Watson Developer Cloud Technical Workshop

Summary

This document lists the steps to follow to create a cognitive application using IBM Bluemix. The application would have a “Natural Language Classifier” service which will be trained using a sample set of texts from tweets indicating complaints, queries or feedbacks. The application will be based on Node-red which will read twitter mentions of a particular handle, use the NLC Service to identify the type of tweet and based on the type of tweet, will automatically reply to that tweet with an appropriate response.

Instructions for creating your cognitive application

- Login to Bluemix - https://console.ng.bluemix.net
    - Register a new Bluemix account if not already done so
- Create a Natural Language Classifier service from Bluemix Catalog
- Open the newly created service and click on “ Access the Beta Toolkit”
- In the toolkit application and click on “Sign in with Bluemix” and complete the sign in process.
- Create a new classifier and add representative text and class for each of them.
    - Click on Training tab at the top.
    - In classes textbox, enter the classification you want to create and press enter. For example, you can create classifications like “Complaint”, “Feedback” or “Query”.
    - Start entering representative text and press enter in the “Text” field. Once the text shows up in the bottom, click on the + next to the text and enter the classification you want for that text and press enter.
    - Repeat the above step for all the representative text you want to use.
    - Alternatively, you can also create a csv file with the questions and classifications and import it to create a new classifier (A sample CSV file attached)
    - Click on Train link and provide a name to the classifier and press “Train”. It will create a classifier with a classification ID. Note the id for use in a later step.

- Create a Node-Red application
    - From Dashboard, select create app.
    - Choose “web” for type of app.
    - Choose to Browse Boilerplates
    - Select Node Red Starter community
    - Provide app-name and create the app.
- Open the node-red application in node-red editor
    - Open node red editor through the URL <appname>.mybluemix.net/red. If your app name was myTwitterBot, open the node-red editor using myTwitterBot.mybluemix.net/red
- Drag and drop a “twitter input” node into node red flow editor. You can find this node under “Social” category. There would be 2 twitter nodes. Mouse over them and the text will show which one is input vs output.
- Configure twitter input node with necessary details
    - Double click twitter input node and use the edit button for Twitter ID to connect to your twitter account. You could create a new twitter account for this workshop or can reuse your existing twitter id (You may want to inform your timeline about the workshop and some test tweets they would see originating from you).
    - Choose to search for all public tweets
    - In “for”, type @<twitterhandle>. So if your twitter handle is nlctest1, enter @nlctest1 in the “for” field. This will ensure that all your twitter mentions will be captured by this node.
    - Provide a name to this node for ease of use.
- Drag and drop a “function” node next to this twitter node. You can find this node under “function” category.
- Double click the function node and enter the following text

    msg.sender=msg.tweet.user.screen_name;

    return msg;
- Give a meaningful name to this function node
- Connect the output from twitter node to the input of this function node.
    - Click on the circle at the right side of twitter node and drag the line and connect it to the left side of the function node.

- Drag and drop a “Natural Language Classifier” node next to the function node.
- Configure the “Natural Language Classifier” node
    - Give a meaningful name
    - Provide the username/password noted earlier in the NLC service in the username and password field.
    - Keep the mode as Classify
    - Provide the classifier id of the classifier created earlier. This can be noted from the toolkit.
- Connect the output of the earlier function node to the input of this NLC node.
- Create a new function node.
- Write logic here to create new twitter message based on the classification of the incoming tweet. For example, if the 2 classifications are “Feedback” and “Complaint”, the following code will create the output message to be tweeted

```
if (msg.payload.top_class == "Feedback") {
   msg.payload="@"+msg.sender+" Thanks for your feedback";
}
else if (msg.payload.top_class == "Complaint") {
   msg.payload="@"+msg.sender+" We are sorry. Our customer rep will contact you shortly"
}
else {
msg.payload="@"+msg.sender+" You can look for more information in our website"
}
return msg;
```

- Connect the output of NLC node to this newly created function node
- Create a twitter output node and configure it
    - In twitter ID, use the same ID that was used in the earlier twitter input node
    - Provide a meaningful name to this node
- Connect the output of the last function node to this new twitter output node.

- Press Deploy button at top right to deploy this application.

Now the application is ready and listening to mentions for our twitter handle. Try sending a tweet from any handle mentioning this target handle with either a feedback or complaint. It need not be the exact text that was used for training NLC. The application will use NLC service to find the classification of tweet and respond with one of the configured responses based on the classification. You can check that by checking the timeline in twitter.

References:

Watson Developer Cloud - http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/

Watson Developer Community - https://developer.ibm.com/watson/

Cognitive Developerworks - http://www.ibm.com/developerworks/cognitive/

Watson youtube channel - https://www.youtube.com/channel/UCxPJljXUHvUd9idyfEHvXqg

Watson Twitter channel - https://twitter.com/IBMWatson