Teaching Case

Conversational AI as an Emerging Technology

Adrian Gardiner
Michael A. Smith
Georgia Institute of Technology
Atlanta, GA 30332, USA

The robots are coming! Will your students be ready?

Abstract

Conversational AI is an emerging technology enabling digital transformation that is exciting and also relevant to future work practices of most – if not all - business students. This paper presents a lab exercise currently being used in an information systems service course to provide students a practicum for this technology. The lab exercise is designed to help inspire business students to reflect on the potential future role of technology within the workplace, and understand the potential disruptive nature of artificial intelligence (AI) to work practices and companies' operations.

Keywords: Conversational AI, chatbots, software agents, emerging technologies, automation, digital transformation.

During COVID-19 restrictions, many professors pre-recorded classroom lectures and allowed students to access them asynchronously. Rather than reverting back to the common model of delivering all material via lectures, in a post COVID-19 environment there is an opportunity to flip or invert the classroom – students are expected to learn new material outside the classroom before class time (Lage, Platt, & Treglia, 2000), thereby freeing up class time for more interactive and group-focused work to extend and enrich the material. With this opportunity in mind, the lab exercise showcased in this paper was developed as part of a series of 'hands-on' exercises that can be used in a flipped-classroom situation to inspire students' interest in and understanding of emerging technologies. The specific emerging technology featured in this paper is Conversational AI, which has the potential to change how employees will work in the future. This technology is also being used with other emerging intelligent automation technologies, such as Robotic Process Automation (RPA), in initiatives aimed at digitally transforming organizations (Overby, 2020).

1. CONVERSATIONAL AI

Deloitte (2019, p.6) defines Conversational AI (conversational agents) as "a programmatic and intelligent way of offering a conversational experience to mimic conversations with real people, through digital and telecommunication technologies." A similar definition comes from Chatbots Magazine (Nuseibeh, 2018): "A conversational agent is a software program which interprets and responds to statements made by users in ordinary natural language. It integrates computational linguistics techniques with communication over the internet." In summary, conversational agents are a dialog system that conducts natural language processing (NLP) and then responds to the query with human language (Vishnoi, 2020).

While Conversational AI is commonly referred to as a chatbot, a distinction can be made between Conversational AI (or conversational agents) and chatbots. According to Oxford Learner's Dictionaries, a chatbot is "a computer program designed to simulate conversation with human users, especially over the Internet." In contrast, Vishnoi (2020) emphasizes that "conversational agents not only advise you, but they also think with you and help you make decisions. For example, they can search through data sets, book your travel on your favorite, and also find the most suitable date." Conversational AI can therefore be seen as a more sophisticated form of (or evolution of) chatbot technology.

Applications of Conversational AI offer companies the opportunity to engage in conversational commerce and to develop value-adding applications. Conversational commerce is described by the conversational AI provider *Automat* as "a new form of ecommerce where brands and consumers communicate through live representatives and AI assistants in order to explore offerings, make transactions and receive services or help with their purchases." An example of a value-adding application developed using Conversational AI technologies is the app *CovidAsha*, which has been credited with saving lives during India's recent surge of COVID-19 infections. This chatbot was used to guide caregivers and ambulance service providers to oxygen and medicine suppliers, as well as to provide information on the availability of COVID-19 vaccines (Pragati, 2021).

Conversational AI is also expected to impact how people will work in the future. For example, Gartner (2019) predicted that 25 percent of digital workers will use virtual employee assistants daily by 2021. Along similar lines, Afshar (2021) suggests that 2021 "marks an inflection point where digital natives move beyond the omnichannel experience and now are expected to be serviced by a growing number of digital concierges -- virtual assistants that provide a new level of personalization in an all-digital world." In this respect, Conversational AI is clearly an emerging technology that will continue to evolve and, over time, applications using such technology will become more widespread within the workplace.

2. POSITIONING THE LAB EXERCISE WITHIN THE BUSINESS CURRICULUM

This lab exercise would make a valuable addition to a variety of different learning modules, across different courses. Learning module topics that this lab exercise could potentially support include emerging technologies in business; Natural Language Processing (NLP); business process design and automation; AI in business; digital transformation; marketing technologies, e-commerce and conversational commerce; and also human-computer interaction and user experience (UX) design.

This lab exercise is not intended to be stand-alone and it is expected that additional material would be provided to students to support a full learning module focused on the previously-mentioned topics. To create a comprehensive learning module, this lab exercise could be augmented by either lecture material, methodologies, and other technologies. For example, for a learning module on business process design and automation, this lab could be paired with another lab focusing on robotic process automation (RPA), which is another emerging technology that is also increasingly being used for digital transformation and process and task automation (Overby, 2020).

3. STUDENTS' LEARNING OUTCOMES

To illustrate applicable students' learning outcomes, we will assume this lab exercise has been included within a learning module focusing on emerging technologies in business. For such a learning module, possible learning outcomes could be:

Upon completion of this learning module in emerging technologies in business, students should be able to:

General emerging technology learning outcomes

- Discuss Conversational AI as an 'emerging technology' – Conversational AI's history, current stage of evolution and sophistication (state-of-the-art), and potential future road map.
- Provide examples of important business use cases for Conversational AI (e.g., customer service; personal assistants/employee enablement; FAQ Assistance & Site Navigation).
- Discuss some of the strengths and weaknesses of Conversational AI within the context of application areas – e.g., conversational commerce, digital transformation, and process automation.

Specific Conversational AI technology learning outcomes

- Explain how Natural Language Processing (NLP) enables Conversational AI.
- Understand the role and nature of intent recognition within software agent conversations.
- Understand the possible use of entity recognition within software agent conversations.
- Design, create and test a basic within software agent conversations.

Completion of this lab exercise will strongly support the latter specific Conversational AI emerging technology students' learning outcomes; however, supplementary material would need to be provided to students in order to satisfy the general emerging technology students' learning outcomes listed previously.

3. CHOICE OF SOFTWARE PLATFORM

Creating robust Conversational AI applications that can be deployed commercially is a complex undertaking. Consequentially, the software to create such software agents is also complex. To simplify the provisioning of the software to students, a cloud provider was chosen to be the Conversational AI software platform – specifically, *SAP Conversational AI* (version June 2021):
https://www.sapstore.com/solutions/42282/SAP-Conversational-AI

Conversational AI is part of SAP's Business Technology Platform. SAP describes this platform as the business technology platform (infrastructure) underpinning their vision of the "intelligent enterprise." (Taylor, 2020).

SAP Conversational AI also offers a low code chatbot building platform. Use of a low code environment frees students with limited technical and coding knowledge from having to deal with many technical issues, thereby allowing them to concentrate on designing and testing their chatbots. SAP Conversational AI also has an active developer and designer community which was used in the lab exercise to source intents to extend the capabilities of the software agent students were developing.

4. CHOICE OF APPLICATION

A number of ideas for an application were initially considered, such as creating a business-oriented chatbot – e.g., a software agent for product ordering; or an application to allow customers to track packages. However, as the students enrolled in the course using this lab exercise were mainly either freshmen or sophomores with limited business education, we chose to focus on an application that they could clearly see was potentially relevant to their own lives – a software agent to advise students where to go on-campus to receive a COVID-19 vaccination.

5. RUNNING THE LAB EXERCISE

This lab exercise is designed to take approximately two-hours to complete, either as an in-class exercise or outside class assignment. While this lab exercise can be completed by a student working alone, it is recommended to be assigned as a small-group exercise (groups of two or three students): there are several steps within the lab exercise (labeled Team Activity) when students can make their own choices (e.g., designing intents) and these choices can be informed and enriched by discussion within a small group of students. [Note: Suggested responses to Team Activities are shown in Appendix B.] In addition, 'having many eyes' while using potentially complex software for the first time may help limit errors that may frustrate students and degrade their experience.

As mentioned previously, this lab exercise would be a valuable addition to several different courses. At the authors' own school, this lab exercise was developed for inclusion within a general service-oriented information systems course that all business students are required to take as part of their business degree requirements. Specifically, the lab exercise was developed to give students a hands-on experience with a 'cool' emerging technology to spark their interest in emerging technologies and information systems. To illustrate emerging technology within business, this lab exercise was paired with another 'hands-on' lab exercise focused on creating and querying a blockchain. In the future, a third lab is planned that will be a practicum on RPA (Robotic Process Automation).

The lab exercise document distributed to students featured the following structure (assuming the lab exercise is to be completed during class time):

- **Purpose**: Inform students where the lab exercise falls within the course and the overall aim of the lab exercise.
- **Introduction**: The topic of Conversational AI was grounded by a contextual discussion of current events or trends in technology. For example, is a recent The New York Times article '*The Robots Are Coming for Phil in Accounting*' (Roose, 2021) to emphasize that Conversational AI and related technologies will likely impact their future work life.
- **What to do before class**: For this lab exercise, students had to create an SAAS account to access *SAP Conversational AI*.
- **Class organization**: Informs students how they will complete the lab – e.g., as an individual or in groups and outlines rules for interacting with other students.
- **What to do during class**: This section contains the instructions for completing the lab exercise. These instructions are replicated in Appendix A.

- **Debrief**: Summarize what students should have learned. This section may also contain supplemental information – e.g., example jobs postings relating to conversational AI jobs.
- **What to do after class**: Inform students what to submit as evidence of lab completion and where to submit it. It is recommended that students are asked to read this section prior to attempting the actual class exercise steps.

6. EXTENDING THE LAB EXERCISE

The domain of the current version of the lab exercise is developing a software agent to advise students on COVID-19 vaccine locations. After completing this exercise, students could be asked to add further skills to the current agent, or simply be asked to develop and demonstrate (or even deploy) to the class their own software agent application. For students to create their own basic software agent, they would need to source or create relevant intents, identify and address meaningful entities within the chosen domain, and design, create, test and deploy skills and conversations. As the latter option involves considerably more time and goes deeper into the technology, this option would be more appropriate for a course focusing more on these technologies – e.g., a course focusing on NLP (natural language processing), or process automation, AI in business, or human-computer interaction and user experience (UX) design. The lab exercise could also be augmented with de-briefing discussion questions. For example, students could be asked to discuss the trustworthy aspects of software agents as an information source. Students could also be asked to source jobs involving chatbot use or creation and present/discuss specifics of the job requirements.

7. REFERENCES

Afshar, V. (2021, April 7). *AI-powered virtual assistants and the future of work*. ZDNet. https://www.zdnet.com/article/ai-powered-virtual-assistants-and-future-of-work/

Automat. (2021). *What is conversational commerce?* Retrieved June 28, 2021, from https://automat.ai/resources/what-is-conversational-commerce

Deloitte (2019). *Conversational AI: The next wave of customer and employee experiences*. Deloitte Touche Tohmatsu Limited. https://www2.deloitte.com/content/dam/Deloitte/au/Documents/strategy/au-deloitte-conversational-ai.pdf

Gartner (2019, January 9). *Gartner predicts 25 percent of digital workers will use virtual employee assistants daily by 2021*. https://www.gartner.com/en/newsroom/press-releases/2019-01-09-gartner-predicts-25-percent-of-digital-workers-will-u

Lage, M., Platt, G., & Treglia, M. (2000). Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment. *The Journal of Economic Education*, 31(1), 30-43.

Nuseibeh, R. (2018, May 11). *What is a chatbot?* Chatbots Magazine. https://chatbotsmagazine.com/what-is-a-chatbot-6dfff005bb34

Overby, S. (2020, August 11). *How Robotic Process Automation (RPA) and digital transformation work together*. The Enterprisers Project. https://enterprisersproject.com/article/2020/8/how-rpa-robotic-process-automation-and-digital-transformation-work

Pragati, K. (2021, June 28). *A team that has never met each other, builds a chatbot, saves lives.* The Hindu. https://www.thehindu.com/sci-tech/technology/a-team-that-has-never-met-each-other-builds-a-chatbot-saves-lives/article35017505.ece

Roose, K. (2021). The robots are coming for Phil in Accounting, *The New York Times*, March 6, 2021.

Taylor, P. (2020, October 13). *Explaining SAP's Business Technology Platform.* https://news.sap.com/2020/10/business-technology-platform-explainer

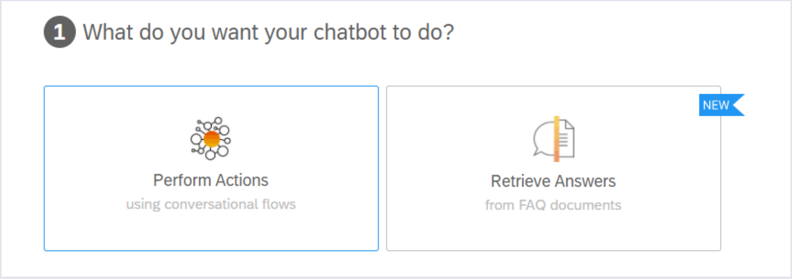
Vishnoi, L. (2020, June 20). *Conversational agent: A more assertive form of chatbots.* Towards Data Science. https://towardsdatascience.com/conversational-agent-a-more-assertive-form-of-chatbots

Appendix A: Conversational AI Lab Exercise

Step 1: Create a new bot project

From the SAP Conversational AI home screen, click **+ NEW BOT** to create a new bot.

Next, click 'Perform Actions'.

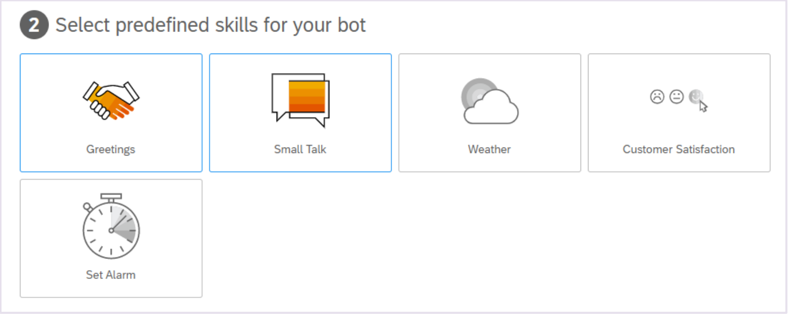


1 What do you want your chatbot to do?

Perform Actions
using conversational flows

NEW
Retrieve Answers
from FAQ documents

We will now select a set of predefined skills for your bot – let's start with selecting **Greetings** and also **Small Talk**.



2 Select predefined skills for your bot

Greetings

Small Talk

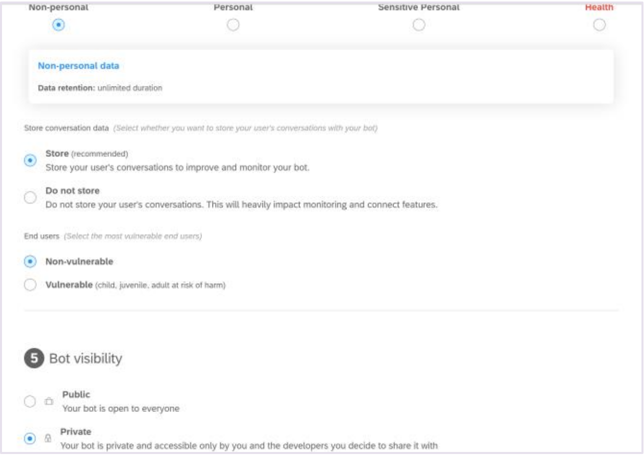
Weather

Customer Satisfaction

Set Alarm

Name your bot *Vaccine-Location-Advisor* and add this description: *This bot advises students where they can go on campus to receive a COVID-19 vaccine*.

Choose the following Data Policy options: Non-personal, Store, Non-vulnerable, and make your bot Private.



● Non-personal ○ Personal ○ Sensitive Personal ○ Health

Non-personal data

Data retention: unlimited duration

Store conversation data *(Select whether you want to store your user's conversations with your bot)*

● **Store** (recommended)
Store your user's conversations to improve and monitor your bot.

○ **Do not store**
Do not store your user's conversations. This will heavily impact monitoring and connect features.

End users *(Select the most vulnerable end users)*

● **Non-vulnerable**

○ **Vulnerable** (child, juvenile, adult at risk of harm)

5 Bot visibility

○ **Public**
Your bot is open to everyone

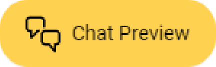
● **Private**
Your bot is private and accessible only by you and the developers you decide to share it with

You have now created a bot.

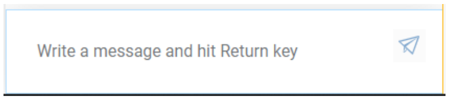
Step 2: Interact with your new bot

As we elected to add skills (Greetings, Small Talk) when we created the bot, it is already able to respond to some simple interactions with the bot user. Let's test this.

Click the Chat Preview button.



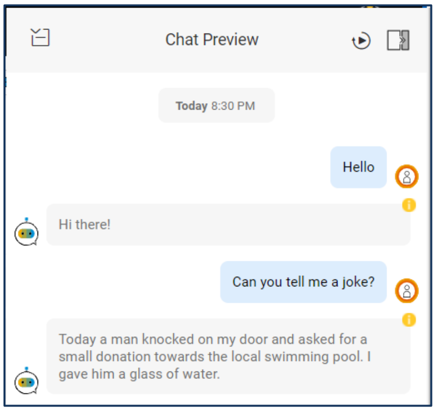
In the interaction textbox (bottom of the screen):



Enter:

Hello [Enter]
Can you tell me a joke? [Enter]

While your bot should be able to respond to these simple interactions, its ability at this stage to carry on a more complex conversation is limited.



Today 8:30 PM

Hello

Hi there!

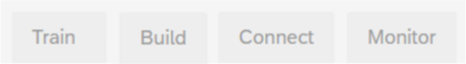
Can you tell me a joke?

Today a man knocked on my door and asked for a small donation towards the local swimming pool. I gave him a glass of water.

We'll now extend the bot's capabilities to develop an application that students can use to get information as to where to go on campus to receive a COVID-19 vaccine.

Step 3: Train your bot

In the Chatbot interface there are several tabs:



These tabs represent the four stages in your bot's life:

Train: Teach your bot what it needs to understand. Here you will create intents – ideas your bot will recognize. Intents are packages of related expressions (i.e., have a common theme) that the bot should recognize coming from the user. For example, an intent could contain a collection of different expressions used to greet someone. The ability of your bot to understand the world is also dependent upon entities. An entity is a keyword that is extracted from an expression. Entities refers to concepts or things that may be referenced in an expression. Examples of an entity are a concept, such as a student's GPA, or the time of day, such as midnight.

Technically, training will establish our bot's Natural Language Processing (NLP) lexicon. The term 'lexicon' is used in the field of NLP to refer to the vocabulary of a person, language, or branch of knowledge. An example of lexicon in a medical domain would be a set of medical terms.

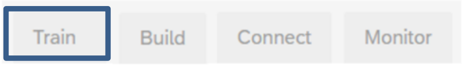
Build: Create your conversational flow. Conversation flows will be defined through creating and connecting skills. Skills define things your bot can do.

Connect: Deploy your bot to one or several messaging platforms.

Monitor: See how users are communicating with your bot, check if the bot is understanding users correctly, and make updates to the bots intents/entities. Monitoring also lets you see what users want and what additional capabilities could be added to the bot.

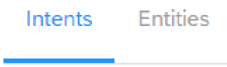
Step 3A: Define intents

Click the **Train** tab. In this tab, you can maintain a bot's **Intents** and recognized **Entities**.



We'll first focus on **Intents**.

Click the **Intents** tab.

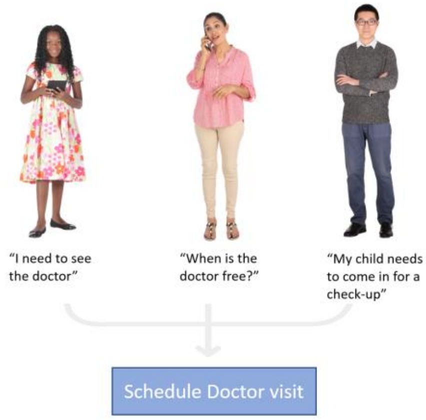


Everything your chatbot understands is in the intents. Each intent corresponds to an action the chatbot user wants to perform.

For example, the intent @ greetings (one of the predefined intents) enables your bot to understand when a user says Hello.

As you cannot predict ahead of time how users may phrase their interactions, each intent will contain a set of expressions that mean the same thing, but are constructed in different ways. For example: for a greeting, one user may say "howdie" while another user may prefer the term "hey". However, both users share the same intent – to stipulate a greeting.

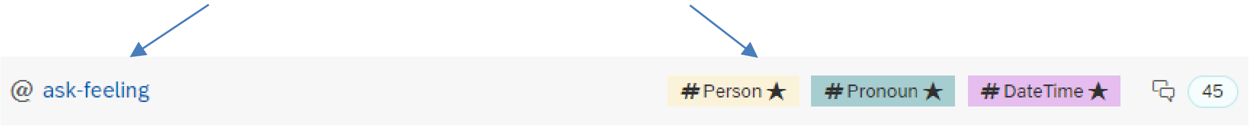
Another example is shown in the following image – which shows three possible ways different users could express a desire to schedule a doctor's visit.


"I need to see the doctor"
"When is the doctor free?"
"My child needs to come in for a check-up"
Schedule Doctor visit

SAP recommends that each intent has between 20-50 expressions (but for this exercise, we'll try to get by with a lot fewer). SAP documentation mentions that some large bots may have more than 10,000 expressions across the defined intents.

As intents are packaged within skills, when you earlier added pre-defined skills to your bot a set of pre-defined intents were also added – which you will see listed on the Intents pane.

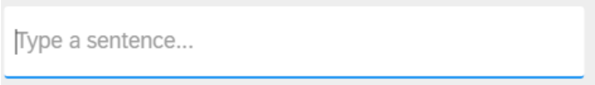
The name of the intent is on the left, with associated entities listed on the right.



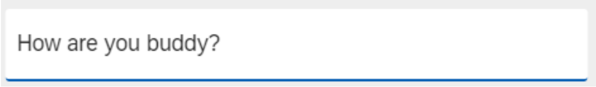
Click on the @ask-feeling intent and you will see a list of the defined 'expressions' the bot will understand as an intent to ask one's feelings (e.g., "All good?").

You can test whether (and which) intent has been activated by an expression entered by a user enter by typing the expression into the test pane. Let's try it. Open the TEST pane by clicking the TEST tab (icon shown to the right).

In the dialog area (located to the bottom-right of the screen):



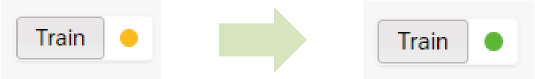
Enter *How are you buddy?* and hit [Enter].



How are you buddy?

The identified intent should be:

Note: if the identified intent differs from *@ask-feeling*, check that the bot engine has been updated. i.e., if the bot engine needs updating (an orange circle), click the Train button to update the engine. (A green circle means the engine has been updated to reflect any recent changes.)

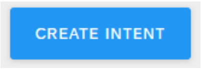


You may sometimes see a message saying your changes have been saved. However, saving changes does not necessarily mean the bot engine has also been updated.

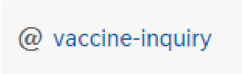
So, the system senses you are asking about how one feels (note: the bot does not actually feel anything!). This pane also shows any identified entities (which will be explained later) within the entered text.

To create our first intent, return to the list of intents and click on the *New Intent* button.

Name this intent: ***vaccine-inquiry***. For the description, enter: *The user is interested in inquiring about vaccine locations on-campus.*

Then click .

You should now see an entry for your new intent in the list along with the existing intents.



-- TEAM ACTIVITY #1 --

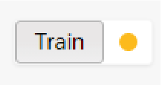
Click on your new intent to open it.

Add 5 expressions to the intent that a *Georgia Institute of Technology* student may type into a chat to indicate they are interested in finding out where they can get a COVID-19 vaccine shot on-campus. SAP Help recommends that you diversify your intents: Use as many different grammatical structures as you can.

As you enter your expressions, the system will suggest additional expressions – e.g.:

You have 6 expressions suggested to enrich your intent

Review any suggested expressions and [add] the ones your team feels have the potential to improve your bot's understanding.

Again, click  to update your bot engine.

Given our bot's domain (i.e., COVID-19 vaccine locations), certain concepts or keywords will likely be more common across your collection of expressions defining the *@vaccine-inquiry* intent – for example, review your set of expressions and see how many feature the term 'where' or 'vaccine'. As these terms are indicative of the *@vaccine-inquiry*, when these terms come up in a conversation this intent may be 'triggered' by the system.

To demonstrate the sensitivity of the system to keywords in expressions entered by users, enter into the TEST pane:

Can I get protection?

If the system reports that 'No intent found', retry with:

<u>Where</u> can I get protection?

For this statement, if several of your expressions in the *@vaccine-inquiry* intent featured the keyword 'where', the system would be more likely to identify the user's intent as *@vaccine-inquiry*.

Although the ultimate aim of a user is to find 'where' on campus they can receive a COVID-19 vaccine, given the richness of language, a user may have started the conversation with the bot by simply entering:

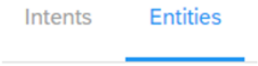
Can I get a vaccination on-campus?

So, add this expression to the *@vaccine-inquiry* intent and re-train the bot engine.

Let's now look at **entities**.

Step 3B: Define entities

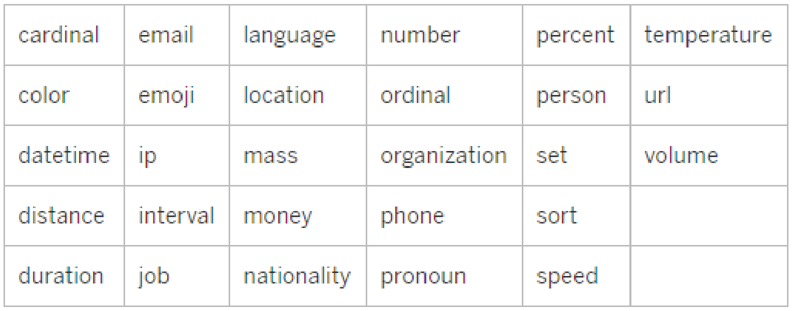
Click the **Entities** tab.



Entities are keywords that you may find within expressions entered by users that refer to things or concepts (e.g., distance, GPA, time).

One of the main uses of entities identified in expressions is for intent detection – i.e., an expression entered by a user referring to a specific entity will raise the probability the system associates that expression with a specific intent that also refers to that entity, or a closely-related thing or concept (as seen by the system).

SAP conversational AI has 28 pre-defined 'gold' entity types (listed in the following table):

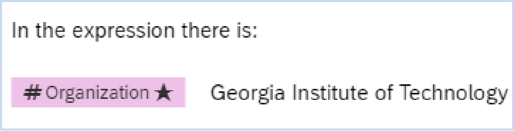


cardinal	email	language	number	percent	temperature
color	emoji	location	ordinal	person	url
datetime	ip	mass	organization	set	volume
distance	interval	money	phone	sort	
duration	job	nationality	pronoun	speed	

To see whether the system recognizes *The Georgia Institute of Technology* as an Organization (one of the gold entity types), click on the TEST icon and enter:

Georgia Institute of Technology

The system should automatically identify *The Georgia Institute of Technology* as an Organization.



In the expression there is:

Organization ★ Georgia Institute of Technology

Note: One could argue that *The Georgia Institute of Technology* is both an Organization and Location – both of which are gold entities. However, in this case, the system emphasizes more the Organization property. Notwithstanding, the system's knowledge graph will likely also feature connections to the location aspect of a University (e.g., it is located in Atlanta). Moreover, as a recognized organization, the system will likely be 'aware' of related properties of tertiary educational organizations, such as it is a University, and has students and faculty.

If you click within the TEST pane "See more in the JSON view", you can see the confidence level the system has attributed to the assertion: *The Georgia Institute of Technology* is an Organization (89% confidence). However, the system is likely to be less confident when a user refers to *The Georgia Institute of Technology* using other terms – e.g., *GA Tech* (61% confidence). The level of confidence may also change depending on the contextual use of the term within a longer expression - e.g., *GA Tech in Atlanta* (90% confidence).

Step 3C: Define custom entities

The domain of our bot is on-campus COVID-19 vaccine locations. While the bot will likely assert the user is referring to the *@vaccine-inquiry* intent when they enter:

Can I get a vaccination?

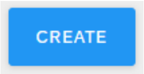
During testing, the bot did not identify any intent (i.e., the bot reported 'No intent found') when the user entered:

Can I get Pfizer?

Note: the bot is somewhat confused about the meaning of the term 'Pfizer'. As Pfizer is an 'organization' (one of the gold entity types), this is how the bot sees (has tagged) this term – not as a reference to an actual brand of vaccine. The bot's inability to adequately distinguish between Pfizer as an organization and Pfizer as a vaccine brand can be addressed by creating a custom entity. Note: an alternate approach would perhaps be to expand the breath of your example expressions within the *@vaccine-inquiry* intent to refer specifically to the Pfizer vaccine.

Let's now create a custom entity for *vaccine*.

Click on .

Name the entity: ***Vaccine***. Keep it as a restricted entity. Click  .

You will now see the entry for the new ***Vaccine*** custom entity listed above the list of gold entities. Click on its icon .

For the 'List of entity values' enter the following:

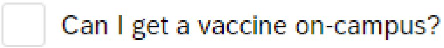
- *johnson and johnson*
- *moderna*
- *pfizer*
- *vaccine*

-- TEAM ACTIVITY #2 --

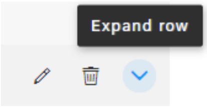
In the 'List of entity values', enter 3 additional terms ('entity values') that a user could potentially enter to refer to a vaccine in a conversation. Hit [Enter] after making each entry and do not worry about capitalization.

We now need to return to the *@vaccine-inquiry* intent to associate our custom entity with keywords within this intent's expressions. By creating this custom entity that includes references to Pfizer, it will allow the system to make that association between occurrences of this custom entity and the *@vaccine-inquiry* intent, thereby increasing the probability that it will associate Pfizer with wanting to be vaccinated.

In the list of intents, click the *@vaccine-inquiry* intent to open it. Find in the list of expressions the first reference to the 'vaccine' concept. For example, it could be something similar to this expression:

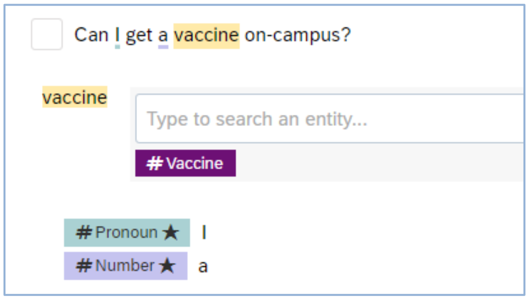


□ Can I get a vaccine on-campus?

Click the 'Expand row' icon for that expression: 

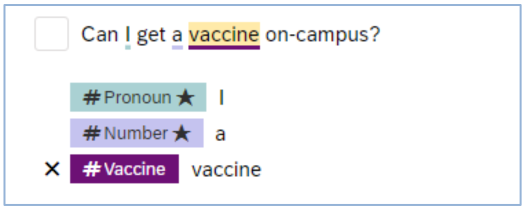
Any entities that have been associated with this expression will be shown.

Associating a custom entity with a keyword within an expression (this is called adding *tagged values*) must be done manually. To do this, use your mouse to highlight the keyword within the expression you would like to associate an entity with. For example:



□ Can I get a vaccine on-campus?

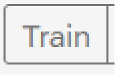
A list of custom entities will be displayed (we only have one). Click the #Vaccine icon to select it. The #Vaccine entity has now been associated with the keyword *vaccination* within this expression.



□ Can I get a vaccine on-campus?

Now, go through all your expressions in the *@vaccine-inquiry* intent and tag them accordingly.

Before re-testing your bot, you will need to update your bot's engine: click the button.



Now re-enter:

> *Can I get Pfizer?*

The bot should now assert that the user is referring to the *@vaccine-inquiry* intent.

Step 3D: Define additional intents

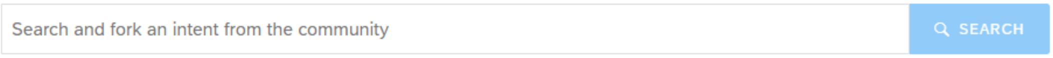
We will now add to our bot several more intents. The need for these intents will become apparent later when we define a conversation.

Add a second intent. Name this intent: ***student***. This intent will not have any expressions.

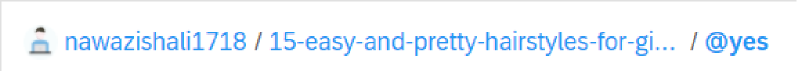
Finally, we'll add two more intents – one for 'Yes', and another for 'No'.

SAP Conversational AI has an active developer community and we'll take the opportunity to adopt intents open-sourced by other developers, rather than develop them ourselves.

In the community intent search box, enter *Yes*, then [Enter].



Find the following intent:



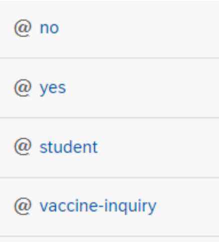
nawazishali1718 / 15-easy-and-pretty-hairstyles-for-gi... / @yes

Click the button to add this intent created by *nawazishali1718* to our list of intents.

Note: ‘forking’ is a developer term for making a copy of something for the purpose of modifying it.

This person has also developed an open-source/shareable ‘No’ intent. So, search for No and also fork that intent. Before going further, surprise yourself as to how many ways one can say No by reviewing the expressions within the @no intent!

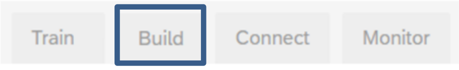
You should now have four new intents listed on the Intents pane (as shown in the following list), in addition to the ones added at the time you first created the bot.



Let’s now focus on building a conversation around our intents and entities.

Step 4: Design and build a conversation

Let’s turn our attention to the **Build** tab.



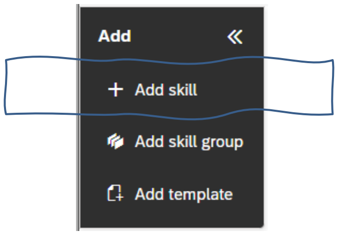
A **skill** is a block of conversation with a clear purpose that your bot can execute to achieve a goal. It may be as simple as the ability to greet someone, but it can also be more complex, like giving movie suggestions based on information provided by the user.

Your skills are defined by triggers, requirements (information your bot must collect), and resulting actions.

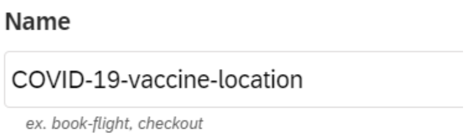
We will add a new skill.

Step 4A: Add a skill (*COVID-19-vaccine-location*)

Click **Add skill**:

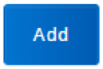


Name the skill: ***COVID-19-vaccine-location***

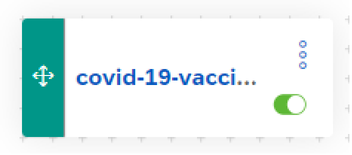


Name

COVID-19-vaccine-location

Click  .

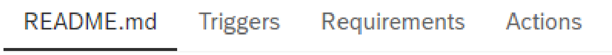
You should now see a container on the Build pane that represents our new skill.



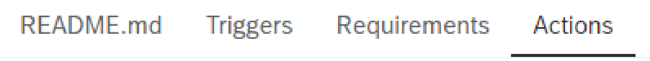
Purpose of this skill: This skill – when executed- will inform the user of the locations on campus they can go to get the COVID-19 vaccine. How this skill factors into the overall conversation will become clearer once you start testing your bot.

Click the name of the skill container (*COVID-19-vaccine-location*) to open it.

Note the tabs:

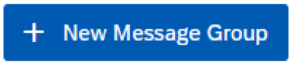


The *README.md tab* is for documentation. Leave it and select the *Actions* tab (last tab).

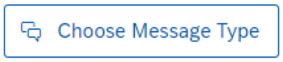


An action is something that your bot executes at a specific point when executing a skill. Examples of possible actions include: Send message to the user; Go to another skill.

Create a new message group – click:

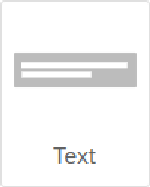


You will now see the ADD CONDITION panel. A condition is a requirement that needs to be satisfied before a message will be sent to the user.

Click 

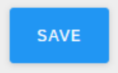
The bot builder offers a number of message forms (e.g., test, image, list, etc.). Messages will be communications sent to the bot user. While perhaps a map with COVID-19 locations marked would be appropriate given our bot's domain, for simplicity, initially we'll opt to provide only a list of vaccine locations on campus.

Click **Text**:

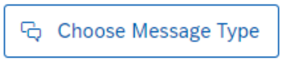


For your message, enter:

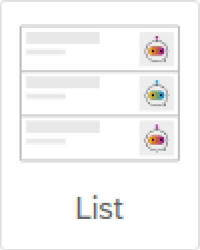
> *At this time, there is only one location on campus where you can receive a COVID-19 vaccine.*

Click  .

Again, choose create a new message group – click:

Click 

This time we will create a **List** message:

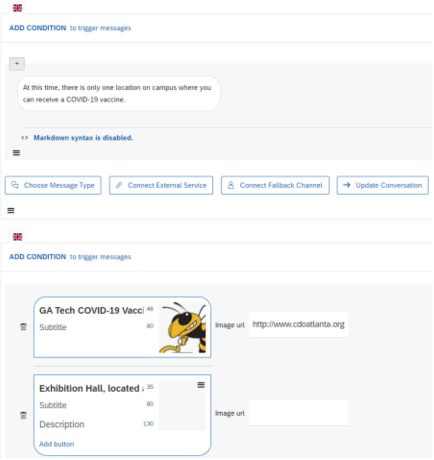


Add a Header>Title: *GA Tech COVID-19 Vaccine Locations*

Add as a Header Image url: http://www.cdoatlanta.org/img/georgiatech.png

Add the following as an Item>Title: *Exhibition Hall, located at 460 Fourth Street*

Your actions pane should now look similar to:



At this time, there is only one location on campus where you can receive a COVID-19 vaccine.

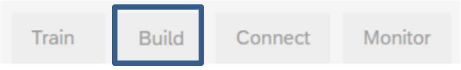
GA Tech COVID-19 Vacci

http://www.cdoatlanta.org

Exhibition Hall, located a

Click SAVE.

Return to the Build pane - if necessary, click



.

Step 4B: Add a second skill (*student-yes*)

Add another skill. The name of this skill is: ***student-yes***.

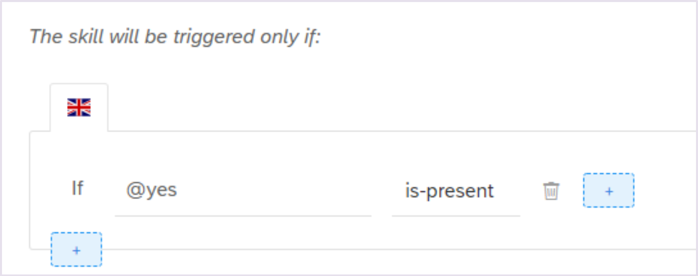
Once the skill container has been added to the Build pane, click its name to open it.

Note: The new skill may have been stacked over the original skill. You can separate them (reposition skill containers) using the handle (as shown to the right). ✥

Now, click on the *Triggers* tab.

Triggers are conditions that determine whether the bot should execute the current skill. If the triggers for the skill are validated, the bot executes this skill over other skills. A condition is a test that evaluates as to either true or false (e.g., whether some phrases or entities must be present in the conversation).

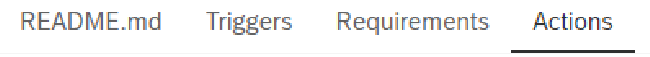
Enter @yes after the 'If'. Click SAVE. The 'is-present' option should be automatically selected.



The skill will be triggered only if:

If @yes is-present

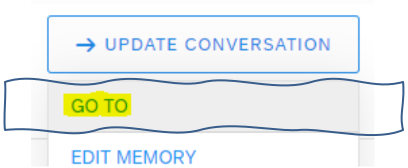
Then click on the *Actions* tab.



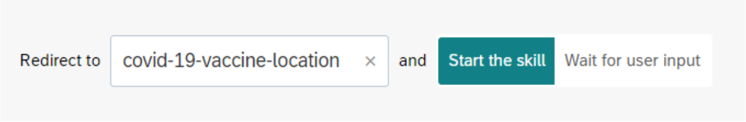
Again, choose create a new message group – click:

Click

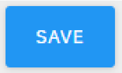
This time, choose: GO TO



Ensure the 'Start the skill' is green (i.e, is chosen as the default action) and choose the ***COVID-19-vaccine-location*** skill.

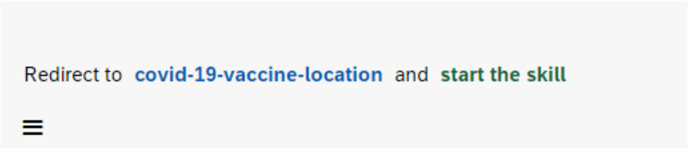


Redirect to covid-19-vaccine-location × and Start the skill Wait for user input



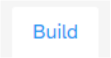
Click .

The Action should now look like the following:



Redirect to **covid-19-vaccine-location** and **start the skill**

Step 4C: Add a third skill (*student-no*)

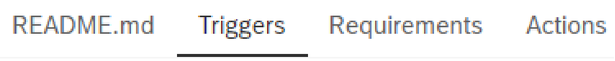


Click the to return to the Build pane.

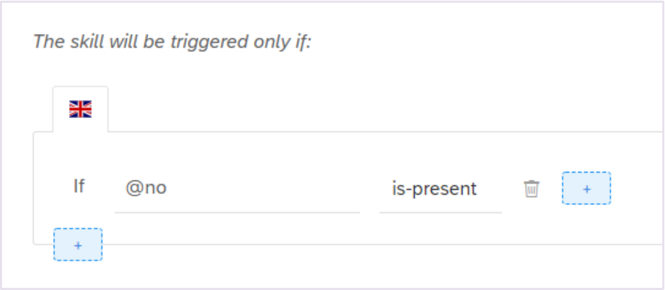
We need now to add the 'no' skill. Add another skill. The name of this skill is: ***student-no***.

Once the skill container has been added to the Build pane, click its name to open it.

This time, click on the *Triggers* tab.



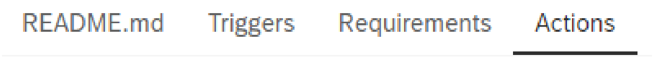
Enter @no after the 'If'. Click . 'is-present' option should automatically be selected.



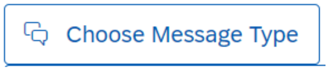
The skill will be triggered only if:

If @no is-present

Then click on the *Actions* tab.



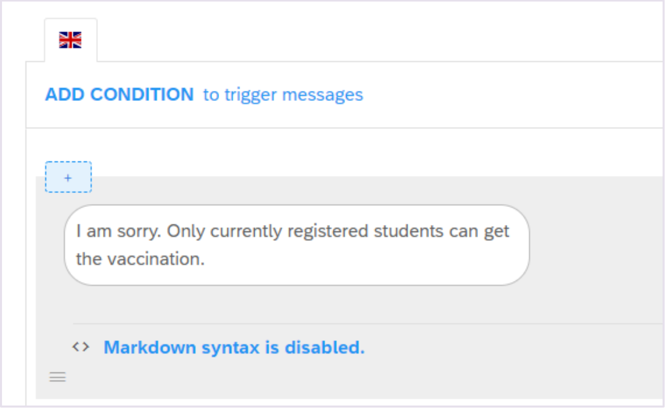
Again, choose to create a new message group – click:

Choose:  and then **Text**.

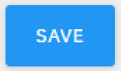
Enter (copy) the following as the text message:

I am sorry. Only currently registered students can get the vaccination.

Your message should look like:


ADD CONDITION to trigger messages
+
Markdown syntax is disabled.

I am sorry. Only currently registered students can get the vaccination.

Click .

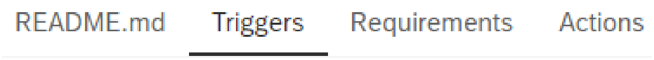
Step 4D: Add a fourth skill (*confirm-registered-student*)

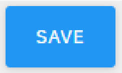
Return to the Build pane.

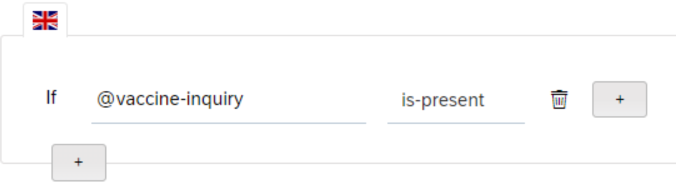
We need to add one more skill. The name of this skill is: ***confirm-registered-student***.

Purpose of this skill: This skill is designed to be the first skill executed. It will initially screen the user to determine whether they are a student at *The Georgia Institute of Technology*. This is an important question as students from other institutions are not eligible to be vaccinated at Georgia Tech.

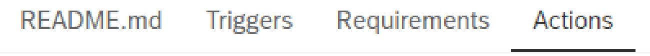
Click to open this skill. This time, click on the *Triggers* tab.



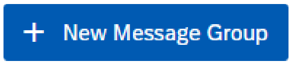
This time enter the intent: @vaccine-inquiry and click .

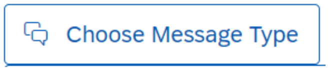


Then click on the *Actions* tab.



Choose create a new message group – click:

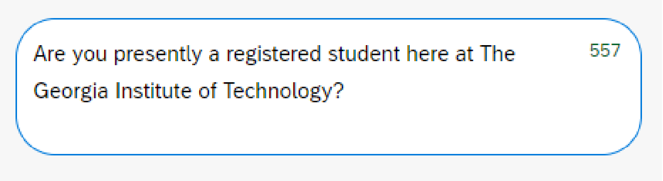


Choose:  and then **Text**.

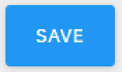
Enter the following as the text message:

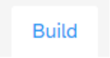
Are you presently a registered student here at The Georgia Institute of Technology?

i.e.,

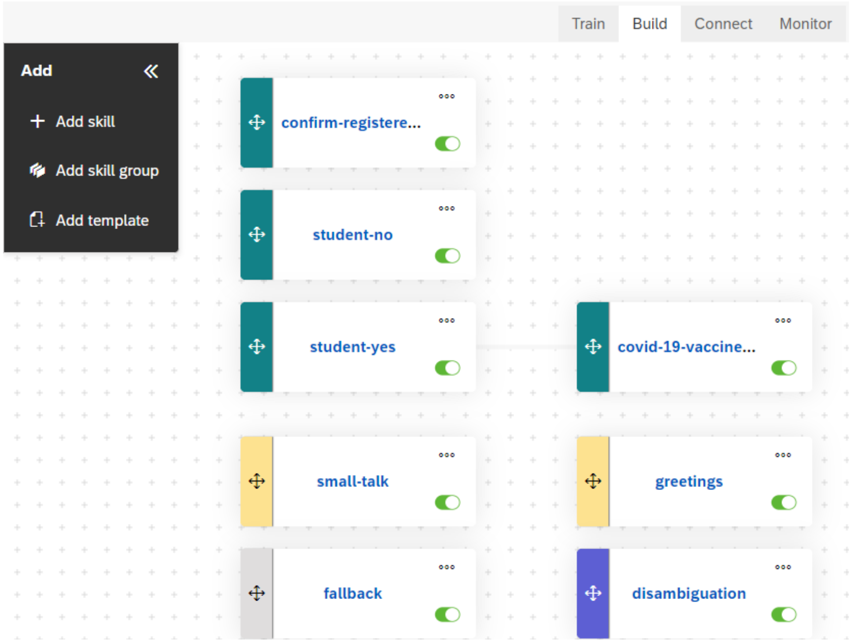


Are you presently a registered student here at The Georgia Institute of Technology?

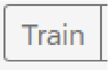
Click .

Return to the Build pane: .

Your Build pane should look now something similar to this image:



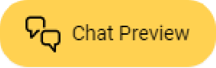
Step 5: Test your bot



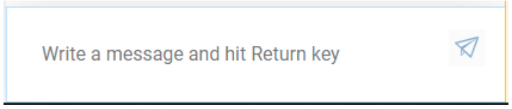
Update your bot’s engine: click the Train button. Wait until the dot turns green.

We will now run our bot to test it (the moment of truth!).

Again, in the bottom-right corner, you will see a gold button – ‘Chat Preview’. Click on it.



In the interaction textbox, conduct the following test conversations (see Test Run 1..5, which follow).



Test Run 1

Say hello to your bot.

Enter: *Can you tell me where on campus I can get a COVID-19 vaccine?*

When asked by the bot, indicate you are not currently a GA Tech student.

Test Run 2

Enter: *Where at Tech can I get the vaccine?*

When asked by the bot, indicate you are currently a GA Tech student.

Test Run 3

Enter: *I want my vaccine, and I want it now!*

When asked by the bot, indicate you are currently a GA Tech student.

-- TEAM ACTIVITY #3 --

If the previous expression does not trigger the *COVID-19-vaccine-location* intent, decide with your team what to do to correct this. Make the necessary change/s, update your bot engine, and re-test.

Test Run 4

Enter: *I am so angry with you! I need a vaccine now!*

When asked by the bot, indicate you are currently a GA Tech student.

This expression may not invoke the *COVID-19-vaccine-location* intent. However, the bot probably picked up on your displeasure. To investigate this further, open the TEST tab and re-enter the statement. Then click on the text: *See more in the JSON view*. Look for the 'sentiment' tag. What sentiment has the bot engine attributed to your statement? Note: **Sentiment analysis** (or opinion mining) is a natural language processing technique used to determine whether data is positive, negative or neutral. Commonly, AI techniques are used to determine sentiment reflected in text.

Test Run 5

-- TEAM ACTIVITY #4 --

Come up with 3 test expressions of your own to indicate your interest in knowing where on campus you can get vaccinated. Make sure you do not use any of the expressions you previously defined for the *COVID-19-vaccine-location* intent. Try to test the boundaries of the bot's understanding in terms of how one may interact with the system (i.e., it's ok if some of your test expressions fail).

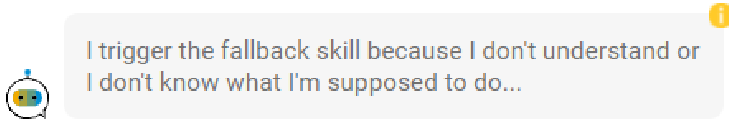
Test Run 6

-- TEAM ACTIVITY #5 --

Challenge Activity

Enter: *I am a student.*

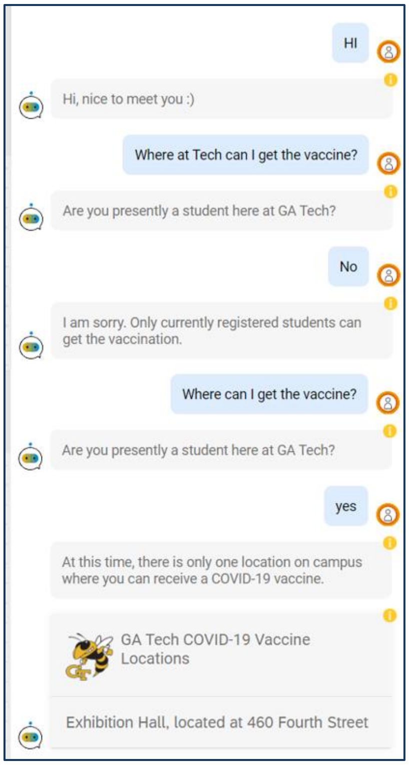
The bot will likely associate no intent with this statement and display the following feedback:



I trigger the fallback skill because I don't understand or I don't know what I'm supposed to do...

Modify the bot so that when this expression is entered the bot will still be able to assist the student in finding the location on campus for COVID-19 vaccination. Note: you are not allowed to add any further expressions to the *@vaccine-inquiry* intent, as doing so would dilute the focus of this intent.

The following is an example of a test transcript:



HI

Hi, nice to meet you :)

Where at Tech can I get the vaccine?

Are you presently a student here at GA Tech?

No

I am sorry. Only currently registered students can get the vaccination.

Where can I get the vaccine?

Are you presently a student here at GA Tech?

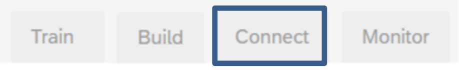
yes

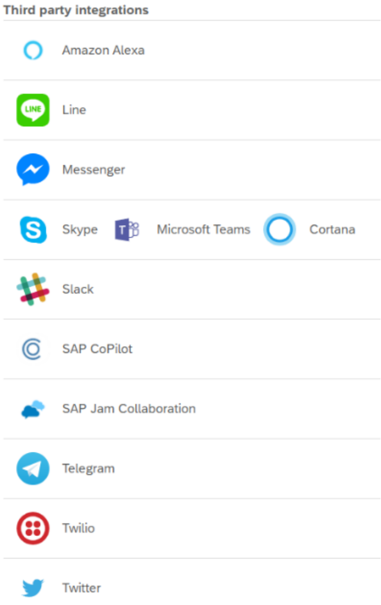
At this time, there is only one location on campus where you can receive a COVID-19 vaccine.

GA Tech COVID-19 Vaccine Locations

Exhibition Hall, located at 460 Fourth Street

Step 6: Deploy your bot





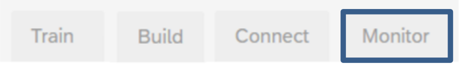
Third party integrations

- Amazon Alexa
- Line
- Messenger
- Skype
- Microsoft Teams
- Cortana
- Slack
- SAP CoPilot
- SAP Jam Collaboration
- Telegram
- Twilio
- Twitter

There are a number of established platforms for AI bots. You can deploy the SAP AI bot to a number of these environments (shown to the right).

Note: there is no need to deploy your bot.

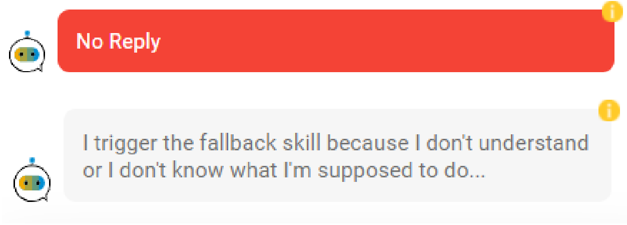
Step 7: Monitor your bot



Once deployed, you can monitor your bot's performance and statistics through the monitor tab. As we have not deployed our bot, there is no need to monitor it.

Troubleshoot (debug) your bot

During a conversation with your bot, as a user, you may experience the following reply:



No Reply

I trigger the fallback skill because I don't understand or I don't know what I'm supposed to do...

Effectively this means the bot does not know what to do. Perhaps it did not recognize the intent of your input, or it did and there was no action associated with that intent.

The best approach here is to reenter you last dialog into the TEST tab. This will verify whether it triggers an intent and which intent that is. If the intent you expect is triggered, then the bug may lie with how the associated skill was configured.

Appendix B: Team Activities

This appendix is for instructors only.

TEAM ACTIVITY #1

Examples of possible expressions:

- Where are the vaccines?
- Where can I get vaccinated at Tech?
- Where are the vaccines on campus?
- Where are immunizations held on campus?
- Where can I be vaccinated?
- Where can I get vaccinations?

TEAM ACTIVITY #2

Examples of possible additional entity values:

- jab
- shot
- protection
- vax

TEAM ACTIVITY #3

An example of an appropriate action would be to return and edit the example expressions within the @ vaccine-inquiry intent to add the following expression: *I want my vaccine.*

TEAM ACTIVITY #4

The possible student responses here will vary as they test the ability of their bot to understand different interaction scenarios. The main thing about this activity is that the designer (i.e., the students) is attempting to test the 'boundaries' of what constitutes an acceptable conversation that will lead to a successful use of the bot. Exploring these boundaries may lead to further refinements within the bot design. These insights will also come from monitoring the bot when deployed and analyzing the captured conversations.

TEAM ACTIVITY #5

Students can remedy this deficiency by creating a separate intent (*@self-identify-as-student*) for users who initially seek to self-identify as a student.

Example expressions for such an intent could be:

- A student
- Student at a school
- Student at the university
- Student at a university
- Enrolled student

This intent could be included within a skill that is triggered when the *@self-identify-as-student* intent is sensed by the bot. The action associated with this skill could be to pass the focus of the conversation to the *confirm-registered-student* skill.