Execution Monitoring

Although the joint plan derived in Section **??** is not conveyed to the user in the demonstration, the actions assigned to the user are assumed to take place in order for the agent's future actions to execute successfully without uncertainty. Our implementation of the PRETCIL framework thus uses this plan for the second purpose of intent recognition to predict how the user will respond to the agent's actions each turn. If the user's action returned from the activity recognition component matches, then we assume that the interaction is going smoothly and execute the agent's next action in the joint plan. If the user's action does not match, then there is a chance that the agent recognized incorrectly and reassesses the completion criteria with the newest observation. This execution monitoring system completes the interaction loop.

3 Challenges for Closed-Loop Interaction

Our implementation of the PRETCIL framework described above was demonstrated at the Twenty-Ninth International Conference on Planning and Scheduling (ICAPS) in July 2019. Approximately fifteen conference attendees watched others interact or directly interacted with the demonstration over the one-hour period that it was on display. Based on the authors' observations of their experiences and feedback, new questions and research challenges emerged involving the integration of the two areas within an interactive domain.

Although we have IRB approval to run interactive sessions with human subjects for experimental purposes, this demonstration was not related to any experiments. We only discuss general observations and experiences to avoid risk of revealing any participant identities.

Brief Explanation of the Demo

The demonstration environment is based on a common toy problem in the planning community called Block Words (also called Blocks World). A table contains stacks of blocks that each have a letter inscribed on them, and the agent is tasked with picking-and-placing blocks until there exists a stack of blocks whose inscribed letters spell a specific goal word when read from top-to-bottom. Actions either pick up a block that is on top of any stack or put down a held block on top of any stack, and an agent can only hold up to one block at a time. Our extension for assitive interaction includes two agents, each able to hold up to one block at a time, that take turns performing actions. Either agent may pass their turn with a no-op action.

The initial block layout and possible words to spell are illustrated in Figure **??**, based on the setup used in comparison experiments with **?**'s (**?**) interactive agent. For the purposes of our demonstration, the user always made the first move after announcing the specific word they wanted to spell. The user and agent took turns until the user was done (whether or not the goal was accomplished). The demonstration was reset between users.

Challenge Topics

Sufficient Information to Interact For the demonstration, two parameter values were set manually when initiating an interactive session: the necessities threshold and the number of turns that the user has for a head-start. The former adjusts the sensitivity of feature selection when generating intermediate goals from the recognized distributions; the threshold $\tau \in [0, 1]$ requires a feature to appear in enough goal criteria that they collectively represent at least τ of the distribution. The latter acts as a delay before the assistive agent begins responding, which allows it to have an observation sequence that is less ambiguous during recognition. Although parameter tuning is a common challenge in many algorithms, especially for machine learning performance, we have identified some impacts for choosing different values.

	h								
	e								
t	r	l	s	m	o	f	w	b	a

father mother master faster later water

Figure 2: The initial block layout (above) and possible goal words to spell (below) in the demonstration.

If the necessities threshold is too low, then more features unique to specific goal criteria are added to the intermediate goal. Although this sounds more robust to accommodate the uncertainty at the beginning of the interaction, the present-day norm of *conjunctive goal conditions* means that there is a greater opportunity for the goal to have contradictions. In our demonstration, at most one block can be placed on top of another. However, lower thresholds allowed words that shared one letter to require both of their preceding letters on top—“mother” and “father” could easily require both the ‘a’ and ‘o’ blocks to be placed on top of the ‘t’ block (in addition to building the stack that spells “ther”) when τ is sufficiently small. The lack of a solution to this goal means that the agent will not be able to find a plan and act that turn, which made it appear less helpful to the user. Likewise, if the necessities threshold is too high, then no features might be found and the intermediate goal is to change nothing—“mother” and “father” may have a combined probability of 0.8 in some cases, but that does not identify any intermediate goal conditions when $\tau = 0.9$. In this case, the *solution of doing nothing has the same consequence as not finding a solution to a goal with contradicting conditions*.

The number of head-start turns can be more drastic. If it is low, such as 0 to begin interacting immediately, then the handful of observations can be very ambiguous such that the assistive agent recognizes a near-uniform distribution over the subset of goal criteria that use those actions at least once in their possible solutions. Even with a reasonable necessities threshold, this distribution can either be spread too thin to find no features for the intermediate goal or be concentrated enough over almost-distinct goal criteria that contradicting unique features are added to the intermediate goal. The latter scenario sometimes selected unique features that did not contradict others, but were generally incorrect so that the agent performed actions that did not make sense to the users. An additional observation would have often pruned those goal criteria from the recognition algorithm, which is why we added the head-start parameter. When it was set too high, though, the user made enough progress that they found the agent's late response less useful.

(Un)Intentional Communication When the autonomous agent in human-robot interactions has its own personal goals, it can communicate its intentions towards these goals to the human via legible planning with low-level motions (**?**) or high-level actions (**?**). However, our assistive agent's personal goal is more abstract: "to help the user with their own goal." So the agent does not have a personal goal until an intermediate one is recognized and computed. **?** (**?**) account for communicating the agent's newfound goals during the negotiation step, but their assistive agent pipelines the interaction process so that no further recognition is performed after negotiating its goals. We assumed that the *cognitive load of frequent negotiations would not be ideal* as PRETCIL loops indefinitely.

However, our implementation's planner assumes what the user will do, which is sometimes reordered or extraneous enough to confuse the user instead. In one instance, the user looked at the debug data to read the assumed actions and mentioned that providing this expectation would have been a useful explanation for the unexpected behavior. Providing explanations for decision making systems (**?**) has been growing in popularity recently, but we need to be careful that these explanations do not constrain the user's freedom to act in accordance to what the machine does (**?**).

Some users at the demonstration already succumbed to such constraints when selecting their own actions to ensure legibility to the recognition algorithm, viewing the demonstration as a puzzle rather than an open-ended interactive experience. Does this *defeat the purpose of closed-loop interaction* if people adjust their own behaviors *to satisfy the algorithms around them rather than act naturally*? Though we mentioned that **?**'s (**?**) assistive agents have more restricted interactions using a library of precomputed plans, this library often contains multiple plans that allow flexibility to the interactive partner (this is the purpose behind their *choice nodes* where the human can take one of several actions). This leads to a research challenge for finding the balance in a hybrid of closed-loop interaction frameworks. If a joint-agent planner finds multiple plans to the intermediate goal, then which plans' action should be used when there are multiple matches to the next observation? That is, when monitoring the execution, which plans are "going according to plan"?

What Information Actually Matters? The most critical ones we identified during the demonstration relate to *using all the available information*. Some plan recognition algorithms already address noisy sensing (**?**) and irrelevant experimental actions while exploring the environment (**?**), but these methods still assume that the observed agent is the only actor in the world. Planning algorithms can address various forms of uncertainty, but we are unaware of any that consider the uncertainty of the goal's validity.

The assistive agent's actions also change the world, and these need to be acknowledged during recognition. We simply encoded them as observations because recognition as planning handles missing observations by assuming actions that can connect two consecutive observations were performed. However, the potential for poorly chosen intermediate goals threw off the recognition algorithms due to the agent's sometimes incorrect actions and state modifications. Accounting for them as noise or experimentation might work pragmatically, but they are conceptually different because these actions have purpose and influence the interactive partner's later actions toward their goal. Furthermore, for long-term interactive systems that cannot be reset like our demonstration, how should observation sequences be modified over time for relevancy to the current interaction only?

When our demonstration's assistive agent computes a joint plan with its intermediate goal, it currently uses the same search heuristics; the state space and set of actions change to address turn-taking. However, the above issues with useless goals present two things to consider. First, when the goal contains contradicting conditions, is there a way to find a plan that satisfies some largest possible subset of conditions so that the agent can do something? Second, if the agent is unable to find a plan, should the agent perform a default action or replan for some default goal? We programmed our assistive agent to perform a no-op, but this led to a few failed demonstrations where the user needed a block that the agent was holding before it failed to find a plan. Even if these users intended to confuse the assistive agent with noisy observations, a default goal of not holding any blocks would at least allow the user to complete the task on their own.

4 Conclusion

For less structured interactions between users and intelligent systems, closed-loop interaction that perceives what people do and decides how to appropriately respond is necessary. We introduced the PRETCIL framework as a cognitive architecture for such interaction and implemented it as an assistive agent for a game. A recent demonstration revealed new research challenges for artificial intelligence methods involved in closed-loop interaction. Future research will explore these challenges, but we encourage the artificial intelligence for human-robot interaction community to consider their own solutions and identify additional problems. *Many novel situations from interactive experiences and integrated frameworks take these traditional algorithms out of their original context*, and we need to address them as we continue to study and create intelligent interactive systems.

Acknowledgements

Richard G. Freedman thanks Steven Levine for his frequently engaging discussions regarding artificial intelligence for closed-loop interaction. The authors also thank the anonymous reviewers for their feedback. A final thank you goes to the ICAPS 2019 demonstration participants.

Maxime dolorem repellat quidem atque ducimus libero eaque impedit quo dicta, aliquam sed exercitationem quidem temporibus nihil eos, at ea iure magni