

Julia McAnallen
Ling 575 – SDS
HW: Basic SDS implementation
February 19, 2019

Q5. Write a short (~1 page) summary of your findings.

How limited is the system-initiative dialog manager you implemented in Q1 and Q2?

The system initiative dialog manager in Q1 and Q2 is limited, though it captures the information necessary, even if the process can be tedious for the user. The system is quite brittle, in that users cannot give information out of order or fill multiple slots in one utterance, e.g. “small pepperoni” would still elicit a request for size. The modifications for Q2 increase functionality somewhat, though only in very high-level ways – allowing the user to cancel the order mid-order, restart the order at the end and have Alexa repeat the last thing she said.

How does moving to a mixed-initiative model address these limitations?

Changing the skill to be more mixed initiative required some modifications in how slots are filled. It requires allowing slots to be filled in a different order and with different types of utterances than were allowed in the Q1 and Q2 FST versions. I mostly handled this by adding additional user inputs to the skill build which could fill multiple slots at the same time, e.g. {pizza size} {pizza type}. Admittedly, my mixed initiative system is still rather limited, since I was not able to incorporate user modifications, e.g. if they wanted to change their crust type.

An additional way I handled user-initiated requests was to create different intents in the Alexa build. If a user asks for a reorder, they will be guided through a different set of questions than those seeking a new order. The same is true for those asking about order status. These latter dialogues were mostly hard-coded with information, since we did not access an external database to lookup customer information or anything like this.

How much effort is there involved in implementing a full mixed-initiative system?

In a way, Alexa is built to handle these more mixed-initiative dialogues, where slots can be filled in different orders, so in some ways this step required less modification to the code than adding universals for Q2. However, I did not make these dialogues fully flexible for the user, which would have required considerably more effort.

Discuss what you tried and/or what did not work as you were implementing these systems, as well as any other insight you may have gained.

In terms of implementation, Q1 was mostly straightforward once I got a handle of the way that the Javascript code handled the intents via the JSON file from the Alexa build (though my learning curve was quite steep in getting started). For Q2 it was quite straightforward to enable

order cancelation, which is actually already built in to Alexa's Hello World model. Adding the start-over universal required disabling auto delegation to the Alexa skill, but other than that, was not too challenging to implement. For this universal, note that I set it up so that the user can only start over when prompted to confirm at the end of the order, not mid-order. Unfortunately, the repeat universal was much more difficult to implement. There is a built in Alexa universal called "AMAZON.RepeatIntent" but I could not figure out how to write a handler that would delegate to this intent when the user says "repeat" or similar. Example code online was quite complex. I was surprised that this was not an easier universal to add. Therefore, instead of relying on this universal, I used validation slots to repeat the question when the user says "repeat" or a related phrase.

Overall, I was surprised at how difficult this assignment was. I think it might be the hardest assignment I've had in the CLMS program! Part of the challenge was in finding clear explanations for how the different pieces of the Alexa skill fit together, and also not finding sample code for how to write handlers for different situations. One thing that surprised me (though perhaps it shouldn't have) is how inflexible skills are by default. I did enjoy the exercise and hope to work on more of them in the future.