Fostering Interdisciplinary Learning for Elementary Students through Developing Interactive Digital Stories

Anisha Gupta[1[0000-0002-6385-8974]], Andy Smith[1[0000-0002-6385-8974]], Jessica Vandenberg[1[0000-0001-6497-1840]], Rasha ElSayed[2[0000-0002-5637-6766]], Kimkinyona Fox[2[0000-0002-5409-2866]], James Minogue[1[0000-0002-1644-7697]], Aleata Hubbard Cheuoua[2[0000-0002-7310-1326]], Kevin Oliver[1[0000-0001-8463-0621]], Cathy Ringstaff[2[0000-0002-2856-1120]], and Bradford Mott[1[0000-0003-3303-4699]]

[1] North Carolina State University, Raleigh NC 27695, USA
{agupta44,pmsmith4,jvanden2,jminogu,kmoliver,bwmott}@ncsu.edu
[2] WestEd, San Francisco, CA 94107, USA
{relsaye,kfox,ahubbar,cringst}@wested.org

Abstract. Recent years have seen growing awareness of the potential digital storytelling brings to creating engaging K-12 learning experiences. By fostering students' interdisciplinary knowledge and skills, digital storytelling holds great promise for realizing positive impacts on student learning in language arts as well as STEM subjects. In parallel, researchers and practitioners increasingly acknowledge the importance of computational thinking in supporting K-12 students' problem solving across subjects and grade levels, including science and elementary school. Integrating the unique affordances of digital storytelling and computational thinking offers significant potential; however, careful attention must be given to ensure students and teachers are properly supported and not overwhelmed. In this paper, we present our work on a narrative-centered learning environment that engages upper elementary students (ages 9 to 11) in computational thinking and physical science through the creation of interactive science narratives. Leveraging log data from a pilot study with 28 students using the learning environment, we analyze the narrative programs students created across multiple dimensions to better understand the nature of the resulting narratives. Furthermore, we examine automating this analysis using artificial intelligence techniques to support real-time adaptive feedback. Results indicate that the learning environment enabled students to create interactive digital stories demonstrating their understanding of physical science, computational thinking, and narrative concepts, while the automated assessment techniques showed promise for enabling real-time feedback and support.

Keywords: Digital storytelling · narrative-centered learning · block-based programming.

1 Introduction

Digital storytelling, which seeks to combine traditional storytelling with digital tools, has received growing recognition as an effective tool for enabling learning in K-12 classrooms [24, 29]. Digital storytelling activities can take many forms, though generally focus on students creating a narrative, documentary, or interactive story around a topic [24,29,32]. Studies have shown digital storytelling to be an effective tool in subjects outside of language arts, promoting exploration of science phenomena [8, 9], as well as enhancing critical thinking and motivation [28]. However, the benefits from digital storytelling-based interventions are not uniform across all students, highlighting the need for more effective support for both students and teachers.

An additional hurdle for utilizing digital storytelling activities in the classroom is the need to overcome a perceived lack of alignment with curricular requirements beyond use in language arts. Notably, there is growing emphasis on computational thinking and science at the elementary level, and teachers who already face dwindling instructional time may view storytelling as a sub-optimal form of teaching [30]. However, there is a growing body of research showing the synergies of the storytelling process, where students create, refine, and present a story, with critical aspects of computational thinking and science [16, 21]. Additionally, with a growing focus on STEM education, many teachers also see the value in integrating language arts skills across multiple disciplines. Given these factors, there is an opportunity for tools and systems supporting storytelling activities if they can be designed to support learning while not overwhelming students and teachers.

InfuseCS is a narrative-centered learning environment designed to engage upper elementary students (ages 9 to 11) in deep, meaningful physical science and computational thinking learning through the creation of interactive science narratives. In InfuseCS, students use a block-based programming interface that enable the creation of narrative programs that drive a virtual stage for storytelling. This interface features a palette of specialized blocks that can be dragged and dropped to form an interactive script—complete with dialogue, scene changes, character entrances and exits, and branching—in the style of a “choose your own adventure.” By engaging with this hands-on, narrative design tool, students not only visualize but also play out their interactive stories, bringing concepts and competencies in physical science and computer science to life. This design builds on best practices for creating block-based programming environments suitable for young learners [10, 31].

This paper reports on an analysis of the interactive narratives students created in a pilot study using InfuseCS. Specifically, we investigate the following two research questions as we look to better support students’ interdisciplinary learning:

> ***RQ1:*** *In what ways are students able to use block-based narrative programming to create interactive narratives, when evaluated from a multi-dimensional perspective?*

> ***RQ2:*** *Do AI-driven automated assessment techniques enable accurate evaluation of student created interactive narratives across multiple dimensions?*

To answer these questions, we use data collected in a pilot study with upper elementary grade students using InfuseCS. Results show students produced a range of stories, exhibiting a variety of storytelling, physical science, and computational thinking concepts. Furthermore, automated assessment techniques were capable of accurately evaluating student stories across multiple dimensions (i.e., storytelling, physical science, and computational thinking), although less accurately than human evaluators. Additional inspection of results from the automated assessment models reveals potential areas of improvement, as well as insights to consider for other learning environments looking to analyze interactive narratives created by upper elementary students.

2 Related Work

Narrative offers a promising tool for engaging students in meaningful problem solving. Narrative not only provides a unique modality to understand the world around us [3] but also to communicate our conceptual understandings to others [2]. In the context of education, digital storytelling activities, in which students create their own narratives, have shown great promise in leveraging narrative to enhance science learning [25], and improve cognitive measures such as visual memory and writing skills [27]. Positive outcomes have also been demonstrated on affective measures such as student engagement and motivation [29], as well as 21st century skills such as argumentation and cooperation [18].

The benefits of digital storytelling interventions are not uniform and depend heavily on the ability of students to construct narratives with the provided tools and resources. This has led to a wide range of research into the design and support of digital storytelling interventions for students across a variety of age ranges. Rao et al. [23] found that priming students, via emotional stimuli or by having the students display varied facial gestures, resulted in statistically richer stories. Relatedly, Chu et al. [5] found that students' storytelling was enhanced through creative, imaginative enactment. Franklin et al., [7], sought to leverage the similarities between storytelling and computational thinking using a modified version of the Scratch environment. Students created animated stories involving 2D movement of sprites, as well as using user inputs to trigger events such as audio clips. The Storytelling Alice system allowed middle school students to manipulate 3D characters and scenery to create stories and motivate students, particularly female students, to learn computer programming [13–15]. Evaluations of these systems tend to focus heavily on interest toward computing and demonstration of computational thinking skills rather than evaluating or supporting storytelling. Other systems have investigated using block-based programming in a tangible, sticker-based block language used as part of an interactive storybook [11], and as a method for introducing computational thinking strategies into English language learning [21]. This work extends these efforts by collecting a corpus of student created interactive science narratives represented as block-based narrative programs and applying automated assessment techniques to evaluate the narratives from a multidimensional perspective.

AI techniques, such as natural language processing, have been explored for automating the evaluation of student generated text within the context of interactive learning environments in a range of applications. Carpenter et al. [4] explored machine learning models for automatically analyzing student reflections written during game-based learning, and found pre-trained ELMo embeddings to be the best representation for improving predictive performance of the models. Park et al. [20] introduced a framework for automating the detection of disruptive talk within in-game chat in the context of collaborative game-based learning, finding that models using pre-trained BERT embeddings performed the best. Ke et al.'s [12] survey on advances in automated essay grading explores recent work for grading longer text responses. These works point to the potential of natural language processing techniques for automatically assessing student created narrative programs.

3 The InfuseCS Learning Environment

InfuseCS is a narrative-centered learning environment designed to promote engaging computationally-rich science learning for students in upper elementary school, while developing proficiency in digital storytelling and computational thinking. Using InfuseCS, students solve physical science problems and develop interactive science narratives reflecting on their problem-solving experiences. These interactive science narratives are driven by programs students create using a block-based programming language to bring their storyworld to life on a virtual stage (Figure 1).

InfuseCS features an overarching backstory about a group of explorers stranded by a tropical storm on a remote Pacific island. Students work to develop solutions to problems the explorers encounter on the island. InfuseCS consists of two main components: *Science Explorer* and *Narrative Designer*. The Science Explorer presents students with problem-solving activities they can select from and permits them to interact with characters and tools in the virtual environment to learn about different types of energy and energy conversions. The science content area, and exercises were designed to align with US science standards for this age group. After completing their science investigation students progress to using the Narrative Designer, which incorporates a block-based programming interface, to create a short interactive science narrative related to the problem-solving prompt and science concepts they explored.

3.1 Science Explorer

Narrative serves an exceptional tool for structuring problem-solving scenarios that can engage students in science investigations. Using the Science Explorer,

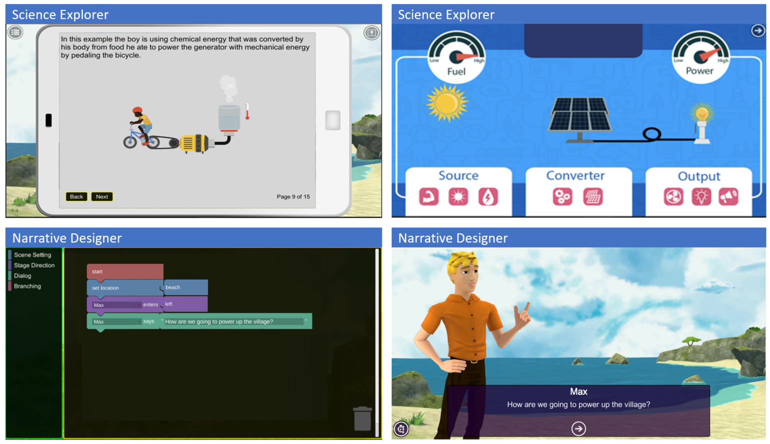

Fig. 1. InfuseCS narrative-centered learning environment.

students select scenarios to explore that center around disciplinary core ideas in physical science, such as developing a solution for powering up their makeshift village on the island. The Science Explorer allows students to identify relevant facts, pose hypotheses, research knowledge gaps, and conduct simulation-based investigations as they work toward their solution.

Fig. 2. Map of locations students can visit on the island.

After viewing the backstory, which appears as an animated sequence, students are presented with a map of the island denoting various locations they can visit in order to gather knowledge necessary for helping the explorers with the problem-solving scenario (Figure 2). The first location introduces students to the explorers stranded on the island, as well as introduces them to some of the science equipment they were able to salvage from their ship. The next locations continue to introduce students to different characters, as well as presents students with interactive media content around types of energy and energy conversion. InfuseCS supports a variety of interactive multimedia content, including workbooks containing text, images, and videos, as well as interactive simulation activities demonstrating scientific phenomena. After developing a solution to the problem, students gain access to the final two locations on the map that serve as a transition to the Narrative Designer component of InfuseCS. Students are first introduced to the narrative blocks they can utilize to create their narrative programs and then presented with an overview of the block-based narrative programming interface, including details on how to iteratively develop and test their interactive narratives.

3.2 Narrative Designer

InfuseCS's Narrative Designer enables students to develop interactive science narratives through an easy-to-use, drag-and-drop interface based around block-based programming (Figure 3). It allows students to incrementally develop, elaborate, and refine their stories in a rapid creative cycle, while supporting testing of their stories through its interactive visualization engine. It provides students access to characters, props, and scenery to populate their storyworlds.

Fig. 3. Iterative narrative program development.

The block-based programming interface of InfuseCS utilizes four types of custom blocks which are designed to enable key aspects of storytelling (Figure 4). This includes setting the location of the scene, arranging characters, enabling dialogue between characters, and supporting branching stories. Since InfuseCS targets upper elementary students (ages 9-11), the custom narrative blocks are designed utilizing best practices from previous research in this age range [6, 7]. These best practices include limiting the programs to a single thread of execution and avoiding event-driven programming. This is accomplished by providing students with a start block, and only executing blocks connected to it. To further limit the potential for syntactic errors, each category of narrative block utilizes a distinct color to leverage visual affordances, and argument blocks are typed so that only the correct type of argument can be used (e.g., blocks denoting locations can only be used with the scene setting block). The blocks also limit the set of characters and locations students can use. To help scaffold students in creating their narrative programs, an initial starter story is provided to encourage students to modify and customize the program rather than being a purely generative activity. There are four block categories available to students: *Dialogue* blocks, *Stage Direction* blocks, the *Ask Audience* block, and *Scene* blocks.

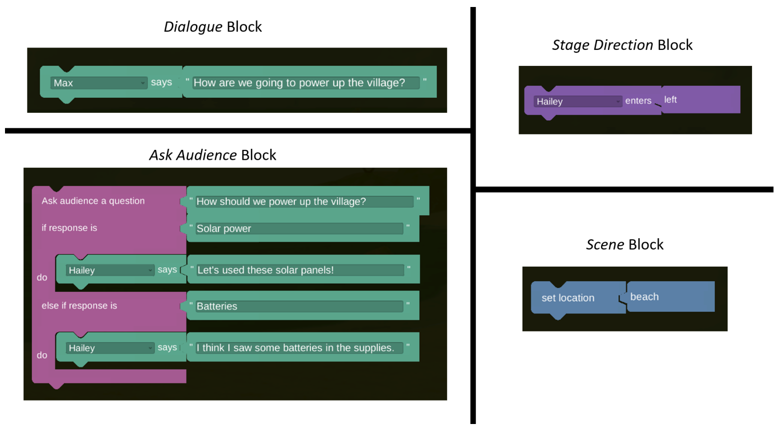

Fig. 4. Custom narrative blocks.

The most used blocks in student narratives are *Dialogue* blocks and *Stage Direction* blocks. *Dialogue* blocks generate lines of character dialog through the combination of a Character block and a Text block. The Character block contains a dropdown list of characters that have been previously introduced to the student earlier in the activity, as well as a Narrator option. *Stage Direction* blocks enable students to place and remove characters from the scene. *Stage Direction* blocks consist of an Enter block that places a character in a scene at a specific position (left, center, right), or an Exit block that removes the defined character from the scene. Like the *Dialogue* blocks, characters are selected via a dropdown menu.

The *Ask Audience* block allows students to incorporate branching into their narratives. Modeled after an If-Then-Else block, the *Ask Audience* block requires students to define three parameters. First, they must define the prompt that will be presented to the viewer of the story. The next two parameters represent the two choices that will be presented as options. Under each choice of the *Ask Audience* block is space for students to add additional blocks defining what will occur if the user chooses that branch.

The final category is *Scene* blocks, which allow students to set the location where the events in the narrative take place. Location blocks are chosen from a predefined set to fit with the remote island scenario of the activity, and to align with art assets for the visualization engine described below.

When first entering the narrative programming interface, students are presented with a starter story that uses at least one of each type of block. This story was designed to be runnable, so that students can experiment with the visualization controls early in the process of developing their story. At any point in the narrative programming process, students can press the run button to play their story (Figure 3). This is accomplished by first converting the block-based representation into Ink script (https://www.inklestudios.com/ink/), a narrative scripting language developed primarily for game applications. Ink script representations of each block are defined using Google's Blockly framework, with a server taking the XML representation of the block program and returning an Ink script. The Ink script is passed to the visualization engine, which presents the animated story using Unity assets. The visualization pauses after each dialog utterance to give the audience time to read, and advances when a button is pressed. When the end of the Ink script is reached, the student returns to the narrative programming interface where they can continue to iterate and revise their story until they are satisfied. Students are encouraged to incorporate the science concepts they explored earlier using the Science Explorer into their stories as they reflect on their problem solving.

4 Pilot Study

A pilot study was conducted using InfuseCS to better understand its capabilities for supporting upper elementary students' interdisciplinary learning. The pilot study consisted of three data collections in Spring 2022. The first two data collections occurred in two traditional classroom settings in the western United States. Participants in these data collections included 57 fourth grade students, ages 9 to 11, and the classroom teachers facilitated the learning experience with InfuseCS in their classrooms. The two teachers had previously attended 8 hours of professional development during which they were introduced to InfuseCS and were guided through best practices regarding how to teach the physical science and computational thinking content as well as suggested pacing. The third data collection took place in a summer camp context in the southeastern United States. Participants in this data collection included 26 rising third through fifth grade students. The research team led this implementation during a single day of the students' 5-day computer science-focused camp. Due to challenges with conducting in-person research due to the COVID-19 pandemic only a subset of the students completed all of the study activities. In the analysis presented in this paper, we include data from the 28 students who completed all of the study activities. Informed parental consent and student assent was obtained from all participants under a human subjects approved protocol for the pilot study.

The research team suggested four 45-minute sessions for the classroom implementations, whereas all sessions for the summer camp group were condensed into a single day (approximately 4 hours total). The implementation flow for both the classroom and camp studies included a brief introduction, completion of a short pre-survey on prior experience with coding and computing, engaging with InfuseCS during which students learned the science content, completing a paper narrative planning worksheet, and finally creating their block-based narrative program within InfuseCS.

5 Results

Students' narrative programming began with an initial starter story (as previously discussed), which provided a runnable narrative program that demonstrated all of the available blocks. Of the 28 students, across both contexts, 3 students interacted with the narrative programming interface, but did not change or add to any of the starter story blocks. Student stories ranged from a minimum of 0 self-selected blocks to a maximum of 59 ($M = 15.0$, $SD = 14.4$), with the bulk of the blocks appearing in the student-created narratives being *Stage Direction* ($M = 5.4$, $SD = 7.3$) and *Dialogue* ($M = 6.3$, $SD = 6.8$) blocks. Students changed locations in their stories using *Scene* blocks approximately 2 times ($M = 2.3$, $SD = 2.0$), with the *Ask Audience* block being used least often ($M = 1.0$, $SD = 0.8$). Students used roughly 2 to 3 characters in their stories ($M = 2.8$, $SD = 1.36$) and ran them approximately 32 times each to visualize their stories ($M = 31.6$, $SD = 30.0$).

> ***RQ1:*** *In what ways are students able to use block-based narrative programming to create interactive narratives, when evaluated from a multi-dimensional perspective?*

To investigate RQ1, we evaluated the students' stories across three dimensions: story structure, science concepts, and computational thinking. For this evaluation, we created a 3-level rubric with evaluative criteria (Table 1). These categories were of interest because InfuseCS aims to support interdisciplinary learning, namely writing in language arts, science knowledge construction and use, and computational thinking (CT) skills and practices. The 0 to 2 rating scale was selected for ease of scoring, indicating minimal to sufficient demonstration of the skills under consideration. The criteria for determining story structure ratings included students' stories, however simple and brief, containing a recognizable beginning, middle, and end. Science concept usage was assessed according to students' use of science concepts and terminology included in InfuseCS's Science Explorer. The criteria for rating computational thinking included both the number and correct usage of blocks.

Two researchers trained on a portion of data, not used in the final analysis, resolving misunderstandings, and further refining the rubric. Both researchers independently rated the 28 student stories, then computed kappa for each category [17]. The Story Structure category achieved moderate agreement with a

Table 1. Narrative evaluation rubric.

	Story Structure	Science Concepts	Computational Thinking
0	No modifications to the starter story, or additions are nonsensical (e.g., "fjioau")	No inclusion of science concepts from the Science Explorer	Two or fewer block types are used
1	Modification of the starter story, but story is incomplete	Science concepts included, but are used inaccurately	3 to 4 block types are included, but misused
2	Story is complete and logical	Science concepts included and used accurately	All blocks are used correctly

kappa of .71 and the Science Concept and Computational Thinking categories achieved strong agreement with kappas of .84 and .83, respectively. We calculated averages for each category: Story Structure ($M = 1.3$, $SD = 0.77$), Science Concepts ($M = 1.1$, $SD = 0.8$), and Computational Thinking ($M = 1.2$, $SD = 0.79$). Across all 28 stories, students earned an average score of 3.7 ($SD = 1.87$) out of a possible 6 points. Figure 5 shows the distribution of story ratings when summing the scores for all three categories.

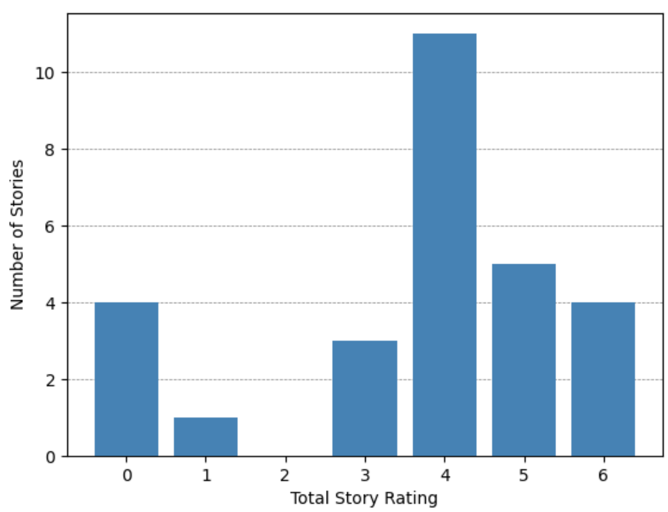

Fig. 5. Distribution of story ratings.

Overall, students performed roughly equally across the 3 areas, showing promise in their ability to integrate the different disciplinary concepts. When looking at the distribution of story ratings, it is notable that no students had a total score of 2, and only 1 student scored a 1. This indicates a significant gap between the low scoring students and the next grouping of students. More investigation is needed to determine if this gap is due to motivational factors, or perhaps a lack of comfort or understanding with the task. Fifty percent of the student stories (14) received a total rating between 3 and 4 points, and predominantly consisted of stories demonstrating varying levels of incorporation of science concepts ($SD = 0.50$), computational thinking ($SD = 0.62$), and story structure ($SD = 0.75$), with only one story scoring equally across all three dimensions of evaluation. Although most of these students demonstrated proficiency in one or two evaluation aspects, their stories require substantial improvement across other dimensions.

To support students in identifying weaknesses within their stories and improving the specific aspects necessary to achieve a higher rating within classroom time constraints, real-time feedback highlighting the areas requiring more attention could prove beneficial. By automating the assessment of stories in each evaluation dimension, students could receive such feedback, or alternatively, teachers could be prompted to guide students who require assistance. In fact, providing feedback across multiple criteria of overall story creation serves the purpose of offering students a rubric, which has been demonstrated to effectively support their learning and academic performance in traditional learning environments [1]. However, due to the creative and open-ended nature of the activity, students might inadvertently overlook the rubric designed to evaluate their stories. Hence, it becomes invaluable to have real-time feedback that not only addresses adherence to all the rubric criteria but also provides an automated assessment of the quality with which they fulfill each individual item on the rubric. Human evaluation of students' stories is both time consuming and labor intensive. Moreover, not all educators have the expertise or training to appropriately evaluate students' stories across all three dimensions [19]. Automating the evaluation of stories can help provide real-time feedback to students, while relieving the workload associated with manual scoring.

> ***RQ2:*** *Do AI-driven automated assessment techniques enable accurate evaluation of student created interactive narratives across multiple dimensions?*

Manual evaluation of students' stories can help us with post-hoc evaluations. However, manual evaluation of natural language text is resource intensive, both in terms of time and effort, and is not well suited for providing real-time feedback for students to improve the quality of their stories. Automated evaluation of student generated narratives across the different dimensions discussed above could be used to drive adaptive feedback in the block-based narrative interface. For instance, if the narrative's science rating is detected to be low, a feedback mechanism could prompt students to integrate appropriate science concepts in their stories. Alternatively, a teacher could be notified to provide assistance to the student. To automate each of the assessments in order to enable such real-time evaluation of student created interactive narratives, we investigated a set of AI-driven models. Given that the Story Structure and Science Concepts categories were largely based on the textual components of the narrative program, we first investigate natural language processing techniques for these categories.

Each student narrative program was converted into a string by concatenating all of the *Dialogue* blocks that the student utilized in their story, as well as parameters from any *Ask Audience* blocks. The resulting string representing the student story was then passed through a text normalization pipeline. A content knowledge-specific system dictionary was generated by extracting all of the text from the Science Explorer that introduced relevant science concepts to the students. Punctuation was removed from the text and the remaining text was split into words and added to the dictionary. Stop words were then removed from the dictionary, followed by lemmatization of the remaining words. Each student story was preprocessed similarly—removing punctuation, splitting text into words, removing stop words, and lemmatizing words that exist in NLTK's English vocabulary. The preprocessed words in the student story are then spell corrected to a word from the system dictionary if they are within a Levenshtein distance of 3. If they do not resemble any word from the system dictionary, they are not altered or removed from the story. The preprocessed words are then concatenated with space delimiters to reconstruct the story.

We generated a bag of words (BoW) feature representation for the stories using a similar process—punctuation was removed, the text was split into words, stop words were removed, and the remaining words were lemmatized. A bag of words feature vector was constructed with an entry for each word in the system dictionary. If a word in the story was found within a Levenshtein edit distance of 3 to a word in the system dictionary, the value in the bag of words feature vector for that word was set to 1. The bag of words feature vector was complemented with word count as an additional feature for each story to construct a baseline feature representation. While the BoW approach provides a straightforward method for identifying keywords and basic features, it is limited in its ability to capture local context and syntax. To address this limitation, we also incorporated advanced language models in our study, including Bidirectional Encoder Representations from Transformers (BERT) and Embeddings from Language Models (ELMo), both of which are pre-trained on large corpora and can effectively capture semantic nuances by understanding the context and relationships between words. These models were selected for their ability to provide a richer feature representation than BoW models alone.

We generated a dataset consisting of the most completed versions of the students' stories, determined by the saved version with the highest number of blocks used. The results for automated science and story ratings were then calculated using pre-trained versions of BERT (uncased distilBERT [26]) and ELMo embeddings [22], both with and without spelling correction applied to the stories. Alongside the BoW and word embeddings techniques, we investigated the impact of augmenting the dataset by subsampling each story by incrementally building the story one block at a time, and adding it to the training dataset with the human annotated rating for the completed story as the corresponding label. Using the resulting training dataset, a support vector machine was trained on the word embedding features to predict the story rating and science rating for the stories. Baseline results were obtained by training another support vector machine on the bag of words features.

Given the limited size of our dataset, we evaluated the support vector machine using leave-one-out cross validation. Accuracy and Cohen's kappa scores are listed for story rating prediction in Table 2, and for science rating prediction in Table 3. From Table 2, we observe the best results for story rating were achieved using BERT embeddings without subsampling, with accuracy 78.57% and Cohen's kappa of 65, indicating very high agreement with human scores. From Table 3, we observe the best results for science rating were achieved using ELMo embeddings with subsampling, with accuracy 57.14% and Cohen's kappa of 34.12, indicating fair agreement with human ratings. For both story and science rating predictions, we observe that the best performing baseline bag of word-based models are outperformed by the best performing word embedding-based models.

Table 2. Predictive performance for automated story rating.

Subsampled	**Representation**	**Accuracy**	$\boldsymbol{\kappa}$	**Acc. with spell correction**	$\boldsymbol{\kappa}$ **with spell correction**
No	ELMo	57.14%	28.51	57.14%	28.51
No	BERT	**78.57%**	**65.00**	75.00%	59.59
Yes	ELMo	64.00%	39.13	64.00%	39.13
Yes	BERT	67.86%	44.62	68.00%	44.62
No	BoW	-	-	46.00%	12.5
Yes	BoW	-	-	57.00%	24.49

Table 3. Predictive performance for automated science rating.

Subsampled	**Representation**	**Accuracy**	$\boldsymbol{\kappa}$	**Acc. with spell correction**	$\boldsymbol{\kappa}$ **with spell correction**
No	ELMo	50.00%	22.99	50.00%	23.14
No	BERT	46.43%	17.65	50.00%	22.99
Yes	ELMo	**57.14%**	**34.12**	54.00%	28.49
Yes	BERT	46.00%	17.16	50.00%	22.53
No	BoW	-	-	50.00%	21.60
Yes	BoW	-	-	32.14%	-0.08

To automate Computational Thinking rating for students' stories, we use a feature vector consisting of the number of *Scene*, *Dialogue*, and *Ask Audience* blocks as input features to a logistic regression model with human-annotated Computational Thinking scores as target labels for prediction. A prediction accuracy of 62.96% was achieved.

6 Discussion and Limitations

InfuseCS enabled students to produce a wide range of interactive narratives, ranging from 2 to over 30 lines of dialogue. As the target of the system is inherently interdisciplinary, the multidimensional rubric we developed allows us to better understand what aspects students struggle with, and where and how different types of support could be incorporated into the intervention. Overall, students performed roughly equal across the three concept areas, showing promise in their ability to integrate storytelling, science, and computational thinking concepts. Furthermore, automated assessment techniques demonstrated potential for predicting computational thinking, science, and story ratings of narrative programs created by elementary school students. Predictions from the models could be used to provide pedagogical support to students based on their predicted scores.

We achieved the best results for predicting story ratings using BERT embeddings on complete versions of students' stories. For science rating prediction, we obtained the best results using ELMo embeddings by training on subsampled stories. Training on subsampled stories did not improve the accuracy of story rating predictions. A possible explanation for this is that incomplete subsampled stories with rating labels corresponding to complete stories might be misleading for the model during training.

Preprocessing the stories through spell correction based on vocabulary used in the learning environment improves the predictive accuracy for science ratings by 4% for models trained on BERT embeddings, and have negative or no effect on the models trained on ELMo embeddings. We also see the best model performance using ELMo embeddings without spellcheck. This might indicate that ELMo embeddings are inherently better at handling misspelled words as compared to BERT embeddings. The spellcheck with respect to vocabulary used in the learning environment is also not reliable in some cases. For instance, while the spellcheck was able to correct "soral" to "solar", "recourses" to "resource", and "energey" to "energy", it incorrectly changed names of characters like "max" and "mia" to "fan" and "oil", since they are not recognized words in the English dictionary and are within a Levenshtein distance of 3 of vocabulary used in the learning environment.

The distinction between the human scoring rubric for computational thinking ratings 1 and 2 is based on correct usage of the block parameters for each block type. Blocks for which students left the required parameters blank or filled them in with text that did not reflect their purpose (such as conditions for Ask Audience blocks) were considered to be incorrectly used. Since a representation of block parameters were not included as input features for the automated computational thinking evaluation, this could have led to stories with computational thinking rating of 1 being mislabeled as a rating of 2.

One of the main limitations of our work is that the results are evaluated on a dataset consisting of only 28 students. With a larger and more diverse dataset, we would have the opportunity to expand and refine our set of rubrics for evaluating student-created stories. This would allow us to go beyond the basic elements of narrative structure, such as assessing whether the story has a beginning, middle, and end, to include more robust and nuanced evaluation metrics, such as rubrics that measure the "interestingness" or "creativity" of the stories. Additional data would also help us better train and evaluate our automated scoring models. Moreover, we concatenated the dialogue blocks from the block-based narrative programs to construct stories in text format that are used for model training and evaluation. Some of the structure of the program is lost in this process, losing information of who said what in the story, if-then logic in Ask Audience blocks, and stage directions. Preserving such information from the stories might help improve predictive performance of our automated scoring models.

7 Conclusion and Future Work

Digital storytelling continues to show great potential for improving student learning across a wide range of ages and subject areas. It offers promise as an effective way of embedding computational thinking into existing subjects such as language arts and science. However, we need to better understand how to best utilize these activities in different learning contexts in a way most beneficial to all students. This includes more sophisticated methods for evaluating students' narratives, as well as supporting their creation.

In this work, we have presented an analysis of student stories created using the InfuseCS narrative-centered learning environment. InfuseCS engages students in science explorations and enables them to create interactive narratives using a structured, block-based programming interface. Data from 28 students who participated in a pilot study using InfuseCS was analyzed to better understand the nature of the narratives students created. The narrative programs showed that InfuseCS enabled students to create narratives demonstrating understanding of concepts focused on storytelling, science, and computational thinking. However, not all students were able to produce narratives of high quality. While most stories received high scores in one or two evaluation dimensions, they exhibited a notable deficiency in some aspect of the evaluation, as determined by human scoring. In an effort to enable real-time support to students on which aspects of their story needs more attention, we investigated automated assessment techniques. The automated assessment of student stories showed promise across all three evaluation dimensions, though further refinements are necessary to reach human level accuracy.

Future studies should investigate the effectiveness of providing different supports driven by robust automated assessment techniques. There is also a need to develop and investigate expanded rubrics for student narrative programs, allowing for more granular analysis and targeted interventions across narrative systems. Follow-up studies featuring think-alouds and interviews with students should be conducted to allow for better interpretation of the choices students make in their stories along with their rationale. Additionally, future research should explore the utility of more advanced large language models for automated story assessment. While this study employed pre-trained BERT and ELMo embeddings, newer models such as GPT-4 offer potentially richer representations of text. These advanced language models could provide a more nuanced understanding of students' narrative programs. Finally, it will be important to work with teachers to develop tools to better enable them to support their students in generating the best narratives possible.

Acknowledgements. This research was supported by the National Science Foundation under Grants DRL-1921495 and DRL-1921503. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Andrade, H., Du, Y.: Student perspectives on rubric-referenced assessment. Practical Assessment, Research, and Evaluation **10**(1), 3 (2005)
2. Avraamidou, L., Osborne, J.: The role of narrative in communicating science. International Journal of Science Education **31**(12), 1683–1707 (2009)
3. Bruner, J.S.: Acts of meaning: Four lectures on mind and culture, vol. 3. Harvard University Press (1990)
4. Carpenter, D., Cloude, E., Rowe, J., Azevedo, R., Lester, J.: Investigating student reflection during game-based learning in middle grades science. In: Proceedings of the 11th International Learning Analytics and Knowledge Conference. pp. 280–291 (2021)
5. Chu, S.L., Quek, F., Tanenbaum, T.J.: Performative authoring: Nurturing storytelling in children through imaginative enactment. In: Proceedings of the 6th International Conference on Interactive Digital Storytelling. pp. 144–155. Springer (2013)
6. Dwyer, H., Hill, C., Hansen, A., Iveland, A., Franklin, D., Harlow, D.: Fourth grade students reading block-based programs: predictions, visual cues, and affordances. In: Proceedings of the 2015 ACM Conference on International Computing Education Research. pp. 111–119 (2015)
7. Franklin, D., Skifstad, G., Rolock, R., Mehrotra, I., Ding, V., Hansen, A., Weintrop, D., Harlow, D.: Using upper-elementary student performance to understand conceptual sequencing in a blocks-based curriculum. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. pp. 231–236 (2017)
8. Henriksen, D.: Full steam ahead: Creativity in excellent stem teaching practices. The STEAM journal **1**(2), 15 (2014)
9. Henriksen, D., Mishra, P., Fisser, P.: Infusing creativity and technology in 21st century education: A systemic view for change. Journal of Educational Technology & Society **19**(3), 27–37 (2016)

10. Hill, C., Dwyer, H.A., Martinez, T., Harlow, D., Franklin, D.: Floors and flexibility: Designing a programming environment for 4th-6th grade classrooms. In: Proceedings of the 2015 ACM SIGCSE Technical Symposium on Computer Science Education. pp. 546–551 (2015)
11. Horn, M.S., AlSulaiman, S., Koh, J.: Translating roberto to omar: Computational literacy, stickerbooks, and cultural forms. In: Proceedings of the 12th International Conference on Interaction Design and Children. pp. 120–127 (2013)
12. Ke, Z., Ng, V.: Automated essay scoring: A survey of the state of the art. In: Proceedings of the International Joint Conference on Artificial Intelligence. vol. 19, pp. 6300–6308 (2019)
13. Kelleher, C.: Supporting storytelling in a programming environment for middle school children. In: Proceedings of the 2nd International Conference on Interactive Digital Storytelling. pp. 1–4. Springer (2009)
14. Kelleher, C., Pausch, R.: Using storytelling to motivate programming. Communications of the ACM **50**(7), 58–64 (2007)
15. Kelleher, C., Pausch, R., Kiesler, S.: Storytelling alice motivates middle school girls to learn computer programming. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1455–1464 (2007)
16. Lee, I., Martin, F., Apone, K.: Integrating computational thinking across the k–8 curriculum. ACM Inroads **5**(4), 64–71 (2014)
17. McHugh, M.L.: Interrater reliability: The kappa statistic. Biochemia medica **22**(3), 276–282 (2012)
18. Niemi, H., Multisilta, J.: Digital storytelling promoting twenty-first century skills and student engagement. Technology, Pedagogy and Education **25**(4), 451–468 (2016)
19. Paiva, J.C., Leal, J.P., Figueira, Á.: Automated assessment in computer science education: A state-of-the-art review. ACM Transactions on Computing Education (TOCE) **22**(3), 1–40 (2022)
20. Park, K., Sohn, H., Min, W., Mott, B., Glazewski, K., Hmelo-Silver, C.E., Lester, J.: Disruptive talk detection in multi-party dialogue within collaborative learning environments with a regularized user-aware network. In: Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 490–499 (2022)
21. Parsazadeh, N., Cheng, P.Y., Wu, T.T., Huang, Y.M.: Integrating computational thinking concept into digital storytelling to improve learners' motivation and performance. Journal of Educational Computing Research **59**(3), 470–495 (2021)
22. Peters, M.E., Neumann, M., Zettlemoyer, L., Yih, W.t.: Dissecting contextual word embeddings: Architecture and representation. arXiv preprint arXiv:1808.08949 (2018)
23. Rao, N., Chu, S.L., Faris, R.W., Ospina, D.: The effects of interactive emotional priming on storytelling: An exploratory study. In: Proceedings of the 12th International Conference on Interactive Digital Storytelling. pp. 395–404. Springer (2019)
24. Robin, B.R.: Digital storytelling: A powerful technology tool for the 21st century classroom. Theory into practice **47**(3), 220–228 (2008)
25. Robin, B.R.: The power of digital storytelling to support teaching and learning. Digital Education Review (30), 17–29 (2016)
26. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
27. Sarıca, H.Ç., Usluel, Y.K.: The effect of digital storytelling on visual memory and writing skills. Computers & Education **94**, 298–309 (2016)

28. Shahid, M., Khan, M.R.: Use of digital storytelling in classrooms and beyond. Journal of Educational Technology Systems **51**(1), 63–77 (2022)
29. Smeda, N., Dakich, E., Sharda, N.: The effectiveness of digital storytelling in the classrooms: A comprehensive study. Smart Learning Environments **1**, 1–21 (2014)
30. Tan, M., Lee, S.S., Hung, D.W.: Digital storytelling and the nature of knowledge. Education and Information Technologies **19**, 623–635 (2014)
31. Weintrop, D., Hansen, A.K., Harlow, D.B., Franklin, D.: Starting from scratch: Outcomes of early computer science learning experiences and implications for what comes next. In: Proceedings of the 2018 ACM Conference on International Computing Education Research. pp. 142–150 (2018)
32. Yang, Y.T.C., Wu, W.C.I.: Digital storytelling for enhancing student academic achievement, critical thinking, and learning motivation: A year-long experimental study. Computers & Education **59**(2), 339–352 (2012)