IBM course AI(mod3)

Chatbot structure

If you've clicked a **Chat** button online and encountered a non-human chatbot, you might have noticed two things:

- When you ask a clear question that relates to the website's purpose, like asking a shopping chatbot "How can I get a refund?" it usually gives you a response related to your question.
- When you ask a question that's unclear or unrelated to the website's purpose, like asking the shopping chatbot "Got any tickets for sister?" the response will be more along the lines of "I'm sorry, I didn't understand your question."

This happens because the chatbot is programmed to answer only limited questions about a particular subject.

Even with this limitation, chatbots are useful in fields ranging from retail sales to immediate care medicine. You can engage with a chatbot online at any time. A chatbot is always ready for your question (even if it cannot answer you). Some institutions use chatbots often because they don't just broadcast ads like TV commercials. When people engage with them, chatbots listen, and they answer repetitive questions that a business would otherwise need to pay humans to handle.

Chatbots work with **small data**. This means their scale is much more limited. A movie chain's chatbot might need to answer questions only about movie titles,

locations, and times, while a more general AI that searches social media might need to answer broad questions about what millions of people think. AI researchers say that chatbots “snack on small data”.

A chatbot has a “frontend” and a “backend”

The frontend of a chatbot is the messaging channel. The frontend interacts with the person who’s asking questions, both listening (or reading) and speaking (or presenting text).

The backend of a chatbot is where the hard work takes place. The backend operates application logic and has enough memory to remember earlier parts of a conversation as dialog continues

---

A chatbot’s backend does the hard work of understanding and responding

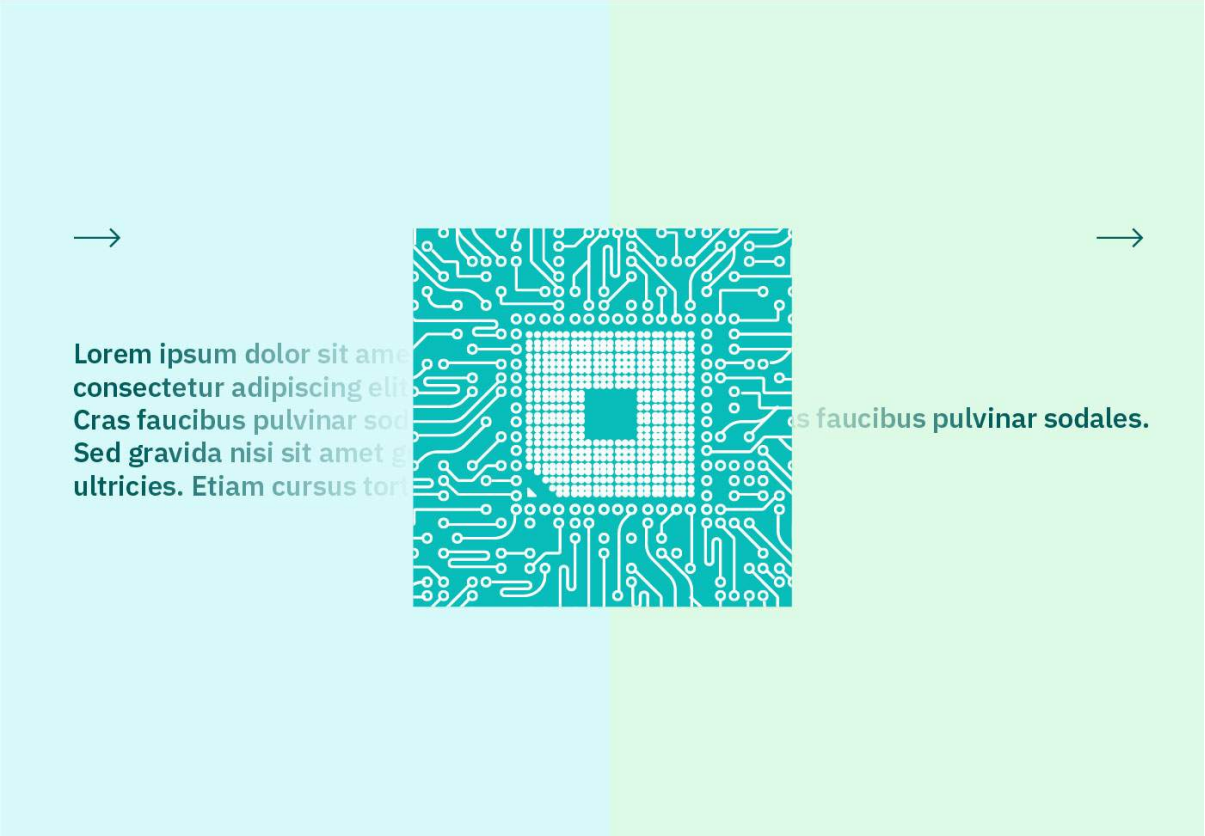
Suppose part of a chatbot’s job is to help you when you've lost or forgotten your password. You might enter, “How do I reset my password,” which would be easy for the chatbot to handle. To describe that action, a chatbot programmer might think in terms, such as:

```
IF question = “How do I reset my password”
THEN reply = “Here’s how to create a new password”
```

But humans are not always easy to understand. They might ask or type:

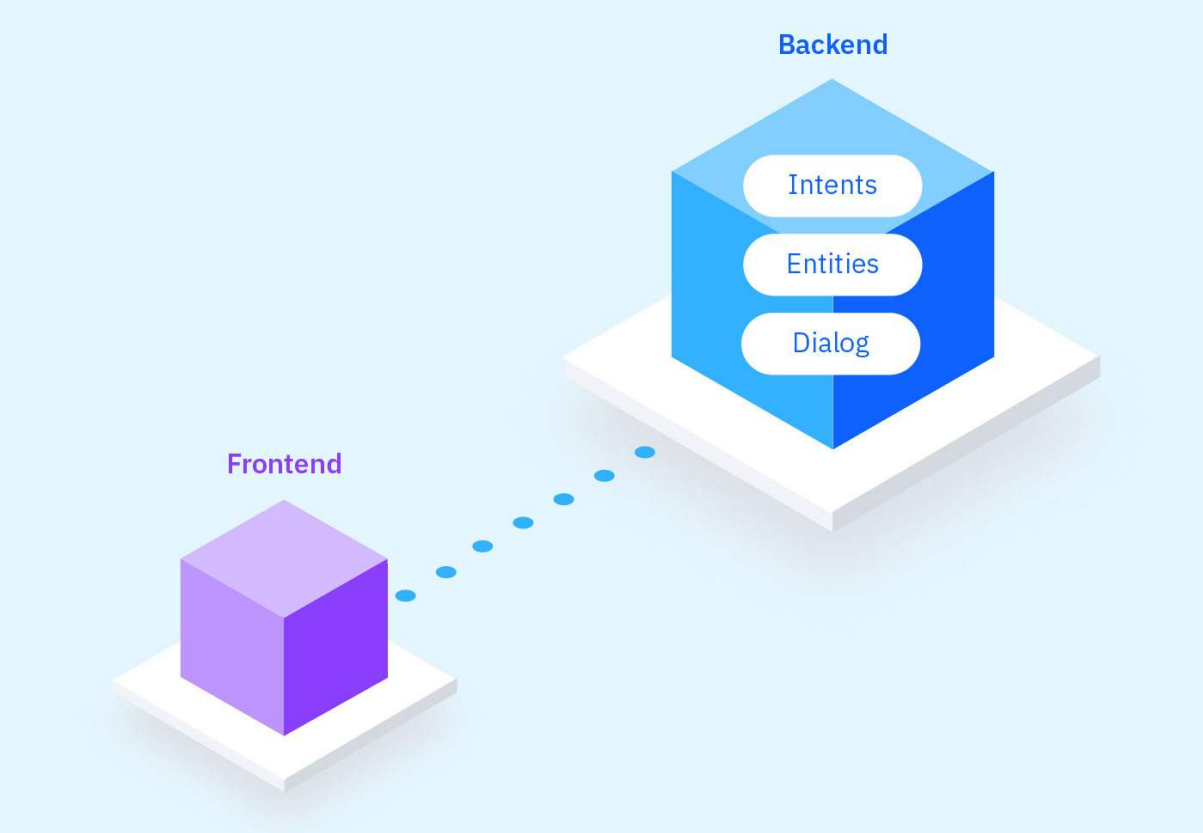
- “How come I can’t log into my account?!”
- “I forgot my @#$ password!”
- “It says my password is wrong.”
- “fergot paswrd”

This tells you that one chatbot response (“This is how to create a new password”) can be triggered by a large number of user queries (including all those listed, and more).



This is a great job for algorithms called **classifiers**. Classifiers can map many different ways of asking a question to a very small set of answers. How small? Some retail chatbots respond to hundreds of different questions with only five or six possible answers. Questions the chatbots can't answer are sent to human customer service representatives.

A chatbot backend usually includes three parts: intents, entities, and dialog



Chatbots **understand** a question by breaking it into parts and relating those parts to things in its memory. A chatbot's goal is to identify what are called **entities** and **intents**, then use what it's found to trigger a **dialog**. What do these terms mean?

Intent

An **intent** is a purpose: the reason why a user is contacting the chatbot. Think of it as something like a verb: a kind of action. A user might intend to file a complaint, to ask for directions, to speak to a salesperson.

An institution might have many customer or member intents that it would like a chatbot to handle. Suppose you've been hired to help create a chatbot for a restaurant chain. One possible intent would be to find out when the restaurants are open. You might first interview a person who previously handled many forms of this question on the phone. Then, your task would be to list all the different ways a caller might ask about when the restaurants are open. The following table provides many examples of possible user inputs that map to this kind of intent.

Intent	**Possible user inputs**
Open	When do you open?
	What are your hours?

	You open now?
	How late are you open?
	Can I walk in at 7 pm?

Once you have a list of the intents you want your chatbot to fulfill, you're ready to continue.

Entity

An **entity** is a noun: a person, place, or thing. This concept was covered earlier in this course.

If a user asks, "What are the hours for the Austin location?", then providing business hours is the **intent** and Austin is the **entity**. A chatbot needs a full list of entities in order to be helpful.

The following table lists examples of entities that map to the intent and possible user inputs of the previous restaurant chain example.

Intent	**Possible user inputs**	**Entities**
Open	When do you open?	Austin
	What are your hours?	Schedule
	You open now?	Time
	How late are you open??	Time
	And so forth; there are many **inputs** mapping to this intent.	And so forth; there are many **entities** implied by this intent.

Now that you've learned about entities and intents, let's talk about dialogs.

Dialog

A **dialog** is a flowchart—an IF / THEN tree structure that illustrates how a machine will respond to user intents. A dialog is what the machine replies after a human asks a question. Even if a human uses run-on sentences, poor grammar, chat messaging expressions, and so on, artificial intelligence allows the NLP to understand well enough to provide a response.

The dialog represents each possible word or phrase a user might enter, the matched response for the chatbot, and the many possible subsequent replies a user might make next. That's too much for an ordinary flowchart to show (you might need three or four dimensions!), so chatbot software condenses each moment of the conversation into a **node.** A node contains a statement by the chatbot and a long, expandable list of possible replies.

Planning this flowchart would be an adventure! You'd need to assign a reply to every possible user input after the chatbot's greeting. For the restaurant hours example, all possible questions about a restaurant's hours would lead to a single reply. This would continue for the next question (perhaps for the restaurant's address), and so on. A large number of possible questions would be mapped onto a small number of possible answers until the conversation is ended. (Spoiler: It helps that today's virtual assistants already have been trained on Wikipedia, so they know, for example, the difference between a wise man and a wise guy.)

Intents, entities, and dialogs make quick work for NLP

In a conventional computer, the program code is stupendously large but wouldn't handle intents, entities, and dialogs very well. A conventional computer would need a separate IF / THEN line for many thousands of ways a question might be asked. Unless a human were to match one of those lines perfectly, the computer would fail.

But an AI system's combination of NLP with intents, entities, and dialog can make quick work of this. NLP analyzes sentence components, then uses processes like passage and evidence scoring to classify the sentence components against possible chatbot responses. The result is that when a human user asks a question, the AI system provides the answer with the highest confidence.

Consider Staples, the office supply company that features the **Easy** button. Suppose a customer asks for a chatbot on Staples' website. The chatbot's frontend receives the customer's inquiry, then forwards it to the chatbot's backend. There, IBM Watson Assistant runs the NLP to understand the human's intentions regarding ordering a product or tracking a shipment.

If the customer says, "I want to reorder black pens," the chatbot figures out what that means. Then, it uses cognitive services to pull up the customer's purchase history. Within seconds, the system is helping the customer buy more pens.

The Staples chatbot only knows five things, but it knows them well. It supports online sales 24 hours a day, giving callers round-the-clock service while leaving human staff free to do other work. This is valuable for both the customers and the business.