Jacob Coles
For LT2216

Lab 2 Report: Basic Dialogue Management

There are many potential improvements to be made in the project noted in this document, some of which have been implemented, and many of which have not. The potential and realised improvements have been noted.

The app doesn't use RASA intents for all the commands; it relies on the grammar object in the React app itself to be able to recognise the intent. We could implement RASA intents across all the actions, and perhaps it would be useful to write a function/module to simplify doing this. As a first step, I have added true and false intents (corresponding to yes/no questions), to RASA. More work is needed to make all the transitions based on intents. We might also wish to implement entities, so as to be able to identify names of individuals.

If something unexpected is said (which doesn't correspond with any of the intents), the system would originally choose the intent with the highest confidence, even if it didn't make sense. I decided to add a check of the confidence level on the intents. If the intent has a confidence of greater than 70% it will choose that intent, otherwise it will ask the user again.

The speech recognition is not particularly good, and often requires the user to repeat things multiple times, and may have some issues with some accents (it was more accurate when I faked an american accent). We could implement some other speech recognition API, for example Google's cloud-based API. A limitation of this is that instead of being able to use pre-built tools for React (using the Web Speech API), we might have to write much more code and another backend service. american accent

The app did not have particularly good error handling; there were many ways for the app to crash. For a few of the transitions I have improved this so that instead of stalling or crashing, it speaks feedback to the user and returns to an appropriate state (instead of going back to the beginning of the state chart). In future we could have richer feedback and better error messages, perhaps also with visual feedback.

If the app hasn't been run in a while, there is often a network error. This caused unexpected crashing, and a reload is required. I'm not fully sure what caused this, perhaps it's to do with Heroku automatically shutting down the service. In future we could have an initial state which ensures the service is running, and if not, will start up the service.

Many of the changes to be made are a combination of improved ease of use, while others would bring improved functionality. We could include timeouts to ensure the system doesn't keep running indefinitely, and more detailed visual feedback so that the user doesn't get confused by the behaviour of the system. For example if an API call is taking a while, the user might want to know that. Some improvements have been made, but there are many ways in which the system could be improved yet.