MenuMate: A Cognitive Approach to Navigating Meals on Campus

Dave Harmon and Jack Cardwell

November 14, 2016

1 Overview

Numerous college students wonder "What can I have to eat?", but oftentimes the information that they seek isn't widely available. Specifically at Dartmouth, we have all encountered the situation where one shows up to FoCo, only to find no satisfactory meals. Dartmouth does publish their weekly menus online, but the web application is difficult to use and hard to access from mobile devices. Our project addresses this situation by utilizing Watson's natural language capabilities to query and process a database composed of the Dartmouth Dining Service Menu from a smartphone. IBM Watson allows us to develop a deep conversation tree so that all that the user has to do is ask "What's for dinner at Ma Thayer on Monday?". Our application hopes to help students find what they are hungry for on campus.

2 Why use COGS?

The main advantage that cognitive systems bring to this solution than the current implementation is the intuitive user interface. The current web application that DDS has in place is difficult to use and is basically non-functional from a mobile device. College students are constantly on their phones, so it makes a lot of sense to target this device to help them locate food on campus. Natural Language Processing saves the user time and frustration by allowing the user to ask a question in their native tongue. In other words, the cognitive system caters to the user's input, instead of making the user conform to the system's inputs. By using a combination of a basic user interface and natural language processing, this application intuitively and efficiently helps students locate on-campus food options.

3 Implementation Details

Our target platform for this cognitive implementation is an iPhone application. Our full application includes a front end application written in Swift, a Cloudant database of meal options for Class of 1953 Commons ("Foco"), a Bluemix application service, and a Bluemix Conversation Service. The cognitive component of this project is done through the Bluemix Conversation Service.

The first step in the implementation was obtaining and parsing a menu that Dartmouth Dining Services provided. The document itself was in PDF format, which initially made it difficult to handle. Given the short scope of time to complete the assignment, we opted to create the database by hand, instead of utilizing a PDF parser and extending it to itemize the menu. This step of the project involved writing roughly eighty documents in a Cloudant service. In total, these documents contained one-thousand lines of data, organized by "day", "meal", and "station".

After setting up the database via IBM Bluemix, the next step was to create a Conversation Service instance in our Bluemix Portal. Upon initialization, we filled in the intents, entities, and dialogue nodes. For the initial roll out of our app, we allowed three intents: breakfast, lunch, and dinner. Then, we created entities for "day" and "station". We setup three dialogue nodes, where each contained a reference to both of the entities, and a unique reference to one of the intents. This was a implementation choice that limits our application to queries that specify a meal time, FoCo Station, and a day. While this may limit the query for the roll out, we decided to save future query formats for future updates.

The final and most time-intensive step of the implementation process was to develop the iOS app in Swift 3, using Xcode 7. Using the Watson Swift SDK, the first step in the process was to link the Conversation Service and Cloudant Database to the controller. We used Pods and Cartage to integrate the Bluemix packages into our workspace. We used three IBM packages in our code: ConversationV1, RestKit, and SwiftCloudant. After linking to these packages, we created a storyboard scene for our single view and then created outlets to the input text bar and the output text view. We spent the majority of our time on this project learning to handle the response from the Cloudant server.

The controller is responsible for taking input from a text bar in the view. Upon receiving a TextFieldDidEndEditing() signal from the view, the controller sends the user query to the Conversation Service instance in Bluemix. The server responds with three tags, if the query was valid. Next, the controller uses these keywords to filter the database and search for the relevant document that contains the menu for that meal on a specific day at a certain station. Finally, the controller takes the database's response and presents it on an updated view to the user.

4   Results

Given the limited time to complete this assignment, we believe that our application is a useful and efficient solution to the problem we hoped to address. Our implementation successfully informs users of their dining options at a given FoCo station on a certain day. In our tests, using our application takes significantly less time than DDS' web application. While our application does successfully complete its mission, it is limited by a narrow conversation dialogue tree and a limited set of our training data: menus. We did not meet all of the specifications that we were hoping to accomplish from our proposal, but we believe that this application is easily scalable to meet this specifications. This application has room for many improvements.

5   Improvements

If given more time to work on this project, we would like to extend the Conversation dialogue tree to infer days from system data. This would allow the user to query without specifying the current day. Perhaps the biggest upgrade to our system would be extending the corpus of menus to include all days of the week at Class of 1953 Commons, as well as other DDS locations. At the time of this project, DDS only emailed us four menus, where each menu is for a specific day. With these two changes, we think that the basic functionality of our app would be improved.

   To extend the basic functionality of our application, we would like to eventually link our app to ChefWatson to handle specific recipe requests. For instance, we would like to be able to tell a user how to make a certain meal at FoCo, if another DDS location doesn't contain the same meal pre-prepared. This could be accomplished by extending our dialogue tree and then integrating the ChefWatson API into our controller. Other improvements could include Speech-to-Text queries from Siri and including an automatic database updater that would pull menus from a DDS server and populate the application's corpus for a given week.

6   Projection

This application may take a little bit of time to catch on, as it may take time to automate the database process and extend the functionality to all DDS

locations. However, given a few months, we believe that this app could be widely successful among Dartmouth students. Most likely, this app would not create revenue from subscriptions, as few Dartmouth students would pay for an app of this nature. However, if this app were to scale out to other colleges across the country, some advertisers may be willing to place small banners at the base of our application. Thus, in a few years, this app could bring in revenue. In summary, we believe that this app will be successful, provided that we make some necessary modifications and improvements.

7   Acknowledgments

We would like to thank Dartmouth Dining Services for granting us access to some menus from their database. Additionally, we would like to thank IBM for granting us free access to Bluemix. Without these two contributions, this project would not have been feasible.