Mr. Interviewer - An Interviewing Chatbot

Anjala Thulasiram
Computer Engineering Department
San Jose State University
San Jose, USA
anjala.thulasiram@sjsu.edu

Renu Dighe
Computer Engineering Department
San Jose State University
San Jose, USA
renu.dighe@sjsu.edu

Varun Potlacheruvu
Computer Engineering Department
San Jose State University
San Jose, USA
tushar.sharma@sjsu.edu

Tushar Sharma
Computer Engineering Department
San Jose State University
San Jose, USA
tushar.sharma@sjsu.edu

***Abstract*— In the modern era of technology, Chatbots and the conversational services they provide are the next big thing. A chatbot essentially is a virtual person who can effectively talk to any human being using interactive textual/voice skills. Currently, there are many cloud-based Chatbot services which are available for development and improvement of this sector such as IBM Watson, Microsoft Bot, AWS Lambda, Heroku and many others. A virtual person is based on Machine Learning and Artificial Intelligence (AI) concepts and due to dynamic nature, there is a drawback in the design and development of these chatbots as they have built-in AI, NLP, programming and conversion services. This paper gives an overview of chatbots.**

Keywords—chatbots, interview, artificial intelligence, natural language processing

I. INTRODUCTION

Nailing that dream job interview is one of the most nerve wracking moments in anyone's career. Leading up to the interview, nervousness and anxiety kick in and people tend to forget the basic questions and answers. This is why we have built a chatbot to serve this very purpose of helping kill those feelings and make sure you do well.

Our system is built using the help of the IBM Watson Assistant which is available on the IBM Cloud catalogue. We developed the engine with the knowledge of intents, entities and dialogs to keep track of how the flow of the conservation occurs in real time. As we are creating this from scratch, our knowledge base is limited to topics like Python, Java or Data Science. Our chatbot is programmed to give you options for you to choose which topic to train under. After typing in one, the user will be asked 5 questions and they will try to answer to the best of their knowledge. If the answer is incorrect, the bot will provide the correct response and how to frame your input better. This gives the user the chance to learn from their mistakes and better themselves for real-life situations in the future.

This application has been put up on the IBM cloud via a Bluemix.net link. It is a simple design which allows the user to clear view of the display and focus on the interview training aspect of the app. We have also integrated the application to work on Facebook Messenger as well as Slack which proves to be a great way to widen the audience pool to ensure everyone on the big platforms can access and use the application.

II. RELATED WORK

We have researched extensively and came to the conclusion that such bots don't exist for serving this purpose of training for interviews. What we came across is a bunch of FAQ bots where the roles are reversed, in the sense that the user asks the question and bot replies with the proper answer. Many bots are related to fields such as restaurants or customer service. Personalized apps to help an individual in different areas like our Chatbot haven't been designed enough so we took this opportunity to push this project through.

III. IBM WATSON ASSISTANT

Watson Assistant Chatbots involve three broad phases namely: Design, Scope and Integrate.

Scope - It is the first step to consider. You need to gather the requirements from the user and also other specifications that might be important to be considered right from the beginning.
Design - Next step is to create an instance of the Watson Assistant to build the required functionalities. The intents and entities of the chatbot are explored in this phase which have been described in the further content.
Integrate - Last step is to integrate the Watson Assistant on platforms to serve it as a solid application. The platforms can be Cloud, Slack and Facebook. This can be further made business adaptable by exploiting more of IBM's tools like IBM Watson Retrieve and Rank Service, Watson Discovery, etc.

Going a bit deeper into the architecture of IBM Watson Assistant, we have been presented with three main building blocks:
Intent - Every bot has certain objectives for which it is being used, intents specify these objectives. For example, our mock interview chatbot has the objective of testing the interviewee so the intent can be '#ready_for_interview'. Intents are recognised by the symbol '#'. Intents generally answer the questions 'what', 'why', 'when', 'where'.
Entity - The difference between intents and entities is quite a fine one. To describe entities, they can be thought of as subjects of intents. So in the intent '#ready_for_interview', the entity is 'interview'. Our chatbot has multiple entities which pose as the answers of the different interview questions. Each entity will have the option to add values which can recognise keywords in the answers to help differentiate between correct and wrong answers. More the number of values better the functionality of the respective

entities. There is also an option to add synonyms to the values to have multiple words with same meanings clubbed in one value. Entities are recognised by the symbol '@'.
Dialog - We now have intents and entities with us. Additionally, we need the dialog module which is essential considering we are building a conversation bot, a chatbot. It is what enables interaction with the users. The dialog module is like a logic tree having many if-then conditions. Dialogs in our interview chatbot are essentially the questions asked by the bot and the responses given when the user answers those questions, as in whether the answer is correct or wrong and if wrong what is the correct answer.

Along with that we also have some more modules like-
Utterance - Utterances in the Conversation API refer to the different questions your end users ask your bot.
Node - Nodes are individual contents of a dialog. A default node is the welcome node which greets the user when he begins using the chatbot. Every node can have child nodes as well to have a nested functionality to the dialog.
Context Variables - They can be defined in the node and can also specify a default value for them. Other nodes can set or change the value of the context variable. We can pass information from the application to the dialog by setting a context variable and passing the context variable to the dialog.
System Entities - They are common entities which can be used for the application. Enabling a system entity makes it possible to quickly populate your workspace with training data that is common to many use cases. For example, @sys-number, @sys-date, @sys-time, etc.

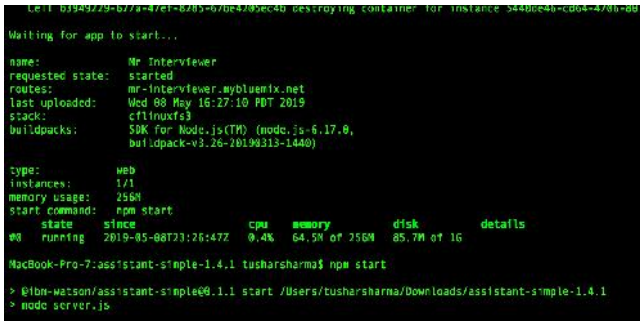
The process flow is somewhat like this - each intent begins a node on the left and the logic flows from the top down through your intents. If a certain intent is triggered by an utterance, its node is opened and the logic continues to entities.
The logic within each node (i.e. through entities) also flows from top to bottom. A specific combination of #Intent and @Entity:value triggers a certain response to a question – this combination is referred to as the response condition.

IV. Design

We have used example starter code for a chatbot app and connected it to the assistant using the v1 API service.

For the front-end we used HTML, CSS and JS with some bootstrap code to design a simple UI for the chatbot app. In the UI, we construct a new DOM element for both messages that the user sends as well as the responses received from the assistant which is then used in a recursive function to add responses to the chat area. An 'adjustInput' function is used as an event listener to resize the green underline under the input box, so it follows the size of the input.

For the app server, we've used node with express, which uses bodyparser to parse the JSON payloads we created into http requests for the assistant. We then deployed the app to the cloud as a cloud foundry application. For this we had to log in to the IBM cloud via the CLI, target a cloud foundry application and space push it to the IBM cloud using IBM Cloud app push.



V. Conclusion

Users look for the best, reliable and user-friendly applications. When we first worked on the concept, we listed a couple of features that we thought would be important for the users. We designed the chatbot after realizing how inadequately people from all backgrounds and experience levels were preparing for potentially life changing interviews. We offer them with questions that boost their interview skills, strengthen responses and gain confidence for their interview. We made use of the IBM Watson Assistant which will instantly begin curating your questions for the interview.

If we had an extended project timeframe and fewer constraints, we could have done it in a much bigger way. If

we would take the project further, the first thing is to take feedback on the application. For instance, how can we use the AI to provide the best possible customized, reliable and accurate feedback to users? How to offer a personalized approach to the experience?

VI. Future Enhancements

Our recent version of the application is a chatbot application which works on the phone and laptop, which is useful for people who are preparing for interviews. Currently our chatbot is trained to do three kinds of interviews. For future enhancements, we can add hundreds of different kinds of interviews to this project so that many more people can benefit from our project. We could add another feature giving the scores for the performance of the users for a particular interview and suggest ways to improve and be prepared.

Acknowledgment

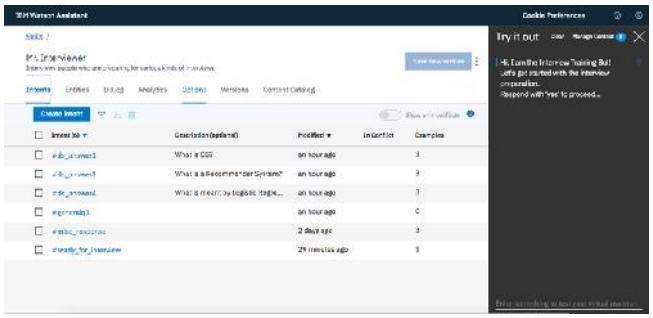
During the project development, we had to take the help and guidance of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much pleasure. We would like to show our gratitude to Prof. Rakesh Ranjan, Course Instructor, San Jose State University for giving us good advice and feedback for the project throughout numerous consultations.

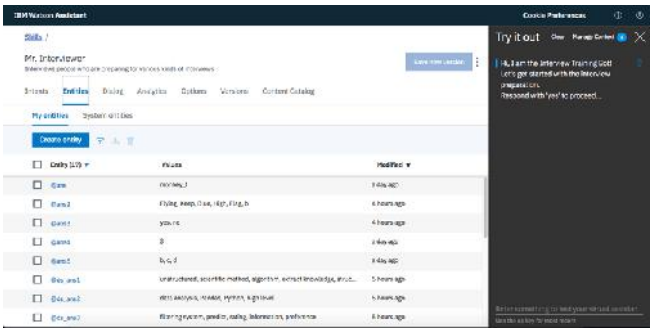
We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in successfully completing this project. Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help wholeheartedly.

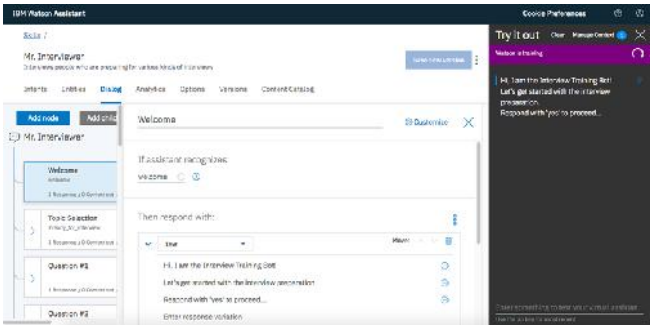
References

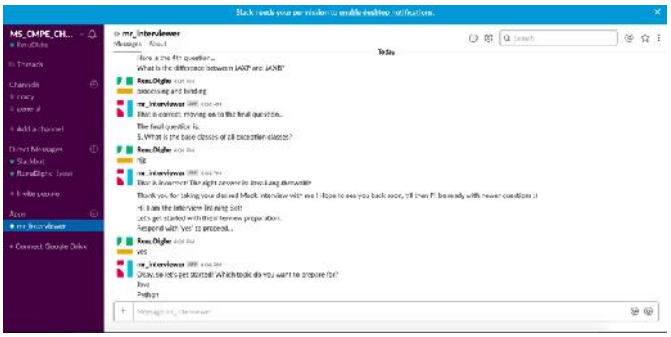
[1]Chatbot-
https://www.ibm.com/watson
[2]Skills-
https://cloud.ibm.com/docs/services/assistant?topic=assistant-skills
[3]Entities-
https://cloud.ibm.com/docs/services/assistant?topic=assistant-entities
[4]Dialogs-
https://cloud.ibm.com/docs/services/assistant?topic=assistant-dialog
[5] Node.js Framework-
https://nodejs.org/en/
[6]Java Questions-
https://www.softwaretestinghelp.com/core-java-interview-questions/
[7] Python Questions-
https://www.edureka.co/blog/interview-questions/python-interview-questions/
[8] Github Link for our project-
https://github.com/SJSU272Spring2019/Project-Group-3

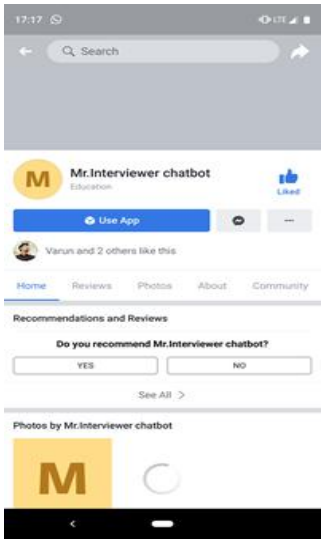
Screenshots










17:17
Search
Mr.Interviewer chatbot
Education
Liked
Use App
Varun and 2 others like this
Home
Reviews
Photos
About
Community
Recommendations and Reviews
Do you recommend Mr.Interviewer chatbot?
YES
NO
See All
Photos by Mr.Interviewer chatbot
M