Practical Worksheet 9

Version: 1.0 Date: 12/04/2018 Author: David Glance

## Learning Objectives

1. Write an application using natural language processing
2. Discover some of the elements needed to write a chatbot

## Technologies Covered

Amazon Comprehend

Python

Boto

## Travel Chatbot

Background

The aim of this lab is to write a program that will do simple question and answers with someone that is reporting a travel itinerary. It will take unstructured conversational text and create an itinerary of where people travelled and what they visited. It will prompt for a comment about the trip and detect the sentiment, responding in an appropriate way.

The documentation for the Amazon Comprehend is <https://docs.aws.amazon.com/comprehend/latest/dg/how-it-works.html>

## [Step 1] Build the itinerary

The program should start with an explanation of what it will do and then ask for the first part of the itinerary.

The program will prompt the user as follows:

> You can tell me “who” is going “where” on what “date” to see “what”

The person can then enter text such as:

John and Mary Jones travelled to Paris on the 15th of October 2018 to see the Eiffel Tower

The program will validate this and then create an itinerary entry:

Person: John

Person: Mary Jones

Travelling to: Berlin

On: 15/10/2018

Reason: see the Eiffel Tower

If the text being entered leaves out any of the components, the program should prompt for that information:

e.g.

> Who is travelling to Paris?

Of course, if the location or date is left out, these should be prompted for as well:

> Where are you going?

> On what date will you be travelling?

> What did you see?

## [Step 2] Sentiment Analysis

Using AWS Comprehend, allow the user to enter a comment on what they thought of the trip and detect the sentiment of the comment. Depending on the sentiment, reply with an appropriate response:

If the sentiment was positive: “That sounds like you had a great time”

If the sentiment was negative: “I am sorry to hear that”

If the sentiment was neutral: “That sounds interesting”

#### Create a Lex Chatbot

## [Step 3] Create a custom slot type

Using the AWS CLI, create a custom slot type using the following json file

{

"enumerationValues": [

{

"value": "tulips"

},

{

"value": "lilies"

},

{

"value": "roses"

}

],

"name": "FlowerTypes",

"description": "Types of flowers to pick up"

}

and the following command:

aws lex-models put-slot-type \

--region region \

--name FlowerTypes \

--cli-input-json file://FlowerTypes.json

## [Step 3] Create an intent

Using the AWS CLI, create the intent using the following json file

{

"confirmationPrompt": {

"maxAttempts": 2,

"messages": [

{

"content": "Okay, your {FlowerType} will be ready for pickup by {PickupTime} on {PickupDate}. Does this sound okay?",

"contentType": "PlainText"

}

]

},

"name": "OrderFlowers",

"rejectionStatement": {

"messages": [

{

"content": "Okay, I will not place your order.",

"contentType": "PlainText"

}

]

},

"sampleUtterances": [

"I would like to pick up flowers",

"I would like to order some flowers"

],

"slots": [

{

"slotType": "FlowerTypes",

"name": "FlowerType",

"slotConstraint": "Required",

"valueElicitationPrompt": {

"maxAttempts": 2,

"messages": [

{

"content": "What type of flowers would you like to order?",

"contentType": "PlainText"

}

]

},

"priority": 1,

"slotTypeVersion": "$LATEST",

"sampleUtterances": [

"I would like to order {FlowerType}"

],

"description": "The type of flowers to pick up"

},

{

"slotType": "AMAZON.DATE",

"name": "PickupDate",

"slotConstraint": "Required",

"valueElicitationPrompt": {

"maxAttempts": 2,

"messages": [

{

"content": "What day do you want the {FlowerType} to be picked up?",

"contentType": "PlainText"

}

]

},

"priority": 2,

"description": "The date to pick up the flowers"

},

{

"slotType": "AMAZON.TIME",

"name": "PickupTime",

"slotConstraint": "Required",

"valueElicitationPrompt": {

"maxAttempts": 2,

"messages": [

{

"content": "Pick up the {FlowerType} at what time on {PickupDate}?",

"contentType": "PlainText"

}

]

},

"priority": 3,

"description": "The time to pick up the flowers"

}

],

"fulfillmentActivity": {

"type": "ReturnIntent"

},

"description": "Intent to order a bouquet of flowers for pick up"

}

using the following command

aws lex-models put-intent \

--region region \

--name OrderFlowers \

--cli-input-json file://OrderFlowers.json

## [Step 4] Create the bot

Use the following json

{

"intents": [

{

"intentVersion": "$LATEST",

"intentName": "OrderFlowers"

}

],

"name": "OrderFlowersBot",

"locale": "en-US",

"abortStatement": {

"messages": [

{

"content": "Sorry, I'm not able to assist at this time",

"contentType": "PlainText"

}

]

},

"clarificationPrompt": {

"maxAttempts": 2,

"messages": [

{

"content": "I didn't understand you, what would you like to do?",

"contentType": "PlainText"

}

]

},

"voiceId": "Salli",

"childDirected": false,

"idleSessionTTLInSeconds": 600,

"description": "Bot to order flowers on the behalf of a user"

}

and the following command

aws lex-models put-bot \

--region region \

--name OrderFlowersBot \

--cli-input-json file://OrderFlowersBot.json

This will take a little bit of time to create so use this command to see if it is ready:

aws lex-models get-bot \

--region region \

--name OrderFlowersBot \

--version-or-alias "\$LATEST"

## [Step 5] Test the bot

You can now test the bot using the following commands:

aws lex-runtime post-text \

--region region \

--bot-name OrderFlowersBot \

--bot-alias "\$LATEST" \

--user-id UserOne \

--input-text "i would like to order flowers"

aws lex-runtime post-text \

--region region \

--bot-name OrderFlowersBot \

--bot-alias "\$LATEST" \

--user-id UserOne \

--input-text "roses"

aws lex-runtime post-text \

--region region \

--bot-name OrderFlowersBot \

--bot-alias "\$LATEST" \

--user-id UserOne \

--input-text "tuesday"

aws lex-runtime post-text \

--region region \

--bot-name OrderFlowersBot --bot-alias "\$LATEST" \

--user-id UserOne \

--input-text "10:00 a.m."

aws lex-runtime post-text \

--region region \

--bot-name OrderFlowersBot \

--bot-alias "\$LATEST" \

--user-id UserOne \

--input-text "yes"

## [Step 6] Create a script to take user input and interact with the bot

Create a shell script to take in input and interact with the bot.

## Submission and Quiz

Submit the python file you wrote in [2] and the shell script from [6]– respond to the quiz

## Respond to the Quiz

[1] A Turing test

[A] Tests whether a chat bot can interpret natural language

[B] Tests whether a human tester can trick an AI into thinking it is communicating with another AI

[C] Tests whether a human tester can distinguish the gender of an AI pretending to be male or female

[D] Tests whether an AI can distinguish the gender of a human tester

[2] Amazon SageMaker:

[A] Is an AMI set up to run AI frameworks

[B] Runs a Jupyter server that can run any custom code

[C] Runs a Jupyter server that runs a set of built-in algorithms

[D] Is another name for AWS’s DeepAR forecasting service