

Lab II: VG report

Anna Katarina Page

February 2021

The first shortcoming that I identified in the app is that after finishing the cycle of creating a meeting or going to the welcome messages for the states for setting a timer or creating a to-do item, the app would just stay there with no way to go back to the beginning to do another action. What I wanted was for the app to go back to the start at completion. In order to implement this, I changed a previous version of my app from having the menu for the transition states in a separate file from the `.dmAppointments` file, to having all of the different states and the transition menu within the `.dmAppointments` file. This means that after confirming the appointment that is being made, or after giving a welcome message for the timer or to-do item, the app simply goes back to the `init` state at the top and the user can begin the process again.

The second short-coming is that there previously was no way to cancel and go back to the beginning half-way through. To try to fix this I added some quitting related terms to the grammar and added the following condition in the `on: RECOGNISED` sections of the appointment scheduler:

```
(1)  cond:  (context) => "cancel" in (grammar[context.recResult] || {}),
      actions:  assign((context) => { return { cancel:  grammar[context.recResult].cancel
      } } ),
      target:  "init" ,
```

This means that at any step of the appointment scheduling process, the user can say one of “quit”, “cancel” or “nevermind” and the app will quit and go back to the initial state. Admittedly, this is not entirely ideal. Ideally, the app would ask for confirmation and in the event that the user does not want to quit, would need to return to the state where the quitting was activated from having retained the information it had received up to that point. I only implemented this simple version in the appointment scheduling chain, but the same could easily be applied within longer versions of the setting a timer or adding a to-do item chains if those were extended.

Another improvement I have made is to extend the grammar for the appointment scheduling to incorporate more options and more intuitive phrasings. Specifically I added additional times and days and made it so that the user can either say e.g. ‘Tuesday’ or ‘On Tuesday’ without failing to recognise it if the user omits the ‘on’. I also extended the times that a meeting could be scheduled for. To do this I had to choose what time of day e.g. ‘at six’ refer to, as in whether they refer to 6:00 or to 18:00. I figured that most meetings people would want to schedule would be during or close to ‘normal business

hours', so I made the defaults for 8, 9, 10, and 11 refer to the morning hours and the defaults for 12, 1, 2, 3, 4, 5, 6, and 7 refer to the afternoon hours. To refer to the other version of the time (e.g. 3am rather than 3pm) the user has to specify "am" or "pm" when indicating the time of the meeting.

An additional limitation of the app is that presently if the user selects the options for setting a timer or for adding a new to-do item, the app provides a welcome message and then immediately returns to the start rather than allowing them to actually follow through with the activity. This could be corrected by extending the state machine and storing the user's responses, though I have not implemented this here as it is out of the scope of the present task.

A final shortcoming that I have not attempted to correct here is that aside from the transitions between states, the app still relies on a simple grammar to recognise names, days and times. This restricts what you can do to what is in the grammar. For something like adding a to-do item, where it would be nearly impossible to predict what the user might want to add, this would be highly impractical and a better method would be to link this to something like rasa as well where a larger range of phrasings and entities could be recognised. I have attempted to extend the grammar somewhat to provide users with more options, but it is still not ideal. Even for the simple appointment scheduling action, the app can only recognise days of the week, so it could not even schedule an appointment for a specific date next month for example. This could be added to the grammar, but it would involve adding 356 lines to the grammar, and it seems that there is probably a better way to do this using a system like rasa.