for taking Levodopa at the same time as the activity is shown below. The plan indicates that the Monday pill is in the correct location and the only action remaining is to place the Friday pill:

```
(alternativePlanFor state8
((preference beforeActivity 0))
((addPill Levodopa 5 3)))
```

Finally, the user can inquire about the robot's actions. For example, the user can ask why the robot said to "Try removing a Levodopa from Wednesday." This question is parsed into an intermediate representation: `(onDate Levodopa Wednesday)`, which is passed to the Explanation Synthesizer along with the associated *preference*; the user prefers to take the medication before an activity. The following is a trace of the reasoning of the Explanation Synthesizer

```
[(IsA Levodopa pill), 'Given']
[(AtLocation pill cabinet), 'ConceptNet']
...
[(IsA Wednesday weekday), 'ConceptNet']
[(IsA Wednesday day), 'ConceptNet']
...
[(prefers user (before pill activity)),
'Given preference']
[(IsA appt activity), 'Given knowledge']
[(atTime appt '1pm'), 'calendar']
[(onDay appt Wednesday), 'calendar']
[(atTime appt afternoon), 'Rule fired']
```

The justification is that `(onDay pill Wednesday) (beforeTime pill afternoon)`. And the final explanation reads in a series of symbolic triples: `(prefers user (before pill activity)) (IsA user activity) (atTime appt '1pm') (onDay appt Wednesday) (IsA '1pm' afternoon)`.

Future Work

The work we describe here sets the foundation for a whole line of work in designing social robots to adapt to users, adhere to user preferences, and provide explanations. The most immediate next step is to build upon the proof of concept we have demonstrated here by integrating the individual parts and evaluating the system on more complex scenarios.

Once we have a fully integrated system, the critical next step is to incorporate feedback from the user. One of the greatest advantages of taking a knowledge-driven approach is that the entire system is inspectable, which will facilitate integrating the user feedback. The user may provide feedback after the robot provides an explanation to the user. An explanation synthesizer extracts key terms from the feedback, interprets the feedback, and determines which component(s) need adjustment. The explanation synthesizer also *validates* its conclusion by verifying with the user. In this ongoing work, if the explanation synthesizer identifies that the feedback is related to the user preferences, the Preference Reasoner will construct a new case that is used by AToM to update the model of the user. Even a single piece of feedback from the user can be sufficient for AToM to learn the user's preferences because analogical learning, which is used by AToM, is data efficient and capable of learning from only a few examples (**?**; **?**).

Related Work

Consideration of *user preferences* is an important aspect of human-robot interaction, as it allows the robot to modify its behaviors according to its understanding of the user. Hiatt, Harrison, and Trafton (**?**) found that people prefer collaborating with robots that adapt their behaviors in this way. However, most approaches to recognizing users' preferences use statistical techniques like reinforcement learning (**?**) and Markov Decision Processes (**?**) to predict preferences. This means that they require large amounts of training data, are not responsive to user feedback, and are not explainable. That is, once trained, such systems predict preferences based on their built-up statistical models; a user cannot state a preference directly or inspect why the robot predicts a particular preference. By incorporating stated user preferences, and moving toward learning preferences by analogy, we attempt to avoid these pitfalls.

There are many forms of *adaptive assistance* in robotics. One approach is shared autonomy, in which the system infers human intentions and adapts how much assistance in provided in controlling a robot (**?**; **?**). This work is focused on assisting people in physically controlling robots, whereas we are working towards an autonomous robot that provides social assistance.

Other work has looked at adapting a robot's behavior based on user preferences. For example, a recursive neural network was used to learn weights pertaining to user preferences, which influence the plans for a robot (**?**). While the preferences did affect the plans used by a robot, the plans are used to improve the robot's navigation. Thus, the user preferences do not relate to assistance provided to the user.

A model of Theory of Mind (ToM) has been proposed to adapt the assistance provided by a social robot (**?**). A stochastic model is used to infer what action a person could be executing. Based on this estimate, they generate a plan to determine with which action the robot should help. While they use ToM to estimate a user's intent (via a set of possible actions), they do not represent a user's preference for how the task should be completed.

One way to understand complex decision making systems is with *interpretable* or *explainable* parts. Explanations can describe proxy methods (**?**; **?**; **?**), representations (**?**; **?**), or be inherently explanation-producing (**?**). In the context of human-robot interaction, explanations can help to communicate and build trust (**?**), justify the robot's actions (**?**) or motions (**?**), or describe *unreasonable* perceptions (**?**). But most of these explanations are generated *after-the-fact* and cannot be used to improve the completion of tasks moving forward.

We propose to use explanations as *feedback* to augment assistive robots. This has been explored for agents playing games, especially *when* to provide explanations (**?**). This approach builds on Rainbow, a self-adaptive system that can correct itself and reuse the same baseline framework (**?**). To our knowledge, this is the first work to propose a knowledge-driven architecture that could use explanations to *improve* robotic reasoning and inference.

Contributions

In this paper, we motivate a knowledge-driven architecture for adaptive assistance. We demonstrate the functionality of the components of this architecture in a task for socially assistive robots (SARs). In future work, we will expand the architecture to incorporate and process feedback and learn user preferences. This paper opens a new area of research in adaptable and interpretable SARs.

Molestias unde ut totam quaerat eum numquam maxime consequatur aperiam similique fugit, sunt rem tempora architecto dicta minima ipsum magnam adipisci accusamus vero eos, dolorum porro deserunt excepturi asperiores blanditiis, numquam nulla deleniti dicta perspiciatis.Expedita ullam quas quos hic rerum dignissimos maiores, unde numquam sed nihil eligendi asperiores illo, blanditiis cupiditate tempore perferendis suscipit ab tenetur pariatur molestias nisi quaerat consequuntur, laudantium cumque eligendi rem doloremque optio animi quo quae, rem ullam totam nobis sint esse assumenda consectetur ex quasi unde vitae?Aut soluta consequatur aliquid officiis quas dicta eum inventore, porro molestias reprehenderit dolorum perspiciatis fugiat impedit assumenda aspernatur, possimus eum iste nisi amet dolor iusto?