Chapter 10: Watson Conversations

Learning Bluemix & Cognitive

Bob Dill, IBM Distinguished Engineer, CTO Global Technical Sales

Chapter 10: Getting started with Conversations

- **Know your goal**
  - First thing's first: Write a story for how a person will interact with this 'conversation' service
  - "In many cases, it makes sense to start small on well-defined questions you know users will ask and expand from there. Experience has told us that it's not until you release into production do you really find out what customers are going to ask."

    Brian Loveys, Program Director for Watson Emerging Technologies

- Define Intents
- Define Entities
- Create your first dialog

- Resources:

https://www.ibm.com/blogs/watson/2016/10/app-developers-guide-creating-first-watson-bot/
https://www.ibm.com/blogs/watson/2016/07/build-chat-bot/
https://www.ibm.com/blogs/watson/2016/08/getting-started-watson-conversation-6-steps/
https://www.youtube.com/watch?v=DmvN6ZJrZE4&cm_mc_uid=&cm_mc_sid_50200000=1479130251
https://www.youtube.com/watch?v=oSNF-QCbuDc&cm_mc_uid=&cm_mc_sid_50200000=1479130251
https://www.youtube.com/watch?v=3HSaVfr3ty0&cm_mc_uid=&cm_mc_sid_50200000=1479130251
https://www.youtube.com/watch?v=wG2fuliRVNk&cm_mc_uid=&cm_mc_sid_50200000=1479130251

- User Story …

Chapter 10: Reuse and all that

- The dialog, intents, and entities for this tutorial came from a great tutorial by Zach Walchuk
- His work can be found here: https://www.ibm.com/blogs/watson/2016/12/build-chat-bot/

Do This First

July 27, 2016
Posted In: Developers + APIs

Build a chat bot in ten minutes with Watson

We recently released the Watson Conversation service to make the difficult task of automatically understanding and responding to customers as simple as possible. The service provides a nifty GUI for training and configuring intents and dialog flow. As part of the launch, we've created some handy getting started resources, including a video, a demo, and a basic tutorial.

Zach Walchuk
Dev Evangelist, IBM Watson

Chapter 10: Defining Intents

- Defining intents
  - An intent is a purpose or goal expressed in a customer's input, such as answering a question or processing a bill payment. By recognizing the intent expressed in a customer's input, the Conversation service can choose the correct dialog flow for responding to it.
- Planning your intents
  - To plan the intents for your application, you need to consider what your customers might want to do, and what you want your application to be able to handle. Choosing the correct intent for a user's input is the first step in providing a useful response. The intents you identify for your application will determine the dialog flows you need to create; they also might determine which back-end systems your application needs to integrate with in order to complete customer requests (such as customer databases or payment-processing systems).
- Procedure to identify the intents you need for your application
  - Gather as many actual customer questions, commands, or other utterances as possible. Using input from real users gives a better picture of the expected input than having experts create lists of possible utterances. Remember that customers might phrase the same kind of request in many different ways. For example, the following examples all represent requests for weather information:
    - Tell me the current weather conditions.
    - Is it raining?
    - What's it like outside?
  - After you have a list of examples, sort them into categories based on the capabilities you want your application to support. These categories represent the intents you will define, and the examples will help your application to identify the intents in new input. As you plan your intents, keep the following guidelines in mind:
  - Do not make your intents too similar. Similar intents can be difficult for the Conversation service to distinguish. If you find that you have several intents that are close in meaning, consider whether you could combine them into a single intent, and then use entities to provide multiple possible responses for that intent.
  - Remember that a customer's input does not have to be an exact match for any of your examples; intents are recognized using natural-language processing.
  - Continue to refine your intents and examples as needed. Do not think of your set of intents as a finished product. It is likely that when you design your dialogs, you will identify additional intents that you need to add. You can also continue to gather input from new customers and use it to add new examples; this iterative process improves your application's ability to recognize intents accurately.

Chapter 10: Defining Entities

- An entity represents a class of object or a data type that is relevant to a user's purpose. By recognizing the entities that are mentioned in the user's input, the Conversation service can choose the specific actions to take to fulfill an intent.
- Planning your entities
  - An entity represents a term or object in the user's input that provides clarification or specific context for a particular intent. If intents represent verbs (something a user wants to do), entities represent nouns (such as the object of, or the context for, an action). Entities make it possible for a single intent to represent multiple specific actions. An entity defines a class of objects, with specific values representing possible objects in that class.
  - For example, suppose your application is a cognitive car dashboard that enables users to turn accessories on or off. Consider the following utterances:
    - Headlights off
    - Turn the radio off
    - Stop the air conditioner
  - These requests all represent the same intent (to turn something off), which you might call #turn_off. You would then use an entity to represent what the user wants to turn off. To do this, you might create an entity called @accessory, and give it the following possible values:
    - headlights
    - radio
    - air conditioner
  - For each value, you can also specify synonyms. For example, you might specify AC and cooling as synonyms for air conditioner; these strings would all be treated as the same value. Listing multiple ways that users might refer to the same value helps improve your application's accuracy.
  - When the user's input is received, the Conversation recognizes both intents and entities. Your dialog flow can then use these to provide the best answer. You should only create an entity for something that matters in terms of changing the way the application responds to an intent.
  - In addition to the entities you define, the Conversation service provides a set of system entities that are already defined and available for you to use. For more information, see Enabling system entities.
  - If you're using a language other than English, see Language considerations.

Chapter 10: Building a dialog

- The dialog component of the Conversation service uses the intents and entities that are identified in the user's input to gather required information and provide a useful response. Your dialog is represented graphically as a tree; create a branch to process each intent that you define.
  - Plan the responses that you want to make to each possible user input.
- The dialog component of the Conversation service provides responses to users based on the identified intents and entities. Create a dialog branch for each intent, to gather any required information and make a helpful response. More details are provided in subsequent topics for these high-level steps.
- Procedure
  - Choose the first intent for which you want to create a dialog branch. Choose an intent that can be processed simply, so that you can become familiar with the tools and concepts.
  - Determine the response that you want to provide for this intent. Depending on the purpose of your application, you might provide a simple text response that answers a question, or you might tell the application to call a back-end application using the information that you have gathered.
  - Identify the information that you need to gather to make a helpful response. For example, if you want to answer questions about weather forecasts, you would need to know the location and the date. If you want to enable users to order a pizza, you might need to know the size, the toppings, and so on. In many cases, you will have entities defined to contain these pieces of information.
  - Define the structure of the context data that your dialog branch can use to pass data back and forth with the application, and to retain data from the beginning of the branch until you provide the response.
  - Build a sequence of dialog nodes that captures the information that you need to enable you to provide a useful response. Think of possible user inputs and identify what additional information you need. For example, if a user asks What is the forecast for Austin?, you would need to ask for the date.
  - Test your dialog branch as you build it. Enter a variety of inputs to verify that the correct intents and entities are recognized and that you can gather the information that you need.
  - Choose the next intent to work on and repeat these steps until you have defined a dialog branch for each intent that your application needs to support.

The Plan: 30 minute Chapters with an hour or two of practice

1. The Story, Architecture for this app	
2. Setting up Bluemix	
3. Building your first Watson App	(Watson Speech to Text)
4. Getting Watson to talk back	(Watson Text to Speech)
5. Understanding Classifiers	(Watson NLC)
6. Creating a custom dialog with Watson	(custom Q&A, session management)
7. Authentication	(puts C2 thru 6 together)
8. Alchemy News	(Watson Alchemy)
9. Visual Recognition and Images	(Watson Visual Recognition)
10. Watson Conversations	(Watson Conversations)
11. Rank & Retrieve	(Watson Alchemy + Rank & Retrieve)
12. Getting started on my first client prototype	Design Thinking, Stories, Architecture, Keeping it simple

Chapter 11: Rank & Retrieve

Learning Bluemix & Cognitive

Bob Dill, IBM Distinguished Engineer, CTO Global Technical Sales