Report: How the dialog system could be optimized

Problem 1 (partly implemented):

The grammar needs to be optimized. It seemed that when the whole dialog uses only one grammar variable, it performed badly. For example, when asking if the appointment will last a whole day or not, and when confirming at the final step, the user needs to say yes or no. The user's utterance could not be recognized when these two questions share the same answers, i.e. yes and no. So I have separated the grammars into several parts, i.e. each question will have its own single grammar variable. This is implemented already in the project.

Furthermore, the grammar should be more flexible and broad, which allows the user to speak in a more natural way. For example, when referring to the "day" of the appointment, the user may only say "Saturday" instead of "on Saturday", or "this Friday", "next Monday", "tomorrow", "the day after tomorrow", etc. The appointment can happen at any time of a day, not only those time points in the current grammar. Also the user may say it in different ways, for example, "fifteen thirty", "five thirty PM", "half past five in the afternoon/evening" which means "15:30".

It would be nice if the system can guide the user to choose among people that have the same first names. For example, when there are two "David" inside the database, the system can ask "Which David do you mean? David Svensson or David Hansson?"

Problem 2 (implemented):

After each question, if the user's utterance is not recognized, an error message will be given, for example "Sorry, I don't understand. Please say yes or no." And then the dialog will be transited back to the start state of that question, which will cause that the system utters again the whole question that contains both the confirmation information of the previous question and the question itself, for example, "OK, you are going to meet David. On which day is the meeting?" Actually, "OK, you are going to meet David." is not needed again. The solution to this is to add an extra state, an confirmation-state, before the question-state, so that the two sentences are separated into these two states. When the user's utterance is not recognized, it will directly go back to the question-state, and avoid the confirmation-state. This is implemented already.

Problem 3 (implemented):

A "stop", "start over" and "go back" functions should be implemented throughout the whole dialog. When the user says "stop", the whole dialog will be stopped and transits to the root menu. When the user utters "start over", the whole dialog will be restarted from the initial state (i.e. creating an appointment from the beginning). When the user says "go back", the dialog will go back to the previous question, which allows the user to correct a wrong input. This is implemented.

I have written three transition conditions in each question of the dialog. However, I think that there should be a better and simpler way to implement this, i.e. since the transition conditions for these three are totally the same, (for the go back function, a context.previous_state variable can be created to record the what the previous state is,) these conditions should be shared

during the whole dialog, rather than writing the same code in each question. But I haven't figured out how to implement it.

Problem 4 (implemented):

Regarding the intent of the user, if the user utters something which gets an intent result with a lower confidence score from RASA, then the system should be able ask the user to specify the intent and try to say it again. This is already implemented in this project, by using "cond: (context) => context.intent.intent.confidence < 0.70" The value was set to 0.7 in this project, i.e. when the confidence score is lower than 0.7, the system will ask the user to utter the intent again. In this way, the system is always very sure about the user's intent, and almost no misunderstanding will happen.