Using Intelligent Texts in a Computer Science Classroom: Findings from an iTELL Deployment

Scott Crossley[1], Joon Suh Choi[1], Wesley Morris[1], Langdon Holmes[1], David Joyner[2] and Vaibhav Gupta[3]

[1] *Peabody College, Vanderbilt University, Nashville, Tennessee, USA*
[2] *College of Computing, Georgia Institute of Tech, Atlanta, Georgia, USA*
[3] *Data Science Institute, Vanderbilt University, Nashville, Tennessee, USA*

Abstract

This study assesses the efficacy of intelligent texts to help students in a computer science class understand and process information about computational thinking and programming. The intelligent texts used in this study were taken from an introductory programming textbook. The texts were ingested into an intelligent text format using the Intelligent Texts for Enhanced Language Learning (iTELL) framework, which converts any type of machine-readable text into an interactive, intelligent text. iTELL asks students to complete constructed response items and summaries, which are scored automatically by large language models (LLMs) specifically trained to generate scores to inform qualitative feedback to students. Survey results indicated that students responded positively to the constructed response and summary items and felt both items helped them learn. An analysis of delta value gain scores between pre-tests and post-tests for students that used iTELL and those that did not use iTELL indicated that iTELL students showed increased learning gains. Regression analyses showed that delta scores for the iTELL students were predicted by the number of scrolls, word scores on summaries, and pre-test proficiency level (low/high). The results indicate that intelligent texts may help computer science students better learn material than traditional texts.

Keywords

Intelligent texts, natural language processing, reading assessment, computer science

1. Introduction

Computational thinking is a critical 21st century skill that can help students navigate an increasingly digital world. It describes the ability to express a problem in terms of steps, such that they could be written out as an algorithm. Teaching computational thinking allows students to explore knowledge in concrete ways, and asking students to code computational thinking into a computer program can provide students with a quick method to check the validity of the knowledge [1].

However, coding is a complex skill that requires sustained effort, a specialized approach, and a diverse skill set. Developing these skills is an iterative process that requires persistence and knowledge well beyond simple syntax [2]. Becoming a proficient programmer requires a combination of various abilities, and merely knowing programming syntax is just the initial step in the challenging process of creating effective programs. The complexity of computer programming and the dedication required to succeed as a programmer means that computer science classes suffer from high failure and dropout rates [3] which has led the computer science community, especially those interested in education, to develop numerous tools and supports to help facilitate student success. The majority of these tools focus on assessing the correctness of assignments in object-oriented programming languages. Typically, these tools use dynamic techniques to provide grades and feedback to students. Some tools use static analysis techniques to compare a student's submission with a reference solution or a set of correct student submissions [4].

Students in computer science classes are also expected to learn about computational thinking and programming approaches, using reading materials. However, printed books are generally considered ineffective at teaching computational thinking and the dynamic nature of programming because they are bound by the static confines of the text [5]-[6]. Specifically, studies indicate that students may fail to comprehend programming

CSEDM'24: 8th Educational Data Mining in Computer Science Education Workshop, July 14, 2024, Atlanta, GA
scott.crossley@vanderbilt.edu (S. Crossley); joon.suh.choi@vanderbilt.edu (J. S. Choi); wesley.morris@vanderbilt.edu (W. Morris); langd.holmes@vanderbilt.edu (L. Holmes); david.joyner@gatech.edu (D. Joyner); vaibhav.gupta@vanderbilt.edu (V. Gupta);

dynamics when explained through static pedagogical materials [7].

The goal of this study is to assess the efficacy of interactive intelligent texts in a computer science class to help students understand and process information about computational thinking and programming. The intelligent texts used in this study are taken from an introduction to computing textbook. They were ingested into an intelligent text format using the Intelligent Texts for Enhanced Language Learning (iTELL) framework. iTELL is a computational framework that converts any type of machine-readable text into an interactive, intelligent text within a web-app. iTELL is based on theories of reading comprehension and provides opportunities for users to generate knowledge about what they read and watch using constructed responses and summary writing. The constructed responses and summaries are scored automatically by large language models (LLMs) specifically trained to generate scores which inform qualitative feedback to students. The feedback from these AI integrations is used in a number of different ways, including to guide learning, correct misconceptions, review missed topics, prepare for upcoming materials, make links between the texts and the real world, and help elaborate on what users have learned. iTELL represents an advanced learning technology based on AI which can be used to improve student learning outcomes in computer science classes.

1.1. Intelligent Texts

Intelligent textbooks have become more popular as advancements in Natural Language Processing (NLP) have made human-machine interaction more accessible [8]. Early digital textbooks offer several advantages over traditional print textbooks, such as videos and hyperlinks, but studies found no significant difference in learning outcomes between digital and print textbooks [9]. However, a recent meta-analysis indicates that the interactive features of intelligent textbooks can moderately improve reading performance [10]. Moreover, college students tend to prefer digital textbooks due to their lower cost and ease of use.

Digital textbooks have been in production for over 30 years with initial texts using principles of knowledge engineering wherein domain experts designed and produced the textbooks [11]. Early work included the development of hypertext technology, which allowed students to navigate the textbooks more efficiently [12]. One of the first web-based interactive textbooks was ELM-ART, an intelligent and interactive textbook introduced in 1996 to teach computer programming [13].

Research on intelligent textbooks has increased in the past decade as computational tools have become more advanced and accessible [8]. Studies have focused on analyzing student behaviors in intelligent textbooks to provide personalized learning experiences. For instance, researchers have developed algorithms that use previous assessment results to recommend optimal learning activities for each student in textbooks [14]. Student behaviors, such as struggling to answer comprehension questions, can also be used to adaptively modify intelligent textbook content and provide remedial materials [15]. Furthermore, experts can extract concepts from intelligent texts that can train machine learning algorithms to personalize learning [16] or part-of-speech taggers can be used to automatically generate comprehension questions [17].

The advent of large language models (LLMs) will further change the developmental landscape for intelligent textbooks. LLMs allow intelligent textbooks to be more interactive and afford more accurate, real-time feedback on student generated responses used to assess text understanding. Additionally, LLMs allow for the integration of AI chatbots that can guide readers through the text and potentially help with any misunderstandings or difficulties the readers may encounter. In short, LLMs enable the creation of intelligent textbooks that can automatically generate content as well as prompt and evaluate reader responses allowing for the automation of scoring and feedback generation.

1.2. iTELL

iTELL is a framework that simplifies the creation and deployment of intelligent texts with integrated interactive features. iTELL includes automated pipelines that leverage Large Language Models (LLMs) with human oversight to generate participatory content like constructed response items and summaries. Additionally, it includes scoring APIs for constructed responses and summaries. iTELL is a domain-agnostic framework that utilizes multiple highly transferable generative LLMs to transform static texts into interactive, intelligent textbooks.

iTELL generates rich clickstream data, allowing for the analysis of user behaviors, particularly related to reading. It uses JavaScript's intersection observer API to determine whether a specific text section is within the user's viewport and logs the observation time for different parts of the text. iTELL also log events within the systems include scrolling and page clicks.

Most importantly, iTELL includes read-to-write tasks that engage the reader in learning. Read-to-write tasks require readers to extract and integrate information from the text into their writing allowing them to construct knowledge as they read [18]-[21]. Read-to-write task have been shown to be effective learning tools. For example, asking readers to summarize what they have written results in strong learning gains [22]-[23]. Additionally, constructed responses, where students provide short written answers, can improve learning comprehension [24]. In iTELL, readers are required to complete at least one constructed response item and write one summary per page. iTELL utilizes multiple fine-tuned and out-of-the-box Large Language Models (LLMs) to support these tasks. Specifically, iTELL uses LLMs to generate short questions and to evaluate readers' constructed responses to those questions. Additionally, iTELL requires readers to submit a summary of each page and uses LLMs to score those summaries and provide feedback to readers to use when revising summaries.

1.3. Current Study

The current study examines an iTELL volume deployed in an Introduction to Computing class. A single volume of iTELL was developed that covered a textbook chapter on control structures (Chapter 3). Students within the

classroom were given extra credit to use the iTELL volume of which about 25% did. The remaining students depended on a static, digital version of the textbook. At the end of each chapter, the students were given a test to assess their knowledge. The research questions that guide this study are the following:

1. Do students think that iTELL is easy to interact with, is understandable, helps them learn, and provides accurate feedback?
2. Do students that completed the iTELL volume show gains from the test on chapter two (no iTELL volume) to the test on chapter 3 (iTELL volume) compared to students who did not complete the iTELL volume?
3. Are data points collected from the iTELL volume related to click-stream data, focus time, and summary scores related to differences in test scores from chapter two to chapter three?

2. Method

2.1. Course

Data for this research is based on an Introduction to Computing class that was taught in the spring of 2024 at a large technology university in the southeastern United States. The course is one of three courses that can fulfill the computer science requirement for all students at the university and is taken by over a couple of thousand students every academic year in one of two variations: in-person and online. The course covers the basics of computing, presupposing no prior programming ability: it begins with the basics of procedural programming, moves through control structures and data structures, and concludes with brief units on object-oriented programming and algorithms. Throughout the course, students complete several hundred small programming problems through the homework assignments, as well as some live coding problems during four timed, proctored tests. Tests and quizzes comprise 52% of students' grades in the class.

2.2. Textbook

The textbook used in the course is called Introduction to Computing, first edition [25]. The textbook is published by McGraw Hill Education and is available in digital format. There are five units in the textbook with each unit comprising between 2 and 5 chapters. The five units are Computing, Procedural Programming, Control Structures, Data Structures, and Object-Oriented Programming.

For this study, we ingested the third unit covering Control Structures into an iTELL volume using iTELL's content management system. The volume comprised an overview page that introduced iTELL and 5 additional pages with each page referencing a chapter from the unit. Each page followed the structure of the chapter in the book including learning objectives, key terms, the chapter prose, and any figures, graphs or tables. However, screenshots of integrated development environments (IDEs) in the textbook that demonstrated Python code and the code output were replaced with a Python interactive sandbox. The sandbox allowed students to enter in their own code and run the code within iTELL. The pages were separated into chunks (i.e., all content under a unique sub-heading) based on related content as selected by the page designer. On average, each page had around 6.6 chunks (SD=1.14). Only the first chunk on a page was visible to a user at the beginning with all subsequent chunks being blurred. Users were required to click on a "chunk reveal" button to unblur the next chunk.

The content management system automatically generates constructed response questions and answers for each chunk with human-in-the-loop. Prior to publishing, the page designer ensured that the questions and answers were accurate. Each chunk had an accompanying constructed response item. There was a 1/3 chance of a constructed response item being presented to a user for each chunk.

2.3. Participants and Procedure

There was a total of 476 students enrolled in the class. Of those enrolled, 121 students elected to use the iTELL version of the textbook and 356 did not. Students were given 1% extra credit (added to their overall course grade) for participating in the study.

These 121 students first provided consent for their data to be used. If the student did not provide consent or was under the age of 18, they were sent directly to the iTELL volume, but no data was collected. If they provided consent, they then completed an intake survey that collected demographic information and individual difference data including age, sex, race/ethnicity, first language, and reading habits and technology use. They were then sent to the iTELL volume. If the student finished the five pages in the volume, participants were asked to complete an outtake survey to describe their experience of working with the iTELL volume. The outtake survey focused on students' perceptions of the digital text's layout, organization, annotation features, and the effectiveness of the summary and short answer tasks.

Of these 121 students, 101 consented to having their data used for analysis. Of those 101 students, 82 completed iTELL including the outtake survey. However, of the 82 students that completed iTELL, 79 reported test scores for units 2 and 3. Of the 356 students that did not use iTELL, 277 reported test scores for units 2 and 3.

2.4. Surveys

2.4.1. Intake Survey

Before interacting with iTELL, students completed a short intake survey to collect demographic data such as age, gender, race or ethnicity, and first language background. In this intake survey, students were also asked to provide information about their interactions with technology and they provided input about their reading habits on electronic and traditional texts. Finally, students provided information about the types of features that they have used before in intelligent texts.

2.4.2. Outtake Survey

Upon completion of the intelligent texts, users completed an outtake survey which included user feedback questions on each feature of iTELL including annotation and

notetaking, the section summary tasks, constructed response items, and overall feedback about the layout and organization of the intelligent text. This survey allowed us to collect data on users' perceptions of how well each of the features stayed relevant to the text, worked correctly, was easy to interact with, and helped improve the users' learning. Users were also prompted to provide short text feedback about each of the features.

2.5. iTELL Data Extraction

For this analysis, we extracted data related to participant focus time, click-stream events, constructed responses, and summaries.

2.5.1. *Focus Time*

For each page focus time was extracted in two different ways. First, focus time was recorded bey subtracting the time that users opened the page and the time the user moved onto the next page. The focus time included all the time spent on constructed responses and on summary scoring. Second, focus time was recorded by how long each chunk in a page was viewed. From this, a total time for all chunks per page and an average time for all the chunks on a page was derived.

2.5.2. Events

A number of events are calculated in iTELL that can be instrumented into predictive variables. These include chunk reveal events (for chunks with and without constructed response items), general clicks on items within the system, periods of time when learners are focusing on the page, and when scrolling. The chunk reveals without constructed responses is not the inverse of the chunk reveals after constructed responses because many students reread chapters either as a choice or as a function of needing to reset the chapter because of a bug in the iTELL system. Around 40% of chapters had some type of rereading (i.e., scrolling upwards of more than 3% of the page content) on the part of students.

2.5.3. Constructed Responses

As part of the iTELL integration, an accompanying constructed response item is generated for each chunk using GPT-3.5-turbo with human-in-the-loop. End users do not see all constructed response items when reading an iTELL volume; instead, each chunk has a 1/3 chance of spawning an accompanying constructed response item, with a minimum of one constructed response item per page. Users are required to submit at least one response to a spawned item before proceeding to the next chunk. Readers' constructed responses are scored for correctness using two separate fine-tuned LLMs, Bilingual Evaluation Under-study with Representations from Transformers (BLEURT) [26] and Masked and Permutated Language Modeling (MPNet) [27], both of which report an accuracy of ~ .80 [28] on question/answer pairs in the Multi-Sentence Reading Comprehension (MultiRC) dataset [29]. The same BLEURT and MPNet models are used to provide feedback to readers who are given the opportunity to revise their constructed responses if needed.

For each participant, we calculated the number of constructed responses they produced and the average score they received for the constructed response on a scale of 1-3 with a 1 representing when the two LLMs agreed the answer was incorrect, a 2 representing when one of the LLMs classified the answer as incorrect and the other classified as correct, and a 3 representing when the two LLMs agreed the answer was correct. Figure 1 demonstrates the interface and the feedback returned for a constructed response scored a 2 by the model. Because of a bug in the code connecting iTELL to its database, a number of constructed responses were not logged and were omitted at random. Of the 395 pages completed by the 79 iTELL participants, 130 of those pages did not have constructed response data logged.

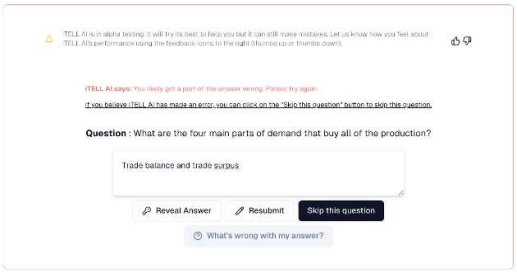

Figure 1: Constructed Response Interface.

2.5.4. Summaries

After reading each page, students were prompted to write a summary of what they read. Algorithmic filters in iTELL ensure that the summaries are between 50 and 200 words long, are written in English, do not include inappropriate language. The iTELL interface does not allow copying and pasting directly from the text. When a student submits a summary, they receive a score on Language Borrowing by calculating the proportion of overlapping trigrams between the summary and the source [30]. They also receive a score on Relevance using cosine similarity between the text embedding of the summary and the text embedding of the source. If they pass these tests, they are scored by two encoder LLMs introduced by Morris et al. [31-[32] on Content (i.e. does the summary reproduce the content of the source) and Wording (i.e. does the summary use correct grammar/syntax and paraphrasing).

These models, based on the Longformer pretrained model [33], were finetuned on a large dataset of sources and summaries that were scored on a six-criteria analytic rubric by expert raters. The six criteria were distilled into two principal components [32]. The Content PCA score includes how well the summary reproduced the main idea and details of the text, how well the summary was organized, and how well the summary used objective voice. The Wording PCA score includes grammar/syntax and how well the summary paraphrased the source using original language. The score predictions are normalized so that 0 represents the mean of the scores in the original training set, with a standard distribution of 1. In a held-out test set of sources that the models had not encountered during training, they reported R^2 values of 0.82 for Content and 0.7 for Wording [32]. An example of the summary interface is provided in Figure 2 for a summary that passed all scoring metrics.

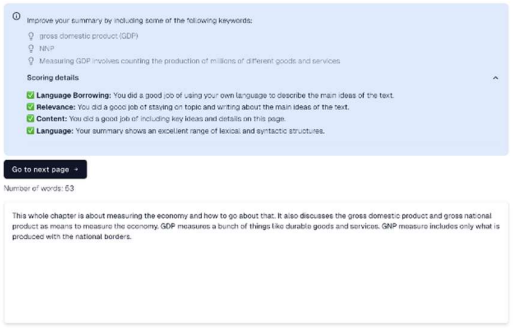

Figure 2: Summary Interface.

For each participant, we calculated the number of summaries they produced and the average score they received on each summary for Content, Wording, Language Borrowing, and Relevance.

2.5.5. Python IDE

The iTELL volume of the *Introduction to Computing* textbook included an integrated development environment (IDE) sandbox that exhibited Python code and allowed students to enter and run their own code. However, logging features for the sandbox data were not implemented at the time of data collection.

2.6. Analyses

We ran three different analyses to address our research questions. For the first research question related to whether users think iTELL is easy to interact with, is understandable, helps them learn, and provides accurate feedback, we ran simple descriptive statistics and graphed out the results. For the second research question related to whether students who completed the iTELL volume show gains from chapter 2 to chapter 3 test scores compared to students who did not use the iTELL, we ran statistical tests to assess differences between the two groups' delta values. Our main statistical metrics are a p value to indicate if an effect exists and a Cliff's Delta value to examine the strength of the effect. For the third research question related to whether data points collected from the iTELL volume related to click-stream data, focus time, and summary scores are predictive of delta values, we conducted a stepwise linear regression using iTELL data as predictors of the delta values. We included a categorical performance variable on the chapter 2 test based on whether students scored above the mean (high) or below the mean (low). This variable was included as a predictor and as a possible interaction to see if there was an effect of iTELL on lower or higher-performing students.

3. Results

3.1. User Survey Data

For the user survey data, we were most interested in student responses to the summarization task, the constructed response items, and their overall satisfaction with iTELL. To provide a simpler representation for survey item visualizations, we combined scores of 4 and 5 into a single category (agree) and all scores of 1 and 2 into a single category of disagree. Scores of 3 were labeled neutral. We conducted follow up ANOVAs to examine if any differences were noted across survey responses by ethnicity or reading frequency.

For the summary task, the mean responses were generally positive ($M > 4$). The lowest responses were for the accuracy of feedback ($M = 4.18$) while the highest responses were for ease of understanding ($M = 4.29$). Students felt that the summary tasks helped them learn ($M = 4.18$). There were no significant differences noted across survey items by race or ethnicity or by reading frequency. Data for this analysis are presented in Figure 3.

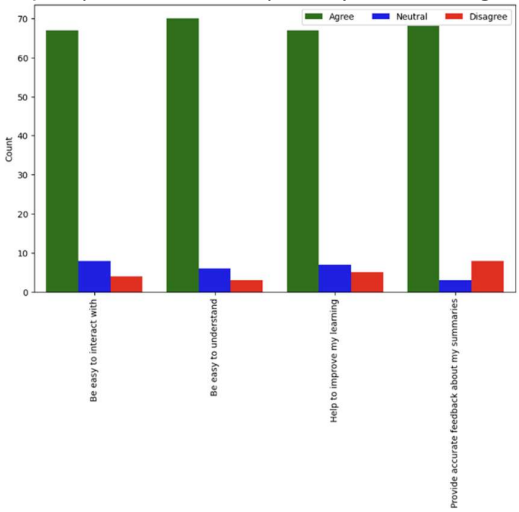

Figure 3: Summary Task Survey Results.

For the constructed response task, the mean responses were generally positive ($M > 4$). The lowest responses were for the accuracy of feedback ($M = 4.01$) while the highest responses were for ease of interaction ($M = 4.32$). Students felt that the summary tasks helped them learn ($M = 4.20$). There were no significant differences noted across survey items by race or ethnicity or by reading frequency. Data for this analysis are presented in Figure 4.

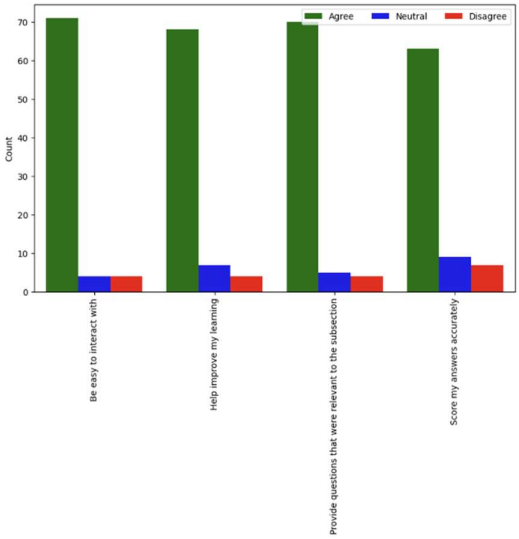

Figure 4: Constructed Responses Survey Results.

For the overall feedback response (Overall, how satisfied were you with this digital text?), the mean responses were generally positive ($M > 4.14$). Additionally, there were no significant differences noted across survey

items by race or ethnicity or by reading frequency. Data for this analysis are presented in Figure 5.

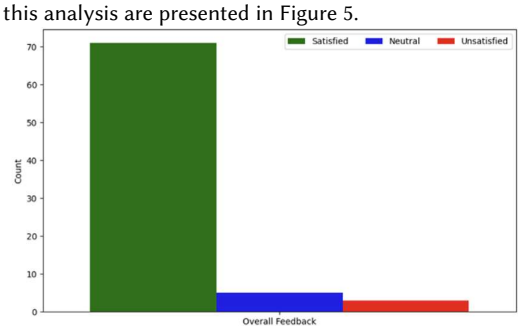

Figure 5: Overall Feedback Survey Results.

3.2. Test Score Differences

For our test score difference analysis, we examined the Delta score between test 2 and test 3 for students who used iTELL and those that did not. Descriptive statistics indicated greater gains in learning for test 2 and 3 by the students who used iTELL (M = .032, SD = .304 than the non-iTELL students (M = -.018, SD = .248). Visual examinations of the data indicated that it was not normally distributed (see Figure 6 histogram). Thus, we conducted a Mann-Whitney U test on delta scores across conditions (iTELL vs. non-iTELL). The differences approached significance (p=0.067; U=12,386) indicating that an effect size is likely. The Cliff's Delta = .132, 95% CI [-0.029, 0.286]) reported a small effect suggesting that students in the iTELL condition scored higher on test 3 (after using iTELL) than they did on test 2 (no iTELL use).

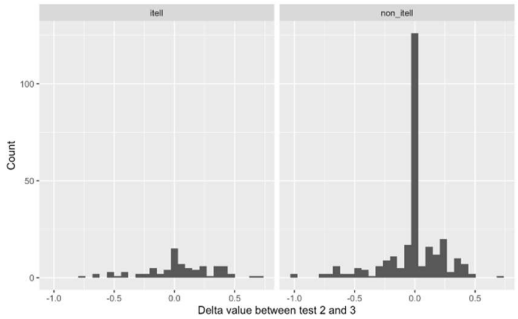

Figure 6: Histogram for delta values between tests

Upon inspection of the data, we noticed that almost every single student who reported delta values of 0 did so because they received 100% on the chapter 2 test and 100% on the chapter 3 test. For the entire class, 36% of the students showed a delta value of zero. For the non-iTELL students, 42% showed a delta value of zero while 18% of the iTELL students showed a delta value of zero. Presuming the students showing delta values of zero were at ceiling, we conducted a post-hoc analysis where we removed these students.

Removing these students dropped the sample size to 227, of which 162 were non-iTELL students, and 65 were iTELL students. Visual examination of the delta scores indicated a normal distribution. An independent samples t-test between the iTELL students (M = .039, SD = .336) and the non-iTELL students (M = -.030, SD = .324) showed no statistical difference, t(114.52) = 1.426, p = .157. However, a small effect size was reported (Cohen's d = .213, 95% CI [-0.077, 0.503]), suggesting that students in the iTELL condition may have improved their test scores after using iTELL more than students who did not use iTELL

3.3. Delta Value Predictions

Table 1
Linear model to predict delta scores (test 2 and test 3)

Variable	Estimate	SE	*t*
(Intercept)	-0.085	0.047	-1.852
Number scrolls	-0.099	0.032	-3.111**
Wording score	-0.077	0.031	-2.499*
Testing level (low)	0.216	0.064	3.382**

* $p < .05$, ** $p < .010$

Three variables were significant predictors in our regression model: number of scrolls, Wording scores on summaries, and the categorical variable related to whether the student scored high or low on test 2. The linear model reported r = .501, R^2 = .251, F (3, 75) = 8.362, p < .001 (see model parameters summarized in Table 1). The coefficients indicated that higher delta scores between test 2 and 3 were predicted by fewer scrolls, lower Wording scores, and performance on test 2 with lower performers on test 2 that used iTELL showing greater gains between test 2 and 3.

4. Discussion and conclusion

In the modern economy, computing skills are increasingly important for students to acquire effectively. Understanding computational thinking and computer programming enables students to solve important, real-world problems in a number of complex ways across a number of domains. However, learning computer skills is difficult and requires sustained efforts, specialized teaching environments, and diverse skills, making success difficult. As a result, many computer science curricula have suffered from high failure and dropout rates [3]. While many supports have been developed to help students succeed, there is some consensus that traditional textbooks are less than effective for acquiring computing skills because computing is a dynamic process that is not captured well in static texts [7]. The goal of this study was to assess the efficacy of interactive intelligent texts in a computer science class to help students understand and process information about computational thinking and programming. Specifically, we assessed an iTELL volume of an introduction to computing textbook that focused on a section related to control structures.

Our assessment of the iTELL volume was an A/B test where about 25% of the class volunteered to use the iTELL volume (for extra credit) and the remainder of the class used a plain digital version of the text. We used test scores from the previous chapter where all students used the digital textbook as a baseline measure of student skills. We then examined students' survey data to better understand their experiences with the iTELL volume. Additionally, we compared delta scores calculated as the difference between the scores on the control structures test and scores on the baseline test to assess potential gains for students who used iTELL. Lastly, we ran a regression model to better understand what features explained gains by students who used the iTELL volume.

The survey results indicated that students' experiences with the AI tools within iTELL were positive. Overall, students felt that the constructed response items and summary tasks were easy to work with and helped them improve their learning. Students also felt the AI feedback was accurate. Student surveys also indicated that the students were satisfied with the iTELL volume overall.

In terms of learning differences between the iTELL and non-iTELL students, a Mann-Whitney U test approached significance and reported a meaningful, but small, relationship between score differences between the baseline test and the test on the control structures chapter (see reported effect sizes). A number of students showed ceiling effects across both tests and removing these students led to similar results. While a *p* value can indicate whether an effect exists, the Cliff's Delta size shows the magnitude of the differences between the iTELL and non-iTELL groups (a small but meaningful effect) and is the main quantitative consideration of the study [34]. The mean score differences indicated that students that used iTELL showed gains of ~5% versus the students that did not use iTELL and the Cliff's Delta indicated that this difference was meaningful. The standard deviation was quite high for both groups, though.

The regression analysis of the iTELL student data indicated that students who showed a greater number of scrolls showed lower delta scores. This may indicate that students who are non-linear readers or students who scroll in smaller increments performed worse. However, much deeper analysis of scrolling needs to be performed to support any meaningful conclusions. Additionally, the regression model indicated that students who scored higher in Wording for their summaries showed lower delta score gains. This may indicate that students who focused on Wording in their summaries at the expense of content may have performed worse. It may also be that students that are better writers gain less from using iTELL then students who are worse writers. This may be supported by the final feature of the regression model which was testing level. The coefficients for testing level indicated that students that performed lower on the baseline test showed greater gains when using iTELL. This indicates that iTELL may work better for low level students, but, again, much more fine-grained testing is needed to support this notion.

Overall, this study finds evidence that intelligent textbooks are an advanced learning technology that can use AI to improve student learning outcomes in an introduction to computing class. This finding builds on previous studies that have argued that traditional, static textbooks may be ineffective in computer science instruction [7]. However, intelligent texts that are interactive and allow for dynamic assessments and greater student engagement may be effective, especially intelligent texts that integrate read-to-write tasks known to lead to increased learning gains [22-24] based on generation effects [35].

While the learning gains are small (see reported effect sizes), it is unlikely that a single text-based intervention would lead to stronger gains in a computer science class. However, in combination with other AI powered tools help computer science students effectively use debuggers and compilers or guide students through complex, multi-step, open-ended problems, intelligent texts may lead to greater gains.

There were a number of limitations to the analyses conducted. First, we were looking at a convenience sample and one in which students were rewarded for using iTELL, so there is likely a self-selection bias [36]. While it is difficult to know the type of students that volunteered for the iTELL condition, Figure 4 indicates that many more students that were at ceiling on scores for Tests 2 and 3 did not volunteer to use iTELL. So, it is likely that students who needed extra credit volunteered versus students who are highly motivated in general. Regardless, a randomized control trial is needed to truly assess iTELL effects in the computer science classroom. There were also problems in iTELL with data documentation. A bug in the constructed response items led to data for over half of the items not being logged. Additionally, no data from the Python sandbox was logged. Thus, it is difficult to disaggregate the effects of these tools on learning in the iTELL environment. We also had a relatively small sample size for the iTELL condition compared to the non-iTELL condition, which makes it difficult to generalize the findings to different populations. Additionally, the standard deviation in scores was quite high for both groups indicating much variation in learning gains. Lastly, the students in the class all come from a technical background (i.e., they are studying at a technical university), which may have affected the outcomes.

Overall, though, the study provides a promising first step in understanding how the use of intelligent texts may lead to learning gains for computer science students. Knowing the importance of computer science skills in the modern economy and the difficulty in obtaining those skills, a variety of educational tools will be needed to address potential learning deficits in the computer science classroom. Intelligent texts may prove to be one of those tools.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant 2112532. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Webb, M., Bell, T., Davis, N., Katz, Y. J., Reynolds, N., Chambers, D. P., ... & Mori, N. (2017). Computer science in the school curriculum: Issues and challenges. In Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing: 11th IFIP TC 3 World Conference on Computers in Education, WCCE 2017, Dublin, Ireland, July 3-6, 2017, *Revised Selected Papers 11* (pp. 421-431). Springer International Publishing.

[2] Jiau, H. C., Chen, J. C., & Ssu, K.-F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. *IEEE Transactions on Education, 52*(4), 555-562. https://doi.org/10.1109/TE.2008.2010983

[3] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Com-puter Science Education, 13*(2), 137-172. https://doi.org/10.1076/ csed.13.2.137.14200

[4] Messer, M., Brown, N. C., Kölling, M., & Shi, M. (2024). Automated grading and feedback tools for programming edu-cation: A systematic review. *ACM Transactions on Computing Education, 24*(1), 1-43.

[5] Bennedsen, J., & Caspersen, M. E. (2005). Revealing the programming process. Paper presented at the *ACM SIGCSE Bulletin*. https://doi.org/10.1145/1047344.1047413

[6] Zhang, X., Zhang, C., Stafford, T. F., & Zhang, P. (2019). Teaching introductory programming to IS students: The im-pact of teaching approaches on learning performance. *Journal of Information Systems Education, 24*(2), 6. Retrieved from http://jise.org/Volume24/n2/JISEv24n2p147.pdf

[7] Gomes, A., & Mendes, A. J. (2007, September). Learning to program-difficulties and solutions. In *International Con-ference on Engineering Education–ICEE* (Vol. 7).

[8] Sosnovsky, S., Brusilovsky, P., & Lan, A. (2022, July). Intelligent textbooks: themes and topics. In International Conference on Artificial Intelligence in Education (pp. 111-114). Cham: Springer International Publishing.

[9] Rockinson-Szapkiw, A. J., Courduff, J., Carter, K., & Bennett, D. (2013). Electronic versus traditional print textbooks: A comparison study on the influence of university students' learning. *Computers & Education, 63*, 259-266.

[10] Clinton-Lisell, V., Seipel, B., Gilpin, S., & Litzinger, C. (2023). Interactive features of e-texts' effects on learning: A systematic review and meta-analysis. *Interactive Learning Environments, 31*(6), 3728-3743.

[11] Brusilovsky, P., Sosnovsky, S., & Thaker, K. (2022). The return of intelligent textbooks. *AI Magazine, 43*(3), 337-340.

[12] Bareiss, R., & Osgood, R. (1993, December). Applying AI models to the design of exploratory hypermedia systems. In *Proceedings of the fifth ACM conference on Hypertext* (pp. 94-105).

[13] Weber, G., & Brusilovsky, P. (2016). Elm-art–an interactive and intelligent web-based electronic textbook. *International Journal of Artificial Intelligence in Education, 26*, 72-81.

[14] Lan, A. S., & Baraniuk, R. G. (2016, June). A Contextual Bandits Framework for Personalized Learning Action Selection. In *Educational Data Mining* (pp. 424-429).

[15] Thaker, K., Zhang, L., He, D., & Brusilovsky, P. (2020). Recommending Remedial Readings Using Student Knowledge State. *International Educational Data Mining Society*.

[16] Wang, Z., Lamb, A., Saveliev, E., Cameron, P., Zaykov, J., Hernandez-Lobato, J. M., ... & Zhang, C. (2021, August). Results and insights from diagnostic questions: The neurips 2020 education challenge. In *NeurIPS 2020 Competition and Demonstration Track* (pp. 191-205). PMLR.

[17] Kumar, G., Banchs, R. E., & D'Haro, L. F. (2015, June). Revup: Automatic gap-fill question generation from educational texts. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 154-161).

[18] Delaney, Y. A. (2008). Investigating the reading-to-write construct. *Journal of English for academic purposes, 7*(3), 140-150.

[19] Grabe, W., & Stoller, F. L. (2019). *Teaching and researching reading*. Routledge.

[20] Nelson, N., & Calfee, R. C. (1998). Chapter I: The Reading-Writing Connection Viewed Historically. *Teachers College Record, 99*(6), 1-52.

[21] Nelson, N., & King, J. R. (2023). Discourse synthesis: Textual transformations in writing from sources. *Reading and Writing, 36*(4), 769-808.

[22] Silva, A. M., & Limongi, R. (2019). Writing to learn increases long-term memory consolidation: A mental-chronometry and computational-modeling study of "Epistemic writing". *Journal of Writing Research, 11*(1), 211-243.

[23] Brown, A. L., Campione, J. C., & Day, J. D. (1981). Learning to learn: On training students to learn from texts. *Educational researcher, 10*(2), 14-21.

[24] Bensoussan, M., & Kreindler, I. (1990). Improving advanced reading comprehension in a foreign language: summaries vs. short-answer questions. *Journal of Research in Reading, 13*(1), 55-68.

[25] Joyner, D. (2016). *Introduction to Computing*. McGraw-Hill Education LLC.

[26] Sellam, T., Das, D., & Parikh, A. P. (2020). BLEURT: Learning robust metrics for text generation. arXiv preprint arXiv:2004.04696.

[27] Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. Advances in neural information processing systems, 33, 16857-16867.

[28] Morris, W., Choi, J., Holmes, L., Gupta, V., & Crossley, S. A. (in press). Automatic Question Generation and Constructed Response Scoring in Intelligent Texts. *Proceedings of the 17th International Conference on Educational Data Mining*.

[29] Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S. and Roth, D. 2018. Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, 2018), 252–262.

[30] Broder, A. Z. (1998). On the resemblance and containment of documents. *Proceedings Compression and Complexity of SEQUENCES 1997* (Cat no 97TB100171), 21–29.

[31] Morris, W., Crossley, S., Holmes, L., & Trumbore, A. (2023, March). Using transformer language models to validate peer-assigned essay scores in massive open online courses (MOOCs). In *LAK23: 13th international learning analytics and knowledge conference* (pp. 315-323).

[32] Morris, W., Crossley, S., Holmes, L., Ou, C., Dascalu, M., & McNamara, D. (2024). Formative Feedback on Student-Authored Summaries in Intelligent Textbooks Using Large Language Models.

International Journal of Artificial Intelligence in Education, 1-22.

[33] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[34] Sullivan, G. M., & Feinn, R. (2012). Using effect size—or why the P value is not enough. *Journal of graduate medical education*, *4*(3), 279-282.

[35] Bertsch, S., Pesta, B.J., Wiscott, R. and McDaniel, M.A. 2007. The generation effect: A meta-analytic review. *Memory & Cognition*. 35, 2 (Mar. 2007), 201–210. DOI:https://doi.org/10.3758/BF03193441.

[36] Tripepi, G., Jager, K. J., Dekker, F. W., & Zoccali, C. (2010). Selection bias and information bias in clinical research. *Nephron Clinical Practice*, *115*(2), c94-c99.