when made (this is described in more detail below). Additionally, unlike most educational software, the present software allows students to type in their own problems onto the electronic worksheet and receive hints and feedback just as what happens when they solve problems taken from the database (the mechanism for this is also described below). However, this content entry feature was not used in the present experiment.

In the present experiment, each question provided to the student involved division of complex numbers of the form (a+bi)/(c+di), where a, b, c, and d are integers. Once the student sees the problem, he or she types in his or her work step-by-step on the electronic worksheet. The worksheet is organized by lines, with one line given for each step. When a student is through typing in a step, he or she clicks on an enter button and the step is evaluated by the AI technology. If the step is correct, the student proceeds to the next step. If the step is incorrect, the worksheet line the step is on is highlighted and the feedback box explains why the step is wrong and how the step should be corrected. When the student completes the problem by entering the correct answer, the students is notified in the feedback box. There is a hint button that students can use. In this case, the hints are tied to the step that the student has recently completed and gives the student information on how to complete the next step. There are three hints available, each at successive levels of detail. For example, in one of the steps in the problems involving division of complex numbers, the general hint tells the students to multiply by 1. The second hint tells the students to try to eliminate the imaginary part of the denominator.

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

The third hint tells the student to multiply by the complex conjugate of the denominator. The actual numbers from the problem are populated into the hint's text.

The hints and feedback capabilities are made possible through AI and are modeled after the concept of intelligent tutoring systems (cf., Brna, Ohlsson and Pain, 1993; Greer, 1995). The present system has an underlying expert model of the problem-solving process. When a student enters a step, the system matches the student's input to the step that is listed in the expert model. A match is considered to be a correct step and a mismatch is considered to be an incorrect step. The system's expert model for more than one pathway to a solution, which is beneficial since there is generally more than one way to solve a problem.

Typically, people who build AI-based systems for education typically enumerate each problem-solving path that is possible for solving the problem. This is done for each specific problem that the system will deliver (cf., Aleven et al., 2006). This becomes particularly cumbersome if the software will ultimately deliver many problems (as would any large-scale educational system) or if the system is intended to be flexible enough to allow students to enter their own homework or test-study guide problems as our system allows.

Therefore, in order to create a more flexible system that can support any problem within a problem class, we wrote our system to operate on generalized problem types where the numbers used in the underlying expert model the AI engine uses are parameterized rather than instantiated.

For example, a typical system might model a simple solution path for adding (2 + 3i) + (3 + 4i) as

Step 1: (2 + 3i) + (3 + 4i)

Step 2: (2 + 3) + (3i + 4i)
Step 3: 5 + 7i.

This would require a separate model for every possible problem that the system would deliver to a student. By parameterizing each variable, we create a system that requires only one knowledge model per problem type plus the particular variable values for each problem. Therefore, our solution path for the same problem looks like

Step 1: (a + bi) + (c + di)

Step 2: (a + c) + (bi + di)
Step 3: evl(a+c) + evl(b+d)i. (evl means to evaluate the sum of a+c)

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

Problem 1: a=2, b=3, c=3, d=4, and so on for each problem to be used.

This method means that the system can generate unlimited problems to present to the students and the AI technology can respond to them since its representation of the problem is generic rather than hardcoded. For each possible step, there are multiple pathways that are permissible and we supplemented the algorithm with mathematical expression evaluators that recognize equivalent inputs (e.g., a+bi and bi+a are mathematically equivalent).

For each step in the process, the possible errors a student could make are enumerated. For each error, there is associated text that describes the error and the way to correct it. Similarly, three hints, each progressively more specific, are also created for each step in the process. The benefit of our parameterized approach to representing the problems is that these hints and feedback can also be written generically and then populated with specifics from the problem. For example, in a standard algebra problem type of ax+b=c, if a person subtracts the value of b from one side of the equation and not the other, we can write the corrective feedback as "You subtracted b from one side of the equation and not the other. You need to subtract b from both sides of the equation." However, rather than saying "b", the system would replace that b with the actual number used in the problem. This format allows for one general piece of feedback to be used in any problem of this type where the user makes this particular mistake. It is this feature that allows the student to enter his or her own problem since the system in not tied to any particular set of numbers.

Procedure

Of the 40 participants, 20 were assigned to each condition. Prior to instruction, each one was screened by being given a pre-test to ensure that they did not already know how to divide complex numbers. No participant was able to do so.

There were two separate teacher-led classes. Students in the teacher-led condition were divided between the two teachers and taught as a group, as they would in a standard classroom setting. Prior to the class, teachers were told that they would be teaching a one-hour class in dividing complex numbers and that students would be taking a post-test after the instruction. Teachers were told to teach the lesson as they normally would, so that they would not be hampered by having to teach in a format or with a curriculum that they were unfamiliar with. This is ecologically valid as well since different teachers have different teaching styles and use different curricula.

The first part of the instructional process was having participants in each group learn the concept. The teachers then gave practice problems and reviewed them with their students. In the

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

experimental condition, participants used the present software. Their session consisted of learning the material, going through the assessment, and doing six practice problems on the electronic worksheet. Afterwards, all participants were given a 20-question post-test.

Results

Of the 20 students who began the teacher-led instruction, five wound up dropping out before completing the post-test. A parent of one of the students who dropped out sent an email to the lead author of the present paper stating that the student dropped out because he was unable to follow the material being taught. This type of phenomenon occurs in school settings as well, although students may feel freer to drop out of optional classes than they would dropping out of required school classes. We were given no explanations for why the other four students dropped out. However, only the data from the 15 students who completed the post-test in the teacher-led condition are included in the present analyses. We note that all 20 students who were in the software-led instruction condition completed the instruction and the post-test.

The answers to the 20 questions on the post-test were scored based on whether the correct answer was given. The mean number of correct answers given by students who were taught by the first teacher is 13.14. The mean number of correct answers given by students who were taught by the second teacher is 12.75. This difference is not statistically significant. Accordingly, the data from the two teachers were combined into a single group for purposes of analysis, yielding a mean number of correct answers for students taught by teachers to be 12.93 or 64.65%. The mean number of correct answers for students taught by the software was 17.75 or 88.75%.This difference between software condition mean score and teacher condition mean score was statistically significant, $t = 2.67$, $df = 33$, $p = .015$.

Although students who used the software scored, on average, 37% higher than those who were taught by teachers, it was not the case that there was an across-the-board improvement for students using the software. A closer examination of individual scores reveals that test scores of students in the present study mirrored those one might expect in a regular classroom, namely some students excelled, some were average and some struggled. Of the 15 students who were taught by teachers, five scored 90% or higher on the post-test, which is equivalent to being in the A-grade range in many schools. Another five scored between 70% and 85%, which places them in the B to C-grade range, which is considered average performance. The remaining five students scored below 50%, which is considered failing. These results roughly mirror the 2019 National Assessment of Educational Progress 12th grade math findings that showed that 40% of students performed below basic level, 35% performed at the basic level and 24% performed at

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

the proficient or advanced level (21% were proficient, which is considered grade level, and 3% were advanced).

In the software group, the post-test scores ranged from 14 to 20 (70% to 100%). Of the 20 students who used the software, 12 scored 90% or higher (A-grade range),six had percent scores in the 80s (B-grade range) and two had percent scores in the 70s (C-grade range). This suggests that, while some students did outperform others, in general, all students performed reasonably well. The implications of these results suggests that the distribution of scores for students taught by teachers was rectangular in nature while the distribution of scores taught by the software was skewed, favoring scores at the higher end of the scale.

This suggests that the software reduced variability of scores as well as shifting them higher. In order to test this difference in variability, we conducted a Levene's Test for Homogeneity of Scores Variance across the two groups. The results was statistically significant, $F(1, 33) = 34.79$, $p < .00001$, indicating that the students using the software showed more consistent performance than did those taught by teachers.

Discussion

The primary purpose of the research was to compare the effectiveness of our AI-based educational software to that of human teachers. Two noteworthy results occurred. First, students taught using our software outperformed those taught by human teachers on a post-test by an average of 37%. Second, there was significantly less variability in the post-test scores of students who used the software than of those who were taught by teachers. This latter result occurred because most students who used the software scored 90% or higher on the post-test and none scored below 70%, whereas the distribution of test scores for students taught by teachers was rectangular with a full one-third of them failing. It may also be noteworthy that five students in the teacher-led groups dropped out, and, while we do not know the reasons for all of the dropouts, one parent did explicitly state that her child dropped out because he could not understand the material. There were no dropouts in the software condition.

While we are not advocating that software replace teachers in schools, the collective results of the present study offer some suggestions as to how software such as ours can be used in conjunction with classroom education. As noted in the Introduction, the majority of students in the United States perform below grade level. Moreover, the critical shortage of certified teachers makes it unlikely that teachers themselves can remedy this problem in the foreseeable future. Absent an effective intervention, we are likely to see a continuation of current results: some students do well, some perform in the average range and some perform at a failing level.

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

Using AI-powered software such as ours may prove to be the needed remedy. As noted earlier, there is an increase in self-directed learning. While our previous research suggested that gifted and talented students could self-learn basic computer programming while average students did better with a teacher, the present results suggest that AI-powered educational software may be able to stand in for a human teacher. We note that while there were low-performing students in the teacher-led classes, there were no low-performing students in the AI-led instruction. This suggests that our AI-powered technology can work for all students, not just the gifted and talented ones. If this is the case, the suggestion is that families and schools can use software such as ours to allow more students, particularly low-performing students, to engaged in self-directed learning as an alternative or supplement to teacher-directed learning and the students will not only not suffer as a result of being taught by software, on average, they will do better.

References

Anderson, J.R. (1982). Acquisition of cognitive skill. Psychological Review, 89, 369-405.

Aleven, V., McLaren, B.M., Sewall, J., and Koedinger, K.R. (2006). The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. Human-Computer Interaction Institute. Pittsburgh, PA: Carnegie Mellon University.

Brna, P., Ohlsson, S. and Pain, H. (1993) (Eds.) Artificial Intelligence in Education. Association for the Advancement of Computers in Education, Charlottesville, USA.

de Kleer, J. and Brown, J.S. (1981). Mental models of physical mechanisms and their acquisition. In J.R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Erlbaum.

Garcia, E. and Weiss, E. (2020). A Policy Agenda to Address the Teacher Shortage in U.S. Public Schools. Washington, DC: Economic Policy Institute

Greer, J. (1995) (Ed.) Proceeding of Artificial Intelligent in Education '95. Charlottesville, VA: Association for the Advancement of Computing in Education.

Leddo, J., Bisht, D., Narla, E., Saranu, R. and Titov, M. (2016). Using Artificial Intelligence to Enhance the Effectiveness of Multimedia-based Instruction. International Journal of Advanced Education and Research, 1(12), 30-36.

Leddo, J., Boddu, B., Krishnamurthy, S., Yuan, K. and Chippala, S. (2017). The Effectiveness of Self-directed Learning vs. Teacher-led Learning on Gifted and Talented vs. Non-gifted and Talented Students. International Journal of Advanced Educational Research, 2(6), 18-21.

ISSN: 2455-8834

Volume:06, Issue:03 "March 2021"

Leddo, J., Guo, Y., Liang, Y., Joshi, R., Liang, I., Guo, W. and Bailey, S. (2019). Artificial Intelligence and Voice-powered Electronic Textbooks. International Journal of Advanced Educational Research, 4(6), 44-49.

Leddo, J. (1994). Did they learn anything?: Finding out with a knowledge assessment tool. Presented at the National School Board Association Technology and Learning Conference. October, 1994.

National Assessment of Educational Progress. (2019). Princeton, NJ: Educational Testing Service.

Quillian, M.R. (1966). Semantic memory. Camridge, MA: Bolt, Beranek and Newman.

Schank, R.C. and Abelson, R.P. (1977). Scripts, Plans, Goals, and Understanding. Hillsdale, NJ: Erlbaum.

