Chatbot for pizza ordering system using Watson Conversation

Anonymous

ABSTRACT

When we order pizza we have either go on-line or we have to call customer service. This process is very tedious as we have to go through all the choices. Due to introduction of automated systems, the customers are finding it easy to interact with a machine, rather than actually talking with a person. This process makes it easy for the customers to interact with the system and they can extract as much information they want from the system. For example a pizza ordering system: the automated service system can provide as much information it wants on the menu which the restaurant provides and all of them in detail.

Our project delivers automatic responses for a pizza ordering system. We are using Watson Conversation Slots with intents, entities and dialogs to achieve a full conversation with the AI. We also deployed our project's instance on Slack application.

1. INTRODUCTION

Overseeing office procedures requires a high level of complexity as the customer expectation grows continuously. With new products coming into the market every day, the customer expectations are heightened and so the companies are looking forward to making shopping productive for them and interactive for customers. IBM Watson combined machine learning, natural language processing, and contextual understanding to create systems that can learn and interact with people [9].

In earlier days, use of social media was very high and by the invention of new technologies it attracted many people's attention. If any individual wanted to search anything they would simply query anything on the social media and would get their result. Hence, building this type of system requires a lot of maintenance and it should be able to query and deliver the desired results. The system had some problems that it wasn't able to query sometimes the data or information which the individual wanted or it couldn't return the result to the user well in time.

There has been some prior work where the authors have introduced the concept of conversation systems to solve those problems. One of the earlier work [8] proposed a chatbot for customers so that when the customers interact with the system it simply gives response on its own. The authors were able to get a decent accuracy means a decent response was given to the customers by the system. The users trained the system with 1 million of twitter data. The other work [7] shows information retrieval. The authors propose a framework for extracting useful information from the conversations among the people. They use this information while implementing the framework so that the chat could be easy but there are some limitations in their work as well.

In this project, we will use the IBM Watson's Conversation instance to develop a chatbot for pizza ordering system. The needed information such as size, type and ingredient choices will be entered in one Conversation node. This will be done with the help of Node.js, that can also handle the asynchronous event driven by JavaScript at the runtime and will also handle inputs from a web application implemented by us.

2. WATSON CONVERSATION

During the course of a conversation, there will be a back and forth conversation between the user and Watson Conversation system. The user will provide input and typically this comes in the form of text or speech. The Watson Conversation service will respond with JSON. The three building blocks of Watson Conversation are Intents, Entities, and Dialog [1].

2.1 Intents

Watson will try and understand the users Intent. An Intent, one of the primary building block, is a purpose or goal expressed by the user's input. An example of Intent is "I want to order pizza", we ask Watson to brief us about something. An intent is an expression of purpose by a user. Another example, "Turn on the lights", "Turn on the radio", and "Start the car" are all examples that express the intention of turning on something. In this case, the intent might be expressed in the Watson Conversation tooling as #turn-on.

2.2 Entities

An entity represents a class of object or a data type that is relevant to a user's purpose. In Watson Conversation, entities are denoted with '@' symbol. In our dataset, we have entities like pizza type, pizza toppings, pizza type, sauce type and extra confirmed. Each entity is made up of three

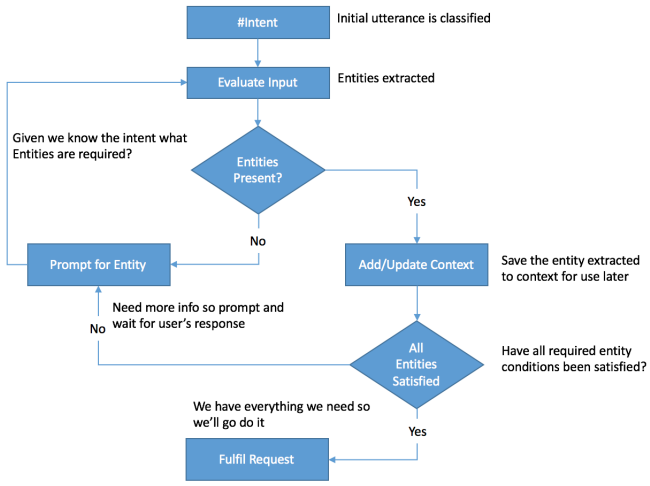

Figure 1: Basic flow chart of how the chatbot works.

components - the name, value and any synonyms for the value. In our dataset, the entity, 'pizza place' has a value take away; it has synonyms like go, home, take out, etc. If the user says, "I want to order a pizza and take it away". Here the intent is #order and the entity is @pizza_place, the value of this entity is take away. The #order intent is very generic as it applies to order movie tickets, order a book, order items on the website and much more. Our entity helps in focusing this order. Entities help Watson identify the exact thing that user is requesting to search.

2.3 Dialogs

The final block is the dialog, which is a logical flow that determines the responses provided when certain conditions are met[1]. The dialogs are basically the responses given by Watson by matching the entities or the intents or both from the input sentence given. Node represents a single interaction in the conversation. A node is triggered when its condition is met.Once triggered it provides any associated responses. There are special conditions, 'welcome', 'anything_else'. The condition when 'welcome' is set, it executes at the start of the conversation. The 'anything_else' executes when none of the above conditions are met. Placing the 'anything_else' condition at the end, ensures that the conversation always returns a response.

Figure 2: Intent and Entity detection by Watson conversation

The dialog uses the intents and entities that are identified in the user's input, plus context from the application, to interact with the user and ultimately provide a useful response. [3] The whole conversation happens in the form of nodes, parent node contains some configurations and then it evaluates its child nodes further to get a response and hence when nothing matches then there is a node to handle such conditions and it always returns the value from that node. Example: "Order a large Hawaiian pizza with alfredo sauce". Here, #order is the intent so the first node will traverse the intent and then it will further go down to traverse the entity and gives a proper response. For more information please refer the figure.

Conversation Slots reduces the number of nodes to a single node for each API call. The node itself is interesting because it allows us to fill in for entries and optional entries like pizza type, pizza size, where we are going to eat it and we can have some handlers for it.

Responses defines how to reply to the user. You can reply with one of these response types:

1. Simple text response
2. Conditional responses
3. Complex response

Simple text response is used if we want to provide a text response, simply enter the text that you want the service to display to the user. It gets boring if the user If your users return to your conversation service frequently, they might be bored to hear the same greetings and responses every time. You can add variations to your responses so that your conversation can respond to the same condition in different ways.

3. DATA SET DESCRIPTION

We have created our own dataset for the intent part and entity part. The 'intents.csv' contains rows of all the intents. For now, we have created five intents like greetings, help, order, exit and reset. IBM Watson uses this intent to recognize the intention of the user typing in. The 'entities.csv' contains the rows to identify the entity. In our dataset, we have entities like pizza type, pizza toppings, pizza type, sauce type and extra confirmed. Each entity is also attached with a value and synonyms.

Every entry in the data starts with the Intent that is what type of service the customer wants from the system ex: order, help. This tells the system what the next step will be example: if the customer chose help then the system will explain in brief how the system works and how the pizza can be ordered [2]. If the customer chose order then the system will recognize the text or speech of the customers and will go on to further process the order of the customers.

4. IMPLEMENTATION

IBM Bluemix provides various cloud services for Infrastructure and Platform. One of the service offered by BlueMix is Watson. Watson is used to build cognitive applications that help enhance, scale, and accelerate human expertise. For this project, we use the conversation interface of Watson.

Conversation is used to add a natural language interface to the application to automate interactions with end users. The basic applications for conversation would be virtual agents

and chat bots that can integrate and communicate on any channel or device. IBM provides a easy-to-use web application to train Watson Conversation service, so as to build natural conversation flows between application and users. This can be easily deployable, scalable and cost effective solutions.

We have created intents, entities and dialogs. There are two ways to create intents on the Watson Conversation [6].

1. We can directly create intents by clicking the create button and add intents and user-examples.

2. We can have a comma separated value, CSV file with intents and corresponding user examples.

For our project, we used the second way of importing the CSV file to the Watson intent. Once, the CSV data is imported, Watson parses the CSV file and creates intents and user examples corresponding to it. The intents are stored in the form of hash-tags. For example, 'greetings' : 'hi', 'how are you?', etc. The created intents were tested using the chat interface of Watson. We have tried examples like "Hi", "How are you?" and get the correct hash-tags. It is interesting to note that, the chat interface of Watson is case-insensitive and it has an inbuilt semantics parser. This is demonstrated when, we typed "hola" and Watson correctly tagged it to #greetings, while we do not have 'hola' in our user examples. The Watson interface also allows additional user-examples through chat testing, if it is not already present.

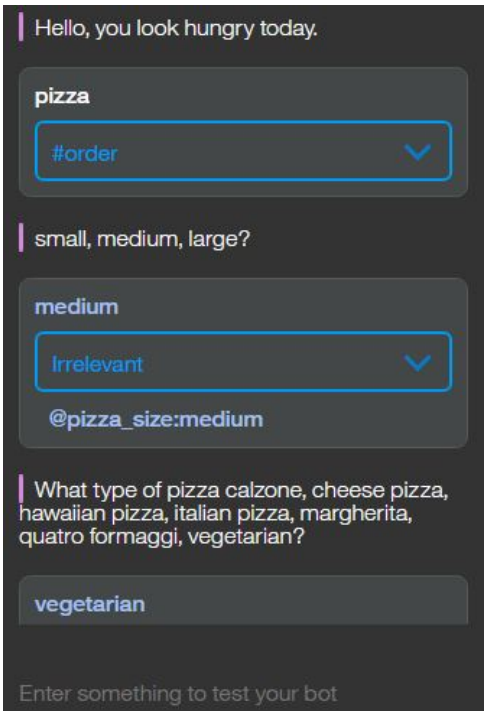

Figure 3: Conversation after adding conditional dialogs

Similarly, for the entity, we have a CSV file, which contains each entity value and the corresponding synonyms. There can be mutliple synonyms for each entity value. Watson parses the CSV file and stores the entity and corresponding synonyms. For example, pizza_size : 'large', 'medium', 'small'. We have tested the intents and entities using the Watson chat. For example, when we type "I want a large pizza", Watson correctly detects the intent as #order and 'large' as the entity value, pizza_size. This is demonstrated in figure 2.

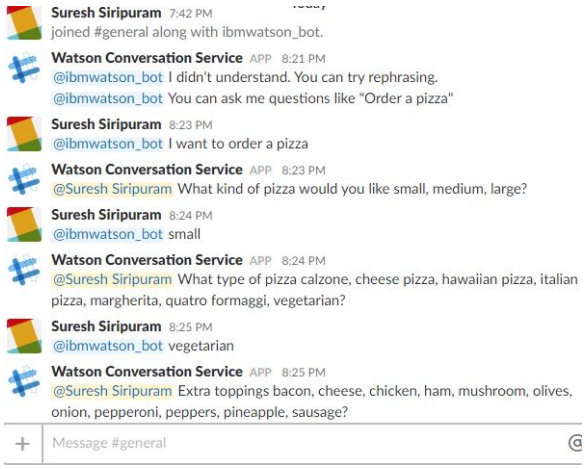

Figure 4: Intent and Entity detection by Watson conversation

For dialog, nodes are created for each of the response the conversation should led to. There are two special conditional nodes in the Conversations tab. Welcome is the node which works at the start of the conversation, it is triggered as soon as the conversation starts. Welcome is set true by default and the response is the first thing in the conversation. Another special conditional node is Anything else. Anything else node is executed when there is no conditions met by the Conversation. It is executed and we could have response like , 'I didn't get you'.

Watson Conversation Node consists of a dialog includes a name, condition, response and post-process. The dialog uses the intents and entities that are identified in the user's input, plus context from the application, to interact with the user and ultimately provide a useful response. [3] We created dialog nodes to consume all the intents and entities with proper responses. There were multiple response and funny language used. This is done so as not to make the application boring to interact to for the users. The anything else node was connected to the help node, creating a parent child node. This is usual for the users and provides for a logical flow to the conversation.

Finally, we tested the chatbot using the Ask Watson chat interface by Watson Conversation. It can be seen from figure 3 that we have responses and relevant intents and entity tags present and response are well structured and have logical flow. There were some words which were wrongly tagged intents, we manually trained Watson, by manually tagging this words.

Slack is a cloud-based team collaboration service. Slack offers various tools and services.[4] It has a large user base. Slack provides messaging and communicating applications. The chatbot was deployed on a Slack application. We cre-

ated a workspace in the Slack and a Slack application. We used the IBM Cloud interface to integrate Slack with Watson. Once we have given Client ID, tokens from Slack to IBM Cloud, we get a request URI and response URL. This URI are entered into slack interface. To test the integration between Slack and IBM Cloud, we authenticate both of them. Now, we test the Watson Conversation chatbot we created on slack interface. This can be seen in figure 4.

5. ETHICAL ISSUES

When designing a new technology we need to have some boundaries. The AI technology exceeds beyond these boundaries, but they need to be monitored. Tech giants believe that now is the right time to talk about the nearly boundless landscape of artificial intelligence [5]. In many ways, this is just as much a new frontier for ethics and risk assessment as it is for emerging technology. Some of the ethical issues in our project are:

5.1 Unemployment

The hierarchy of labor is concerned primarily when it comes to automation of system[5]. It may seem that our project, autonomous ordering pizza is taking away a lot of customer service jobs. But the reality is autonomous artificial intelligence system actually helps in improving the quality of human jobs. Our project gives rise to the development of globalized lifestyle, rather than making people do the routine job everyday. A customer service employee may not remember all the available pizza toppings to explain it to the customer. In such cases, AI is very useful. It can even skip choices based on the availability of pizza types and toppings.

5.2 Misbehaving and interacting rudely with customers

An AI system cannot understand the feelings and emotions of a person. So, it doesn't know how to treat customers based on their emotions. Rather the AI system will act according to its rule base. This issue could lead to bad service rating from customer side. For instance, when a customer is angry about an issue, and if the AI said something crude about it, then the conversation between them will be rough. It may happen that the place ends up loosing those customers.

5.3 Privacy for customers

After placing pizza order, the customers should not see advertisements for coupons on similar pizza types. This situation will happen only when the AI system stored the customers details and sold it to advertising agencies. Also, sometimes it may happen that the AI stores the card details of the customers and by mistake it reaches to other entities who tend to misuse their information. So, improving on the privacy features would be great so that any sensitive information could be prevented on being misused/

5.4 Misinterpreting orders

The chances are high that a AI may not understand all slangs of a language and error is certain. For instance, if the customer said, "I want sausage topping for my pizza" and the AI interpreted it as, "I want sauce for my pizza". This is a big flaw in AI ordering system as they work based on thresholds and they may not ask again if it falls under a particular threshold. Also, sometimes it may be possible that people who speak different language which the AI doesn't understand then in this case it may be hard to fulfill any orders.

6. CURRENT STATUS AND FUTURE WORK

In the current implementation, we have created the intents, entities and dialogs for the project from the pizza dataset obtained. We have also integrated the chatbot with Slack API. The intents, entities and dialogs have been uploaded and traversed by Watson and they are listed. We also have tested these CSV files and they are giving us correct outputs as we have shown in figure 3.

The future work for now is to develop an accessible ZeroUI - People nowadays do not want to open a separate application for ordering a pizza. We will go on and further analyze what can be done more and added in this conversation system. Also, we are planning to implement our Watson Conversation instance on Facebook Messanger and Whatsapp.

7. REFERENCES

[1] Chatbot and watson https://developer.ibm.com/courses/all/chatbots-watson-lets-talk-national-parks/.

[2] Data from pizza ordering systems https://github.com/ibm/watson-conversation-slots-intro/blob/master/data/watson-pizzeria.json.

[3] Ibm bluemix documentation https://console.bluemix.net/docs/services/conversation+/dialog-overview.html.

[4] Slack api documentation https://api.slack.com/getting-started.

[5] J. Bossmann. https://www.weforum.org/agenda/2016/10/top-10-ethical-issues-in-artificial-intelligence/.

[6] A. Hathibelagal. Create intelligent chatbots on android with ibm watson https://code.tutsplus.com/tutorials/create-chatbots-on-android-with-ibm-watson–cms-29387.

[7] F. Radlinski and N. Craswell. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, pages 117–126, New York, NY, USA, 2017. ACM.

[8] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3506–3510, New York, NY, USA, 2017. ACM.

[9] J. Young. Making customer service easy with watson conversation https://www.ibm.com/blogs/watson/2017/02/staples-making-customer-service-easy-watson-conversation/.

APPENDIX

.1 Appendix A - Individual Contributions

.1.1 Contribution made by Akash:

- Came up with the design layout for the project

- Created dataset for entities in the Conversation Slot
- Wrote about dataset description and ethical issues in paper
- Wrote about dataset description and ethical issues in presentation
- Wrote the introduction part in the paper
- Setup a slack workspace and created an application

.1.2 Contribution made by Dipesh:

- Came up with the design layout for the project
- Created dataset for intents in the Conversation Slot
- Wrote about intents, entities and dialogs in Watson conversation in paper
- Wrote about intents, entities and dialogs in Watson conversation in presentation
- Wrote the abstract part in the paper
- Integrate Watson Conversation with Slack API

.1.3 Contribution made by Suresh:

- Came up with the design layout for the project
- Created dataset for dialog in the Conversation Slot
- Wrote the implementation part and future work in paper
- Wrote the implementation part and future work in presentation
- Wrote the current status and future work in the paper
- Deployed and tested the Slack API for ordering pizza

.2 Appendix B - User Specification

We have created a Watson Conversation instance which is capable of interacting with customers, who are interested in ordering pizza. For now this instance can be accessed from Bluemix Dashboard. It can also be accessed using Slack API for a more interactive session. All the user has to do is type in his responses to the questions asked by Watson.