PopBots: Designing an Artificial Intelligence Curriculum for Early Childhood Education

Randi Williams, Hae Won Park, Lauren Oh, Cynthia Breazeal

Personal Robots Group, MIT Media Lab

20 Ames Street, Cambridge, Massachusetts, 02142

{randiw12,haewon,laurenoh,cynthiab}@media.mit.edu

Abstract

PopBots is a hands-on toolkit and curriculum designed to help young children learn about artificial intelligence (AI) by building, programming, training, and interacting with a social robot. Today's children encounter AI in the forms of smart toys and computationally curated educational and entertainment content. However, children have not yet been empowered to understand or create with this technology. Existing computational thinking platforms have made ideas like sequencing and conditionals accessible to young learners. Going beyond this, we seek to make AI concepts accessible. We designed PopBots to address the specific learning needs of children ages four to seven by adapting constructionist ideas into an AI curriculum. This paper describes how we designed the curriculum and evaluated its effectiveness with 80 Pre-K and Kindergarten children. We found that the use of a social robot as a learning companion and programmable artifact was effective in helping young children grasp AI concepts. We also identified teaching approaches that had the greatest impact on student's learning. Based on these, we make recommendations for future modules and iterations for the PopBots platform.

Figure 1: PopBot Components. The social robot is made of a smartphone,LEGO blocks, motors, and sensors. The blocks-based programming interface resides on a tablet.

Children growing up in the era of artificial intelligence (AI) will have fundamentally different relationships with smart technologies than those who first encountered AI

later in life. New AI-enabled interfaces that support interaction through gesture, touch, and speech allow younger and younger children to access digital content and services. However, young children do not yet understand how AI-enabled devices, such as smart toys, work. It is important that they do, however, so that children can use them constructively and safely. Although there are a growing number of resources to learn about AI, most of these curricula target students in high school or beyond, and very few are appropriate for non-programmers. For this reason, we designed an early AI curriculum to help young children learn about artificial intelligence by enabling them to build, program, train and interact with their own social robots.

The PopBots Platform and Curriculum consists of a social robot toolkit, three hands-on AI activities, and associated assessments for young children to explore machine learning, reasoning, and generative algorithms. In this paper, we discuss the tools we developed and the results of an evaluative study with 80 four-to-six year olds. We investigated differences in how well children understood AI concepts based on age and gender as well as children's hands-on interaction with the PopBots toolkit components. While factors like prior technical skills and age can certainly impact how much children can learn, we hypothesized that their interaction with the toolkit would really drive their understanding.

Background

Children and AI

Today, we have robots in our homes and intelligent agents in our pockets. Children within their first years of life can now interact with tablets and toys that have magnitudes more computing power than personal computers even just a decade ago. Personal background, individual experiences, social and cultural factors (like how parents talk about technology) all impact on how children understand AI-enabled devices as mental, psychological, and emotional entities (Turkle 1984; Druga et al. 2017; 2018). As children gain exposure and understanding of AI technology, their reasoning about these devices becomes more thoughtful and nuanced (van Duuren and Scaife 1996; Severson and Carlson 2010; Druga et al. 2017).

Technological literacy is becoming increasingly important as children grow up with intelligent machines. Previous

studies have shown that children do not have a clear understanding of how smart toys and other intelligent technologies work (McReynolds et al. 2017; Druga et al. 2017). A study on children's interactions with smart toys found that they tell toy robots personal information not realizing that the toy can record their conversations (McReynolds et al. 2017). Furthermore, studies exploring children's conformity to smart toys report that children can be too trusting of these devices and can be influenced by them (Williams et al. 2018; Belpaeme et al. 2018). By empowering children with hands-on experiences that enable them to better understand how AI works, our hope is that parents and educators can help children gain a healthy appreciation for its abilities and limitations and develop an appropriate relationship with it.

Artificial Intelligence Education

AI education goes beyond computational thinking, it explores how computers sense, think, act, learn, make decisions, create, perceive, and make sense of things. In higher education, students can take courses where they program and test algorithms and systems that cover the full range of AI concepts (Imberman 2003; Kumar, Kumar, and Russell 2006; Martin 2007; Klassner and McNally 2007; Koski, Kurhila, and Pasanen 2008; Talaga and Oh 2009; Burgsteiner, Kandlhofer, and Steinbauer 2016). AI courses in higher education emphasize the importance of students building their own projects. Instructors often use robots in these courses to help students concretize their understanding (Kumar 2004; Koski, Kurhila, and Pasanen 2008; Talaga and Oh 2009).

Younger children, who are concrete thinkers and active learners, especially benefit from hands-on approaches to learn STEM. In our work, we use a social robot as a powerful platform that children relate to as a social and psychological other.

The first idea to teach children about AI came from Cynthia Solomon and Seymour Papert in 1971. They wanted children to explore AI through LOGO programming and the Turtle robot (Papert and Solomon 1971). Today, researchers are just beginning to implement some of those ideas. Machine Learning for Kids (Lane 2018), AI Programming with eCraft2Learn (Kahn and Winters 2018), and Cognimates (Druga 2018) are three online platforms that allow children to build projects using AI services by programming in block-based languages. However, the programming interfaces are designed for children ages 7 and up. Beyond these examples, there are also a few blog posts and Kickstarter[1] campaigns where instructors use video and non-programming activities to describe AI to children (Slavin 2016; Chen 2017; Milford 2018). In our work, we focus on even younger children to help them to learn about and create things with AI themselves. To our knowledge, there is no platform or curriculum in existence today that truly allows children 4–7 years of age to program, train, build, and learn about AI.

---

[1]Kickstarter, a crowdfunding platform. http://www.kickstarter.com/

Computational Thinking in Early Childhood Education

In 1974, Perlman developed the first computational thinking (CT) platform, TORTIS, to teach children about computational thinking and "most important of all...that learning is fun" (Perlman 1974). Today, a number of platforms exist for children ages 4+ including robots, educational apps, websites, board games, and hardware. Through different mediums, children learn computational thinking ideas (e.g., sequencing, conditionals, decomposition, etc.) by playing games, solving puzzles, and creating art. We took a number of design cues from these learning-to-code systems to develop our early AI curriculum.

Research with other CT platforms has found that young children learn best by seeing ideas repetitively, engaging with interactive activties, and putting computational ideas in a social framing. KIBO (Sullivan, Elkin, and Bers 2015; Elkin, Sullivan, and Bers 2016; Bers 2017), ScratchJr (Flannery et al. 2013; Leidl, Bers, and Mihm 2017) and SoRo (Social Robot Toolkit) (Gordon, Breazeal, and Engel 2015) help young children develop computational thinking skills with creative, open-ended activities. KIBO is a robot, for children ages 4–7, that uses tangible programming blocks with icons for pre-readers. Researchers designed a curriculum and assessments for the KIBO robot and found that younger children could learn computational thinking, but that their understanding was not as complete as their older counterparts because they needed more repetition (Sullivan 2016). ScratchJr is a free app for home use, developed by researchers at Tufts University that allows children ages 5–7 to explore computational thinking by programming stories and games on a tablet (Flannery et al. 2013). Researchers found that the most popular and effective features of ScratchJr were those where children could bring themselves into the project by adding their own recordings and pictures (Leidl, Bers, and Mihm 2017). Finally, SoRo is a programming interface for children 4–8 years old where children learn about computational thinking concepts by interacting with a social robot (Gordon, Breazeal, and Engel 2015). The tangible programming interface allowed children to compose programs by placing stickers on a sheet of paper and showing them to the robot to "teach" it a social rule. Researchers found that children could explore computational concepts through the social lens of predicting the robot's mental state and "teaching" emotive behaviors and social skills.

PopBot Curriculum Design

We built constructionist ideas into our AI curriculum because this is the most effective way to help young children learn (Papert 1980; Resnick 2017). The four principles underpinning our designs are:

- **Hands-on learning**. Make the toolkit interactive and let children guide themselves through activities.

- **End-to-end learning**. Have children play a role in every step of developing a full system, from training to operating a fully functional system.

- **Transparency and tinkerability**. Choose algorithms and give feedback that exposes as many logical and reasoning steps as possible.

- **Creative exploration**. Embed the AI algorithm in fun and creative activities. Give children agency to make something personally meaningful.

The PopBot platform (Figure 1) consists of an Android phone with the PopBot app, a tablet with the PopBlocks app, a LEGO WeDo 2.0[2] set with motors, and LEGO blocks. The PopBlocks App, with a blocks-based programming interface and a separate interface for each AI activity, runs on the tablet. The programming blocks are all picture-based to accommodate children who cannot read yet. The programmable output blocks of the robot include the robot's motor movements, TTS speech and sounds, facial expressions, eye color, and LED colors. The robot's input blocks include sensors for tilt, proximity, light, touch, and speech recognition. In addition, there are blocks that support control operations such as delays and for loops. Although the robot is programmable, it also has autonomous functionality to play an active role in the curriculum where it can explain its thinking to the child. The autonomous portion of the robot's "mind" becomes a metaphor through which children can examine the current state of the different algorithms. In addition, the tablet logs all information including the time, the current activity the child is using, and any buttons the child presses.

AI Curriculum

We used PopBots to teach children three AI concepts: knowledge-based systems, supervised machine learning, and generative AI (the last to show that machines can also be creative). Young children's ability to understand these concepts has not been previously explored.

Knowledge-Based Systems (or expert systems) are a common form of AI that contain two main components: a way to represent knowledge and a way to act on that knowledge. Learning about knowledge-based systems allows children to see how robots can learn something and then use it to make future decisions.

In the knowledge-based systems activity, children teach the robot the three rules of the game Rock-Paper-Scissors using the interface in Figure 2a. After children program the rules, the robot reads the rules back to them. Children are encouraged to try teaching the robot correct and incorrect rules. Next, they program the robot how to react to winning or losing by using the programming interface (Figure 2b). Finally, they can play the game against the robot – they tell the robot their move using the buttons in Figure 2c. During the game, the robot uses a state probability transition matrix to predict the child's next move based on their last three moves. If the robot's guess for a move is greater than chance (33%) then the robot will say "I think you will put X, so I will put Y because Y beats X." Otherwise, the robot says "I'm not sure what you'll put. I'll just guess." As the game
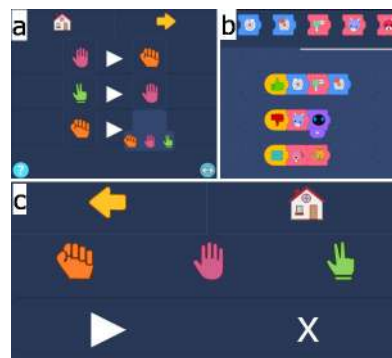
Figure 2: Screenshot of the Rock-Paper-Scissors activity. (a) Interface for teaching the rules (e.g. the top line reads "paper beats rock"). (b) Interface for programming how the robot to respond to wins, losses, and draws. The top line reads "if we win, then spin, next play a happy animation, next stop spinning." (c) Interface for playing rock, paper, scissors against robot.

progresses, the robot encourages the child to keep playing, "I'm getting good at this. The more you play me, the better we will get."

Supervised Machine Learning is a common AI technique for personalized recommender systems, like YouTube Kids. It involves forming a knowledge base by learning from examples. This topic allows children to see how robots can learn patterns based on a training set.

This activity involves teaching the robot how to classify healthy and unhealthy foods based on a number of features. The robot is pre-preprogrammed with information about 20 different foods such as color, which food group it belongs to, the number of calories per 100g, and the amount of sugar per 100g. Discrete features such as food group and color were mapped to numerical values that take proximity into account. For example, the fruit food group is closer to the vegetable food group than the dairy food group. The supervised machine learning algorithm uses k-nearest neighbors with k=3. When there are no labeled items, the robot says, "I don't know where anything goes yet." When the number of labeled items is less than or equal to k, the algorithm only compares to the one food that is the closest match.

Children train the robot to recognize healthy and unhealthy foods as indicated by the thumbs up and thumbs down interface shown in Figure 3. They can also program the robot how to respond to the foods. When the robot guesses which group a food item goes it says, "X is a lot like Y, so X goes in the same group as Y." Children can also use a help command to have the robot further explain its reasoning for grouping foods. Children can experiment with the number and kinds of foods in the training set to see how this impacts how well the robot classifies foods.

Generative Music AI is very different from the other two examples. We chose this activity to show children that robots do not always follow the rules, they can also be creative in their own way. Children experience real world examples of
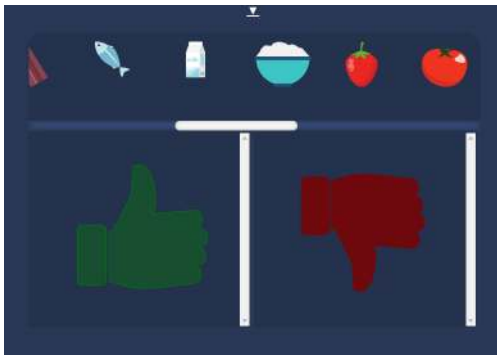
Figure 3: Screenshot of supervised machine learning interface. Children label foods as healthy (thumbs up) or unhealthy (thumbs down) by dragging them to the appropriate box. Children test a food by simply clicking it.

Figure 4: Screenshot of the generative AI activity: music remix. (a) Interface for describing music emotions with tempo and chord progression direction. (b) Interface for creating new songs.

generative AI in camera apps that use style transfer filters to remix photographs.

First, children discuss how tempo and chord progressions (whether music goes up or down) translate to emotions in music. For example, happy songs have a faster tempo and go up. Figure 4a is the interface that children use to program different parameter combinations to teach "musical emotions" to the robot. Then, when children use the play button to play music to the robot (Figure 4b), the robot plays back a remix.

To change the tempo of the song, the generative music algorithm uses simple rules. If the tempo needs to go faster, then it divides long notes into several rapid notes. To make the tempo slower, it does the opposite. The algorithm for generating chord progressions uses a probability transition matrix to add a sequence of notes that goes up in a consonant interval or down in a dissonant one. The robot only changes the middle of the song but keeps the rest of the notes the same. For example, a song input of A B A B with a positive mood may change the tune to A B *E C E* A B.

Assessments

We developed a set of assessments for each activity delivered as multiple choice questions on either a tablet or on paper. These questions probe children's understanding of the algorithm's basic functionality, edge cases, and initialization. The full materials for the assessment and the rationale for the answers can be found at

https://figshare.com/s/78c50547a770fed113d9.

Knowledge-Based Systems (KB)

1. We teach the robot the normal rules. Then, Sally plays rock and the robot plays paper. Who does the robot think has won? Sally or the robot? (Robot)

2. Sally plays paper five times. What does the robot think she will play next? Rock, paper, or scissors? (Paper)

3. The robot thinks that Sally will play paper next. What will the robot play so that it can beat Sally? Rock, paper, or scissors? (Scissors)

4. We changed the rules so that they are all opposite rules (paper beats scissors). Sally plays scissors and the robot plays paper. Who does the robot think has won? Sally or the robot? (Robot)

Supervised Machine Learning (SL)

1. The robot does not know anything about foods. You put strawberries and tomatoes into the good group. Which group will the robot think chocolate goes? The good group or the bad group? (Good)

2. What food does the robot think is most like a tomato? Strawberry, banana, or milk? (Strawberry)

3. You put ice cream in the good category and bananas in the bad category. What category will the robot put corn? The good category or the bad category? (Bad)

Generative Music (GM)

1. Priya asks the robot to play back with the bars in the middle. Does the robot play the same song or a different song? (Same)

2. Priya asks the robot to play back with the bars to the right. Does the robot play the same song or a different song? (Different)

3. Does the robot's song have to have some of the same notes as the input? (Yes)

Evaluative Study Procedure

In order to evaluate the curriculum, we delivered it as a one-week module in Pre-Kindergarten (Pre-K, usually age 4-5 years) and Kindergarten (K, usually age 5-6 years) classrooms. We used the curriculum assessments to measure how much of the material children understood and recorded robot interaction data from groups of children as they went through the curriculum. We used these two pieces of information to examine differences in assessment performance based on children's experience, age, grade, and amount of interaction with the toolkit.

Studies were conducted during Spring 2018 with children from five classrooms across four schools in the Greater Boston Area (Table 1). We collected data from 80 children between the ages of 4–6 years. Three classrooms were public afterschool programs, while the other two were in a private school. Only two classrooms contained Pre-K children. Classroom E had a mix of Pre-K and Kindergarten children, and classroom A had all Pre-K. In terms of previous

Variable		Count (%)
Age	4	10 (12.5%)
	5	46 (57.5%)
	6	24 (30%)
Gender	Female	39 (48.8%)
	Male	41 (51.2%)
Grade	Pre-K	27 (33.8%)
	K	53 (66.2%)
Total		80

Table 1: Participants by age, grade, gender.

	ED[a]	ELL[b]	Avg. Age	Gender (% Fem.)	N
A	Not available		4.59	47.1%	17
B	Not available		5.05	50.0%	22
C	14.3%	2.2%	5.37	42.1%	19
D	51.4%	32.9%	5.50	33.3%	6
E	38.9%	48.4%	6.00	37.5%	16

Table 2: Participant data by classroom. Including the propotion of students in the school who are economically disadvantaged[a] as defined by the household income of their parent/guardian(s) and who are who are English language learners[b]. Classrooms E and D were private schools and did not have available socioeconomic status data.

exposure to robotics and computational thinking, one student from Classroom D played Minecraft[3], one student from classroom C used a LEGO WeDo, and children in classroom B discussed robots in their classroom. None of the other children reported experience in robotics or computational thinking platforms.

Procedure

The order of the curriculum remained consistent across the different classrooms. First, we did pre-assessments followed by an introduction to programming with PopBots. This was followed by three AI activities in order: Knowledge-Based Systems (Rock-Paper-Scissors), Supervised Machine Learning (Food Classification), and Generative AI (Music Remix). Each activity lasted 10–15 minutes. Immediately after each activity, students answered the corresponding AI assessment questions. The researcher would show a picture and read the question aloud. Children answered individually on tablets or on paper with minimal interference from the researchers. In addition to quantitative data from the assessments and tablet logs, we also video recorded the sessions, collected data about children's interaction with the toolkit, and recorded children's responses to the assessment questions. In most classrooms, children worked in small groups of 4 or 5 peers. The exact amount of time spent on the activities varied for each classroom depending on the needs of the teachers. In classrooms C-E, activities were conducted over five consec-

---

[3]Minecraft, an adventure game in which players construct their own world. Can be programmed with simple macro codes. https://minecraft.net

Variable		KB	SL	GM	Overall
Age	4	59.3%	66.7%	58.3%	59.3%
	5	73.0%	73.1%	49.4%	66.6%
	6	75.0%	70.0%	54.5%	70.2%
	F	1.65	0.227	6.54	0.917
	p	0.20	0.798	0.701	0.404
Grade	PreK	63.3%	61.5%	53.3%	58.7%
	K	76.8%	74.6%	50.6%	70.9%
	F	5.88	3.79	0.126	6.54
	p	0.018*	0.0568	0.725	0.013*

Table 3: Student's performance on the assessments by age and grade. Significance level at $\alpha = 0.05$

utive days. In classrooms A and B, we only had two days in each classroom, so we did two activities each day. Half of Classroom B did not complete the generative AI activity because we ran out of time. Another complication was that sometimes children were absent, however this did not affect them negatively since activities did not build on each other.

Results

To validate the POP AI assessments, our measure of success was the number of correct responses on the assessment questions. We looked at differences in assessment performance by age and grade, gender, and classrooms using Chi-square analysis on individual assessment questions and one-way Anova tests on assessment averages. Then, we looked at children's tablet interaction logs and quantified the number of interactions and amount of time children spent doing different portions of each activity. Since children worked in groups that often changed from day to day, we summarized tablet information into averages for each school. To understand the relationship between average tablet interaction and assessment performance, we calculated Spearman's correlation coefficient.

Differences in Age, Grade, and Gender

Overall, we found that older children performed better on the assessments than younger children, see Table 3. The average score across all 10 assessment questions for Pre-K children was 58.7% vs. K children at 70.8% (F=6.54, p=0.013). Across the three activities, Pre-K children usually performed worse than K children.

On the knowledge-based systems assessment (questions KB1–4 on the Rock-Paper-Scissors activity), there was a significant difference in scores between Pre-K and K children (63.3% vs. 76.8%, F=5.88, p=0.018). The biggest difference occurred on KB3, "The robot thinks that Sally will play paper next. What will the robot play so that it can beat Sally?". This was the only question in this activity where there was a significant difference between Pre-K (58.3%) and K children (85.4%, $\chi^2(2, 65)=4.594$, p=0.032). The most common incorrect answer was "rock"; almost 40% of the Pre-K children gave this answer. One child reasoned that the robot would put rock because that is what the robot liked
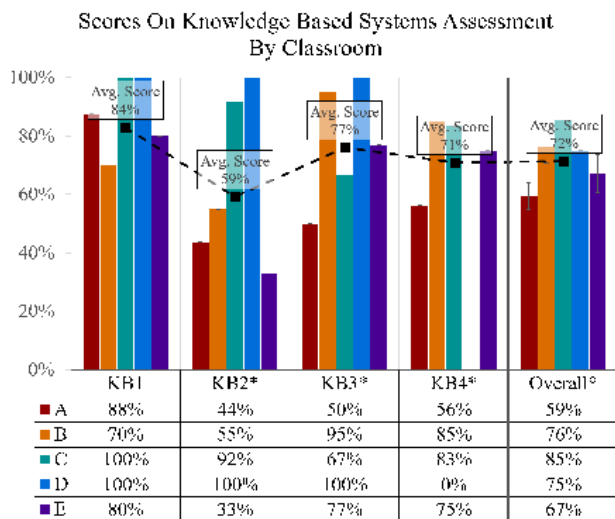
Figure 5: Student's performance on the knowledge based systems assessment by classroom. The line graph shows the average across all classrooms.

Figure 6: Student's performance on the supervised machine learning assessment by classroom. The line graph shows the average across all classrooms.

to play, suggesting that younger children may have been unprepared to connect the robot's predictions to how it would reason about which move to play next.

On the supervised machine learning assessment (questions SL1–3 for the food classification activity), there was a big difference in understanding on SL2 about nearest neighbor associations. Only 25% of four 4-year-olds got this question correct while nearly 90% of the 5 and 6 year olds did ($\chi^2(2, 55)$=14.164, p<0.01). K children were also significantly more likely to get this question correct (97.6%) than Pre-K children (38.5%, $\chi^2(1, 55)$=25.385, p<0.01). Children of all ages struggled with SL1, only 45% got it right, "You start the robot and put strawberries and tomatoes into the good group. Which group will the robot think chocolate goes? The good group or the bad group?" Counterintuitively, older children were less likely to get this question right ($\chi^2(2, 55)$=8.623, p=0.013).

Finally, all children struggled with the generative music assessment (questions GM1–3, music remix activity). Children only seemed to observe the most basic case – that when someone plays a song for the robot, the robot will play a song back that usually sounds different (GM1, 83.3%). The worst question was GM3, "Does the robot's song have to have some of the same notes as the input?" On this particular question, only 14% of children selected the correct answer, showing that the activity most likely conveyed the opposite of the AI concept. We did not observe any significant differences in age on this question. We also did not find any significant differences between gender on any of the assessments.

Differences by Classroom and Tablet Interaction

We only found differences related to age and/or grade on three of the assessment questions. However, looking at assessment performance by classroom we found differences
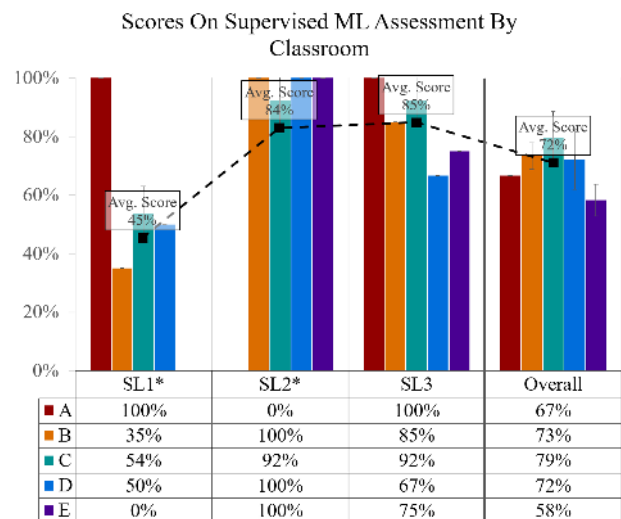
on 7 assessment questions that were often related to children's interaction with the tablets. Classrooms C and D did every activity as a class, passing one tablet and one robot around the room, while other classrooms worked in small groups each with their own robot. Figure 5 shows that classrooms C and D did much better on question KB2 than the other classrooms ($\chi^2(2, 65)$=13.48, p<0.01). This is likely due to the fact that passing the tablet around gave children more time to observe how the robot makes predictions. On this question we found a weak positive correlation (r_S=0.40) between the amount of time children spent playing Rock-Paper-Scissors against the robot and their assessment score. We also found a strong positive correlation between time spent playing against the robot and getting KB3, another question about the robot making predictions, correct (r_S=0.7). Similarly on KB4, a question about reversing the rules taught to the robot, there was a positive correlation between the amount of time spent training (r_S=0.3) and programming (r_S=0.82) the robot and answering correctly.

On the supervised machine learning activity, we found that trying different training sets really made a difference in children's understanding. Classrooms B-E did poorly on SL1, a question about how the robot categorizes objects when it only has positive examples. There was a strong, negative relationship between training and testing lots of foods and assessment performance. During the studies we observed that Pre-K classrooms spent more time testing different training sets while Kindergarten students taught the robot as many foods as possible before testing it. This made the Pre-K students more likely to discover the initialization case that SL1 inquires about. On question SL2, classroom A (the all Pre-K classroom) did much worse than all of the other classrooms. Initially, we believed that younger children had a harder time with this question. However, all of the Pre-K children in classroom E got this question correct.
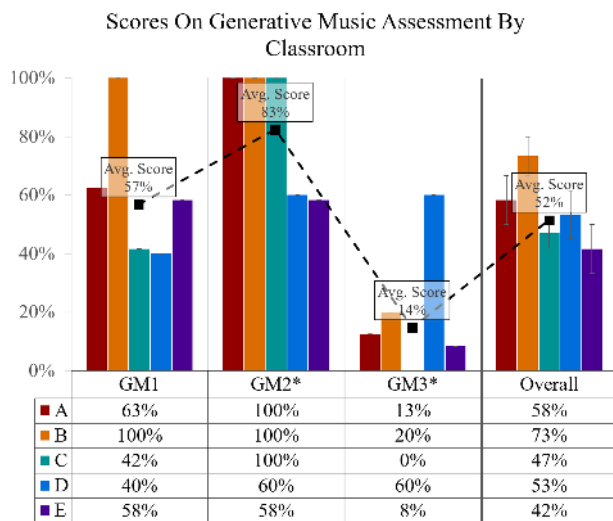
Figure 7: Student's performance on the generative music assessment by classroom. The line graph shows the average across all classrooms.

Therefore, the difference in performance that seemed to be caused by age was really driven by the fact that most of the children in classroom A chose the incorrect answer "banana." One child in this classroom rationalized that tomatoes and bananas both had the seeds on the inside, not on the outside like the strawberry, making them more similar. This suggests that the low score may have been caused by the classroom learning about fruits and seeds at some point earlier in the year rather than an inability to understand the AI concept.

Finally, most classrooms only seemed to understand pieces of the Generative Music activity, except for classroom D where almost half of the students answered every question correctly. The main difference between classroom D and others was that groups in classroom D recorded and played their own songs for the robot while other classrooms only used pre-programmed songs. Recording a new song may have helped students pay more attention to the notes that they played in comparison to the ones the robot played back.

Discussion

Through building, training, and programming a social robot, children encountered three AI concepts and were then able to demonstrate their comprehension on AI assessments for each activity. We found that age sometimes impacted children's understanding, particularly when there were multiple reasoning steps like in the knowledge-based systems activity. We also saw that when children had more time and guidance to thoroughly explore activities, they performed better on the assessments.

Beyond demonstrating how AI concepts can be made accessible to young children, this work also includes key design considerations for future AI curricula for other non-programmers with limited robotics experience.

Connect to Students. In our activities, children deconstructed their own methods of solving problems into simple ideas to communicate to the robot. Finally, they observed how simple ideas merged into intelligent behaviors in the robot's mind. Thus, through straightforward and relatable analogies, non-programmers could make connections to the robot's mind and understand the AI algorithms. Future work should look towards how to help teachers make effective connections for students, and how to design a curriculum flexible enough to meet the needs of different students.

Open the Black Box. Children learned by interacting with a robot that *explained* its reasoning. Children who played more games against the robot had more encounters with the robot's reasoning and could better understand the concepts. Students learn better when they have a chance to ask the system "Why did you do that?"

Interactive Feedback. Many researchers have found that young children understand programming concepts best with lots of sensory feedback (Leidl, Bers, and Mihm 2017; Sullivan 2016). We chose to start with these three AI examples because they have fast, transparent feedback loops. However, this is not the case with all AI concepts, and we need to further explore how to design activities around less intuitive ideas.

Conclusion

This paper discusses the design, development, and assessment of a novel computational toolkit and curriculum that uses a programmable, social robot to guide young children's exploration of AI concepts. After just 15 minutes of interaction, children demonstrated comprehension of various AI concepts. We found that children's ability to understand AI concepts correlated positively with the extent to which they were able to explore the AI concepts through interacting with the activities. Further work is needed to design an effective AI curriculum that covers more material and can be adapted to other contexts, such as classrooms with older, non-programming students and non-expert teachers. We believe that there are hands-on ways to make other AI concepts (e.g., planning, perception, reasoning, and deep learning) accessible to young children. We hope that this will empower children with AI literacy, inspire them to create projects using AI technology, and augment children's reasoning metacognitive skills (thinking about thinking).

Acknowledgements

References

Belpaeme, T.; Kennedy, J.; Ramachandran, A.; Scassellati, B.; and Tanaka, F. 2018. Social robots for education: A review. *Science Robotics* 3(21):eaat5954.

Bers, M. U. 2017. *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.

Burgsteiner, H.; Kandlhofer, M.; and Steinbauer, G. 2016. Irobot: Teaching the basics of artificial intelligence in high schools. In *AAAI*, 4126–4127.

Chen, S. 2017. How to explain AI to kids - the eliza effect.

Druga, S.; Williams, R.; Breazeal, C.; and Resnick, M. 2017. Hey google is it ok if i eat you?: Initial explorations in child-agent interaction. In *Proceedings of the 2017 Conference on Interaction Design and Children*, 595–600. ACM.

Druga, S.; Williams, R.; Park, H. W.; and Breazeal, C. 2018. How smart are the smart toys?: children and parents' agent interaction and intelligence attribution. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, 231–240. ACM.

Druga, S. 2018. Cognimates.

Elkin, M.; Sullivan, A.; and Bers, M. U. 2016. Programming with the kibo robotics kit in preschool classrooms. *Computers in the Schools* 33(3):169–186.

Flannery, L. P.; Silverman, B.; Kazakoff, E. R.; Bers, M. U.; Bontá, P.; and Resnick, M. 2013. Designing scratchjr: support for early childhood learning through computer programming. In *Proceedings of the 12th International Conference on Interaction Design and Children*, 1–10. ACM.

Gordon, G.; Breazeal, C.; and Engel, S. 2015. Can children catch curiosity from a social robot? In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 91–98. ACM.

Imberman, S. P. 2003. Teaching neural networks using lego handy board robots in an artificial intelligence course. *ACM SIGCSE Bulletin* 35(1):312–316.

Kahn, K., and Winters, N. 2018. AI programming by children.

Klassner, F., and McNally, M. 2007. Demonstrating the capabilities of mindstorms nxt for the AI curriculum. In *American Association for Artificial Intelligence*.

Koski, M.-I.; Kurhila, J.; and Pasanen, T. A. 2008. Why using robots to teach computer science can be successful theoretical reflection to andragogy and minimalism. In *Proceedings of the 8th International Conference on Computing Education Research*, 32–40. ACM.

Kumar, A.; Kumar, D.; and Russell, I. 2006. Non-traditional projects in the undergraduate ai course. In *ACM SIGCSE Bulletin*, volume 38, 479–480. ACM.

Kumar, A. N. 2004. Three years of using robots in an artificial intelligence course: lessons learned. *Journal on Educational Resources in Computing (JERIC)* 4(3):2.

Lane, D. 2018. Machine learning for kids.

Leidl, K. D.; Bers, M. U.; and Mihm, C. 2017. Programming with scratchjr: a review of the first year of user analytics. *Siu-cheung KONG The Education University of Hong Kong, Hong Kong* 116.

Martin, F. G. 2007. Real robots don't drive straight. In *AAAI*

Spring Symposium: Semantic Scientific Knowledge Integration, 90–94.

McReynolds, E.; Hubbard, S.; Lau, T.; Saraf, A.; Cakmak, M.; and Roesner, F. 2017. Toys that listen: A study of parents, children, and internet-connected toys. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 5197–5207. ACM.

Milford, M. 2018. What is artificial intelligence.

Papert, S., and Solomon, C. 1971. Twenty things to do with a computer. *Cambridge, MA*.

Papert, S. 1980. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Perlman, R. 1974. Tortis (toddler's own recursive turtle interpreter system).

Resnick, M. 2017. *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*. MIT Press.

Severson, R. L., and Carlson, S. M. 2010. Behaving as or behaving as if? childrens conceptions of personified robots and the emergence of a new ontological category. *Neural Networks* 23(8-9):1099–1103.

Slavin, T. 2016. What is artificial intelligence.

Sullivan, A.; Elkin, M.; and Bers, M. U. 2015. Kibo robot demo: engaging young children in programming and engineering. In *Proceedings of the 14th international conference on interaction design and children*, 418–421. ACM.

Sullivan, A. A. 2016. *Breaking the STEM Stereotype: Investigating the Use of Robotics to Change Young Childrens Gender Stereotypes About Technology and Engineering*. Ph.D. Dissertation. Tufts University.

Talaga, P., and Oh, J. C. 2009. Combining aima and lego mindstorms in an artificial intelligence course to build real world robots. *Journal of Computing Sciences in Colleges* 24(3):56–64.

Turkle, S. 1984. *The Second Self: Computers and the Human Spirit*. New York, NY, USA: Simon & Schuster, Inc.

van Duuren, M., and Scaife, M. 1996. "Because a robot's brain hasn't got a brain, it just controls itself": Children's attributions of brain related behaviour to intelligent artefacts. *European Journal of Psychology of Education* 11(4):365.

Williams, R.; Machado, C. V.; Druga, S.; Breazeal, C.; and Maes, P. 2018. My doll says it's ok: a study of children's conformity to a talking doll. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, 625–631. ACM.