Creating Your Own Conversational Artificial Intelligence Agents Using Convo, a Conversational Programming System

by

Jessica Zhu

S.B., Massachusetts Institute of Technology (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

© Massachusetts Institute of Technology 2021. All rights reserved.

Author . . . .
Department of Electrical Engineering and Computer Science
May 14, 2021

Certified by . . . .
Harold Abelson
Class of 1922 Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Creating Your Own Conversational Artificial Intelligence Agents Using Convo, a Conversational Programming System

by

Jessica Zhu

Submitted to the Department of Electrical Engineering and Computer Science
on May 14, 2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Smart assistants like Amazon's Alexa, Apple's Siri, and Google's Google Home have become commonplace in many people's lives, appearing in their phones and homes. Despite their ubiquity, these conversational AI agents still largely remain a mystery to many, in terms of how they work and what they can do.

To lower the barrier to entry to understanding and creating these conversational AI agents for young students, I expanded on CONVO, a conversational programming agent that can respond to both voice and text inputs. I created a simple and intuitive user interface for students to input training data, create programs, and test the conversational AI agents they create. To further assist anyone in using CONVO, I also produced a couple of video and PDF tutorials that outline how to use CONVO. Additionally, I also developed a curriculum to teach students about key concepts in AI and conversational AI in particular, including the Big 5 AI Ideas and the difference between constrained and unconstrained natural language models.

I ran a 3-day workshop in partnership with MIT's eSPARK program, with a total of 15 participating middle school students. Through the data collected from the pre- and post-workshop surveys as well as a mid-workshop brainstorming session, I was able to explore how students' perceptions, understanding, literacy, and visions of conversational AI agents changed. During the workshop, students were able to create their own conversational AI agents. I also found that after the workshop, students tended to think that conversational AI agents were less intelligent than originally perceived, gained confidence in their abilities to build these agents, and learned some key technical concepts about conversational AI as a whole. Based on these results, I am optimistic about CONVO's ability to teach and empower students to develop conversational AI agents in an intuitive way.

Thesis Supervisor: Harold Abelson
Title: Class of 1922 Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank Jessica Van Brummelen, without whom this thesis would not exist. Your constant guidance and patience kept me on track and gave me the support I needed to get started on a field in which I had no prior experience. Thank you for always answering all of my questions, providing me with valuable resources, and co-teaching with me at my workshops. I've been able to learn so much from your experience and insights, and it's been a privilege working with you. I wish you the best on your doctoral journey!

I would also like to thank my advisor, Hal Abelson, for providing me with direction and a vision for this thesis. Your passion for teaching inspires me to work harder, and I'm incredibly grateful for the opportunity to not only work with you for my research but also for 6.UAT.

Many thanks to the entire MIT App Inventor team, especially Jeff Schiller for helping me put out numerous fires and responding to all of my panicked I'm-not-sure-what-I'm-doing emails. To my fellow peers, Murielle and Joy, thank you for providing me with a sense of camaraderie, as we embarked on our separate but interconnected user studies together. To Daniel, Liz, Nicole, and Lucy, thank you for volunteering to help me test Convo and run my workshops. To Selim, thank you for always being so kind and patient and willing to help. To the rest of the App Inventor team, even though we've never met in person, thank you for providing a place to provide support, share ideas, and celebrate victories.

Finally, I would like to thank my friends and family for keeping me company and sane during this past year. I'm eternally grateful for all of the countless video calls, grocery store trips, and socially distanced hikes that were a welcome break from the stresses and uncertainty of life.

Contents

1 Introduction 15

1.1 Key Terms and Definitions 16

1.2 Scenarios: The Vision for Convo 17

1.2.1 Susan and Janet's Adventure Story 17

1.2.2 Billy's Calculator 21

2 Background 27

2.1 Convo 27

2.2 Constrained vs. Unconstrained Natural Language Models 28

2.3 Related Work[1] 29

3 Technical Implementation 33

3.1 Rasa 33

3.1.1 Integrating Convo with Rasa 36

3.1.2 HTTP API 36

3.2 Creating a New User Interface 37

3.2.1 Training Intents and Entities 39

3.2.2 Connecting Intents to Procedures 41

3.2.3 Talking to Convo 43

3.3 Technical Challenges 44

3.3.1 Resource Constraints 44

3.4 Summary of Technical Contributions 45

---

[1] This section was adapted from [44].

4 Workshop Curriculum 47

5 Evaluation 53
5.1 Research Question 1 54
5.1.1 Pre-Workshop Student Ideas on Conversational AI Agents . . 55
5.1.2 Post-Workshop Student Ideas on Conversational AI Agents . . 57
5.2 Research Question 2 60
5.2.1 Pre-Workshop Student Attitudes on Conversational AI Agents 60
5.2.2 Post-Workshop Student Attitudes on Conversational AI Agents 61
5.3 Research Question 3 63
5.3.1 Pre-Workshop Student Literacy of Conversational AI Agents . 63
5.3.2 Post-Workshop Student Literacy of Conversational AI Agents 65
5.4 Research Question 4 68

6 Conclusion 71
6.1 Key Findings[2] 71
6.2 Future Work 72

A Figures 75

B Workshop Materials 79
B.1 Pre-Workshop Survey 79
B.2 Post-Workshop Survey 85
B.3 CONVO Tutorial 1 91
B.4 CONVO Tutorial 2 96
B.5 Kahoot Quiz 102

---

[2] This section was adapted from [44].

List of Figures

1-1 Susan fills out the intent card as shown to teach CONVO how to recognize when Janet wants to start the adventure story. 18

1-2 Susan fills out three more intent cards that map to the three actions Janet must take to complete the adventure story. Each card contains the name of the intent CONVO should trigger when Janet says something similar to the corresponding given intent phrases. 18

1-3 The left card is the raw text that Billy enters into the intent card on CONVO's *Create Intents* page to indicate to CONVO how to recognize when Billy wants a math problem solved. The right card is how the text is transformed after Billy completes the highlighting of the entities. Note that a new card is not generated, the card on the left simply transforms into the card on the right after the entities are labelled. . 23

2-1 The previous version of CONVO. It was able to understand a very constrained set of user inputs. 28

3-1 The inner pipeline components that allow Rasa to perform intent and entity recognition using relatively little training data. 35

3-2 The homepage of the CONVO website. It contains information about conversational AI and how they work, guides to each of the three other pages on the site, and links to the video tutorials on how to use CONVO. 38

3-3 The *Create Intents* page of the CONVO website. Here, students can enter in any training data they'd like for CONVO to learn about. The Group ID is specific to the workshop, and can be left blank. Once all the data has been entered, students may click the *Train!* button. . . 38

3-4 The *Program* page of the CONVO website. Students create their own programs and connect them to the intents they trained on the *Create Intents* page here. There is a sidebar on the left side of the page with a list of phrases that CONVO can understand. 39

3-5 The *Talk to Convo* page of the CONVO website. Here, students may speak or type to CONVO in a conversational manner, without needing to follow any preset list of valid commands. CONVO will do its best to figure out the intent of what was said, and run any corresponding programs in response. 39

3-6 Billy uses the highlighting feature to indicate to CONVO about the entities that represent the *first* and *second* numbers to add within each intent phrase. 40

3-7 The sidebar of the *Program* page with things someone can say to CONVO. The bottom command contains an example of how a student can indicate to CONVO that it wants to connect an intent to a procedure. 42

5-1 Students selected from a checklist of common use cases how they envisioned conversational AI agents being used in the pre-workshop survey. 57

5-2 Averages of student perceptions of conversational AI agents based on a 7-point Likert scale in the pre- and post-workshop surveys. 58

5-3 Students selected from a checklist of common use cases how they envisioned conversational AI agents being used in the post-workshop survey. 59

5-4 Averages of student attitudes towards conversational AI agents based on a 7-point Likert scale in the pre- and post-workshop surveys. . . . 62

5-5 A graphic of the words used by students to describe AI in the pre-workshop survey. Larger words correspond to a larger word frequency. Figure generated from `https://www.wordclouds.com/`. 64

5-6 An example of a filled out idea template where the idea was to create a calculator that could sum together two numbers. Each student was asked to come up with at least one idea. (a) First of a two page slide template asking students for the purpose and goals of their idea, as well as the intents and entities they wished Convo to understand. (b) Second of a two page slide template that asked students for some example conversations they envisioned a user having with Convo after learning about the training data from (a). 70

A-1 A student came up with the idea named *Time*, which allows the user to ask Convo about the time at a certain location. In this example, the student was able to correctly identify that New York, Florida, and my location were possible entities in the intent phrases a user might speak to Convo. Based on the entities, Convo would respond with the current time in that specific location. 76

A-2 A student came up with an idea called *Convo Feels Your Emotion*, which allows the user to tell Convo how they are feeling, and Convo responds back accordingly. In this slide, the student included the syntax in which to indicate the presence of entities in an intent phrase, and also provided the responses Convo would give for different emotions. 77

A-3 A student came up with a way for Convo to respond to a user asking about Harry Potter characters. This idea was aptly named, *Harry Potter*. 77

List of Tables

4.1 Overview of workshop schedule and curriculum. Each session was 2 hours long, spaced 1 week apart. 48

5.1 Students rank how much they agree or disagree with statements involving different descriptors of conversational AI agents based on a 7-point Likert scale in the pre-workshop survey. 55

5.2 Students rank how much they agree or disagree with statements involving different descriptors of conversational AI agents based on a 7-point Likert scale in the post-workshop survey. 58

5.3 Student attitudes based on a 7-point Likert scale towards conversational AI agents in the pre-workshop survey. 61

5.4 Student attitudes based on a 7-point Likert scale towards conversational AI agents in the post-workshop survey. 62

5.5 Students were asked to select all of the true statements from a series of statements about CONVO. These statements targeted specific concepts in conversational AI that we wanted students to learn. 67

Chapter 1

Introduction

In less than a decade, 'smart assistants' like Alexa, Siri, and Google Home have quickly become household names [15, 23]. These conversational AI agents are able to understand and respond to human speech and text by learning and training on large amounts of data. As more people around the world slowly integrate more of these agents into their daily routines, it is only natural that we should educate others, especially young students, on the capabilities and limitations of such technologies.

In this thesis, I expanded on Convo, a conversational programming agent that can respond to both voice and text inputs, created by Kevin Weng [41]. This previous version of Convo was effectively a conversational AI agent with one specific task: allowing students to create simple programs through speech or text [41]. My new version of Convo allows students to not only create programs, but also to create their own conversational AI agents that can perform various tasks, such as saying the weather or telling a fun fact. To accomplish this, I implemented a simple and intuitive user interface for inputting data, developed a workflow to train and create the agents, and came up with a curriculum to teach students how to use Convo.

To determine the effects that interacting with Convo has on students, I ran a 3-day workshop in the form of a class with 15 middle school students participating. The main goals of this workshop were to explore how students' perceptions, understanding, and literacy of conversational AI agents would change after interacting and building one themselves. I also wanted to gain insight into the ideas and visions that students had for the future of these agents. The curriculum I developed for my workshop can be found in Chapter 4, and my full results (including pre- and post-workshop survey data and student project ideas) are analyzed in Chapter 5.

Overall, students were successfully able to create their own conversational AI agents during the workshop. I also found that, after the workshop, students tended to think that conversational AI agents were less intelligent than originally perceived, gained confidence in their abilities to build these agents, and learned some key technical concepts about conversational AI as a whole. Based on these results, I am optimistic about CONVO's ability to teach and empower students to develop conversational AI agents in an intuitive way.

1.1 Key Terms and Definitions

To understand this thesis, there are a couple of key terms that must be defined first. I will often refer to these terms in the upcoming sections.

- **Intent:** Something you want a conversational AI agent (CONVO) to be able to recognize. Example intents: *greeting someone, asking a question.*
- **Intent Phrase:** A word or phrase that is an example of what someone might say to trigger that intent. Example intent phrases for the intent greeting someone: *hello, hey there, hi.*
- **Entity:** A specific piece of data you want to extract from an intent. Example entities: *name, time, city.*
- **Entity Phrase:** A word or phrase that represents the entity you want to extract. Example entity phrases for the entity city in the intent asking for the weather: *Boston, San Francisco, New York City.*
- **Training Data**: All of the intents, intent phrases, entities, and entity phrases.
- **Procedure:** A set of actions you wish the conversational AI agent (CONVO) to perform. In a programming setting, this is equivalent to a *function* or *method.*

1.2 Scenarios: The Vision for CONVO

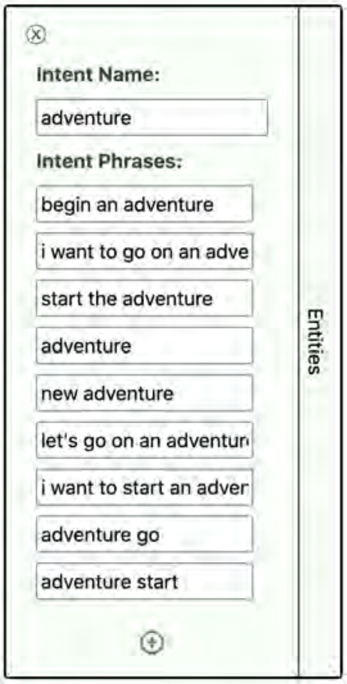
To showcase the vision for CONVO, I present two scenarios, one starring two sisters, Susan and Janet, and one starring Billy, a clever student. These scenarios are fictional, but they are a way to demonstrate the abilities of CONVO in different settings and use cases. I will be using some terminology in the descriptions of these scenarios that is defined previously in Section 1.1.

1.2.1 Susan and Janet's Adventure Story

In the first scenario, Susan wants to make an adventure story game for her younger sister Janet to play when she's bored. Susan navigates to the CONVO web app and brainstorms ideas for her story. Her goal is for Janet to figure out how to escape a locked room by performing certain actions.

To start, Susan tells CONVO how to recognize when Janet wants to begin the adventure by typing in **intent phrases** and an **intent** name into one of the cards on CONVO's *Create Intents* page. Susan tells CONVO about some ways that Janet might want to indicate interest in starting the adventure story, including: *begin an adventure* and *I want to go to on an adventure.* CONVO is then able to take these example **intent phrases** and extrapolate such that even if Janet doesn't say something that matches exactly to a given **intent phrase**, as long as it is similar enough, CONVO will be able to understand it properly. The full text that Susan provides can be found in Figure 1-1.

Susan also needs to create a unique intent for each action she wants Janet to take. One such action will be flipping on the light switch. Susan tells CONVO that **intent phrases** like *flip on the lights, lights on, turn on lights, turn on the light switch*, and *flip the light switch* should trigger the **intent** named *turn on lights.* She types these phrases in a new card on CONVO's *Create Intents* page. Since there are two more actions (picking up the key and unlocking the door) that Janet must take before completing the adventure story, Susan repeats the same process for those actions as well. The completed cards can be found in Figure 1-2.



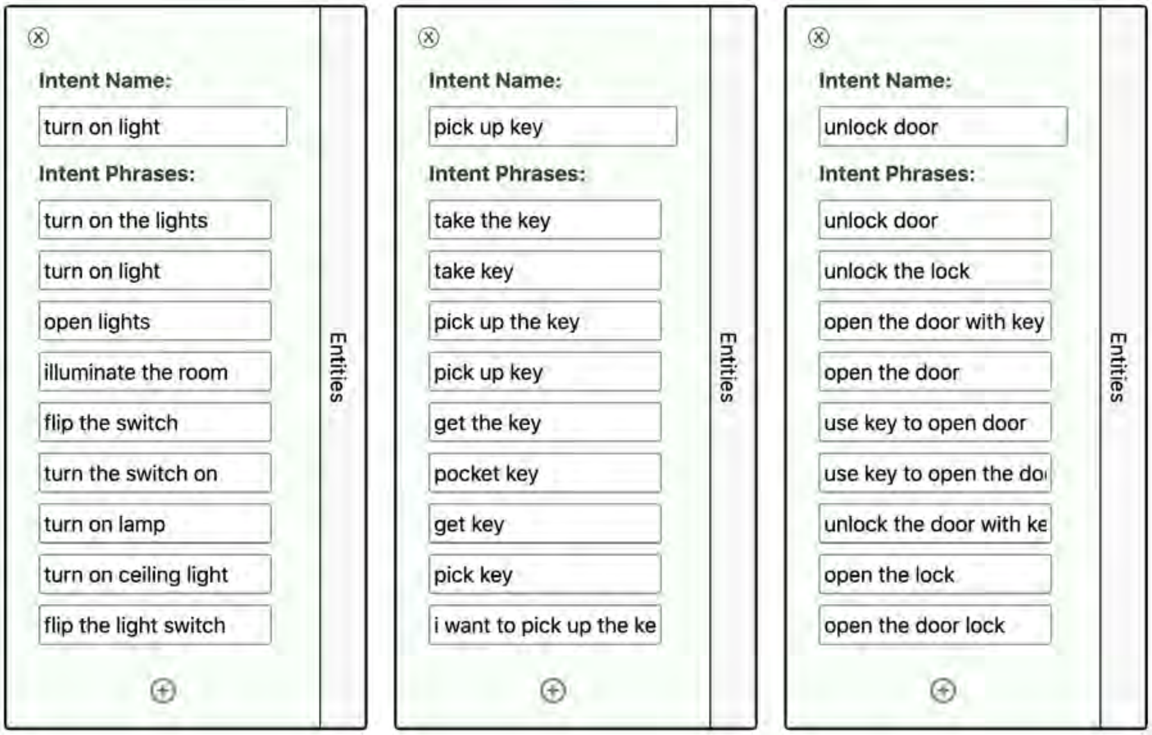
Intent Name:

adventure

Intent Phrases:

begin an adventure
i want to go on an adve
start the adventure
adventure
new adventure
let's go on an adventur
i want to start an adver
adventure go
adventure start

Figure 1-1: Susan fills out the intent card as shown to teach Convo how to recognize when Janet wants to start the adventure story.



Intent Name:

turn on light

Intent Phrases:

turn on the lights
turn on light
open lights
illuminate the room
flip the switch
turn the switch on
turn on lamp
turn on ceiling light
flip the light switch

Intent Name:

pick up key

Intent Phrases:

take the key
take key
pick up the key
pick up key
get the key
pocket key
get key
pick key
i want to pick up the ke

Intent Name:

unlock door

Intent Phrases:

unlock door
unlock the lock
open the door with key
open the door
use key to open door
use key to open the do
unlock the door with ke
open the lock
open the door lock

Figure 1-2: Susan fills out three more intent cards that map to the three actions Janet must take to complete the adventure story. Each card contains the name of the intent Convo should trigger when Janet says something similar to the corresponding given intent phrases.

Once all the **training data** has been entered, Susan clicks the *Train!* button on the *Create Intents* page to train CONVO. The next step is to connect the four **intents** to four **procedures**. Starting with the first **intent**, Susan wants CONVO to begin the story when Janet triggers the *adventure* **intent**. Susan does this by typing in the following command exactly on CONVO's *Program* page.

> CONVO: Hi, I'm Convo! What would you like to do? To get started, you can create a procedure by saying "Create a procedure". If you want to run a procedure, say "Run" and the name of the procedure.
>
> SUSAN: create a procedure
>
> CONVO: What do you want to call the procedure?
>
> SUSAN: start adventure
>
> CONVO: What do you want to happen in the procedure first? You could make me say something. See the sidebar for more options.
>
> SUSAN: say Here's a short adventure for you! You are currently in a dark room and you must try to escape. What do you do first?
>
> CONVO: Added action to the procedure! Do you want to do anything else?
>
> SUSAN: done
>
> CONVO: I finished creating the procedure. You can say, "run start adventure" to play it.
>
> SUSAN: connect the intent adventure to the procedure start adventure
>
> CONVO: I connected the intent adventure to the procedure start adventure. What do you want to do now?

Susan also connects each of the action **intents** to their corresponding **procedures**. She begins by creating a **procedure** that she names *turn on lights response* that will say the next portion of the story. Similar to the *start adventure* **procedure**, Susan programs Convo as follows.

SUSAN: create a procedure

CONVO: What do you want to call the procedure?

SUSAN: turn on lights response

CONVO: What do you want to happen in the procedure first? You could make me say something. See the sidebar for more options.

SUSAN: say Good job! You found the light switch and turn it on. You see a key on the table. What do you do next?

CONVO: Added action to the procedure! Do you want to do anything else?

SUSAN: done

CONVO: I finished creating the procedure. You can say, "run turn on lights response" to play it.

SUSAN: connect the intent turn on lights to the procedure turn on lights response

CONVO: I connected the intent turn on lights to the procedure turn on lights response. What do you want to do now?

When this **intent** is triggered, Convo runs the **procedure** *turn on lights response* and gives a clue for the next action. Susan repeats these steps for the other two actions she wants Janet to perform to finish setting up the escape-room story experience for Janet.

Janet interacts solely with Convo's *Talk to Convo* page. She can speak or type anything she likes, and if Convo determines that something that she says is similar enough to an **intent** that it recognizes, Convo triggers that **intent**. As an example, even though the specific phrase *turn on the lights* might not have been used as an **intent phrase** to train the **intent** *turn on lights*, it is similar enough that Convo triggers that **intent**. Now, whenever Janet says the correct action, CONVO responds with the pre-programmed message Susan told it to say. If at any time Janet tries to perform an incorrect action, or an action that CONVO doesn't recognize, CONVO responds with the default answer of: *"I didn't understand what you were saying. Please try again."*. On Janet's end, her conversation (spoken or typed) with CONVO might look something like this:

JANET: Hey CONVO, tell me an adventure story.

CONVO: Here's a short adventure for you! You are currently in a dark room and you must try to escape. What do you do first?

JANET: Scream for help.

CONVO: I didn't understand what you were saying. Please try again.

JANET: Turn on the lights.

CONVO: Good job! You found the light switch and turn it on. You see a key on the table. What do you do next?

JANET: Pick up the key.

CONVO: You picked up the key. The door to the room is in front of you. What next?

JANET: Unlock the door.

CONVO: Congrats! You used the key to unlock the door and you're now free.

1.2.2 Billy's Calculator

For the second scenario, Billy is a middle school student and doesn't like mental math. His calculator is also broken, and he wants to be able to solve math problems by just asking for the answer.

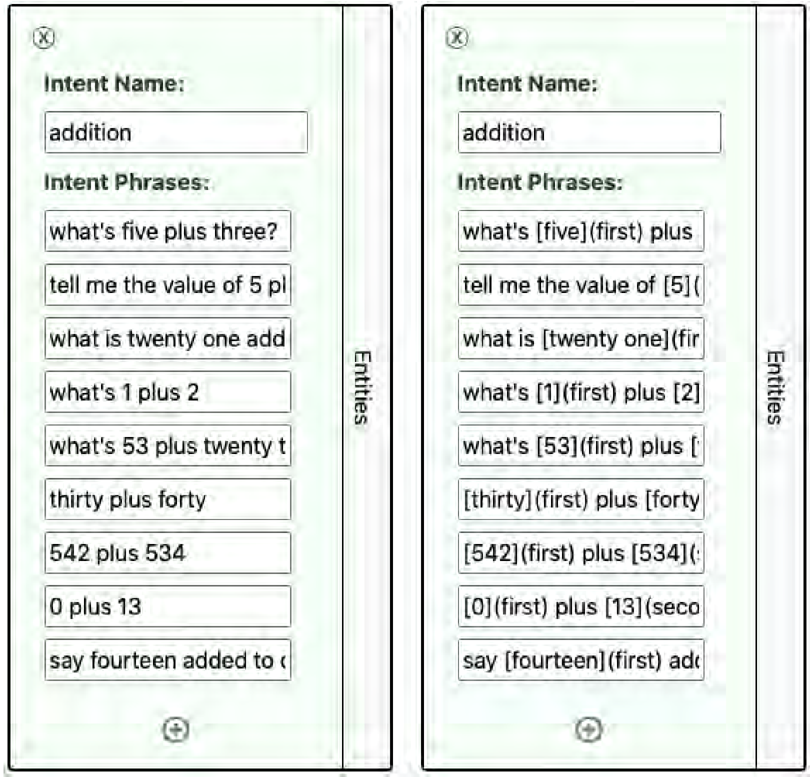
Billy goes to CONVO's website and creates an agent that understands just one **intent**: *addition*. Since there are an infinite number of ways Billy might ask CONVO to sum two numbers together, Billy can't provide Convo with a list of exact questions to look out for. Instead, he tells Convo about some ways he might ask for the sum of two numbers and Convo is able to generalize from these examples. Billy enters these examples in as **intent phrases** on Convo's *Create Intents* page. These include: *what's five plus three?, tell me the value of 5 plus 3, what is twenty-one added to 16?, what's 1 plus 2*, and *what's 53 plus twenty-two?*. However, since Billy wants Convo to be able to extract the specific numbers he is trying to sum together, he also tells Convo to look for two **entities**, the two numbers he wants to sum. He names the first entity *first* and the second entity *second*. He then uses Convo's highlighting feature to highlight the words that correspond to these entities. Once he's finished, his new **intent phrases** are: *what's [five](first) plus [three](second)?, tell me the value of [5](first) plus [3](second), what is [twenty-one](first) added to [16](second)?, what's [1](first) plus [2](second)*, and *what's [53](first) plus [twenty-two](second)?*. The full text Billy enters before and after highlighting can be found in Figure 1-3.

After training Convo on all of the **training data**, Billy moves to Convo's *Program* page and connects his **intent** to a **procedure** named *sum two*. In this **procedure**, Billy creates a new variable called *sum* that contains the summation of the values of the *first* and *second* entity values. The **procedure** ends with Convo speaking the final resulting value of the variable *sum*. His conversation to perform these tasks is below.

CONVO: Hi, I'm Convo! What would you like to do? To get started, you can create a procedure by saying "Create a procedure". If you want to run a procedure, say "Run" and the name of the procedure.

BILLY: connect the intent summing two numbers together to the procedure sum two

CONVO: I connected the intent summing two numbers together to the procedure sum two. I created a variable for the entity first that is set to a value of 0 at step 1. This default value will



addition

what's five plus three?
tell me the value of 5 pl
what is twenty one add
what's 1 plus 2
what's 53 plus twenty t
thirty plus forty
542 plus 534
0 plus 13
say fourteen added to

addition

what's [five](first) plus
tell me the value of [5](
what is [twenty one](fir
what's [1](first) plus [2]
what's [53](first) plus [
[thirty](first) plus [forty
[542](first) plus [534](
[0](first) plus [13](seco
say [fourteen](first) ad

Figure 1-3: The left card is the raw text that Billy enters into the intent card on Convo's *Create Intents* page to indicate to Convo how to recognize when Billy wants a math problem solved. The right card is how the text is transformed after Billy completes the highlighting of the entities. Note that a new card is not generated, the card on the left simply transforms into the card on the right after the entities are labelled.

be overridden if I detect a different value for this entity when I recognize this intent. I created a variable for the entity second that is set to a value of 0 at step 2. This default value will be overridden if I detect a different value for this entity when I recognize this intent. The procedure, sum two, did not exist previously, so I created a new procedure. What do you want to do now?

BILLY: edit sum two

CONVO: I am on the first step of the procedure, where I am creating a variable called first and setting its value to 0. What do you want to do? You can say 'done' if you are finished editing.

BILLY: create a variable called sum

CONVO: What should be the initial value?

BILLY: 0

CONVO: I added the action after step 1. I am at step 2 in the procedure where I am creating a variable called sum and setting its value to 0.

BILLY: next step

CONVO: I am on step 3 which is the last step of the procedure, where I am creating a variable called second and setting its value to 0. What do you want to do? You can say 'done' if you are finished editing.

BILLY: add the value of the variable first to variable sum

CONVO: I added the action after step 3. I am at step 4 in the procedure where I am adding the value of variable first to variable sum.

BILLY: add the value of the variable second to variable sum

CONVO: I added the action after step 4. I am at step 5 in the procedure where I am adding the value of variable second to variable sum.

BILLY: say the value of the variable sum

CONVO: I added the action after step 5. I am at step 6 in the procedure where I am saying the value of the variable sum.

BILLY: done

CONVO: Done with editing procedure sum two.

Billy has now created his own personalized math problem-solving agent. When on CONVO's *Talk to Convo* page, Billy can simply ask CONVO some math questions out loud and get the results spoken back to him without having to get up from the couch or even move a finger.

BILLY: Hey CONVO, what's forty plus sixteen?

CONVO: 56.

BILLY: What about 32 plus 59?

CONVO: 91.

BILLY: Tell me what 93 plus one hundred and eight is.

CONVO: 201.

Of course, while there are an infinite number of possibilities for a conversational AI agent created using CONVO, I will be using these two scenarios throughout this thesis to demonstrate the vision for CONVO and how I achieved it. First, I discuss related work and some key concepts to understand in Chapter 2. Then, I explain all the technical details and challenges behind the changes I made to CONVO in Chapter 3. Chapter 4 contains all of the curriculum and logistics details of my workshop, and Chapter 5 analyzes and derives insights from all of the data I obtained. Finally, I list some key takeaways and suggestions for future work in Chapter 6.

Chapter 2

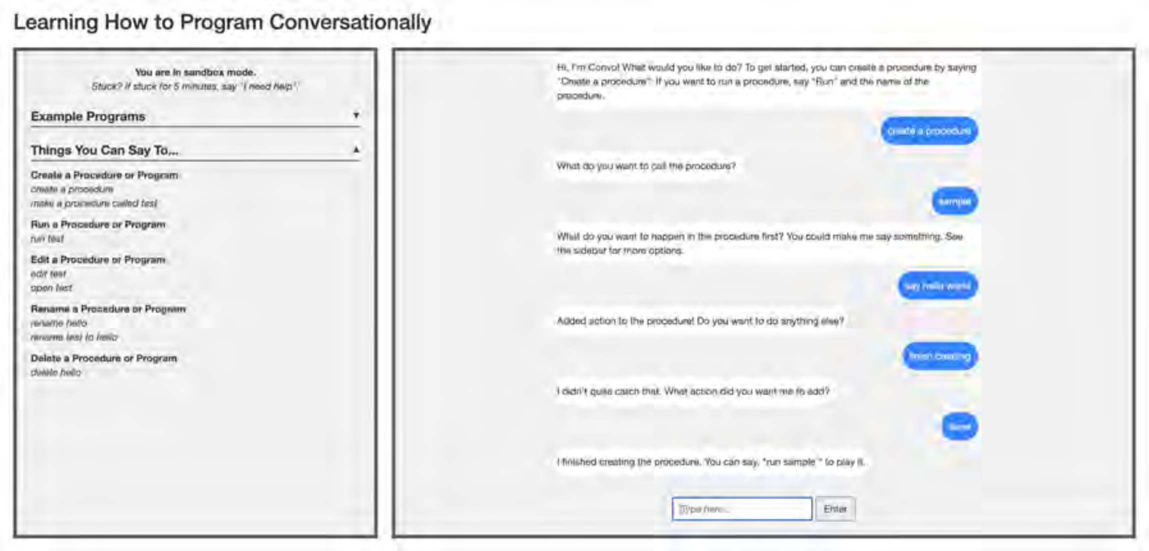
Background

In this thesis, I improved upon an existing system created by Kevin Weng, discussed at length in his master's thesis [41]. Named CONVO, the previous iteration of the system was a conversational programming agent that aimed to lower the barrier to entry for students to learn how to program [41]. In this chapter, I give an overview of some key conversational AI concepts I utilize in CONVO, discuss the previous iteration of CONVO more in detail, and outline similar work to that which I've done for CONVO.

2.1 Convo

Before I discuss the improvements I made to CONVO, I give a brief overview of the previous system and its capabilities and limitations. As mentioned before, CONVO is a voice-first conversational programming agent that uses natural language to teach and allow users to create their own programs [41]. It was developed and deployed in 2019 by a small team of researchers as part of the MIT App Inventor team, and its user interface is shown in Fig. 2-1. The goal of their work was to lower the barrier to entry to programming by allowing students to create programs by simply communicating with a conversational AI agent. They hoped to empower students to use NL to create an impact by solving problems in their lives and communities [38].

CONVO was based off of Grice's "conversational maxims", three conversational and voice-first design principles: *conciseness*, *correctness*, and *relevancy* [19]. In


Learning How to Program Conversationally

You are in sandbox mode.

Stuck? If stuck for 5 minutes, say "I need help".

Things You Can Say To...

Create a Procedure or Program
create a procedure
make a procedure called test

Run a Procedure or Program
run test

Edit a Procedure or Program
edit test
open test

Rename a Procedure or Program
rename hello
rename test to hello

Delete a Procedure or Program
delete hello

Hi, I'm Convo! What would you like to do? To get started, you can create a procedure by saying "Create a procedure". If you want to run a procedure, say "Run" and the name of the procedure.

create a procedure

What do you want to call the procedure?

sample

What do you want to happen in the procedure first? You could make me say something. See the sidebar for more options.

say hello world

Added action to the procedure! Do you want to do anything else?

finish creating

I didn't quite catch that. What action did you want me to add?

done

I finished creating the procedure. You can say, "run sample " to play it.

Figure 2-1: The previous version of Convo. It was able to understand a very constrained set of user inputs.

addition to these maxims, Convo also followed the principle of natural language understanding flexibility, in which industry leaders like Google and Amazon state that conversational AI agents should be able to understand how humans talk, not how we read or write (should understand synonyms, over-answering, and subtextual meaning) [21, 18, 38].

While honoring the principles Convo was built on, I expanded on abilities of Convo in this proposed thesis. I transformed Convo into a conversational AI agent that could be used to create other conversational AI agents.

2.2 Constrained vs. Unconstrained Natural Language Models

One of the key conversational AI concepts covered in this thesis is the difference between constrained and unconstrained NL models in the context of a conversational AI agent. An agent using a constrained NL model can only understand a constrained set of user commands. To communicate, a user has to learn how the computer understands language, and the user must communicate via a set of narrowly defined words and patterns. On the technical side, the previous version of CONVO had only supported constrained NL models, which meant that users had to input specific phrases to let the agent know what actions they would like performed. These 'instruction manuals' appeared in an accessible sidebar panel on CONVO's user interface. While this reduced potential ambiguity in CONVO's understanding of the user's intents, this model limited the user's natural speech patterns as they had to communicate using a set of fixed phrases and commands.

In contrast, an agent that supports unconstrained NL is able to understand a much wider array of user utterances. Instead of trying to match a user utterance to a specific pattern, the computer uses an ML model to try to recognize an *intent* instead. In this scenario, more of the burden of communicating is placed on the computer itself, so the user has less of a learning curve and can communicate more naturally.

Despite the distinctions, it is important to note that the constrained and unconstrained classifications of a NL model are not binary, but rather, lie on a spectrum. Alan Nichol, one of the co-founders of Rasa, outlines the different levels of conversational AI, ranging from command line apps to adaptive assistants (least to most "human") [32]. To teach students about this idea, I designed my improvements to CONVO around this distinction.

2.3 Related Work[1]

While there has been plenty of interest in conversational AI agents from a consumer standpoint [15, 23], research and investment into AI and conversational AI education is becoming increasingly important [27, 37, 11]. Furthermore, conversational AI technology enables many opportunities, such as hands-free interaction, constant agent availability, and voice-based automation of menial tasks, in many different contexts, including the classroom. For example, researchers have designed agents to help students manage emotions during learning, teach history, and quiz students [30, 29, 22].

---

[1] This section was adapted from [44].

Other agents, like Betty's Brain and Zhorai, are teachable themselves, drawing on the learning-by-teaching paradigm [9, 26, 14]. Still others draw on conversational AI agents' abilities to lower the barrier to entry for people to develop skills, like programming [38, 31, 44].

Despite the need for conversational AI education, and the evident utility conversational AI agents can provide, high-utility conversational AI agent development interfaces, like 'Actions on Google', often have steep learning curves [34, 8, 17]. Furthermore, current low-barrier-to-entry conversational AI agent development interfaces, like 'Alexa Blueprints', generally lack many of the features high-utility interfaces include [34, 7]. These low-barrier-to-entry interfaces are generally not developed to educate people about how conversational AI agents work either. Nonetheless, one conversational AI agent interface with this purpose includes Conversational AI in MIT App Inventor, which has been used in K-12 settings to teach students about AI as they develop conversational AI agents [10]. This interface has been shown to be an effective tool in teaching AI literacy concepts and students to program; however, the programming itself is done through a visual code-block interface, rather than a speech- or conversational-based interface [12]. We posit that by utilizing conversational AI to program (e.g., having students *tell* a system to "create a procedure"), students could learn further about conversational AI agents and how AI works during the development process itself. For instance, students could learn about the suitability of constrained (e.g., regular expression matching) versus unconstrained (e.g., deep learning classification) natural language (NL) systems through using both systems when conversing with the conversational AI agent [20, 16].

Recently developed interfaces that utilize conversational programming include SUGILITE, which is an agent that automates tasks through NL and GUI interactions, and CONVO, which is an agent that develops Python-based programs through NL conversation [25, 38, 39]. SUGILITE allows users to demonstrate a GUI-based task and describe the task in words. The multimodal inputs (GUI and speech interactions) increase SUGILITE's performance and allow users without significant programming experience to automate tasks [25]. CONVO instead uses purely NL input (e.g., speech) to allow users to program conversation-based tasks. For example, a user may program CONVO to play a 20-questions game by conversing with it in NL [38].

Chapter 3

Technical Implementation

In Sections 1.2.1 and 1.2.2, I introduced two scenarios in which Susan, Janet, and Billy create and use their own conversational AI agents for dramatically different purposes. In this chapter, I discuss the details behind the technical implementation of Convo and how I achieved the goal of allowing students to easily create their own conversational AI agents in the context of the scenarios. Specifically, I outline three main contributions: integrating Rasa (described in Section 3.1) with Convo, improving the user interface to allow for training data input, and creating a way to connect intents to procedures.

3.1 Rasa

In order to perform intent and entity recognition in Convo, I utilized Rasa, an open source, machine learning framework that helps automate text and voice-based assistants [4]. Rasa is responsible for the tasks of "learning" about the training data, recognizing user intent, and extracting entities from user input.

In Susan's adventure story scenario, she trained Convo to recognize the intents *adventure, turn on light, pick up key*, and *unlock door*. Likewise with Billy, he trained Convo on the intent of *addition* and the entities *first* and *second*. Behind the scenes, Convo uses Rasa to create a natural language model that understands these intents and entities.

Furthermore, when Janet speaks to CONVO on its *Talk to Convo* page, CONVO uses Rasa to interpret what she says. If Rasa judges her intent to match any known intent that it was trained on previously with a confidence higher than some set threshold (I used 70%), Rasa will let CONVO know and CONVO will run the associated procedure. Otherwise, Rasa lets CONVO know that there was no good match and CONVO returns the predetermined message of "I didn't understand what you were saying. Please try again." The same is true for Billy's case. If Rasa doesn't judge what Billy says to it on the *Talk to Convo* page as close enough to its learned intent of *addition*, it will tell CONVO to respond with the same predetermined message.

Next, I dive deeper into just exactly how Rasa is able to both train a natural language model and use it to parse for intents and entities. There are many ways to approach this, as there are many options for pipeline components to choose from.

For CONVO specifically, I use the pipeline in Figure 3-1 to perform the necessary intent recognition and entity extraction steps. I begin by using BERT, a pre-trained natural language understanding model, which allows us to provide fewer training examples and still achieve robust results [13]. Next, we have the Language Model Tokenizer and Language Model Featurizer, which first converts the user input into a vector and then creates tokens for user messages, responses, and user intents (if present) [3]. With these tokens, we then use the Count Vector Featurizer (based off of `sklearn`'s `CountVectorizer`), which "creates a bag-of-words representation of user messages, intents, and responses" [3]. Finally, we use the Dual Intent Entity Transformer (DIET) Classifier to extract the entities, intents, and intent rankings for the user message [3].

The first step, intent recognition, classifies the user's request (*I'm bored, I want to go on an adventure*) as one of a fixed set of possible intents (e.g. *adventure, turn on light, pick up key, unlock door*). I use the intent classifier `Sklearn`, which relies on the `spaCy` library to represent each word as as a word embedding [42]. This system works based on the assumption that we will only be dealing with the English language, do not support multiple intents per message, and do not have many domain specific terms. The benefits of using this classifier are that we do not have to provide much

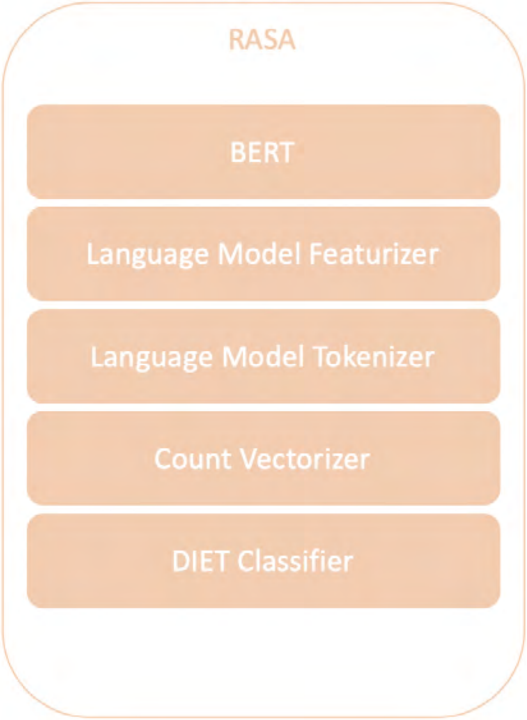


Figure 3-1: The inner pipeline components that allow Rasa to perform intent and entity recognition using relatively little training data.

training data to achieve robust intent predictions, since the word embeddings are already pre-trained [42].

The next step, entity recognition, involves extracting any relevant information from the user's input (e.g. the first and second numbers to add). Rasa developers suggest using different libraries for different types of entities. To understand both words and numbers, I use a combination of the `spaCy` and `Duckling` libraries [43]. The alternative method is to use a component that requires us to train the model from scratch, so we would have to fully annotate all training data ourselves. Although this method requires more time and training data, it is also more flexible in the different entities the NLU model can recognize [43].

All together, Rasa is a powerful conversational AI platform that has the ability to create any conversational AI agent given the right training data [4]. When used in conjunction with CONVO, Rasa and what it can achieve become much more accessible to young students who are eager to learn about conversational AI.

3.1.1 Integrating Convo with Rasa

At a high level, Rasa is simply another server in the network of servers that make up Convo. I added Rasa as a service in the Docker container that runs Convo and mounted it at port 5005 (since its default port, 5000, was already taken by the Flask app) [41]. When starting up Convo, I start the entire Docker container, which runs a command to start up each individual service within it.

3.1.2 HTTP API

To communicate with the Rasa server, I sent requests via its `HTTP API` endpoints [6]. I used Python's `requests` library to make the proper `GET`, `PUT`, and `POST` requests from the Flask backend server [5]. To call a particular endpoint, I simply appended the URL of the server with the corresponding suffix and passed in any relevant data. Below, I outline more details on why I used each specific endpoint.

- `/status`: A `GET` request that checks on the status of the Rasa server. If the server is up and running properly, it returns a `200` request as well as the name of the currently loaded model and the version number. I didn't use this endpoint explicitly in my code, but found myself referencing it often during the testing process of Convo.

- `/model/train`: A `POST` request that trains a new Rasa model. It takes in the specific pipeline for the Rasa model to train on, as well as all of the training data (intents, entities) that should be learned. The newly trained model is saved as a file to a local `/models` directory. I called this endpoint whenever a user clicked the *Train!* button on Convo's *Create Intents* page.

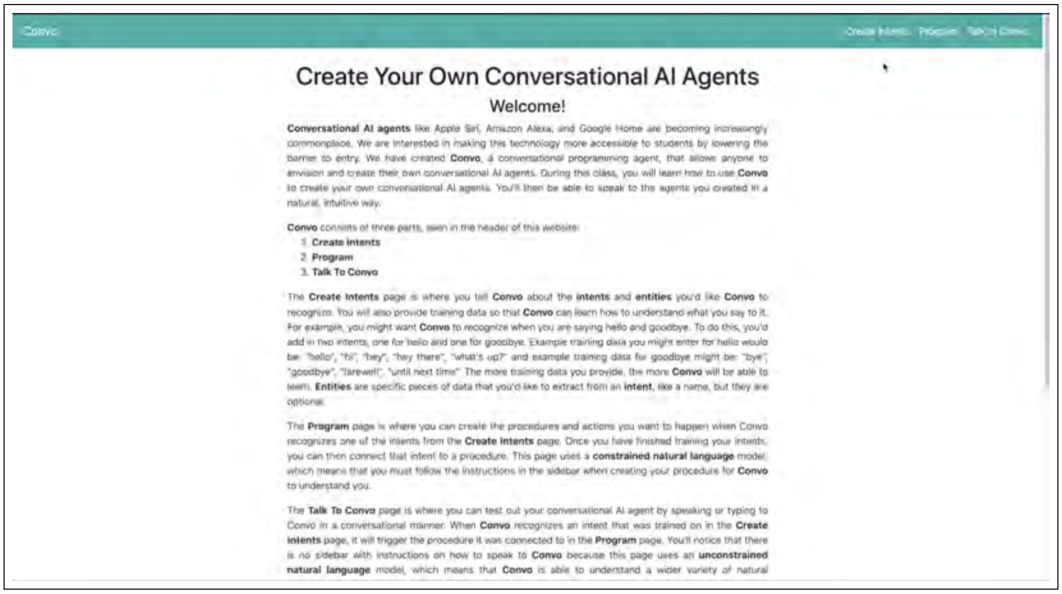
- `/model`: A `PUT` request that loads a specified model into the Rasa server. By default, upon restarting, the Rasa server uses the most recently generated model in the `/models` directory. However, to use a different model, or to update the existing model without restarting the server, I had to manually take that model's filename and override the default model. I used this endpoint every time I trained a new model, to ensure that the Rasa server always had the most recently trained model at hand.

- `/model/parse`: A `POST` request that uses the currently loaded model in the Rasa server to parse through text and perform intent and entity recognition. It returns the most likely intent, as well as confidence rankings for all known intents. If the intent contains entities, Rasa also returns its best guess for the values of those entities. I called this endpoint every time a user typed something into CONVO's *Talk to Convo* page.

3.2 Creating a New User Interface

Other than simply revamping the styling of CONVO, the new user interface also provides a much smoother user experience and facilitates the creation of conversational AI agents. The new user interface was initially designed by Jessica Van Brummelen, and I took the steps to bring her design to light. I chose to use React to build out this new version of CONVO because of its wide developer community and ease of use for both developers and users [33].

There are four pages to CONVO, as seen in Figures 3-2, 3-3, 3-4, and 3-5. Three of the pages are entirely new, and one, the *Program* page, is based off of the previous version of CONVO and contains all of the same functionality. When creating their conversational AI agents, Susan and Billy proceed from reading information and watching tutorials about CONVO on the homepage to inputting their training data and training CONVO on the *Create Intents* page and then creating their procedures and connecting them to their intents on the *Program* page. Once that is finished, Janet and Billy can then interact with the newly created agents on the *Talk to Convo* page. I discuss each of these pages and their purpose more in detail in the following sections.



Create Your Own Conversational AI Agents

Welcome!

Conversational AI agents like Apple Siri, Amazon Alexa, and Google Home are becoming increasingly commonplace. We are interested in making this technology more accessible to students by lowering the barrier to entry. We have created **Convo**, a conversational programming agent, that allows anyone to envision and create their own conversational AI agents. During this class, you will learn how to use **Convo** to create your own conversational AI agents. You'll then be able to speak to the agents you created in a natural, intuitive way.

Convo consists of three parts, seen in the header of this website:

1. **Create Intents**
2. **Program**
3. **Talk To Convo**

The **Create Intents** page is where you tell **Convo** about the **intents** and **entities** you'd like **Convo** to recognize. You will also provide training data so that **Convo** can learn how to understand what you say to it. For example, you might want **Convo** to recognize when you are saying hello and goodbye. To do this, you'd add in two intents, one for hello and one for goodbye. Example training data you might enter for hello would be: "hello", "hi", "hey", "hey there", "what's up?" and example training data for goodbye might be: "bye", "goodbye", "farewell", "until next time" The more training data you provide, the more **Convo** will be able to learn. **Entities** are specific pieces of data that you'd like to extract from an **intent**, like a name, but they are optional.

The **Program** page is where you can create the procedures and actions you want to happen when Convo recognizes one of the intents from the **Create Intents** page. Once you have finished training your intents, you can then connect that intent to a procedure. This page uses a **constrained natural language** model, which means that you must follow the instructions in the sidebar when creating your procedure for **Convo** to understand you.

The **Talk To Convo** page is where you can test out your conversational AI agent by speaking or typing to Convo in a conversational manner. When **Convo** recognizes an intent that was trained on in the **Create Intents** page, it will trigger the procedure it was connected to in the **Program** page. You'll notice that there is no sidebar with instructions on how to speak to **Convo** because this page uses an **unconstrained natural language** model, which means that **Convo** is able to understand a wider variety of natural

Figure 3-2: The homepage of the Convo website. It contains information about conversational AI and how they work, guides to each of the three other pages on the site, and links to the video tutorials on how to use Convo.

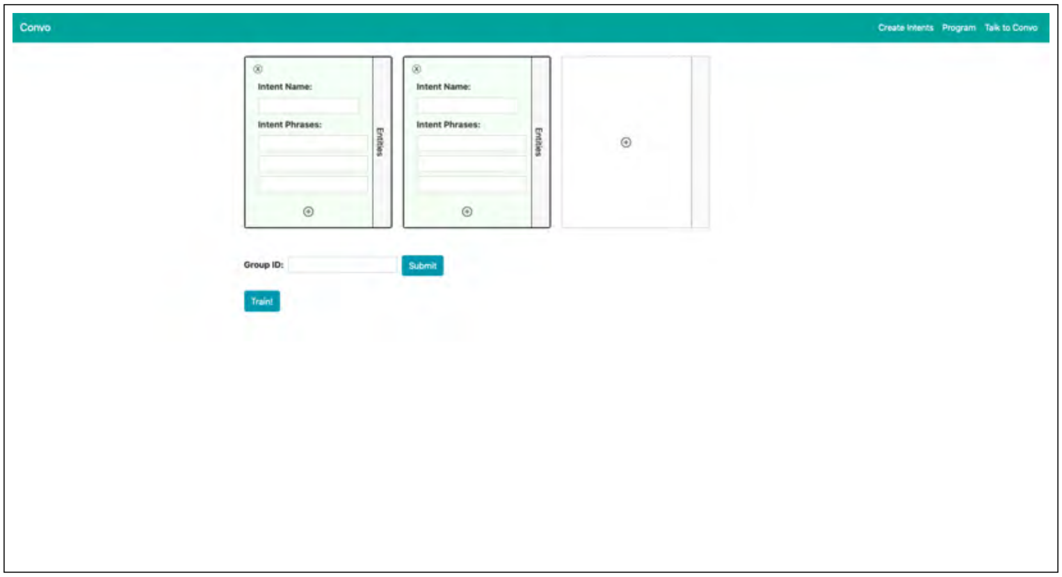
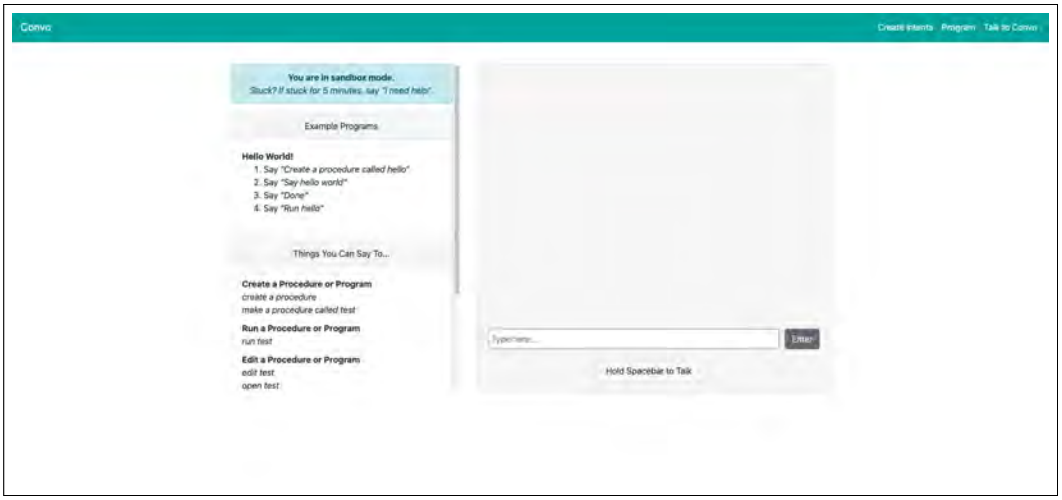


Figure 3-3: The *Create Intents* page of the Convo website. Here, students can enter in any training data they'd like for Convo to learn about. The Group ID is specific to the workshop, and can be left blank. Once all the data has been entered, students may click the *Train!* button.



You are in sandbox mode.
Stuck? If stuck for 5 minutes, say "I need help".

Hello World!
1. Say *"Create a procedure called hello"*
2. Say *"Say hello world"*
3. Say *"Done"*
4. Say *"Run hello"*

Create a Procedure or Program
create a procedure
make a procedure called test

Run a Procedure or Program
run test

Edit a Procedure or Program
edit test
open test

Figure 3-4: The *Program* page of the CONVO website. Students create their own programs and connect them to the intents they trained on the *Create Intents* page here. There is a sidebar on the left side of the page with a list of phrases that CONVO can understand.

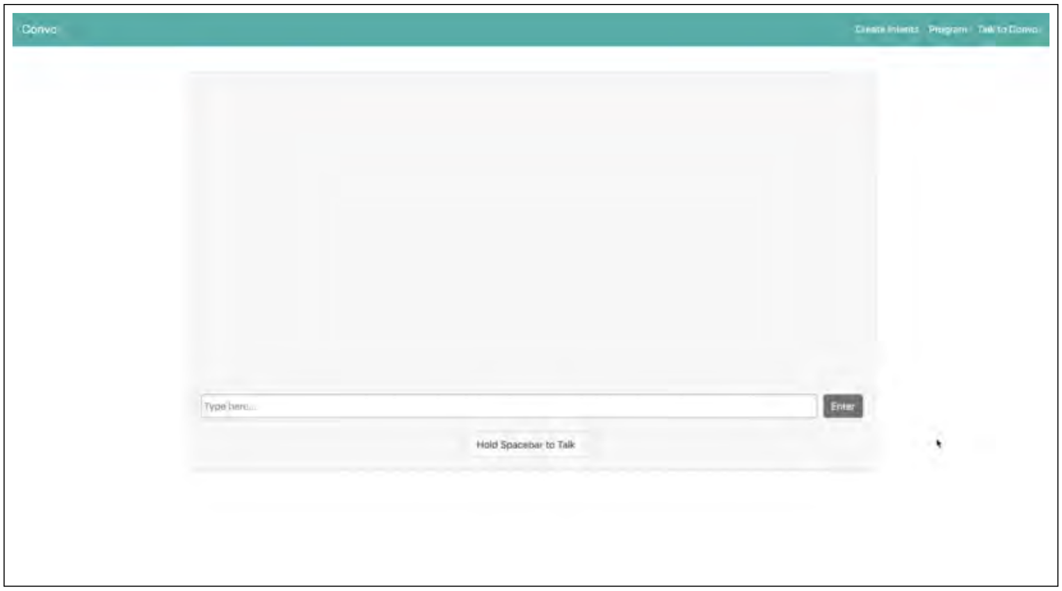
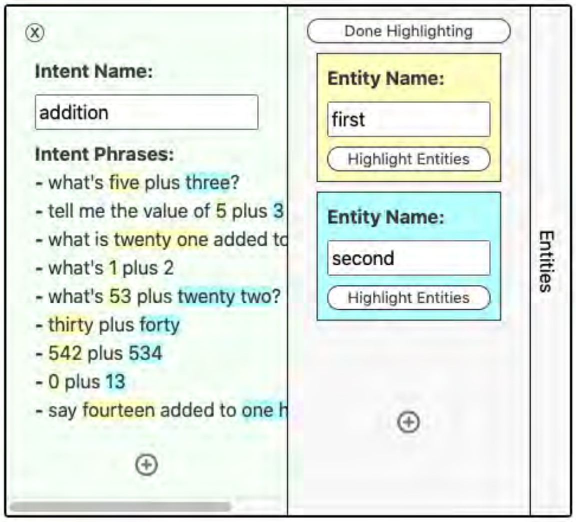


Figure 3-5: The *Talk to Convo* page of the CONVO website. Here, students may speak or type to CONVO in a conversational manner, without needing to follow any preset list of valid commands. CONVO will do its best to figure out the intent of what was said, and run any corresponding programs in response.

3.2.1 Training Intents and Entities

To allow Susan and Billy to teach CONVO about the intents and entities they wished it to learn, I created the *Create Intents* page, seen in Figure 3-3. On the page, Susan and Billy fill out the cards with their training data. There is an input space for the name of the intent, as well as places for the intent phrases, or ways a user might say something that would trigger that intent. Susan and Billy can also easily add more intents, delete unwanted ones, and add more intent phrases. Examples of filled out cards were previously shown in Figures 1-1 and 1-2.

In Billy's case, he also wants to indicate to Convo that there are certain pieces of information he'd like it to learn to extract from a user's intent phrases. He can do this in one of two ways. For the first method, Billy expands the card by clicking on the *Entities* tab, exposing the highlighting mechanism I developed. He names his entities, and then uses the appropriate highlight color (toggled by clicking on the corresponding *Highlight Entities* button) to use his cursor to select the relevant words or phrases in his intent phrases. Each entity is represented by a different color for visual clarity. There can be a maximum of six different entities for one intent, which should be more than plenty. When Billy is done highlighting, he clicks the *Done Highlighting* button to finish. His example card might look something like Figure 3-6.



Intent Name: addition

Intent Phrases:
- what's five plus three?
- tell me the value of 5 plus 3
- what is twenty one added t
- what's 1 plus 2
- what's 53 plus twenty two?
- thirty plus forty
- 542 plus 534
- 0 plus 13
- say fourteen added to one h

Entity Name: first

Entity Name: second

Figure 3-6: Billy uses the highlighting feature to indicate to Convo about the entities that represent the *first* and *second* numbers to add within each intent phrase.

The second way to indicate entities to Convo is to manually type in the proper syntax directly in the intent phrase inputs. The way that Rasa (and, consequently, Convo) recognizes entities from the training data is by having the entity word or phrase placed in hard brackets, followed by the entity name in parenthesis immediately afterwards, with no space in between. An example of this can be seen in the right half of Figure 1-3. Clicking *Done Highlighting* after highlighting results in the same end result. Susan and Billy can also manually edit their intent phrases after highlighting to fix any potential mistakes this way.

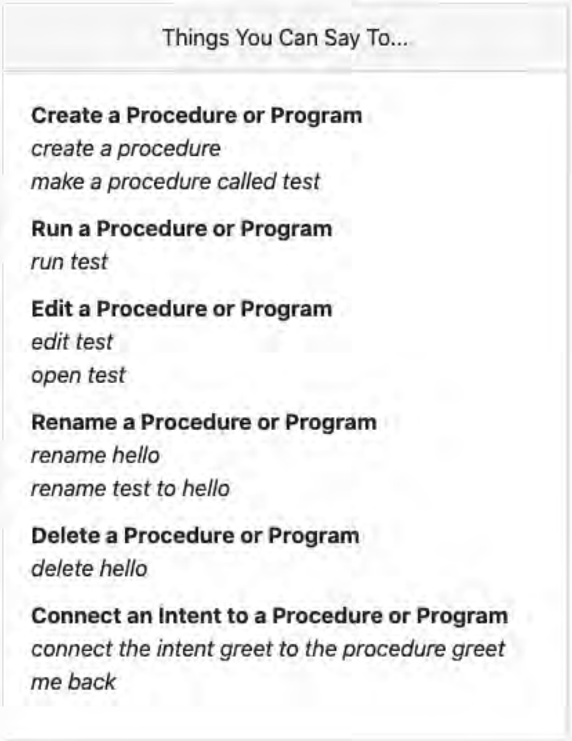
Now that Susan and Billy have both finished filling out all the training data they wish for CONVO to know, they must tell CONVO to actually learn by clicking the *Train!* button on the page. A spinner will appear to indicate that CONVO is in the process of creating a new model based on the new training data, and the spinner will be replaced by the word *Done* when the model has finished being created. The input space for a *Group ID* on the page is optional and was created for the purpose of the workshop. I discuss this further in Section 3.3.

For the actual implementation of this page and the cards, I simply created React components that could keep track of the user-entered data in each of its fields, and I used styled components to incorporate dynamic styling of the cards [1]. I also chose to save all data to the user's local browser storage, so that if they switched to a different page of CONVO or refreshed the page, their training data would not disappear.

3.2.2 Connecting Intents to Procedures

Once Susan and Billy finish teaching CONVO about all their training data, they move onto the next step of creating procedures and connecting them to their intents. These actions are all performed on CONVO's *Program* page, which is on a different tab of the CONVO website. Susan and Billy may choose to speak or to type to CONVO on this page.

To connect an intent to a procedure, Susan and Billy must type in the exact phrase, "connect the intent [intent name] to the procedure [procedure name]," where [intent name] and [procedure name] are replaced by the actual names of the intent and procedure to be connected. If the command succeeds, CONVO will know to run the procedure when the intent is triggered on CONVO's *Talk to Convo* page. The pattern instruction and an example can be seen in Figure 3-7.



Things You Can Say To...

Create a Procedure or Program
create a procedure
make a procedure called test

Run a Procedure or Program
run test

Edit a Procedure or Program
edit test
open test

Rename a Procedure or Program
rename hello
rename test to hello

Delete a Procedure or Program
delete hello

Connect an Intent to a Procedure or Program
connect the intent greet to the procedure greet me back

Figure 3-7: The sidebar of the *Program* page with things someone can say to CONVO. The bottom command contains an example of how a student can indicate to CONVO that it wants to connect an intent to a procedure.

There are a couple of responses Susan or Billy might receive when trying to connect an intent to a procedure. First, if they have a typo or they do not match the given pattern exactly, CONVO will not understand what they are trying to do and return a canned response of "I didn't understand what you were saying. Please try again." In another case, if somehow the intent Susan or Billy is trying to connect doesn't exist, perhaps something like *say goodbye*, CONVO will respond with "An intent with the name, say goodbye, has not been created."

If the command is valid, CONVO will respond with the base message of "I connected the intent [intent name] to the procedure [procedure name]. What do you want to do now?" There are additional statements CONVO might make, depending on if the procedure already existed or if there are entities associated with the intent.

For the former case, if the procedure to be connected did not already exist, CONVO will automatically create a new procedure with the given procedure name, as a shortcut. For the latter, which is not mutually exclusive to the former, CONVO will create new variables that have the same names as the entities within the given procedure if they do not already exist. These variables will be set to a default value of 0, but this value will be overridden if CONVO detects a different value in a user utterance on the *Talk to Convo* page. The purpose of this is so students like Susan and Billy can use the values of their entities within their procedures. For example, Billy would use the variables *first* and *second* to represent his entities with the same name. Then, in his procedure, he would add the values of the variables *first* and *second* together to get his final desired sum. If there are no entities associated with an intent, no variables will ever be created as a result of connecting an intent to a procedure.

Once an intent and a procedure have been connected, it is not possible to delete that connection. It is possible, however, to override connections, so if a student connected the same intent to two procedures, the procedure connected after the other would prevail. Multiple intents can be connected to the same procedure.

In coding up the functionality to connect intents to procedures, I followed the pattern of how CONVO already works, first using a dialog manager to match a user's utterance to a specific `regex` formula, then creating a goal, and finally checking to make sure all I had all of the necessary information (intent name, procedure name, entities where applicable) before executing that goal [41].

3.2.3 Talking to CONVO

My final main technical contribution to CONVO is the creation of the *Talk to Convo* page. This page, as seen in Figure 3-5, contains only a chatbox in which to speak or type to CONVO. The main difference between this page and CONVO's *Program* page is that CONVO is using Rasa to understand meaning and formulate a response here. There is no 'instruction manual' like there is on the *Program* page because CONVO (via Rasa) will interpret the speaker's intent instead of trying to match what the speaker says to an exact pattern. It is on this page that Janet interacts with CONVO, and how Billy communicates with CONVO to use the agent that he created. They may choose to speak or type to CONVO on this page.

3.3 Technical Challenges

I worked on the implementation of CONVO starting in late November up until early March. During this time, I had to familiarize myself with the existing CONVO system, make my changes, and test them all to ensure that everything was working smoothly. Many of the technologies used in CONVO were new to me, including the use of Docker containers and deploying in production via an Nginx server[41]. I relied on the support of Jessica Van Brummelen, Jeffrey Schiller, and the internet in getting a final working version of CONVO.

3.3.1 Resource Constraints

One large technical challenge to overcome was figuring out how to balance hosting an appropriately sized workshop with the immense amount of resources running Rasa required. Because of the way Rasa is designed, each instance of Rasa can only support parsing one model and training one model at a time. These constraints proved tricky to overcome for several reasons.

First, with only one instance of Rasa, anytime multiple users tried to use the *Talk to Convo* page at the same time (as they would in the workshop), CONVO would have to constantly be replacing the model that Rasa was using to parse the user utterances since each user/browser would have its own model, or conversational AI agent. While possible, this could lead to many issues with race conditions and also added latency for users. For these reasons, as well as a time constraint, I chose not to implement this version of CONVO.

The second issue of Rasa only being able to train one model at a time would mean that given one instance of Rasa, if multiple users tried to train their models concurrently (as they would in a workshop), they would keep overwriting each other's processes. Additionally, training times for models varied from a minute up to five minutes. As such, it would be unreasonable to ask all students to wait for each other to finish training, since some students would have no time left to create procedures.

The obvious solution seems to simply be to add more instances of Rasa to CONVO.

This would be correct, except that each instance of Rasa required a resident memory allocation of 2.8Gb. With a total of 15 students attending the workshop, the resource demands would increase proportionally. I ended up having the CONVO server resized to have 16 cores and 64GB of memory. I also decided to pair up students so I ended up running eight instances of Rasa in parallel during the workshop. Paired students had the opportunity to work with each other, and they discussed amongst themselves about when they would finish training. To assign these students to their own instances of Rasa, I gave each pair their own unique Group ID, which they were asked to enter in on CONVO's *Create Intents* page. The Group ID routed each student's `HTTP` requests to the corresponding port of that instance of Rasa.

3.4 Summary of Technical Contributions

The implementation work described in this chapter took place over a period of 3 to 4 months and involved a standard code review process. As a review of this chapter, I summarize my key technical contributions to CONVO as follows:

- **I integrated Rasa with CONVO.** This involved investigation into the capabilities of Rasa and how to use its HTTP API to communicate. Rasa is the backbone behind CONVO's *Talk to Convo* page and allows CONVO to understand utterances that have not been pre-programmed into CONVO.

- **I developed a flow to connect intents with procedures.** On CONVO's *Program* page, students can tell CONVO to run a certain procedure whenever its connected intent is triggered on CONVO's *Talk to Convo* page. Students do this by entering in a specific phrase that much match a pattern that CONVO recognizes exactly.

- **I created a new user interface for CONVO.** In addition to giving CONVO an overall more modern and sleek appearance, the new interface includes a simple and intuitive way for students to enter in the training data they wish CONVO to learn. The training data includes intents, intent phrases, entities, and entity examples.

Chapter 4

Workshop Curriculum

To test my curriculum and the effectiveness of Convo, I held a series of three 2-hour long workshops, titled “Create Your Own Conversational AI Agents,” over a span of three consecutive Saturdays (March 13th, 20th, and 27th, 2021). I co-taught with Jessica Van Brummelen, and had assistance from Elizabeth Harkavy, Daniel Dong Young Kim, Lucy Xinyu Liu, and Nicole Pang. The workshops were held as a part of SPARK, a student-run MIT ESP (MIT Educational Studies Program) initiative aimed at teaching 7th and 8th graders [2]. Due to the ongoing coronavirus pandemic, all classes were conducted remotely. SPARK took care of sourcing students from across the country and planning all virtual classroom logistics via Zoom.

Because of computation resource constraints outlined in Section 3.3.1, I set the maximum enrollment for my class to be 15. There were no prerequisites listed to be a student in my course. On each of my three teaching days, I had an attendance of 13 to 15 students, many of whom eagerly participated by typing into the Zoom chat. In order to encourage audience participation in the remote setting, I tried to include many interactive activities, like a Kahoot quiz, and asked many questions to the students [40]. An overview of each day’s activities can be found in Table 4.1.

The curriculum I taught in the workshop was based off of a previous workshop where students were also taught about conversational AI and tasked with programming Amazon’s Alexa to perform tasks [12]. I used the same Big AI Ideas as a framework for explaining what AI and specifically conversational AI is and isn’t [28].

In my pre- and post-workshop surveys, I also used some of the same questions aimed at assessing students' AI literacy competency [11].

The new materials and ideas I introduced mainly surrounded applying the Big AI Ideas to conversational AI specifically, discussing the spectrum of unconstrained and constrained natural language, and showcasing what Convo can do. This involved creating new slide decks, editing video tutorials and demos, and creating follow-along tutorials for students. The specific PDF tutorials and links to the video tutorials I created can be found in Appendix B.3 and B.4. I also went into more depth about the various new features I added to Convo, and broke down all of the steps that went on behind the scenes to achieve a working conversational AI agent.

The high level goals I wanted to accomplish through this workshop were for students to have a better understanding of conversational AI and for them to also be empowered to create their own conversational AI agents to solve problems in their everyday lives and communities. To achieve this, I wanted to make sure that students not only understood the concepts of conversational AI but that they also had enough time to experience building a conversational AI agent themselves.

Table 4.1: Overview of workshop schedule and curriculum. Each session was 2 hours long, spaced 1 week apart.

Workshop Schedule and Curriculum

Day 1	• Complete the pre-questionnaire survey (see Appendix B.1 for the full list of questions). • Introduce students to the five Big AI Ideas: Perception, Representation and Reasoning, Learning, Natural Interaction, and Societal Impact. Discuss with students whether or not certain technologies should be classified as AI based on the five ideas. • Introduce students to conversational AI, using the previous Big Five AI Ideas. Discuss the difference between constrained and unconstrained natural language models, giving examples on the spectrum. • Debut CONVO and explain its role as a conversational AI agent. Play introduction and tutorial videos so that students have an idea of how CONVO works and what it can do. • Recap all concepts with a fun, interactive Kahoot quiz (see Appendix B.5 for the full list of questions and possible answers).

Day 2	• Review main ideas from the previous week: the Big Five AI Ideas, conversational AI, and Convo. • Perform a live demo of Convo demonstrating the workflow and some simple intents that Convo can recognize. • Ask students to fill out a template for any conversational AI ideas they would like to create. Example ideas were available, and students modified a Google Slides presentation in real time. • Brainstorm for ideas with students by giving a presentation on potential problems we might want to solve. • Provide students with informative PDF tutorials (see Appendix B.3 and B.4 for the full tutorials).
Day 3	• Let students complete tutorials with staff on standby to help debug if any questions or issues arose. • Let students create and train their own conversational AI agents. • Complete the post-questionnaire survey (see Appendix B.2 for the full list of questions).

Originally, I had planned to have the students complete the tutorials on Day 2 of the workshop. However, we ran into technical difficulties and we were forced to improvise. To make up for any potential lost time during this class, I allowed the students to attempt the tutorials on their own time during the week in between Day 2 and Day 3 of the workshop; however, only one or two students made attempts at the tutorials. This was not an issue, as there was plenty of time for students to complete the tutorials in class on Day 3 and still have time for their own projects.

Chapter 5

Evaluation

The data collected from the workshop provide valuable insights into student literacy, ideas, and attitudes on conversational AI. Specifically, I used the data to answer the following research questions:

- How does learning about and interacting with Convo affect students' perceptions and understanding of conversational AI agents? (Section 5.1)
- Does learning about and interacting with Convo empower students to create their own conversational AI agents? (Section 5.2)
- What AI literacy skills can students learn from using Convo? (Section 5.3)
- What do students envision creating with a system like Convo? (Section 5.4)

To address these questions, I examined the changes (if any) that occurred in students from the start of the class to the end. To substantiate the results, I used the Wilcoxon Signed-Rank Test to measure the magnitude of change between the pre- and post-workshop survey data [36]. I also analyzed the project ideas that students developed after a brainstorming session.

At the start of the workshop, all students and parents were provided with a link to a consent form that allowed the student's data to be used for the purpose of my research. In total, 12 out of 15 students submitted signed and completed forms, though not all of them completed both the pre and post workshop questionnaires.

I received 14 responses for the pre-workshop survey, but only 12 students had also submitted a consent form. The post-workshop survey had a lower response rate, with a total of 9 responses but only 7 were from students who had also submitted a consent form. In an ideal scenario, all 15 students would have completed all consent forms and all surveys. Despite not reaching this goal, I believe that the data obtained is sufficient for a pilot study and can motivate further research in this area.

The students' ages were between 11 and 14 years, and all of them were either in 7th or 8th grade. The distribution of the 12 students who completed both the consent form and the pre-workshop survey is: 11 years (1 student), 12 years (4 students), 13 years (5 students), 14 (2 students). There were also 4 students who identified as female, and 8 that identified as male. The breakdown of the 7 students in the post-workshop survey is: 11 years (1 student), 12 years (2 students), 13 years (2 students), and 14 years (2 students) and 4 female and 3 male.

There was also variance in students' prior programming experience. In the pre-workshop survey, 10 out of 12 students indicated that they had at least some form of experience, either with block-based (e.g. Scratch, MIT App Inventor) or text-based programming (e.g. Python, Java), leaving only 2 out of 12 students with no programming experience at all. Of the students with prior programming experience, 7 out of 10 had done some type of text-based programming, 6 out of 10 had some block-based programming experience, 4 out of 10 had taken some type of course on programming, and 4 out of 10 had completed a project or app. There was no requirement for students to have any type of programming experience before taking my course.

5.1 Research Question 1

I begin by examining the question of **how learning and interacting with CONVO affects students' perceptions and understanding of conversational AI agents**. To answer this, I look at student data on how they might characterize a conversational AI agent as well as what they envision these agents being used for in the future. Together, I refer to the combination of these questions as student ‘ideas’ on conversational AI agents.

For the first type of question, I asked students to answer a series of *Persona* questions as outlined in a similar study regarding Amazon’s Alexa [12]. These questions asked students to rank a series of statements about conversational AI agents on a 7-point Likert scale. The statements targeted specific sentiments the students might feel towards conversational AI agents, and students were asked to score each statement based on how much they agree or disagreed with it.

The second question asked students to check off all the ways in which they envisioned they might use conversational AI agents. The options included common use cases for conversational AI today but also allowed students to add their own ideas if they weren’t already listed. This question was based off of a previous unpublished study [24]. The exact survey questions can be found in Appendix B.1 and B.2.

5.1.1 Pre-Workshop Student Ideas on Conversational AI Agents

We begin by examining the data on students’ perceptions and understanding of conversational AI agents by looking at how they scored various statements on a 7-point Likert scale. Due to an error where the question was not marked as required, only 10 out of a potential 12 students filled out this part. The results can be seen in Table 5.1.

Table 5.1: Students rank how much they agree or disagree with statements involving different descriptors of conversational AI agents based on a 7-point Likert scale in the pre-workshop survey.

Statement	**Mean**	**Range**
Conversational AI Agents are intelligent.	6	2-7
Conversational AI Agents are friendly.	4.5	2-7
Conversational AI Agents are alive.	2.8	1-6
Conversational AI Agents are safe.	4.5	2-6
Conversational AI Agents are trustworthy.	4.8	2-7

Conversational AI Agents are human-like.	3.8	1-6
Conversational AI Agents are smarter than me.	5.7	1-7

The data in Table 5.1 indicates that at the start of the workshop, on average, students slightly disagreed with the assertion that conversational AI agents are ‘alive’, and they were pretty neutral about conversational AI agents being ‘friendly,’ ‘safe,’ ‘trustworthy,’ and ‘human-like.’ What is more interesting is the fact that students tended to agree with the statements that conversational AI agents are ‘intelligent’ and ‘smarter than me.’

In terms of the actual distribution of scores, there was one student who chose a score of 6 or 7 for every statement as well as one student who chose a score of 1 or 2 for each statement. Given this, for the statements where conversational AI agents are described as ‘intelligent’ and ‘smarter than me,’ every student with exception of the one who chose 1 or 2 for every statement gave it a score of 6 or 7. This is a strong indicator that before the workshop, the vast majority of students either *agreed* or *strongly agreed* with the ideas that these agents were intelligent and smarter than them. Additionally, for the assertion that conversational AI agents are safe, every student with exception of the one who chose 6 or 7 for every statement assigned a score of 4 or below. This is an indication that the majority of students tended to disagree to some degree that these agents are safe. I can only speculate as to why this might be the case, but it seems that with the increasing reach that these agents have into our daily lives, there is also increasing concern over many ethical issues surrounding boundaries [35]. For the rest of the statements, the large range and relatively even distribution of scores indicates that even within themselves, the students had differing opinions on the validity of these statements.

Next, we move onto the question asking students about how they envisioned using conversational AI agents themselves, and we view the gathered data below in Figure 5-1. There were a total of 12 students who answered this question in the pre-workshop

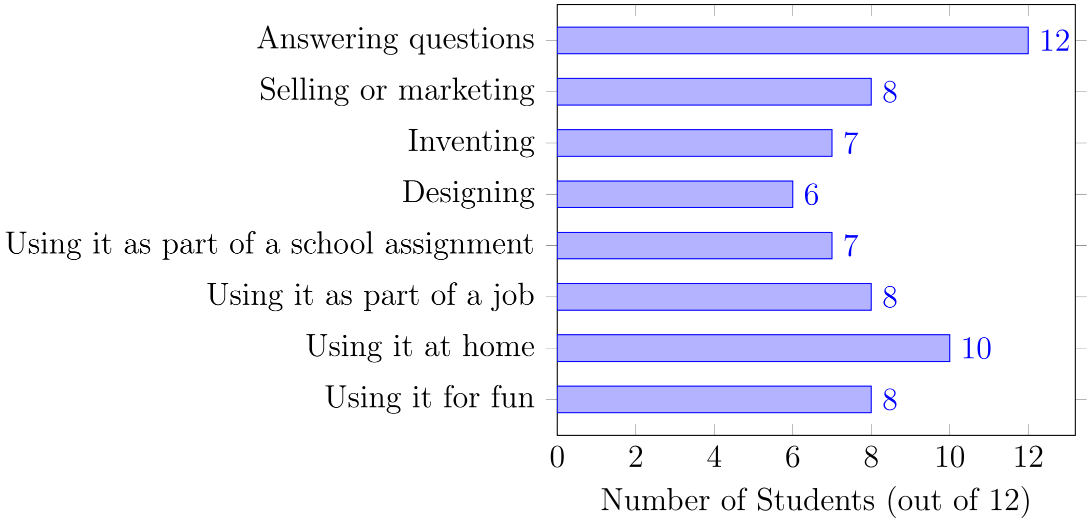


Figure 5-1: Students selected from a checklist of common use cases how they envisioned conversational AI agents being used in the pre-workshop survey.

questionnaire.

The most commonly selected use case for conversational AI agents was 'answering questions,' and the second most popular option was 'using it at home.' This is perhaps unsurprising, since most of the more famous conversational AI agents (Apple's Siri, Amazon's Alexa, Google Home) are mainly known for their abilities to answer questions in a home setting [15, 23]. Every listed option was selected by at least half of the students to be a potential use case, demonstrating the breadth and impact that these students believe conversational AI agents can have. This is only furthered by the fact that there was an option (not pictured) that stated "I do not care much about using this technology" that was selected by zero students.

5.1.2 Post-Workshop Student Ideas on Conversational AI Agents

For the post-workshop survey results, we examine the data from a total of 7 students who scored the same statements on conversational AI agents using the same 7-point Likert scale as in the pre-workshop survey. Information on the average and range of scores can be found in Table 5.2. A direct comparison between the average scores of the pre- and post-workshop surveys is seen in Figure 5-2.

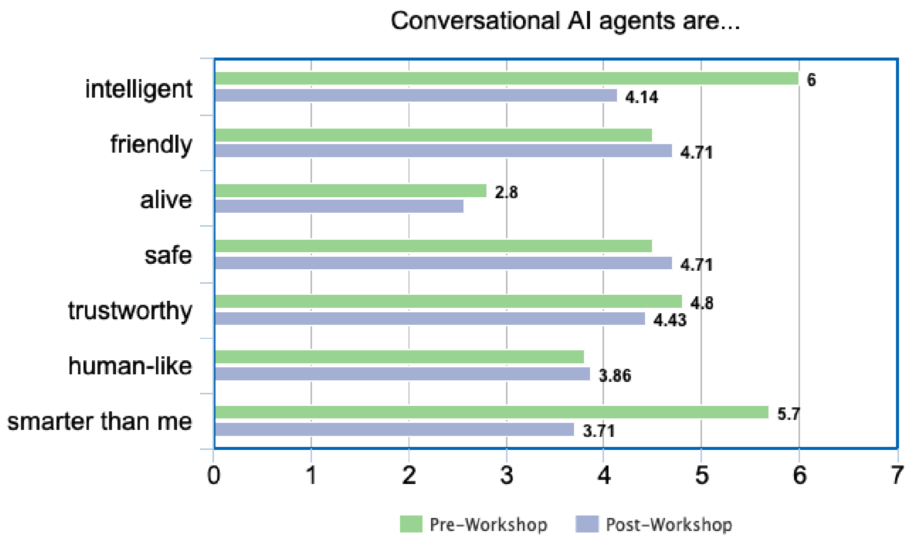


Figure 5-2: Averages of student perceptions of conversational AI agents based on a 7-point Likert scale in the pre- and post-workshop surveys.

Table 5.2: Students rank how much they agree or disagree with statements involving different descriptors of conversational AI agents based on a 7-point Likert scale in the post-workshop survey.

Statement	**Mean**	**Range**
Conversational AI Agents are intelligent.	4.14	1-7
Conversational AI Agents are friendly.	4.71	3-6
Conversational AI Agents are alive.	2.57	1-6
Conversational AI Agents are safe.	4.71	2-7
Conversational AI Agents are trustworthy.	4.43	2-7
Conversational AI Agents are human-like.	3.86	2-5
Conversational AI Agents are smarter than me.	3.71	1-6

The post-workshop survey results indicate that on average, students felt more or less pretty neutral about every single statement, with a slight disagreement with the assertion that conversational AI agents are safe. This is a large change from the pre-workshop survey results, where most students *agreed* or *strongly agreed* with the statements that conversational AI agents are intelligent ($\bar{x}_{diff} = -1.9$, $Md = -2$,

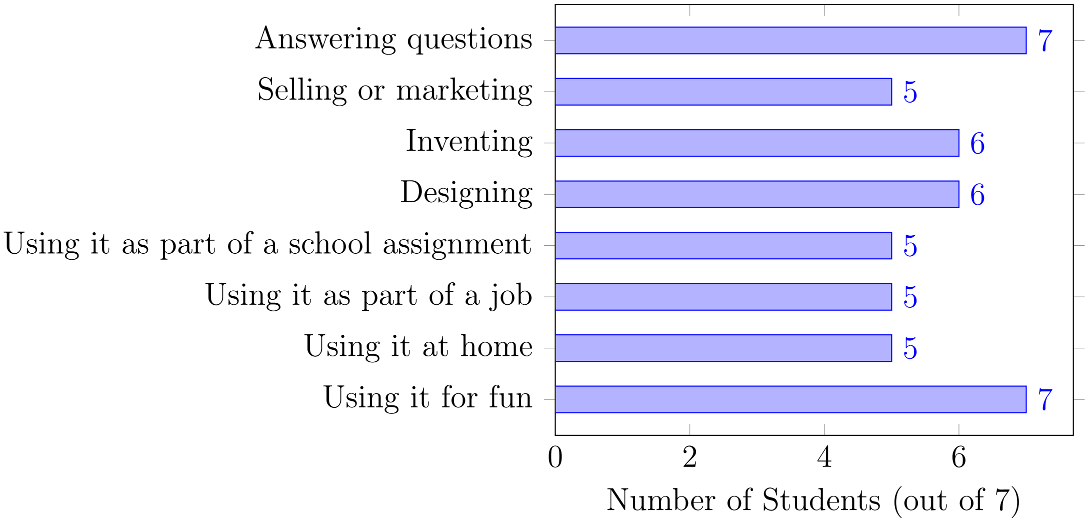


Figure 5-3: Students selected from a checklist of common use cases how they envisioned conversational AI agents being used in the post-workshop survey.

$|Z| = 2.15$, $p = 0.016$) and smarter than them ($\bar{x}_{diff} = -2.0$, $Md = -2$, $|Z| = 2.21$, $p = 0.013$). I attribute this shift in sentiment to the fact that students learned about just how much human work has to go into making a conversational AI agent seem 'smart.' For example, when working with Convo, students had to input data, train a model, and tell Convo exactly what to do in very specific scenarios. If students accidentally made one mistake or typo during the whole process, it was likely that Convo (or any 'smart' assistant) would not know what to do [44].

As seen in Figure 5-3, the most popular use cases (chosen by all students) for conversational AI agents selected by students after the workshop were 'answering questions' and 'using it for fun.' Similar to the pre-workshop survey results, students remained strongly in agreement that they envisioned these agents answering questions. All other options also received a larger percentage of student votes, indicating the versatility and expanded viewpoints the students had of conversational AI agents after the workshop. Like in the pre-workshop survey, zero students indicated that they did not care about using this technology, a clear sign that students were and remained interested in using conversational AI agents in their everyday lives.

Based on the data I gathered from the pre and post workshop questionnaires, I conclude that from learning and interacting with Convo, students decided that conversational AI agents were decidedly less intelligent and not smarter than them. Additionally, the students may have also expanded their views on what conversational AI agents were or could be used for in their daily lives.

5.2 Research Question 2

In this section, I examine **if learning and interacting with Convo empowers students to create their own conversational AI agents**. To do this, I compare the pre and post workshop results of two factors: how confident students are in being able to create their own conversational AI agents and how interested they are in creating their own conversational AI agents. The former reflects a technical understanding of the steps required, and the latter reflects the motivation to take those steps to actually build a conversational AI agent. Together, I refer to these factors as the students' 'attitudes' towards conversational AI agents.

To quantify student attitudes, I ask the same two questions on both the pre and post workshop surveys. The first asks students how confident they are in being able to create their own conversational AI agents. I used a 7-point Likert scale where a point value of 1 corresponds to *not at all confident* and 7 corresponds to *extremely confident*. The second question asks students how interested they are in creating their own conversational AI agents. Again, I also use a 7-point Likert scale here, where a point value of 1 corresponds to *not at all interested* and 7 corresponds to *extremely interested*. The exact questions on the surveys can be found in Appendix B.1 and B.2.

5.2.1 Pre-Workshop Student Attitudes on Conversational AI Agents

There were a total of 12 pre-workshop survey results regarding student attitudes on conversational AI agents. The questions and results can be found below in Table 5.3.

Table 5.3: Student attitudes based on a 7-point Likert scale towards conversational AI agents in the pre-workshop survey.

Question	**Mean**	**Range**
How confident do you feel in being able to create your own conversational AI agent?	3.83	2-6
How interested are you in creating a conversational AI agent?	5.92	4-7

An average value of 3.83 for the first question means that before the workshop, students leaned more on the side of being not confident when it came to creating a conversational AI agent. The distribution was pretty even, with 5, 3, and 4 students out of 12 giving scores less than 4 (less confident), equal to 4 (neutral), and greater than 4 (more confident), respectively.

The second question results demonstrated that overall, students were already very interested in creating conversational AI agents, with an average score of 5.92. Not a single student indicated that they were not interested (represented by a score of 3 or lower), and 5 out of 12 students gave a score of 7, indicating that they were *extremely interested* in creating a conversational AI agent. Most likely, selection bias is at play here, since students were able to choose which classes from SPARK they wished to attend. Regardless, it is truly amazing how young students today are interested in and have the opportunity to explore topics like conversational AI.

5.2.2 Post-Workshop Student Attitudes on Conversational AI Agents

For the post-workshop survey, there were a total of 7 students who responded to the same questions regarding confidence and interest in conversational AI agents. The data can be found below in Table 5.4. A comparison of the pre- and post-workshop data is seen in Figure 5-4.

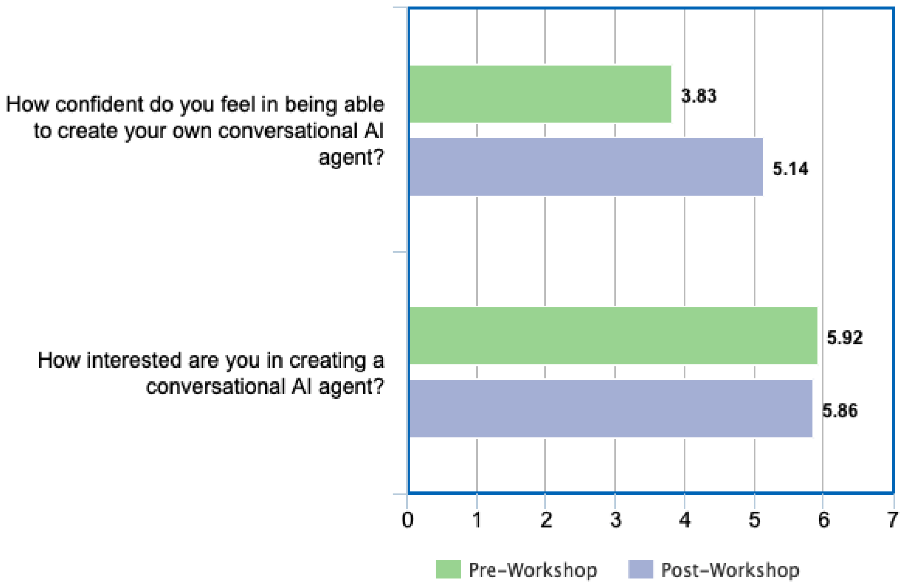


Figure 5-4: Averages of student attitudes towards conversational AI agents based on a 7-point Likert scale in the pre- and post-workshop surveys.

Table 5.4: Student attitudes based on a 7-point Likert scale towards conversational AI agents in the post-workshop survey.

Statement	**Mean**	**Range**
How confident do you feel in being able to create your own conversational AI agent?	5.14	3-6
How interested are you in creating a conversational AI agent?	5.86	4-7

From these results, we see that at the end of the workshop, students generally felt much more confident in their abilities to create their own conversational AI agents ($\bar{x}_{diff} = 2.0$, $Md_{diff} = 2$, $|Z| = 2.03$, $p = 0.021$). In terms of the distribution, 5 out of 7 students said that they were confident in their abilities (giving a score higher than 4), while only one remained slightly not confident (giving a score lower than 4). Additionally, an examination of the pre-workshop results of the 7 students who also filled out the post-workshop survey shows that across those students, their confidence levels all either remained the same or increased.

As for student interest in conversational AI agents, the post-workshop survey results indicate that there was little to no change before and after the workshop ($\bar{x}_{diff} = 0.0$, $Md_{diff} = 0$, $p > 0.05$). This is not too unexpected, given that student interest started off very high. Since student interest remained high even after the workshop, I am not concerned with the ability of my curriculum and CONVO to evoke interest. It would, however, be interesting to see the change in scores for this question if the workshop had been run with students that were originally not interested in conversational AI agents. I explore this more as a possible avenue of future work in Section 6.2.

Overall, because of the increased levels of student confidence and the maintenance of high interest levels in conversational AI agents, I do think that CONVO empowers students to build more projects involving conversational AI.

5.3 Research Question 3

In this section, I explore the question of **what AI literacy skills students can learn from using CONVO**. To do this, I ask a series of questions based on the *Conception* questions outlined in a previous paper that assessed student understanding after interacting with Amazon's Alexa [12]. I also ask some questions that specifically test student knowledge of some of the more technical details of conversational AI, such as the steps to creating a conversational AI agent, as well as questions surrounding the training data and constrained/unconstrained models of CONVO.

5.3.1 Pre-Workshop Student Literacy of Conversational AI Agents

To start, I asked students what they thought AI was in just 3 words or short phrases. The aggregation of all 12 of the responses can be found in Figure 5-5. Common themes among the words and phrases were that computers and technology were smart and intelligent. Some students commented on how "weird" it was, and others gave responses that indicated that AI was still very much a black box topic to them.

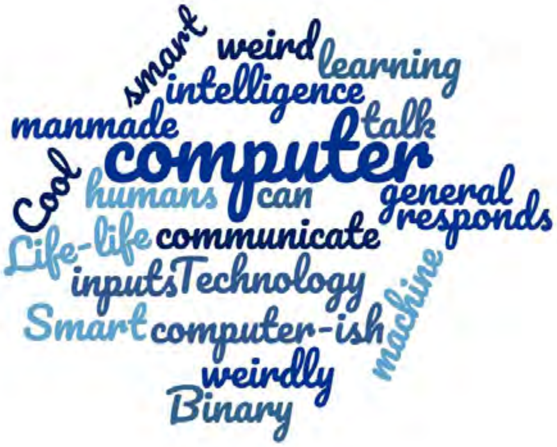


Figure 5-5: A graphic of the words used by students to describe AI in the pre-workshop survey. Larger words correspond to a larger word frequency. Figure generated from `https://www.wordclouds.com/`.

> [AI is...] computer-ish, they talk weirdly, and they are in general weird.

Next, I asked students to briefly describe how they thought conversational AI agents worked. For this prompt, only 5 out of 12 students responded with something related to code and/or programming. Even for those responses, they were often very vague and lacked any type of specificity. Some other responses mentioned responding to human inputs and some responses simply stated “I don’t know.”

> AI works by using code and other stuff to create some sort of machine.

Finally, I asked students to give 3 examples of what they thought AI could do. The students provided a wide array of responses, from “solving math problems” to “turning lights on and off” to even “helping the military.” There was plenty of variety in the answers, and with the exception of one response (“exterminate us!”), all answers were valid and a possible use case of AI.

> “ [I think AI can...] drive cars, do my homework, and search the web. ”

In the pre-workshop survey, I also asked students to arrange a collection of 5 steps in order. These steps are a bit specific to CONVO, but are generalizable to the creation of any conversational AI agent. In the correct order, they are:

1. Collecting/inputting the data
2. Training the data
3. Creating the procedures
4. Connecting the procedures to intents
5. Testing on the app

Note that steps 3 and 4 are interchangeable, but step 2 must come after step 1 and step 5 must come last. Student performance on this question was quite varied, with only 2 (out of 12) students selecting the correct first step but 10 (out of 12) students selecting the correct last step. A total of 7 students (58%) correctly identified that you need to collect and input the data before training. Zero students got the entire procedure correct, even accounting for possible shifts of steps 3 and 4. Of course, since this question was asked prior to any class material, I did not expect students to perform well on this question. Rather, I wanted to use these results as a baseline for the existing level of technical knowledge students had on how to create a conversational AI agent.

5.3.2 Post-Workshop Student Literacy of Conversational AI Agents

Similar to the pre-workshop survey, I asked students to describe, in their own words, what they thought conversational AI agents were. A total of 7 students gave responses, and no student said something that was inaccurate. Three students specifically stated that you must program the agents with human-given data, and all seven students mentioned that these agents talk back to you. Overall, I would say that students gave much more technical responses and it was clear that they had grasped the idea of conversational AI agents (and AI in general) being something that relies heavily on humans for data and input.

> “A conversational AI agent is an AI agent that understands processes, and responds to texts, or vocal inputs in natural ways.”

The next question, how conversational AI agents work, is the same question I asked in the pre-workshop survey. Student responses to this question were much more technical here as well, with many students using new terminology like “training,” “data,” and “entities.” While some answers were still vague, no answer was incorrect, and every single student mentioned needing a human to code or program something for the agent to respond properly. This is an indication to me that student understanding and literacy of conversational AI increased over the course of the workshop.

> “[Conversational AI agents work] by putting data in and training the AI.”

The last short answer question for students asked them for 3 examples of what conversational AI agents can do. Note that this is different from the question in the pre-workshop survey, where I asked students about what AI (in general) can do. Students gave a much narrower scope of answers, with many students stating responses in line with “answering questions” and “communicating with people.” Other students gave examples that many of today’s “smart assistants” commonly perform in the home, like setting a timer or telling you the weather [15, 23]. It is possible that students adopted a more narrow viewpoint on the uses of conversational AI agents after being exposed to the extensive work that goes into programming one. Especially after working on some simple examples and their own projects, students were likely to realize that it is very hard to create an agent that is ‘good’ enough to converse naturally at the level that a human can.

> “ A lot of things. Chat with you. Set a timer. Tell you the weather. And some other stuff. ”

In the post-workshop survey, I also again asked students to arrange the steps of creating a conversational AI agent in order. Out of a total of 7 students, 5 correctly selected the first step and 5 correctly selected the last step. Three students nailed the exact order of all 5 steps, and 2 other students were only off by one step. Note that one person simply selected 1-2-3-4-5 on the randomly ordered steps in the questionnaire. Comparing these results to those in the pre-workshop survey, we see that there is a definite increase in student accuracy in arranging the steps.

Finally, the last set of data I examine for this section is a series of true/false statements. Students were asked to select all statements that were true, and their results are shown in Table 5.5. The purpose of these questions was to examine if students were able to understand a few key conversational AI concepts. We see that students performed very well on the first two questions but poorly on the last three. The first two questions were related to the idea that more training data means a more accurate resulting agent while the last three questions surrounded the concept of constrained and unconstrained natural language models. While students seemed to understand the first idea, it looks like students had trouble fully grasping the latter idea. I propose a couple of ideas for expanding on this for future work in Section 6.2.

Table 5.5: Students were asked to select all of the true statements from a series of statements about CONVO. These statements targeted specific concepts in conversational AI that we wanted students to learn.

Statement	**Correct (out of 7)**
When training an intent, it is better to have fewer training examples. (False)	7
When setting an entity, it is better to have more training examples. (True)	6

In the Program mode, Convo would understand it when you say something different but similar to a command in the sidebar, for example replacing the word "procedure" with "function". (False)	3
In the Talk to Me mode, Convo would understand it when you say something different but similar to a command in the sidebar, for example replacing the word "procedure" with "function". (True)	2
Convo can recognize intents in both the Program and Talk to Me modes. (False)	3

In conclusion, despite the relatively small number of students who answered the post-workshop questionnaire, the increased technicality and overall accuracy in several key conversational AI concepts from student responses is an indicator that it is likely that student literacy in the realm of technical knowledge increased as whole as a result of the curriculum and interactions with Convo.

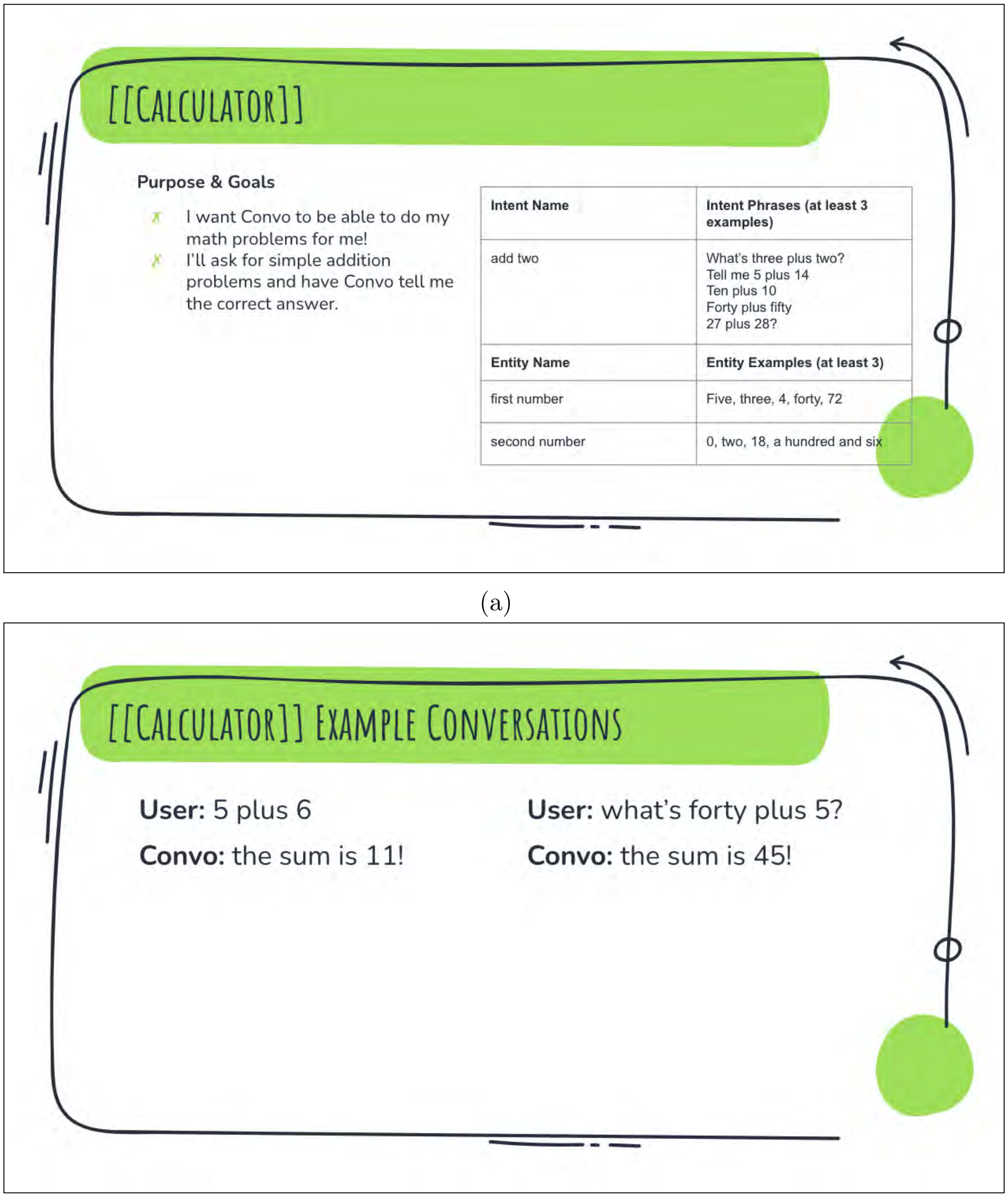
5.4 Research Question 4

The final research question I wanted to answer was examining **what students envisioned creating with Convo**. To accomplish this, I asked students to brainstorm ideas of what they wanted to create using Convo. I wanted to strike a balance between making sure that these ideas were creative and innovative but also realistic. As a result, I decided to host the idea-creating session in the middle of the workshop. That way, students would have been introduced to the Big Five AI ideas and some core conversational AI concepts, but would not have had a chance to get too bogged down in the technicalities of actually using Convo to create a conversational AI agent.

Students were asked to format their ideas by following a specific slide template in a Google Slides presentation. This template asked students to come up with a name for their idea, list out a couple goals they wished to achieve, outline the intents, intent phrases, entities, and entity examples they wished to train, and finally, provide some example conversations a user might have with Convo after it's been trained. Even though it was quite a bit of information, I wanted to ensure that students were thorough with their ideas and that they would be set up for success when it came time to actually implementing their ideas. I provided blank slide templates as well as example completed templates (see Figure 5-6) to the students and gave them about 45 minutes to add their own. I and my co-teachers were on standby the whole time to answer questions and provide feedback.

In total, we received 10 unique ideas from students, with a wide array of functions. While many students chose to create agents that performed tasks that were similar to the examples I had shown or given, like responding to someone saying 'Good morning' or telling a joke, other students came up with more original ideas. For example, some students had ideas involving time, such as asking for the date and time in a certain country. Another student came up with an idea where based on your current mood or emotion, Convo would reply with a corresponding phrase. Yet another student wanted to showcase her love of Harry Potter, and so wanted to come up with a way for Convo to respond accordingly when someone asked about a certain character or quote from the series. To view the exact slides made by these students, please visit Appendix A-1, A-2, and A-3.

Overall, it seems like while some students were more creative with their ideas, most of the students came up with cases where the interaction with the conversational AI agent was utility or command based, such as asking for the time or asking for a story or fun fact. This is likely due to the fact that most of the conversational AI agents these students had been exposed to in the past specialized in these types of narrower, task-focused interactions. Based on this, it might be worthwhile to hone in on these types of examples as possible entry points for students to begin exploring more about what conversational AI agents can do.



[[CALCULATOR]]

Purpose & Goals

- I want Convo to be able to do my math problems for me!
- I'll ask for simple addition problems and have Convo tell me the correct answer.

Intent Name	Intent Phrases (at least 3 examples)
add two	What's three plus two? Tell me 5 plus 14 Ten plus 10 Forty plus fifty 27 plus 28?
Entity Name	**Entity Examples (at least 3)**
first number	Five, three, 4, forty, 72
second number	0, two, 18, a hundred and six

(a)

[[CALCULATOR]] EXAMPLE CONVERSATIONS

User: 5 plus 6

Convo: the sum is 11!

User: what's forty plus 5?

Convo: the sum is 45!

(b)

Figure 5-6: An example of a filled out idea template where the idea was to create a calculator that could sum together two numbers. Each student was asked to come up with at least one idea. (a) First of a two page slide template asking students for the purpose and goals of their idea, as well as the intents and entities they wished Convo to understand. (b) Second of a two page slide template that asked students for some example conversations they envisioned a user having with Convo after learning about the training data from (a).

Chapter 6

Conclusion

I began this thesis by introducing my goal of empowering students to create their own conversational AI agents and how I achieved it. I outlined two example scenarios of why Susan, Janet, and Billy might want to use CONVO and how they would go about doing it. Then, in Chapter 2, I presented any relevant work and context needed to understand the changes I made to CONVO. In Chapter 3, I went into detail about the technical implementation and contributions I made to CONVO. Then, I discuss the curriculum and logistics of the user study I held for middle school students in Chapter 4. The results from that workshop are discussed in Chapter 5. I outline the key findings from that workshop below in Section 6.1.

6.1 Key Findings[1]

- **Learning about and using CONVO empowered students be more confident in their abilities to create their own conversational AI agents.** Student responses on the pre- and post-workshop surveys indicated a difference in how confident they were in their abilities to create their own conversational AI agents.

- **Students' perceptions of AI's intelligence shifted.** At the start of the

---

[1] This section was adapted from [44].

workshop, students overwhelmingly agreed with the sentiments of conversational AI agents being ‘intelligent’ and ‘smarter than them’. However, by the end of the workshop, students’ opinions had changed drastically, with students generally disagreeing with the same two sentences.

- **Students were able to come up with original and creative ideas for potential use cases of conversational AI agents.** During the workshop brainstorming session, students came up with ideas ranging from a Harry Potter trivia bot to a mood tracker.

- **Students were able to learn some key concepts about conversational AI.** From class discussion and also the post-workshop survey results, it is clear that students were able to gain knowledge about the Big Five AI ideas. Students also showed improved knowledge in the areas of providing training data to agents and the steps required to create an agent.

- **Students were able to create their own conversational AI agents.** Almost all students were able to complete the tutorials, which involved creating two separate conversational AI agents. Some students were able to venture even farther, and create their own original conversational AI agents.

- **Convo is a useful tool that can act as a starting block for students to learn more about conversational AI and conversational AI agents in particular.** Conversational AI agents are quite complex, and creating them is often a very involved and obscure process. Through this workshop, middle school students were able to learn about and create conversational AI agents, a promising step to empowering all students of any age or background to do so as well.

6.2 Future Work

While the current iteration of Convo has certainly shown promising results for its ability to help students understand and create conversational AI agents, there is still plenty of room for improvement and future work. I outline a few ideas below.

The first proposal is to run the same workshop with a larger number of students to obtain more conclusive results in the impact that Convo has on young students. Since this the user study conducted for this thesis was only a pilot study, I had a smaller number of students and thus a smaller amount of data to work with as well. A larger study would be able to provide more data and more confidence in the effect that Convo can have on student perceptions, ideas, and literacy of conversational AI.

Additionally, as I discussed in Section 5.2, there was no discernible change in student interest in creating their own conversational AI agents. However, even at the beginning of the workshop, student interest was already high and it remained similarly high after the workshop. I propose running the same workshop with a more random sample of students, instead of a self-selected group that is already interested in the topic. It would be insightful to see if the curriculum and working with Convo could spark interest in students that were previously uninterested in conversational AI.

In Section 5.3, I discussed how students did not seem to fully understand the concept of constrained vs. unconstrained natural language. To expand on this, I propose modifying the curriculum to add in more activities surrounding this concept. I would also emphasize the difference in Convo's *Program* and *Talk to Convo* pages, providing live demos for how one utterance might be understood by Convo on one page but not on the other.

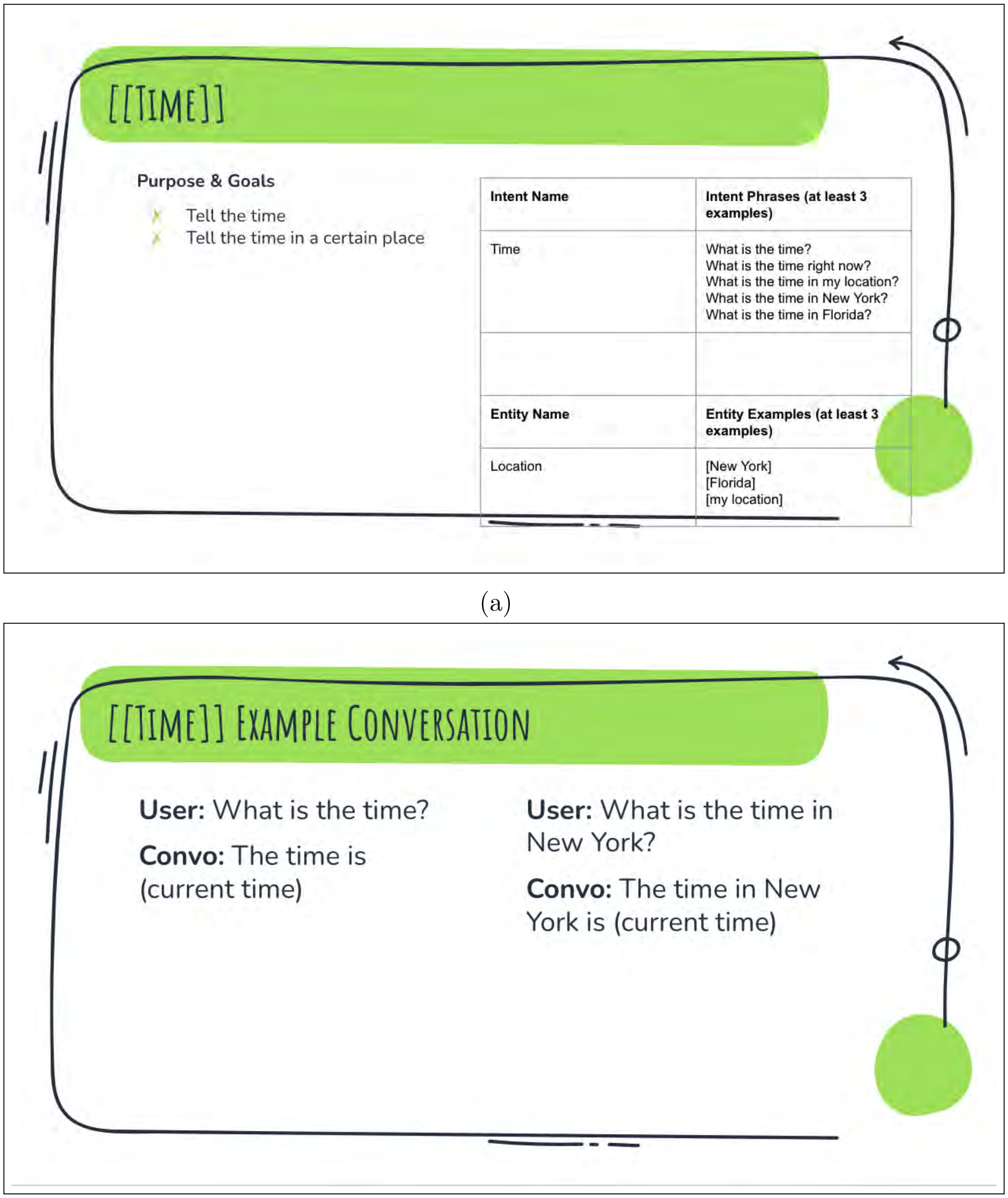
Another idea is to take another look at the avenues of communication with Convo. To start, in the post-workshop survey, 5 out of 7 students indicated that they preferred using text only for both Convo's *Program* and *Talk to Convo* pages, and the remaining two students indicated that they had no preference between typing and speaking. These results indicate that with the current version of Convo, students lean more towards typing as a way of communicating with both a conversational programming agent and a conversational AI agent as well. It would be interesting to explore more about why this might be the case, and whether or not improving the accuracy of the speech-to-text functionality of Convo might change this preference.

Finally, I propose expanding on the programming abilities of Convo. There are two main parts to this goal. The first is more straightforward, and it involves adding in new actions that Convo may perform in a procedure (e.g. multiplication, string concatenation). The next is the addition of a state to the Convo interface. Currently, Convo is able to support the creation of simple procedures, but it cannot share information across procedures. For example, if a student wanted Convo to modify a variable in one procedure and then use that same modified variable in another procedure, she would not currently be able to do so. The use of a 'state' would allow the student to create global variables that could then be used in any procedure. Additionally, this 'state' would allow the student to provide order to various intents. As an example, currently, Convo allows Janet to escape from the room Susan created by simply guessing the correct final step of unlocking the door. The addition of a 'state' would allow Susan to ensure that Janet must successfully complete all previous steps before she is able to invoke the final, room-escaping intent.

All in all, the results from the workshop show that Convo has great potential in how it can impact young students' perceptions, ideas, and literacy of conversational AI and conversational AI agents. Convo aims to lower the barrier to entry for students to learn about and create their own conversational AI agents, and I hope that Convo can bring us one step closer to empowering anyone to build agents that can solve the problems of tomorrow.

Appendix A

Figures



[[TIME]]

Purpose & Goals

- Tell the time
- Tell the time in a certain place

Intent Name	Intent Phrases (at least 3 examples)
Time	What is the time? What is the time right now? What is the time in my location? What is the time in New York? What is the time in Florida?
Entity Name	**Entity Examples (at least 3 examples)**
Location	[New York] [Florida] [my location]

(a)

[[TIME]] EXAMPLE CONVERSATION

User: What is the time?

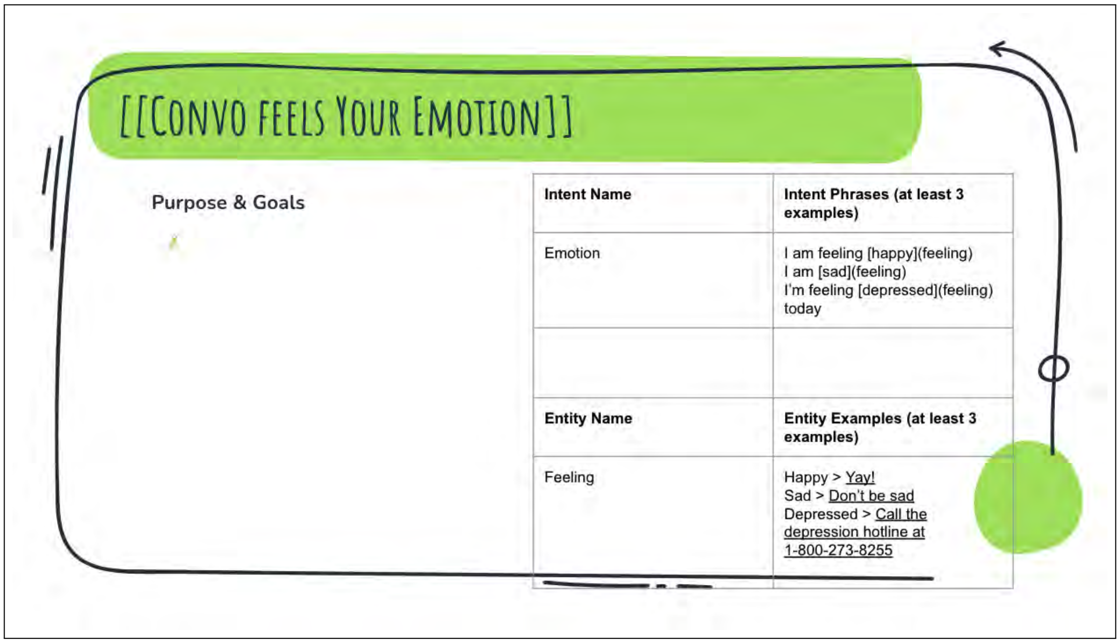
Convo: The time is (current time)

User: What is the time in New York?

Convo: The time in New York is (current time)

(b)

Figure A-1: A student came up with the idea named *Time*, which allows the user to ask Convo about the time at a certain location. In this example, the student was able to correctly identify that New York, Florida, and my location were possible entities in the intent phrases a user might speak to Convo. Based on the entities, Convo would respond with the current time in that specific location.

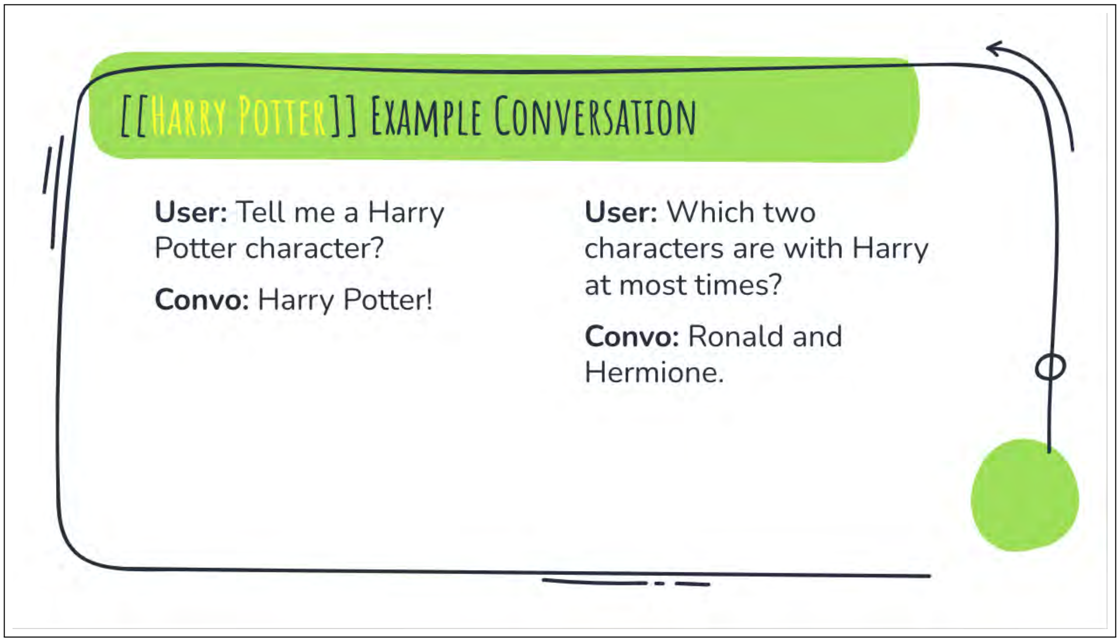


[[Convo feels Your Emotion]]

Purpose & Goals

Intent Name	Intent Phrases (at least 3 examples)
Emotion	I am feeling [happy](feeling) I am [sad](feeling) I'm feeling [depressed](feeling) today
Entity Name	**Entity Examples (at least 3 examples)**
Feeling	Happy > Yay! Sad > Don't be sad Depressed > Call the depression hotline at 1-800-273-8255

Figure A-2: A student came up with an idea called *Convo Feels Your Emotion*, which allows the user to tell Convo how they are feeling, and Convo responds back accordingly. In this slide, the student included the syntax in which to indicate the presence of entities in an intent phrase, and also provided the responses Convo would give for different emotions.



[[Harry Potter]] Example Conversation

User: Tell me a Harry Potter character?

Convo: Harry Potter!

User: Which two characters are with Harry at most times?

Convo: Ronald and Hermione.

Figure A-3: A student came up with a way for Convo to respond to a user asking about Harry Potter characters. This idea was aptly named, *Harry Potter*.

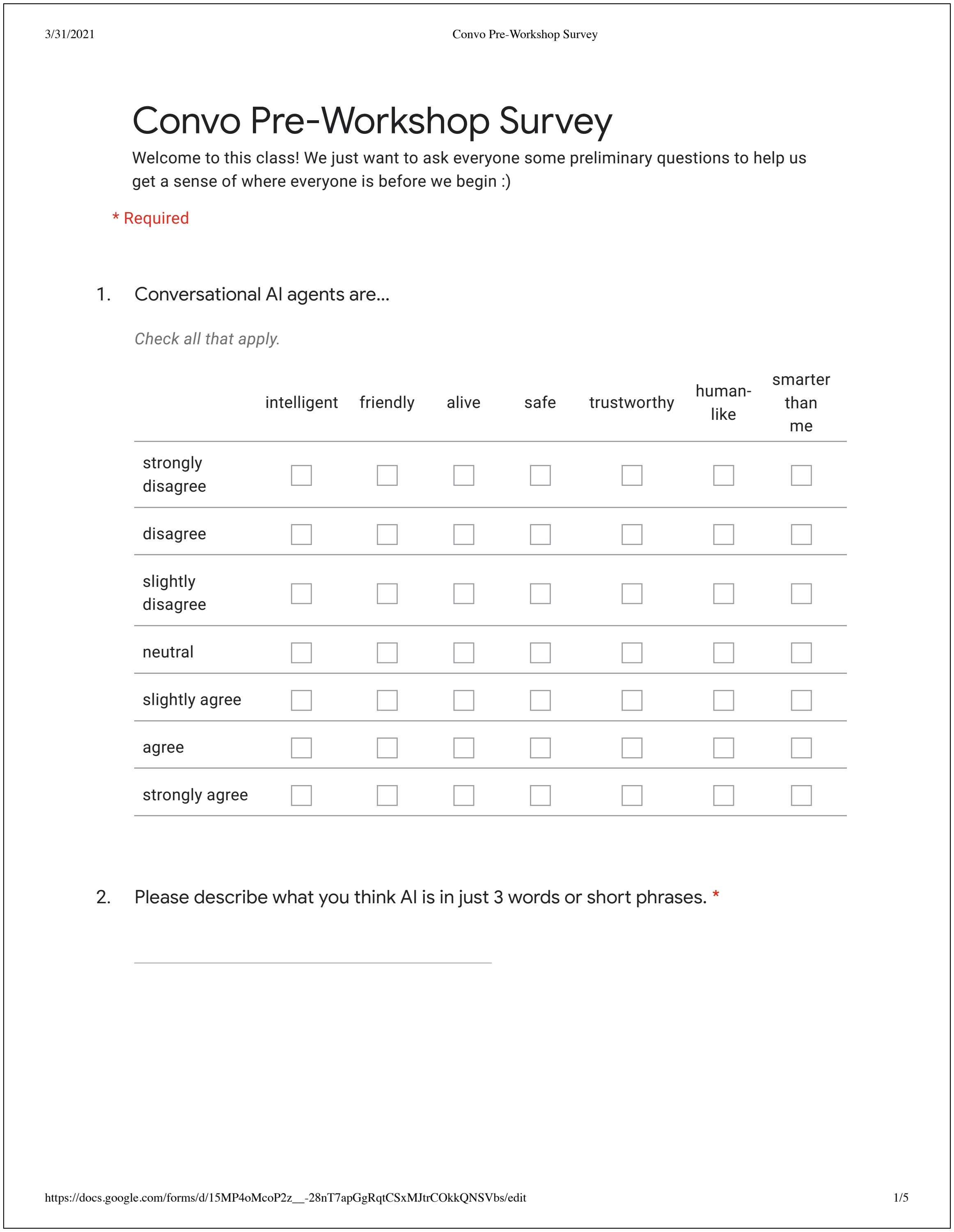
Appendix B

Workshop Materials

This appendix contains all of the workshop materials provided to the students, including the pre and post questionnaires and the CONVO tutorials.

B.1 Pre-Workshop Survey

This survey was provided to students via a Google Form at the beginning of the first day of the workshop.



3/31/2021 Convo Pre-Workshop Survey

Convo Pre-Workshop Survey

Welcome to this class! We just want to ask everyone some preliminary questions to help us get a sense of where everyone is before we begin :)

* Required

1. Conversational AI agents are...

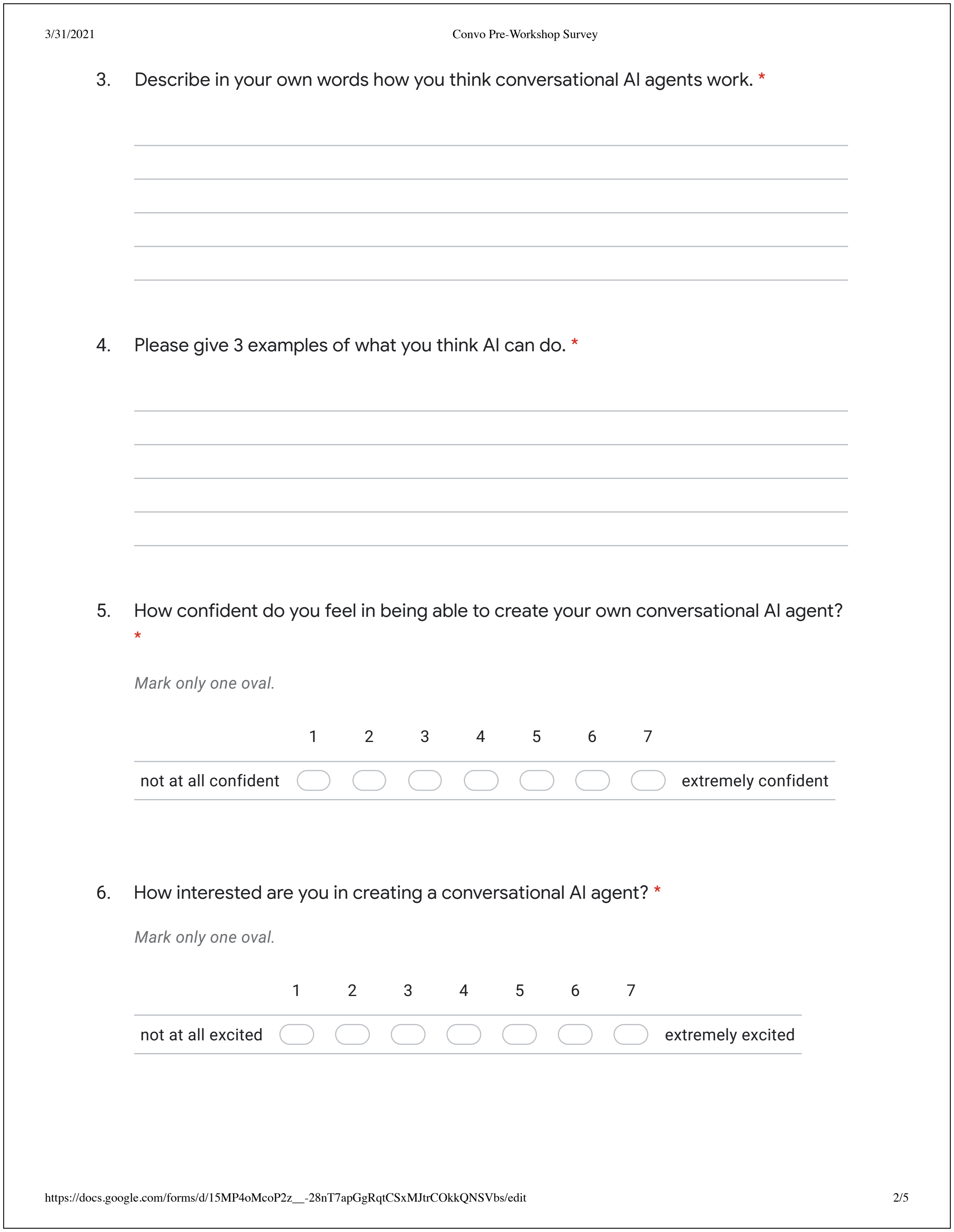
Check all that apply.

	intelligent	friendly	alive	safe	trustworthy	human-like	smarter than me
strongly disagree	☐	☐	☐	☐	☐	☐	☐
disagree	☐	☐	☐	☐	☐	☐	☐
slightly disagree	☐	☐	☐	☐	☐	☐	☐
neutral	☐	☐	☐	☐	☐	☐	☐
slightly agree	☐	☐	☐	☐	☐	☐	☐
agree	☐	☐	☐	☐	☐	☐	☐
strongly agree	☐	☐	☐	☐	☐	☐	☐

2. Please describe what you think AI is in just 3 words or short phrases. *

____

https://docs.google.com/forms/d/15MP4oMcoP2z__-28nT7apGgRqtCSxMJtrCOkkQNSVbs/edit 1/5



3/31/2021 Convo Pre-Workshop Survey

3. Describe in your own words how you think conversational AI agents work. *

4. Please give 3 examples of what you think AI can do. *

5. How confident do you feel in being able to create your own conversational AI agent? *

Mark only one oval.

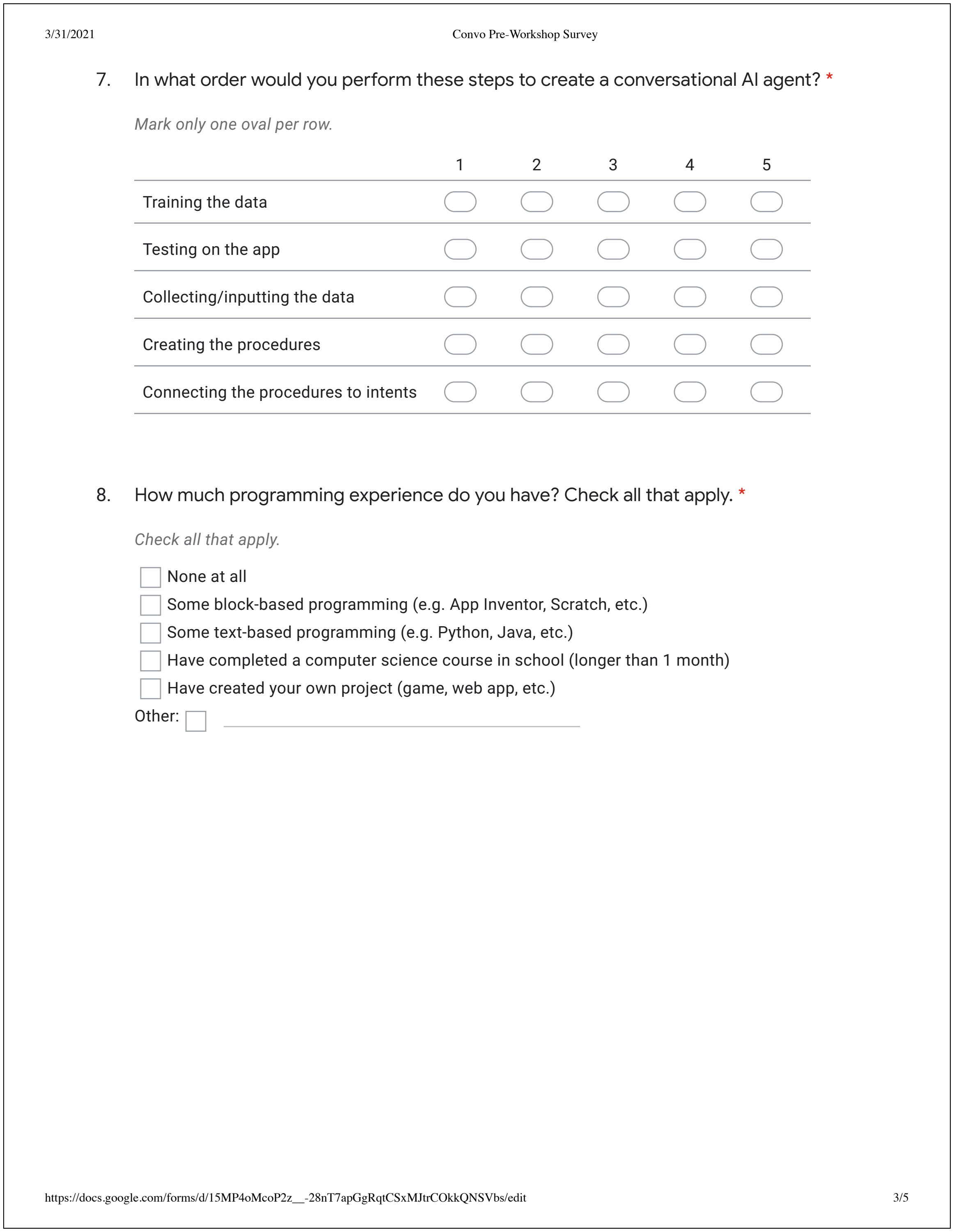
	1	2	3	4	5	6	7	
not at all confident	○	○	○	○	○	○	○	extremely confident

6. How interested are you in creating a conversational AI agent? *

Mark only one oval.

	1	2	3	4	5	6	7	
not at all excited	○	○	○	○	○	○	○	extremely excited

https://docs.google.com/forms/d/15MP4oMcoP2z__-28nT7apGgRqtCSxMJtrCOkkQNSVbs/edit 2/5



7. In what order would you perform these steps to create a conversational AI agent? *

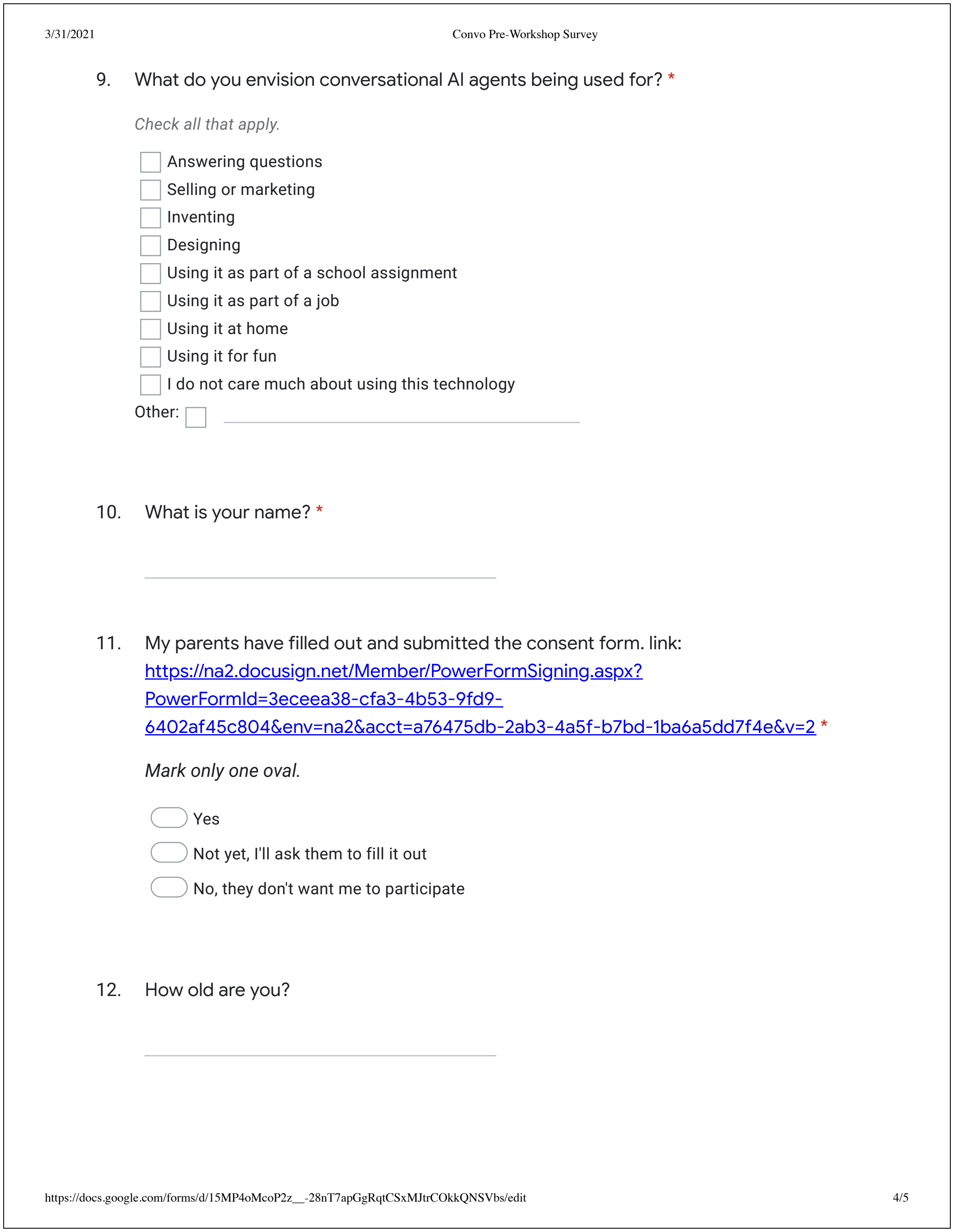
Mark only one oval per row.

	1	2	3	4	5
Training the data	○	○	○	○	○
Testing on the app	○	○	○	○	○
Collecting/inputting the data	○	○	○	○	○
Creating the procedures	○	○	○	○	○
Connecting the procedures to intents	○	○	○	○	○

8. How much programming experience do you have? Check all that apply. *

Check all that apply.

☐ None at all
☐ Some block-based programming (e.g. App Inventor, Scratch, etc.)
☐ Some text-based programming (e.g. Python, Java, etc.)
☐ Have completed a computer science course in school (longer than 1 month)
☐ Have created your own project (game, web app, etc.)
Other: ☐ ____________________



9. What do you envision conversational AI agents being used for? *

Check all that apply.

☐ Answering questions
☐ Selling or marketing
☐ Inventing
☐ Designing
☐ Using it as part of a school assignment
☐ Using it as part of a job
☐ Using it at home
☐ Using it for fun
☐ I do not care much about using this technology
Other: ☐ ____________________

10. What is your name? *

____

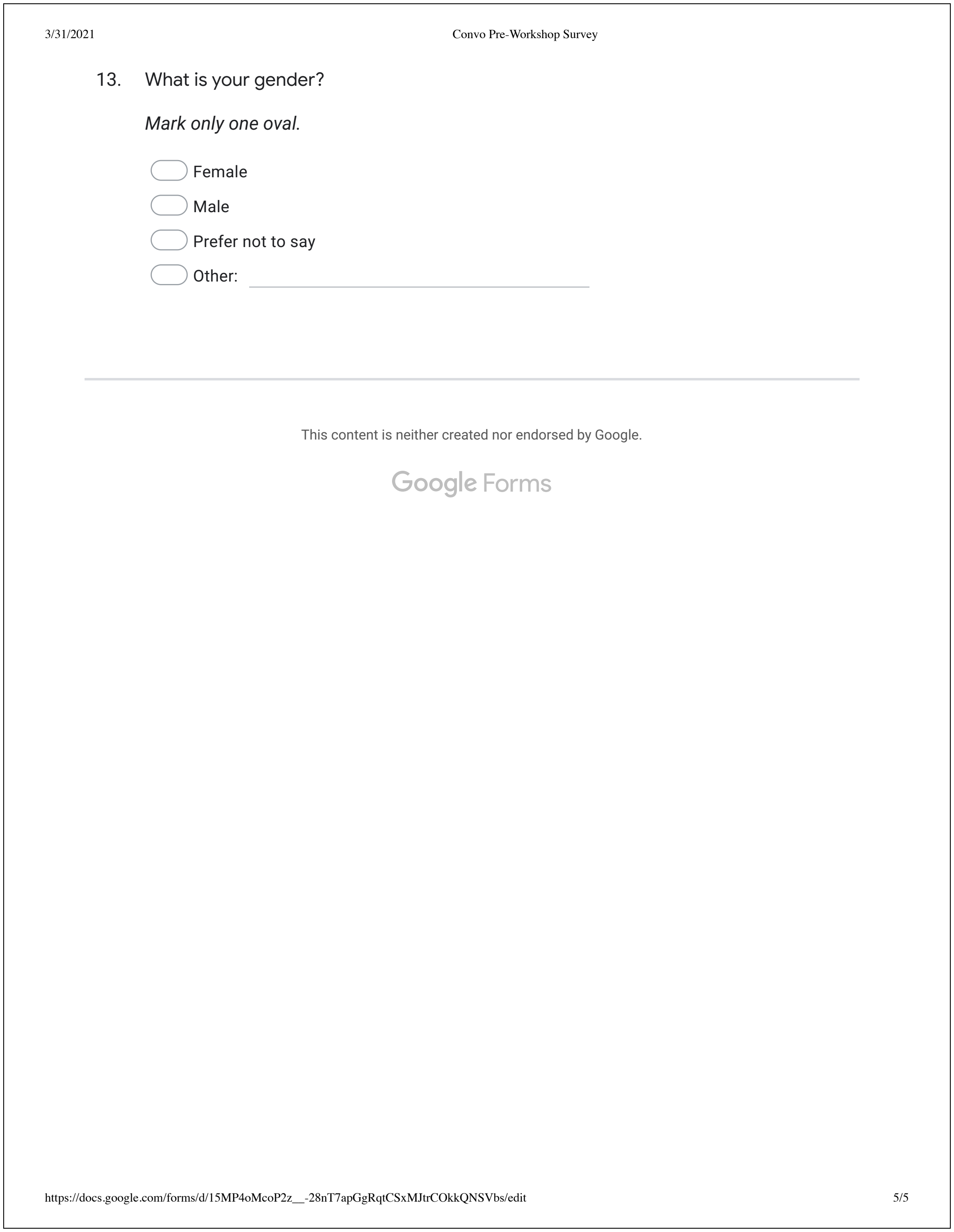
11. My parents have filled out and submitted the consent form. link: https://na2.docusign.net/Member/PowerFormSigning.aspx?PowerFormId=3eceea38-cfa3-4b53-9fd9-6402af45c804&env=na2&acct=a76475db-2ab3-4a5f-b7bd-1ba6a5dd7f4e&v=2 *

Mark only one oval.

○ Yes
○ Not yet, I'll ask them to fill it out
○ No, they don't want me to participate

12. How old are you?

____



13. What is your gender?

Mark only one oval.

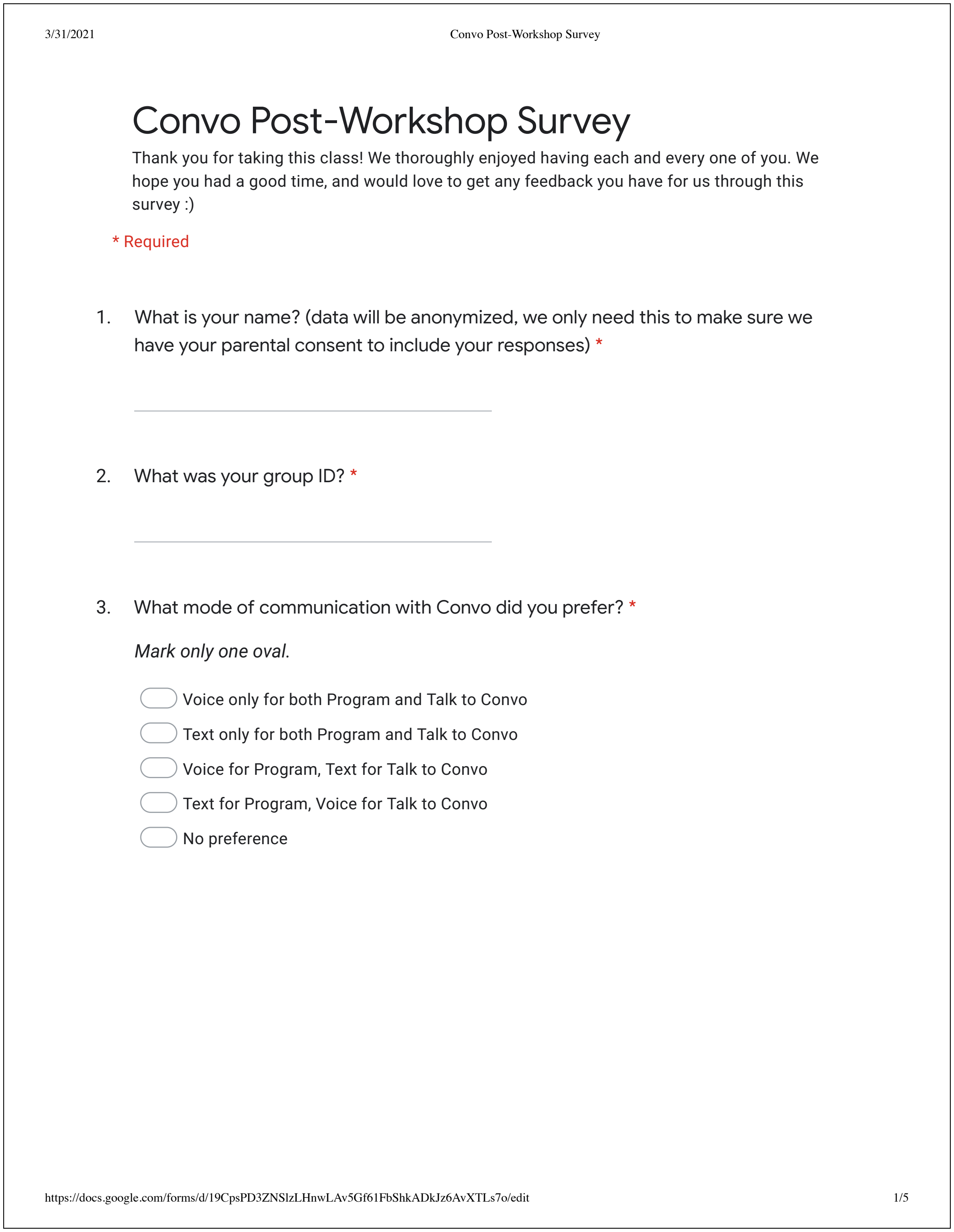
○ Female
○ Male
○ Prefer not to say
○ Other: ____________________

This content is neither created nor endorsed by Google.

Google Forms

B.2 Post-Workshop Survey

This survey was provided to students via a Google Form at the end of the last day of the workshop. Many of the questions are intentionally the same as those in the pre-workshop survey.



Convo Post-Workshop Survey

Thank you for taking this class! We thoroughly enjoyed having each and every one of you. We hope you had a good time, and would love to get any feedback you have for us through this survey :)

* Required

1. What is your name? (data will be anonymized, we only need this to make sure we have your parental consent to include your responses) *

____

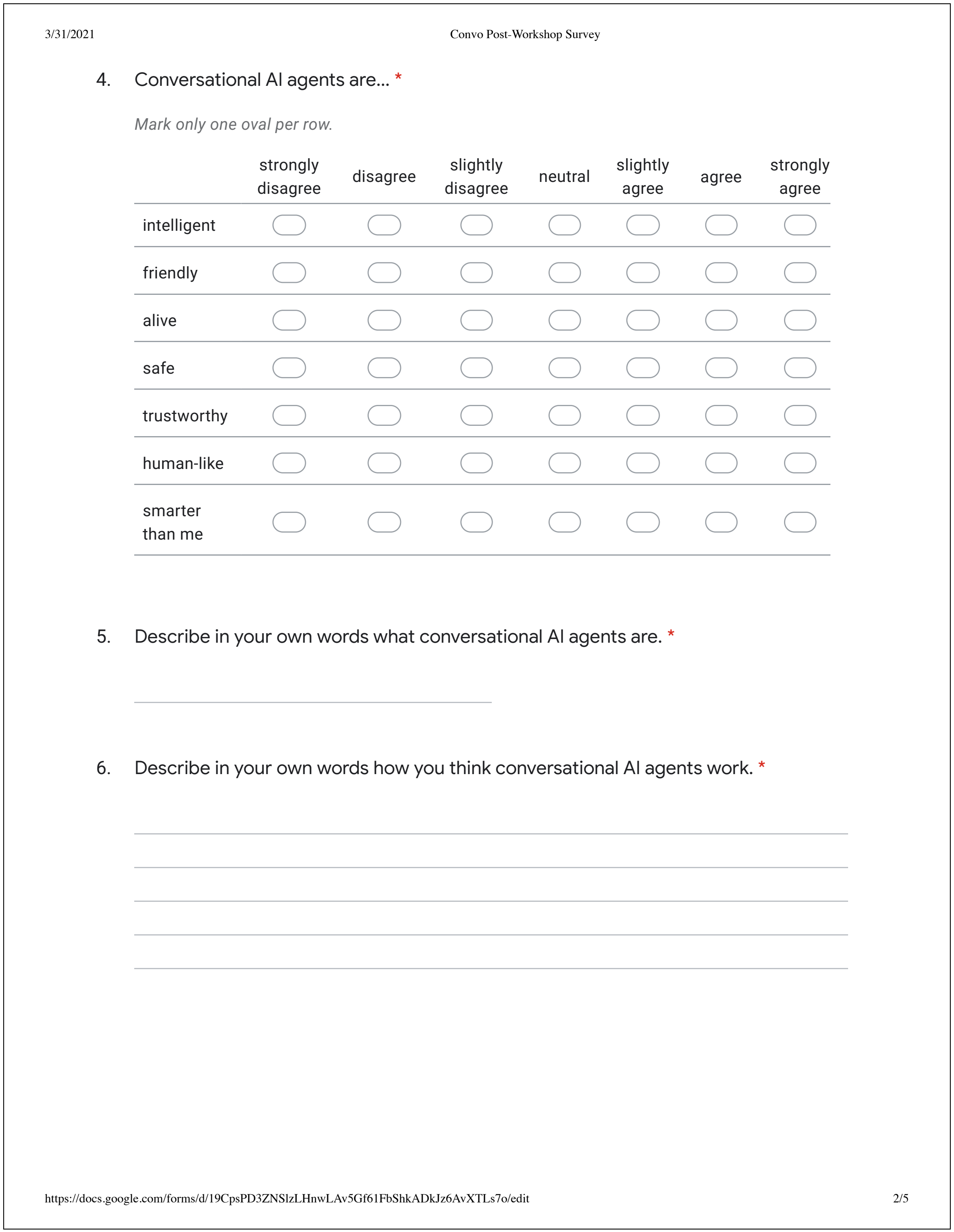
2. What was your group ID? *

____

3. What mode of communication with Convo did you prefer? *

Mark only one oval.

○ Voice only for both Program and Talk to Convo
○ Text only for both Program and Talk to Convo
○ Voice for Program, Text for Talk to Convo
○ Text for Program, Voice for Talk to Convo
○ No preference



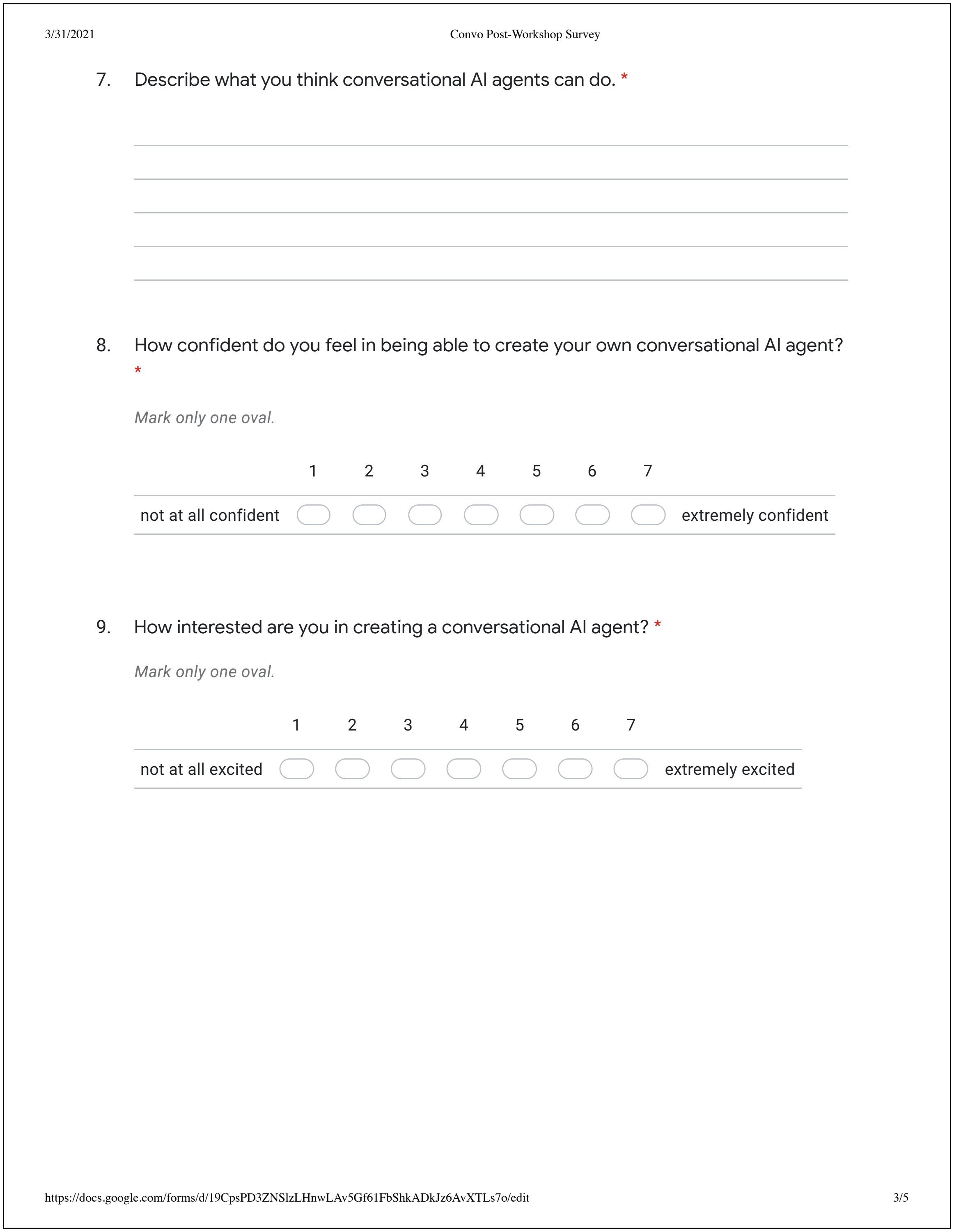
4. Conversational AI agents are… *

Mark only one oval per row.

	strongly disagree	disagree	slightly disagree	neutral	slightly agree	agree	strongly agree
intelligent	○	○	○	○	○	○	○
friendly	○	○	○	○	○	○	○
alive	○	○	○	○	○	○	○
safe	○	○	○	○	○	○	○
trustworthy	○	○	○	○	○	○	○
human-like	○	○	○	○	○	○	○
smarter than me	○	○	○	○	○	○	○

5. Describe in your own words what conversational AI agents are. *

6. Describe in your own words how you think conversational AI agents work. *



3/31/2021 Convo Post-Workshop Survey

7. Describe what you think conversational AI agents can do. *

8. How confident do you feel in being able to create your own conversational AI agent? *

Mark only one oval.

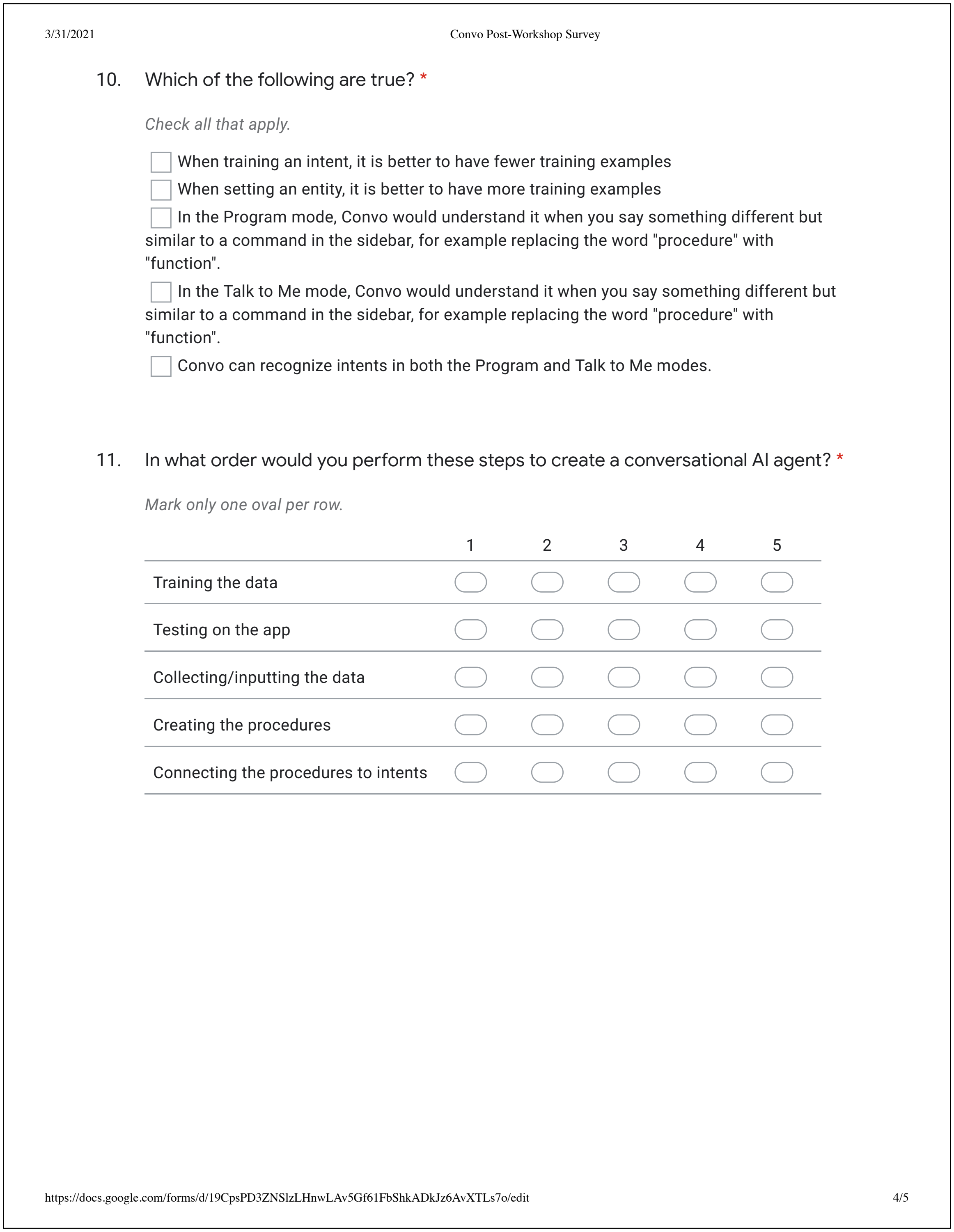
	1	2	3	4	5	6	7	
not at all confident	○	○	○	○	○	○	○	extremely confident

9. How interested are you in creating a conversational AI agent? *

Mark only one oval.

	1	2	3	4	5	6	7	
not at all excited	○	○	○	○	○	○	○	extremely excited

https://docs.google.com/forms/d/19CpsPD3ZNSlzLHnwLAv5Gf61FbShkADkJz6AvXTLs7o/edit 3/5



3/31/2021 Convo Post-Workshop Survey

10. Which of the following are true? *

Check all that apply.

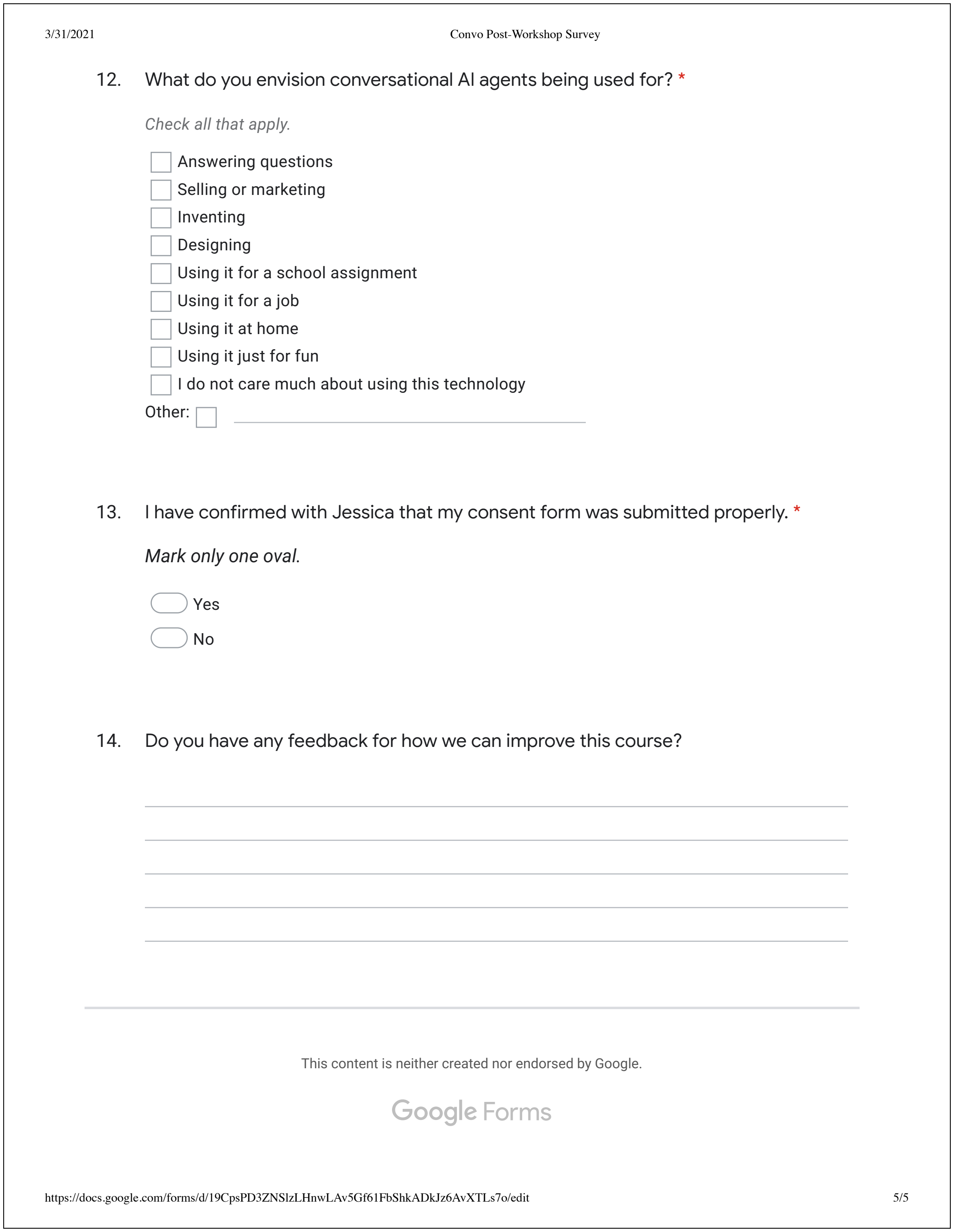
☐ When training an intent, it is better to have fewer training examples
☐ When setting an entity, it is better to have more training examples
☐ In the Program mode, Convo would understand it when you say something different but similar to a command in the sidebar, for example replacing the word "procedure" with "function".
☐ In the Talk to Me mode, Convo would understand it when you say something different but similar to a command in the sidebar, for example replacing the word "procedure" with "function".
☐ Convo can recognize intents in both the Program and Talk to Me modes.

11. In what order would you perform these steps to create a conversational AI agent? *

Mark only one oval per row.

	1	2	3	4	5
Training the data	○	○	○	○	○
Testing on the app	○	○	○	○	○
Collecting/inputting the data	○	○	○	○	○
Creating the procedures	○	○	○	○	○
Connecting the procedures to intents	○	○	○	○	○

https://docs.google.com/forms/d/19CpsPD3ZNSlzLHnwLAv5Gf61FbShkADkJz6AvXTLs7o/edit 4/5



3/31/2021 Convo Post-Workshop Survey

12. What do you envision conversational AI agents being used for? *

Check all that apply.

- ☐ Answering questions
- ☐ Selling or marketing
- ☐ Inventing
- ☐ Designing
- ☐ Using it for a school assignment
- ☐ Using it for a job
- ☐ Using it at home
- ☐ Using it just for fun
- ☐ I do not care much about using this technology

Other: ☐ ____________________

13. I have confirmed with Jessica that my consent form was submitted properly. *

Mark only one oval.

- ○ Yes
- ○ No

14. Do you have any feedback for how we can improve this course?

____

____

____

____

____

This content is neither created nor endorsed by Google.

Google Forms

https://docs.google.com/forms/d/19CpsPD3ZNSlzLHnwLAv5Gf61FbShkADkJz6AvXTLs7o/edit 5/5

B.3 Convo Tutorial 1

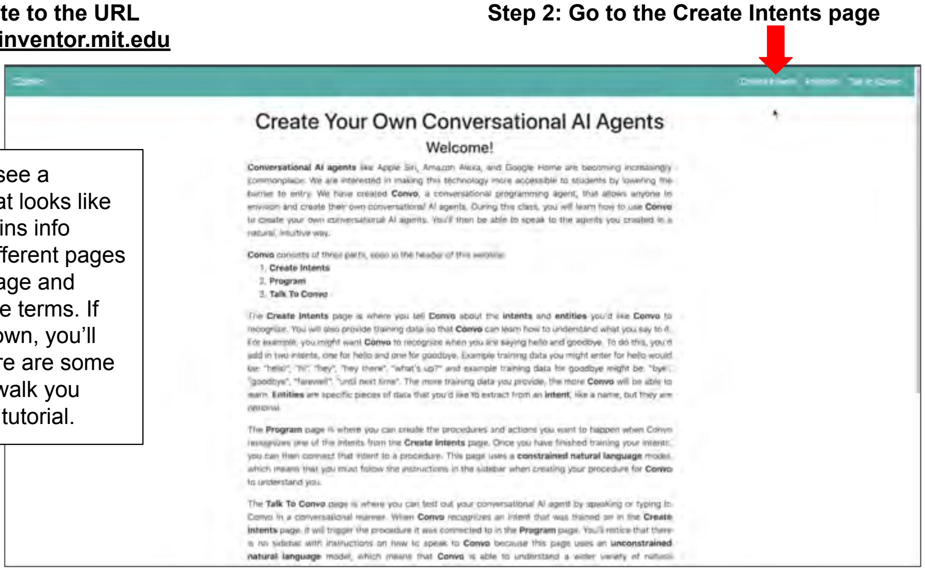
A PDF of part 1 of the Convo tutorial. This tutorial covers the basic workflow of creating a simple conversational AI agent that can respond when a user says hello or goodbye. Key concepts like intents, training, unconstrained/constrained natural language models, and connecting intents to procedures are covered. A video version of the tutorial can also be found at this link: `https://youtu.be/K-NppKDKzDY`.

Convo Tutorial Part 1

Let’s teach Convo how to greet us and say goodbye!

Step 1: Navigate to the URL
userstudy.appinventor.mit.edu

Step 2: Go to the Create Intents page



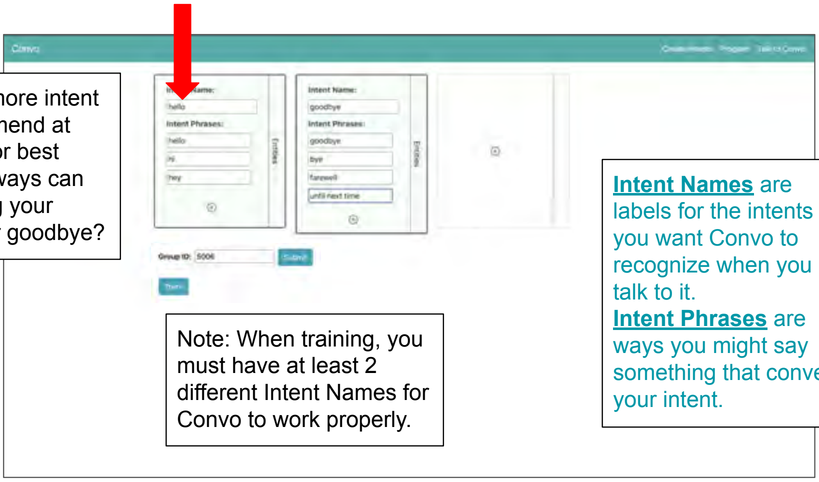
Create Your Own Conversational AI Agents

Welcome!

1. **Create Intents**
2. **Program**
3. **Talk To Convo**

You should see a webpage that looks like this. It contains info about the different pages of the webpage and defines some terms. If you scroll down, you’ll see that there are some videos that walk you through this tutorial.

Step 3: Enter in the intents and intent phrases



Intent Name: hello
Intent Phrases: hello, hi, hey

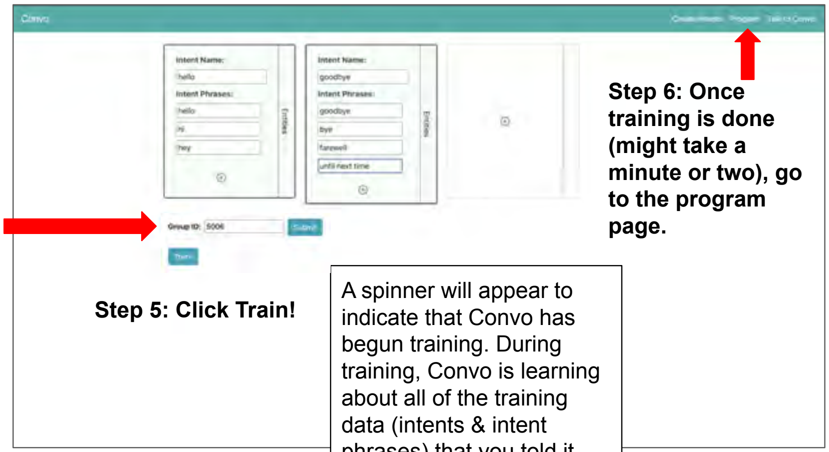
Intent Name: goodbye
Intent Phrases: goodbye, bye, farewell, until next time

Group ID: 5006

Feel free to add in more intent phrases, we recommend at least 15 per intent for best results. How many ways can you think of showing your intent to say hello or goodbye?

Note: When training, you must have at least 2 different Intent Names for Convo to work properly.

Intent Names are labels for the intents you want Convo to recognize when you talk to it.
Intent Phrases are ways you might say something that conveys your intent.



Intent Name: hello
Intent Phrases: hello, hi, hey

Intent Name: goodbye
Intent Phrases: goodbye, bye, farewell, until next time

Group ID: 5006

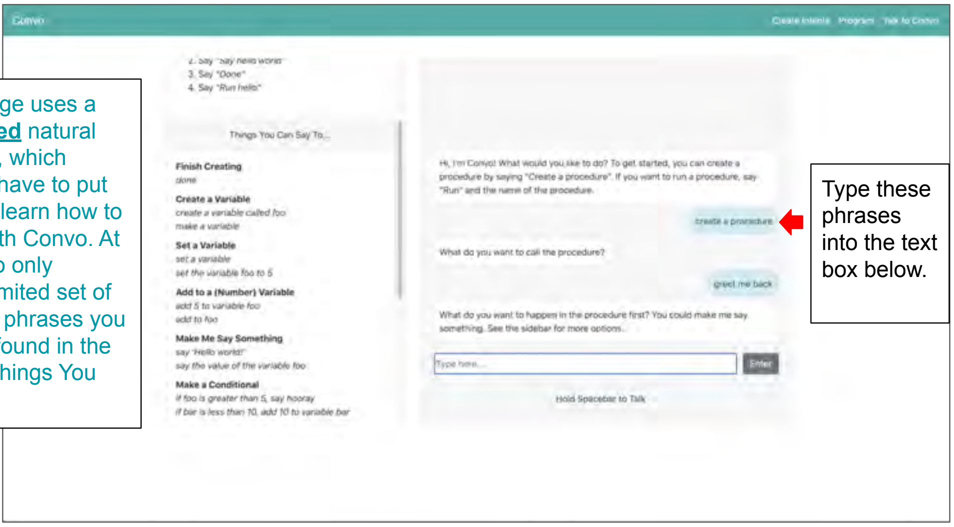
Step 4: Put in your Group ID (given to you in class) and click Submit.

Step 5: Click Train!

Step 6: Once training is done (might take a minute or two), go to the program page.

A spinner will appear to indicate that Convo has begun training. During training, Convo is learning about all of the training data (intents & intent phrases) that you told it about.

Step 7: Create the procedure you want Convo to do when you greet it.



2. Say "Say hello world"
3. Say "Done"
4. Say "Run hello"

Things You Can Say To...

Finish Creating
done

Create a Variable
create a variable called foo
make a variable

Set a Variable
set a variable
set the variable foo to 5

Add to a (Number) Variable
add 5 to variable foo
add to foo

Make Me Say Something
say 'Hello world!'
say the value of the variable foo

Make a Conditional
if foo is greater than 5, say hooray
if bar is less than 10, add 10 to variable bar

Hi, I'm Convo! What would you like to do? To get started, you can create a procedure by saying "Create a procedure". If you want to run a procedure, say "Run" and the name of the procedure.

create a procedure

What do you want to call the procedure?

greet me back

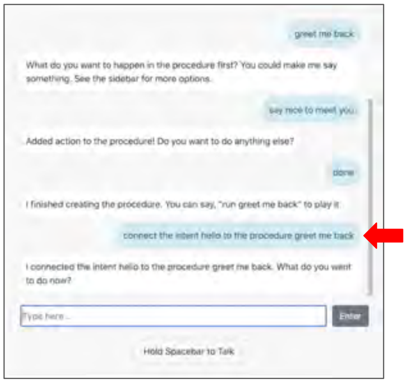
What do you want to happen in the procedure first? You could make me say something. See the sidebar for more options.

The Program page uses a more **constrained** natural language model, which means that you have to put in some work to learn how to communicate with Convo. At any point, Convo only understands a limited set of commands. The phrases you can say can be found in the sidebar under 'Things You Can Say To…'

Type these phrases into the text box below.

Step 8: Finish adding actions to the procedure.

We start off by telling Convo that we want to create a procedure. Then, we name our procedure 'greet me back'. You can choose a different name if you wish. We want Convo to say 'nice to meet you' when it detects that we are saying hello. We finish by typing 'done'.



greet me back

What do you want to happen in the procedure first? You could make me say something. See the sidebar for more options.

say nice to meet you

Added action to the procedure! Do you want to do anything else?

done

I finished creating the procedure. You can say, "run greet me back" to play it

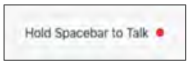
connect the intent hello to the procedure greet me back

I connected the intent hello to the procedure greet me back. What do you want to do now?

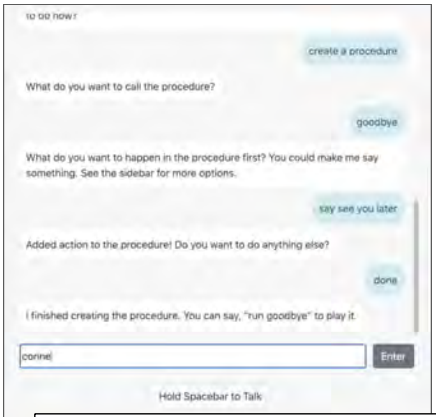
Step 9: Connect the procedure to the corresponding intent.

Connecting a procedure to an intent will make Convo run the procedure you just created whenever the connected intent is triggered on the Talk to Convo page.

You can choose to either type or talk to Convo. To talk, either hold the record button, or hold your spacebar (click outside of the chatbox first). When recording, the red light should turn on:



Step 10: Create the procedure you want Convo to do when you say goodbye.



to do now?

create a procedure

What do you want to call the procedure?

goodbye

What do you want to happen in the procedure first? You could make me say something. See the sidebar for more options.

say see you later

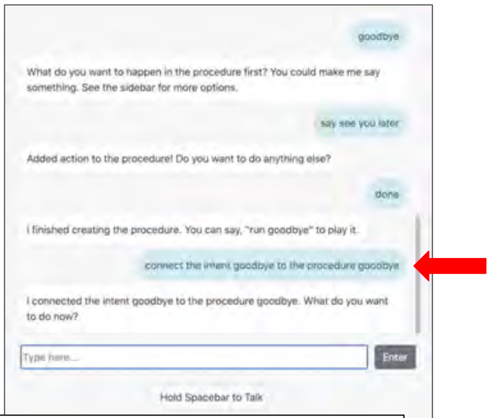
Added action to the procedure! Do you want to do anything else?

done

I finished creating the procedure. You can say, "run goodbye" to play it.

connel

Step 11: Connect the goodbye procedure to the intent goodbye.



goodbye

What do you want to happen in the procedure first? You could make me say something. See the sidebar for more options.

say see you later

Added action to the procedure! Do you want to do anything else?

done

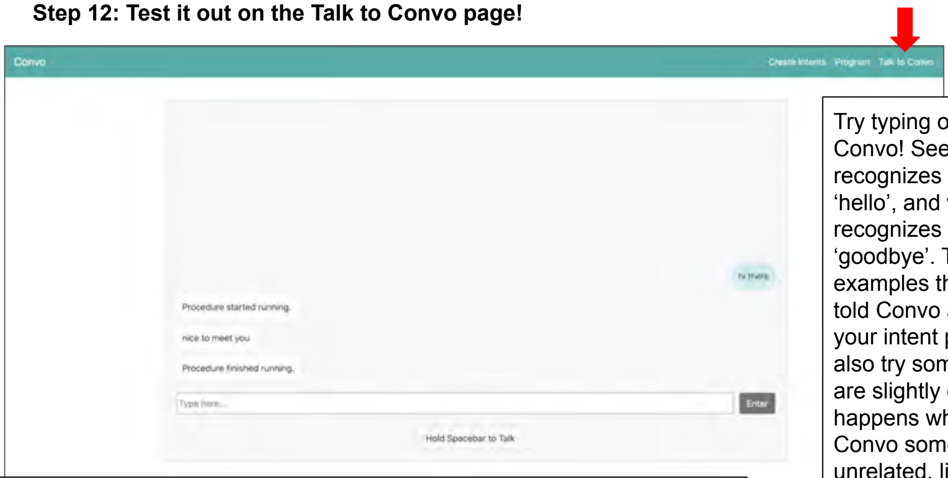
I finished creating the procedure. You can say, "run goodbye" to play it.

connect the intent goodbye to the procedure goodbye

I connected the intent goodbye to the procedure goodbye. What do you want to do now?

This is the same process as creating the procedure 'greet me back'. You can choose to have Convo say something different, or you can also add in more actions from the choices in the sidebar. Connecting the intent to the procedure executes (runs) this procedure (or set of actions) whenever the intent 'goodbye' is detected by Convo.

Step 12: Test it out on the Talk to Convo page!



hi there

Procedure started running.

nice to meet you

Procedure finished running.

Try typing or talking to Convo! See what Convo recognizes as the intent 'hello', and what it recognizes as the intent 'goodbye'. Try saying examples that you directly told Convo about through your intent phrases, but also try some things that are slightly different. What happens when you tell Convo something super unrelated, like "cats and dogs"?

The Talk to Convo page uses a more **unconstrained** natural language model, which means that there's no sidebar of acceptable commands. Convo will do its best to interpret what you say and match it to a known intent.

B.4 Convo Tutorial 2

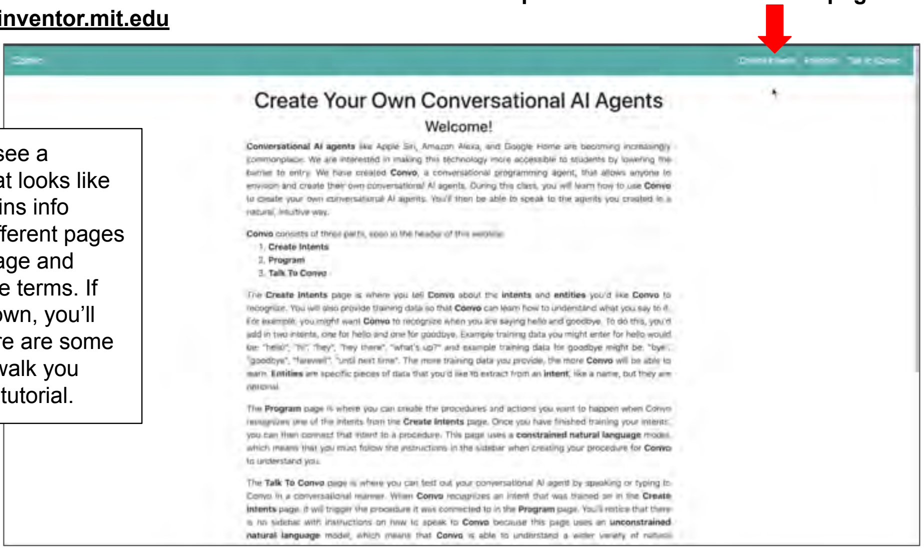
A PDF of part 2 of the Convo tutorial. This tutorial assumes that the user already understands all of the key concepts from tutorial 1, in Appendix B.3. This portion of the tutorial covers entities – what they are, how to use them, and how to teach Convo about them. It also goes into how to create a more complex procedure with logic statements. A video version of the tutorial can also be found at this link: `https://youtu.be/Yio48CwVaR4`.

Convo Tutorial Part 2

Let’s teach Convo to tell us the weather for a certain city!

Step 1: Navigate to the URL <u>userstudy.appinventor.mit.edu</u>

Step 2: Go to the Create Intents page.



Create Your Own Conversational AI Agents

Welcome!

Conversational AI agents like Apple Siri, Amazon Alexa, and Google Home are becoming increasingly commonplace. We are interested in making this technology more accessible to students by lowering the barrier to entry. We have created **Convo**, a conversational programming agent, that allows anyone to envision and create their own conversational AI agents. During this class, you will learn how to use **Convo** to create your own conversational AI agents. You'll then be able to speak to the agents you created in a natural, intuitive way.

Convo consists of three parts, seen in the header of this website:

1. **Create Intents**
2. **Program**
3. **Talk To Convo**

The **Create Intents** page is where you tell **Convo** about the **intents** and **entities** you'd like **Convo** to recognize. You will also provide training data so that **Convo** can learn how to understand what you say to it. For example, you might want **Convo** to recognize when you are saying hello and goodbye. To do this, you'd add in two intents, one for hello and one for goodbye. Example training data you might enter for hello would be: "hello", "hi", "hey", "hey there", "what's up?" and example training data for goodbye might be: "bye", "goodbye", "farewell", "until next time". The more training data you provide, the more **Convo** will be able to learn. **Entities** are specific pieces of data that you'd like to extract from an **intent**, like a name, but they are optional.

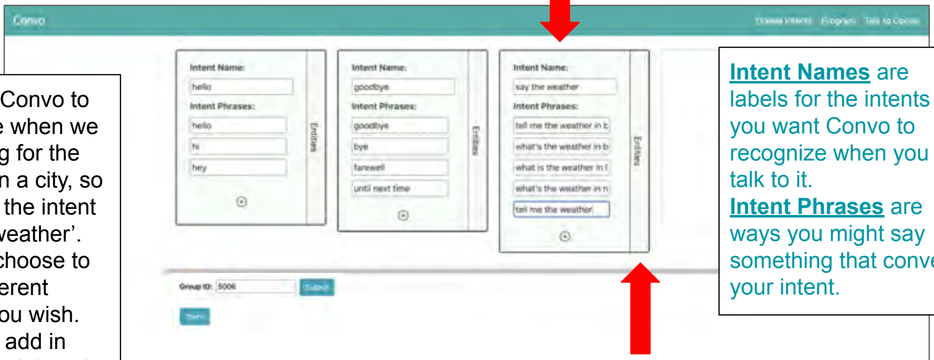
The **Program** page is where you can create the procedures and actions you want to happen when Convo recognizes one of the intents from the **Create Intents** page. Once you have finished training your intents, you can then connect that intent to a procedure. This page uses a **constrained natural language** model, which means that you must follow the instructions in the sidebar when creating your procedure for **Convo** to understand you.

The **Talk To Convo** page is where you can test out your conversational AI agent by speaking or typing to Convo in a conversational manner. When **Convo** recognizes an intent that was trained on in the **Create Intents** page, it will trigger the procedure it was connected to in the **Program** page. You'll notice that there is no sidebar with instructions on how to speak to **Convo** because this page uses an **unconstrained natural language** model, which means that **Convo** is able to understand a wider variety of natural

You should see a webpage that looks like this. It contains info about the different pages of the webpage and defines some terms. If you scroll down, you'll see that there are some videos that walk you through this tutorial.

Step 3: Enter in the intents and intent phrases.

We want Convo to recognize when we are asking for the weather in a city, so we name the intent ‘say the weather’. You can choose to use a different name if you wish. Then, we add in ways we might ask for the weather of a city, using various city names.



Intent Name: hello
Intent Phrases: hello, hi, hey

Intent Name: goodbye
Intent Phrases: goodbye, bye, farewell, until next time

Intent Name: say the weather
Intent Phrases: tell me the weather in b, what's the weather in b, what is the weather in l, what's the weather in n, tell me the weather

Group ID: 5006

Intent Names are labels for the intents you want Convo to recognize when you talk to it.
Intent Phrases are ways you might say something that conveys your intent.

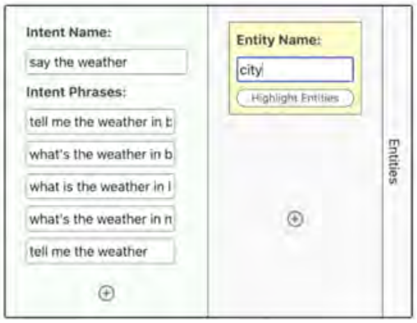
Step 4: Open the Entities tab.

Like in part 1 of the tutorial, we recommend adding *at least 15 intent phrase examples* for Convo to be more accurate.

Step 5: Add an entity.

We want Convo to be able to extract the ‘city’ from what we say, so that it can give us the weather for that specific city. To do this, we need to tell Convo to look for cities, and we do this by creating an entity.

As with intent phrases, Convo will also be more accurate the more entity examples you give it. We recommend at least 15 unique entity values (in this case, we used Boston twice, which only counts as 1 unique entity value).



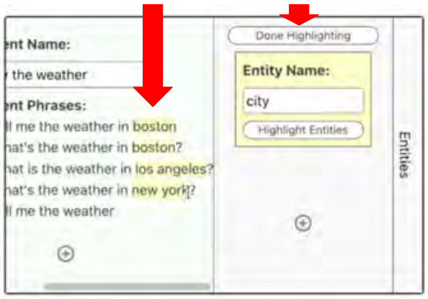
Intent Name: say the weather
Intent Phrases: tell me the weather in b, what's the weather in b, what is the weather in l, what's the weather in n, tell me the weather

Entity Name: city

Entities are pieces of information we want Convo to be able to extract from what we tell it.

You can have multiple entities per intent. You would need multiple entities if you are trying to extract multiple pieces of information, such as getting the *date* in addition to the *city* for a weather forecast. The current limit is 6 entities per intent.

Step 6: Highlight Entities. **Step 7: Click Done Highlighting.**



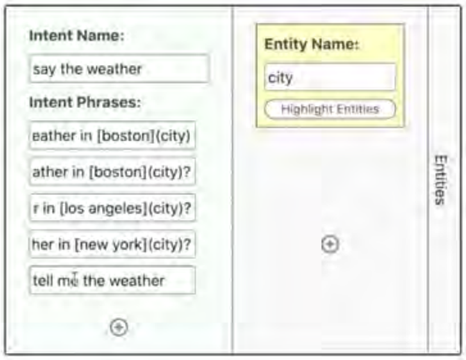
nt Name:
the weather
nt Phrases:
l me the weather in boston
hat's the weather in boston?
hat is the weather in los angeles?
hat's the weather in new york?
l me the weather

Entity Name:
city

Simply drag your cursor over words that you'd want Convo to understand as a 'city'. Please be careful during this step since there's currently no undo button (at this step). If you do make a mistake, no worries. Simply click Done Highlighting and manually edit the Intent Phrases. The entity value should be in hard brackets [], followed by the entity name in parenthesis (), with no space in between.

(Alternatively, you could directly enter in your intent phrases with the entity syntax, [*boston*](*city*), and you can skip the highlighting entirely.)

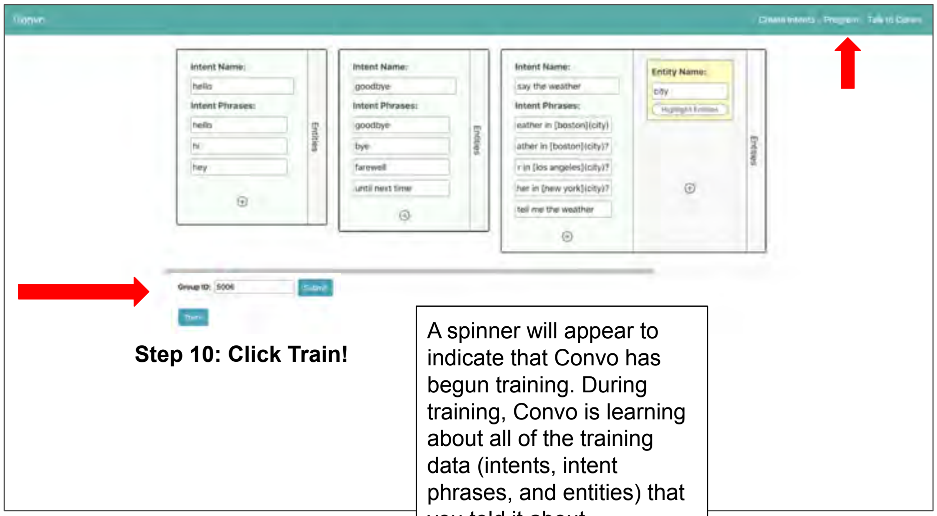
Step 8: Fix any mistakes.



Intent Name:
say the weather
Intent Phrases:
eather in [boston](city)
ather in [boston](city)?
r in [los angeles](city)?
her in [new york](city)?
tell me the weather

Entity Name:
city

Step 11: Once training is done, go to the program page.



Intent Name:
hello
Intent Phrases:
hello
hi
hey

Intent Name:
goodbye
Intent Phrases:
goodbye
bye
farewell
until next time

Intent Name:
say the weather
Intent Phrases:
eather in [boston](city)
ather in [boston](city)?
r in [los angeles](city)?
her in [new york](city)?
tell me the weather

Entity Name:
city

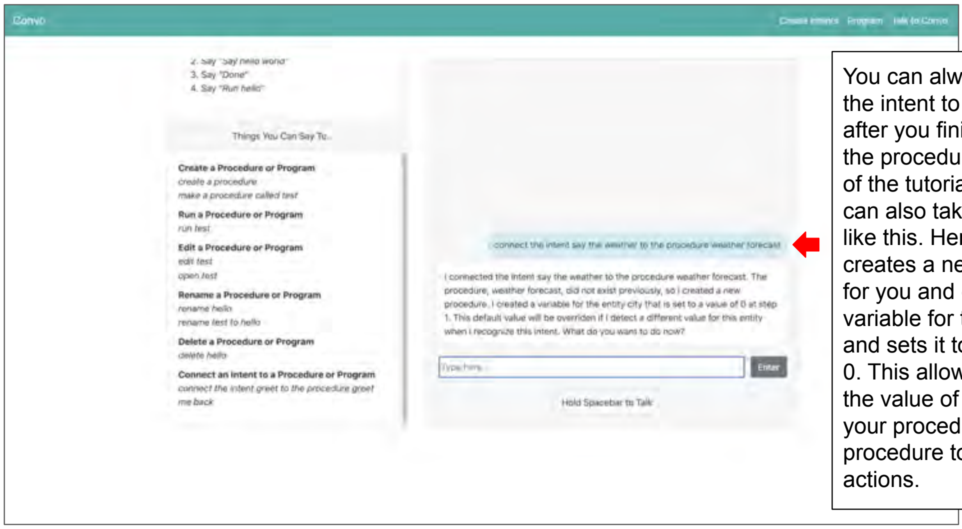
Group ID: 5006

Step 9: Put in your Group ID (given to you in class) and click Submit.

Step 10: Click Train!

A spinner will appear to indicate that Convo has begun training. During training, Convo is learning about all of the training data (intents, intent phrases, and entities) that you told it about.

Step 12: Connect the intent to a procedure.



2. Say "Say hello world"
3. Say "Done"
4. Say "Run hello"

Things You Can Say To...

Create a Procedure or Program
create a procedure
make a procedure called test

Run a Procedure or Program
run test

Edit a Procedure or Program
edit test
open test

Rename a Procedure or Program
rename hello
rename test to hello

Delete a Procedure or Program
delete hello

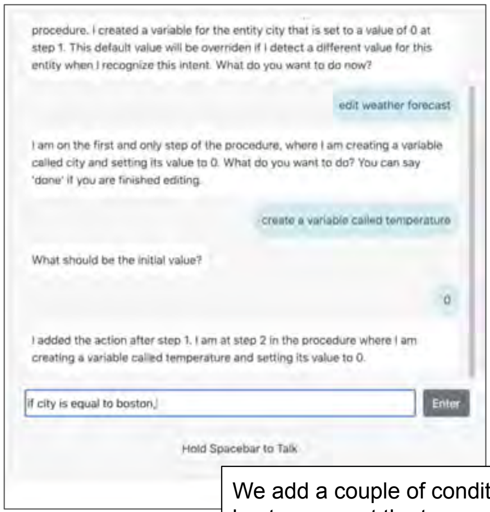
Connect an Intent to a Procedure or Program
connect the intent greet to the procedure greet me back

connect the intent say the weather to the procedure weather forecast

I connected the intent say the weather to the procedure weather forecast. The procedure, weather forecast, did not exist previously, so I created a new procedure. I created a variable for the entity city that is set to a value of 0 at step 1. This default value will be overriden if I detect a different value for this entity when I recognize this intent. What do you want to do now?

You can always connect the intent to a procedure after you finish making the procedure like in part of the tutorial, but you can also take a shortcut like this. Here, Convo creates a new procedure for you and creates a variable for the entity and sets it to a value of 0. This allows you to use the value of city within your procedure. Edit the procedure to add actions.

Step 13: Finish adding actions to the procedure.


procedure. I created a variable for the entity city that is set to a value of 0 at step 1. This default value will be overriden if I detect a different value for this entity when I recognize this intent. What do you want to do now?

edit weather forecast

I am on the first and only step of the procedure, where I am creating a variable called city and setting its value to 0. What do you want to do? You can say 'done' if you are finished editing.

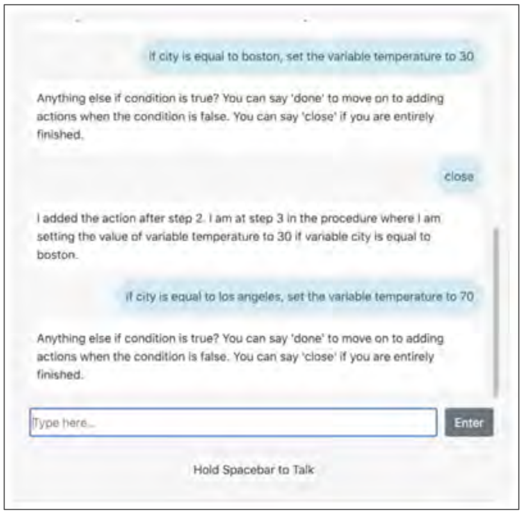
create a variable called temperature

What should be the initial value?

0

I added the action after step 1. I am at step 2 in the procedure where I am creating a variable called temperature and setting its value to 0.

if city is equal to boston,


if city is equal to boston, set the variable temperature to 30

Anything else if condition is true? You can say 'done' to move on to adding actions when the condition is false. You can say 'close' if you are entirely finished.

close

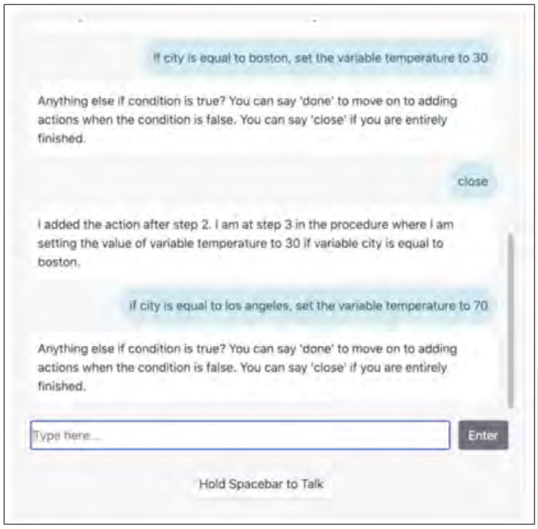
I added the action after step 2. I am at step 3 in the procedure where I am setting the value of variable temperature to 30 if variable city is equal to boston.

if city is equal to los angeles, set the variable temperature to 70

Anything else if condition is true? You can say 'done' to move on to adding actions when the condition is false. You can say 'close' if you are entirely finished.

We add a couple of conditions. If Convo detects that a user is asking about boston, we set the temperature to a value of 30. If the city is los angeles, we set the temperature to a value of 70. You may choose to do different actions here.

Step 13 continued: Finish adding actions to the procedure.



if city is equal to boston, set the variable temperature to 30

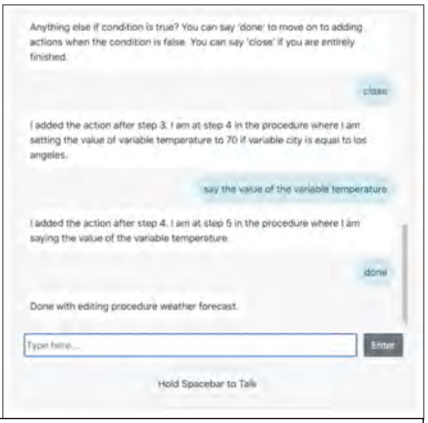
Anything else if condition is true? You can say 'done' to move on to adding actions when the condition is false. You can say 'close' if you are entirely finished.

close

I added the action after step 2. I am at step 3 in the procedure where I am setting the value of variable temperature to 30 if variable city is equal to boston.

if city is equal to los angeles, set the variable temperature to 70

Anything else if condition is true? You can say 'done' to move on to adding actions when the condition is false. You can say 'close' if you are entirely finished.



Anything else if condition is true? You can say 'done' to move on to adding actions when the condition is false. You can say 'close' if you are entirely finished.

close

I added the action after step 3. I am at step 4 in the procedure where I am setting the value of variable temperature to 70 if variable city is equal to los angeles.

say the value of the variable temperature

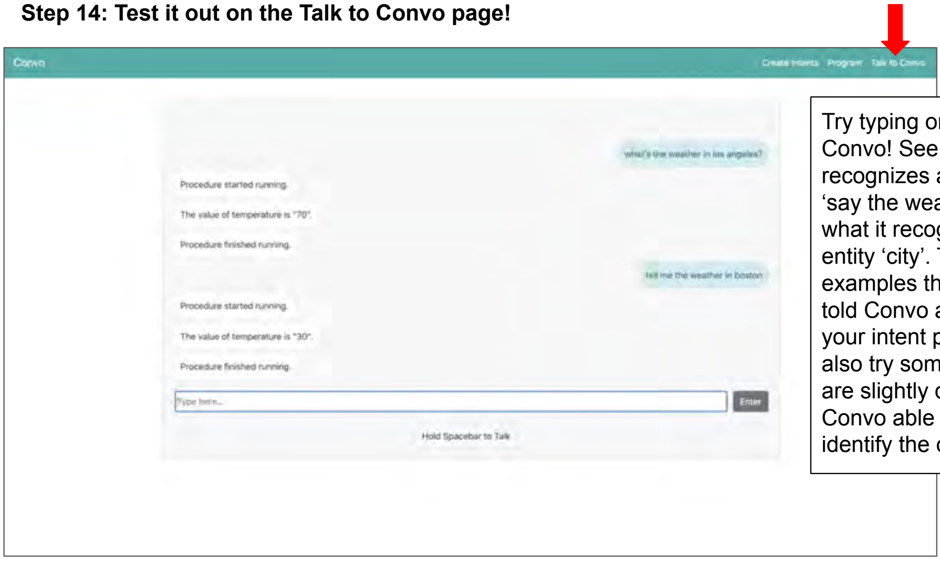
I added the action after step 4. I am at step 5 in the procedure where I am saying the value of the variable temperature

done

Done with editing procedure weather forecast.

We finish by saying the value of the variable temperature, which will be 30 if the city is boston, 70 if the city is los angeles, and 0 in all other cases. We indicate that we are done with adding actions by saying ‘done’. Move to the Talk to Convo page to proceed.

Step 14: Test it out on the Talk to Convo page!



what's the weather in los angeles?

Procedure started running.

The value of temperature is "70".

Procedure finished running.

tell me the weather in boston

Procedure started running.

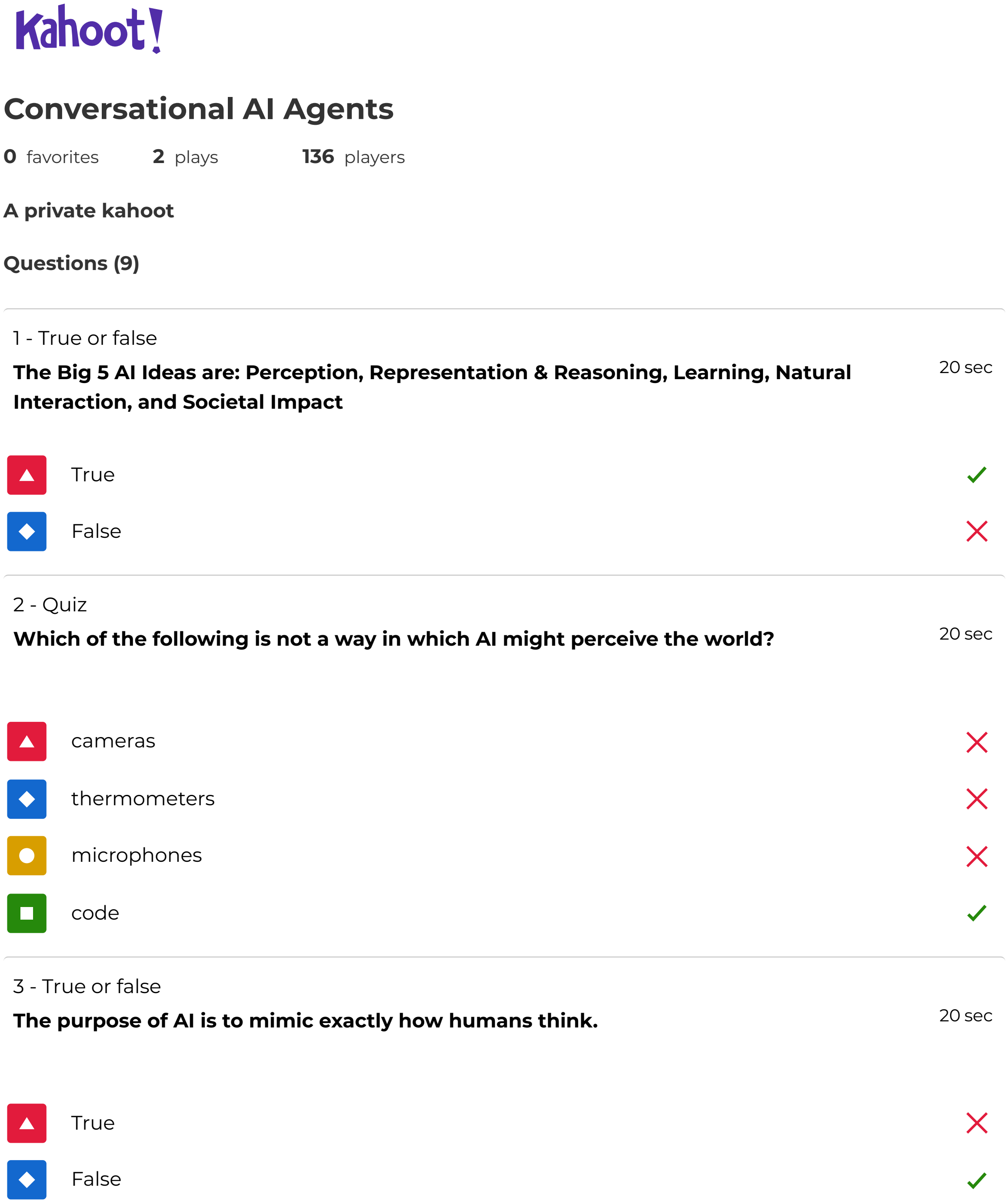
The value of temperature is "30".

Procedure finished running.

Try typing or talking to Convo! See what Convo recognizes as the intent ‘say the weather’, and what it recognizes as the entity ‘city’. Try saying examples that you directly told Convo about through your intent phrases, but also try some things that are slightly different. Is Convo able to always identify the correct city?

B.5 Kahoot Quiz

I had students participate in this Kahoot quiz at the end of the first day of the workshop to recap some of the main key concepts I had covered [40]. The full list of questions as well as all possible answers can be found below.



Kahoot!

Conversational AI Agents

0 favorites **2** plays **136** players

A private kahoot

Questions (9)

1 - True or false

The Big 5 AI Ideas are: Perception, Representation & Reasoning, Learning, Natural Interaction, and Societal Impact 20 sec

▲ True ✓

◆ False ✗

2 - Quiz

Which of the following is not a way in which AI might perceive the world? 20 sec

▲ cameras ✗

◆ thermometers ✗

● microphones ✗

■ code ✓

3 - True or false

The purpose of AI is to mimic exactly how humans think. 20 sec

▲ True ✗

◆ False ✓

4 - Quiz

Today's AI agents are...

20 sec

▲	smarter than a human child	✗
◆	always a positive force for society	✗
●	unbiased	✗
■	none of the above	✓

5 - True or false

Conversational AI agents use either a constrained or unconstrained natural language model

20 sec

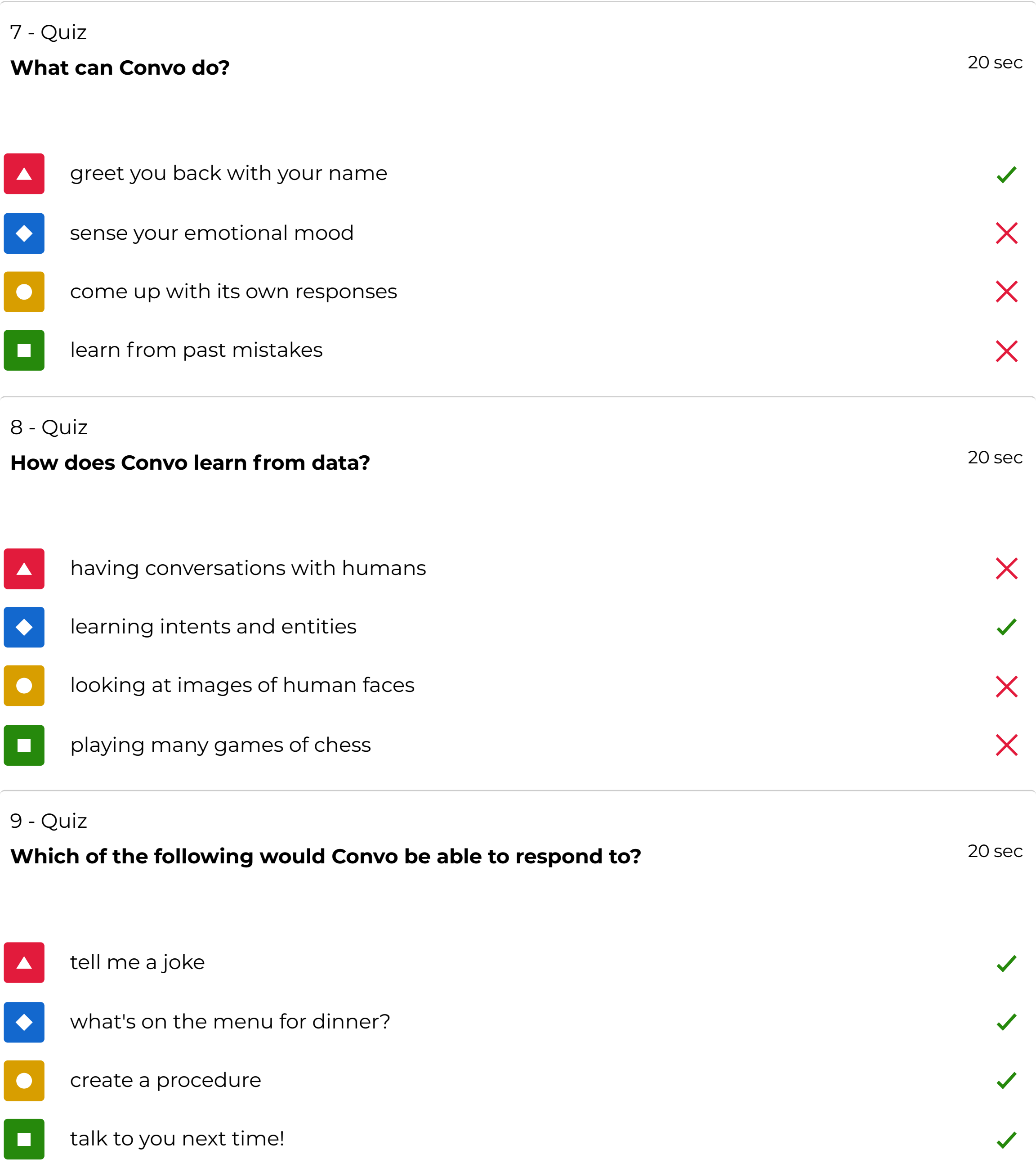
▲	True	✗
◆	False	✓

6 - Quiz

What is Convo?

20 sec

▲	a chatbot that can also program	✓
◆	a conversational programming agent	✓
●	an all-knowing AI assistant	✗
■	a tool to help students learn about conversational AI	✓



7 - Quiz

What can Convo do?

20 sec

▲	greet you back with your name	✓
◆	sense your emotional mood	✗
●	come up with its own responses	✗
■	learn from past mistakes	✗

8 - Quiz

How does Convo learn from data?

20 sec

▲	having conversations with humans	✗
◆	learning intents and entities	✓
●	looking at images of human faces	✗
■	playing many games of chess	✗

9 - Quiz

Which of the following would Convo be able to respond to?

20 sec

▲	tell me a joke	✓
◆	what's on the menu for dinner?	✓
●	create a procedure	✓
■	talk to you next time!	✓

Bibliography

[1] Basics. `https://styled-components.com/docs/basics`, Last accessed on 2021-04-20.

[2] Spark! `https://esp.mit.edu/teach/Spark/index.html`. Accessed: 2021-05-31.

[3] Components, 2020. `https://rasa.com/docs/rasa/components`, Last accessed on 2020-11-15.

[4] Rasa, 2020. `https://rasa.com/`, Last accessed on 2020-09-26.

[5] requests 2.25.1, 2020. `https://pypi.org/project/requests/`, Last accessed on 2021-04-20.

[6] Rasa - server endpoints (1.0.0), 2021. `https://rasa.com/docs/rasa/pages/http-api#tag/Model`, Last accessed on 2021-04-20.

[7] Amazon. Alexa skill blueprints. `https://blueprints.amazon.com/`, 2020. Accessed: 2021-04-24.

[8] Amazon. Alexa skills kit. `https://developer.amazon.com/en-US/alexa/alexa-skills-kit`, 2020. Accessed: 2021-04-24.

[9] Gautam Biswas, James R Segedy, and Kritya Bunchongchit. From design to implementation to practice a learning by teaching system: Betty's brain. *International Journal of Artificial Intelligence in Education*, 26(1):350–364, 2016.

[10] Jessica Van Brummelen. Tools to create and democratize conversational artificial intelligence. Master's thesis, Massachusetts Institute of Technology, Electrical Engineering and Computer Science Department, June 2019.

[11] Jessica Van Brummelen, Tommy Heng, and Viktoriya Tabunshchyk. Teaching tech to talk: K-12 conversational artificial intelligence literacy curriculum and development tools, 2020.

[12] Jessica Van Brummelen, Viktoriya Tabunshchyk, and Tommy Heng. "alexa, can i program you?": Student perceptions of conversational artificial intelligence before and after programming alexa, 2021.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[14] David Duran. Learning-by-teaching. evidence and implications as a pedagogical mechanism. *Innovations in Education and Teaching International*, 54(5):476–484, 2017.

[15] Geoffrey A. Fowler. I live with alexa, google assistant and siri. here's which one you should pick. `https://www.washingtonpost.com/technology/2018/11/21/i-live-with-alexa-google-assistant-siri-heres-which-you-should-pick/`. Accessed: 2021-04-06.

[16] Judith Good and Kate Howland. Programming language, natural language? supporting the diverse computational activities of novice programmers. *Journal of Visual Languages & Computing*, 39:78–92, 2017.

[17] Google. Actions on google. `https://developers.google.com/assistant`, 2020. Accessed: 2021-04-24.

[18] Google. Conversation design, 2020. `https://developers.google.com/actions/design/`, Last accessed on 2020-09-26.

[19] Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975.

[20] Martin G Helander. *Handbook of human-computer interaction.* Elsevier, 2014.

[21] Amazon Inc. Design process, 2020. `https://developer.amazon.com/fr/designing-for-voice/design-process`, Last accessed on 2020-09-26.

[22] Tsuyoshi Komatsubara, Masahiro Shiomi, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Can a social robot help children's understanding of science in classrooms? In *Proceedings of the Second International Conference on Human-Agent Interaction*, HAI '14, page 83–90, New York, NY, USA, 2014. Association for Computing Machinery.

[23] Kate Kozuch. Alexa vs. google assistant vs. siri: Which smart assistant is best? `https://www.tomsguide.com/us/alexa-vs-siri-vs-google,review-4772.html`. Accessed: 2021-04-06.

[24] Irene Lee. Unpublished NSF annual report on the GUTS y Girls project., 2010.

[25] Toby Jia-Jun Li, Tom Mitchell, and Brad Myers. Interactive task learning from GUI-grounded natural language instructions and demonstrations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 215–223, Online, July 2020. Association for Computational Linguistics.

[26] Phoebe Lin, Jessica Van Brummelen, Galit Lukin, Randi Williams, and Cynthia Breazeal. Zhorai: Designing a conversational agent for children to explore machine learning concepts. In *AAAI*, pages 13381–13388, 2020.

[27] Duri Long and Brian Magerko. What is ai literacy? competencies and design considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–16, New York, NY, USA, 2020. Association for Computing Machinery.

[28] Duri Long and Brian Magerko. What is ai literacy? competencies and design considerations. New York, NY, USA, 2020. Association for Computing Machinery.

[29] Naja A. Mack, Dekita G. Moon Rembert, Robert Cummings, and Juan E. Gilbert. Co-designing an intelligent conversational history tutor with children. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, page 482–487, New York, NY, USA, 2019. Association for Computing Machinery.

[30] Elizabeth Katalina Morales-Urrutia, Jose Miguel Ocaña, and Diana Pérez-Marín. How to integrate emotions in dialogues with pedagogic conversational agents to teach programming to children. *Innovative Perspectives on Interactive Communication Systems and Technologies*, page 66, 2020.

[31] MIT News. Teaching hong kong students to embrace computational thinking. `https://news.mit.edu/2016/teaching-hong-kong-students-computational-thinking-coolthink-at-jc-1115`, Last accessed on 2021-04-23.

[32] Alan Nichol. 5 levels of conversational ai: 2020 update, 2020. `https://blog.rasa.com/5-levels-of-conversational-ai-2020-update/`, Last accessed on 2020-09-26.

[33] Ubaid Pisuwala. The benefits of reactjs and reasons to choose it for your project. `https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html`, Last accessed on 2021-04-23.

[34] Daniel Rough and Benjamin Cowan. Poster: Apis for ipas? towards end-user tailoring of intelligent personal assistants. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–2, 2020.

[35] Elayne Ruane, Abeba Birhane, and Anthony Ventresque. Conversational ai: Social and ethical considerations. 12 2019.

[36] Gail Sullivan and Anthony Artino. Analyzing and interpreting data from likert-type scales. *Journal of graduate medical education*, 5:541–2, 12 2013.

[37] David Touretzky, Christina Gardner-McCune, Fred Martin, and Deborah Seehorn. Envisioning ai for k-12: What should every child know about ai? *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9795–9799, Jul. 2019.

[38] J. Van Brummelen, K. Weng, P. Lin, and C. Yeo. Convo: What does conversational programming need? In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–5, 2020.

[39] Jessica Van Brummelen. Conversational agents to democratize artificial intelligence. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 239–240, 2019.

[40] Alf Inge Wang. Dozens of studies show learning benefits of using kahoot!, 2020. `https://kahoot.com/blog/2020/07/01/dozens-of-studies-show-learning-benefits-of-kahoot/`, Last accessed on 2021-04-14.

[41] Kevin Weng. Examining conversational programming design needs with convo, a voice-first conversational programming system using natural language. Master's thesis, Massachusetts Institute of Technology, Electrical Engineering and Computer Science Department, May 2020.

[42] Tobias Wochinger. Rasa nlu in depth: Part 1 – intent classification, 2019. `https://blog.rasa.com/rasa-nlu-in-depth-part-1-intent-classification/`, Last accessed on 2020-09-26.

[43] Tobias Wochinger. Rasa nlu in depth: Part 2 – entity recognition, 2019. `https://blog.rasa.com/rasa-nlu-in-depth-part-2-entity-recognition/`, Last accessed on 2020-09-26.

[44] Jessica Zhu and Jessica Van Brummelen. Teaching students about conversational ai using convo, a conversational programming agent. unpublished, 2021.