Understanding Task-Oriented Dialogs through Abductive Reasoning

Tolga Könik
Computer Science Department
Stanford University, Stanford, CA 94305

Abstract

We present a computational account for reasoning about dialogs that occur during collaborative task performance. Our system builds upon knowledge structures of an existing cognitive architecture and extends those structures to reason from the perspective of other agents. Our approach abductive inference to support reasoning with incomplete knowledge since dialogs usually do not contain sufficient information to deductively infer the mental state of the participants of the dialog. Our abductive inference engine is guided by coherence heuristic and domain constraints. Our framework uses both task knowledge and domain independent dialog knowledge. We demonstrate the feasibility of our approach with an empirical study using symbolic dialogs in a medical emergency response domain. We conclude that our system behaves robustly to missing information and general dialog level knowledge improves the performance of the system.

Introduction

Social exchange through complex dialogs is one of the main distinctive characteristics of human intelligence. The ability to use dialogs to communicate our beliefs and our capacity for building consensus on common goals is key to our existence as a collective society. In this paper, we focus on an important class of dialogs, communicative acts executed for the purpose of collaborating on performing tasks. We present a computational account on the problem of *understanding task-oriented dialogs*, which involves extracting information about the mental state of the agents that participate in communicative acts while performing tasks collaboratively.

The ability to make sense out of dialogs is not only an important requirement of engaging in dialogs, but we frequently utilize this ability acting as passive observants. For instance, when we watch a movie, we associate ourselves with its characters, reason from their perspective, predict their metal state, and try to make sense out of their actions. In this paper, we focus on this observation aspect of dialogs.

Understanding dialogs requires several high-level cognitive functions to work in unison. Clearly, a major such function is low level processing of natural language, which is not our focus in this paper. Instead, we assume that the raw dialog data is already converted to a symbolic representation with known semantics. We want to use that semantic data to extract higher-level inferences about the mental state of the participants of the dialog.

In this paper, we propose a cognitive framework for dialog understanding. We take the theoretical position that dialog understanding is an abductive process that is robust against missing information. Reasoning during dialog understanding involves introducing default assumptions based on coherence heuristics. To support this position, we implemented this framework and experimentally demonstrate that it derives expected inferences. We demonstrate the feasibility of our proposed system on hand-generated symbolic dialogs from a medical emergency response domain. We show that our system can make new plausible inferences about the mental state of the participants of a dialog and it is robust to missing information. We demonstrate that the performance of the system degrades gracefully with increase of missing information. We also hypothesize that task-oriented dialog understanding requires two levels of knowledge; knowledge about the task domain and domain independent knowledge about conducting dialogs. Our implementation uses these two levels of knowledge and our experiments show that general dialog level knowledge improve performance of dialog understanding.

We begin our discussion with the description a motivating task domain and the problems we expect our system to solve in this domain. After that, we present our approach to dialog understanding, including long-term and short-term representational structures and a method for abductive reasoning that uses them. We then describe the systems operation in the task domain and show experimental results. Next, we discuss connections to related work and conclude with final remarks and directions for further research.

A Problem Domain

Before we present our system, we describe a medical emergency response domain to make the kind of problems we expect our system to solve concrete. The domain involves a novice field medic discussing treatment options with a remote computer assistant. The field medic is conducting physical actions to treat the patient, while it is communicating the situation with the computer aid. In this domain both the medic and the computer aid can have information about the patient that the other agent does not have. The computer aid does not observe the patient's situation like the medic does but it may have other information like medical history of the patient or readings from a diagnostic device. The goal of dialog is both to share the relevant information that the other party may not know and to form common goals and a common plan for a treatment.

Table 1 shows an example dialog in this domain that we will use throughout the paper. In this example, the medic shares the situation of the patient, (e.g. "there is a bleeding"), and the computer aid makes a proposal about the goals the medic should adopt (.e.g. "stop bleeding by applying a tourniquet.").

This scenario requires that the computer aid is able to understand as well as generate utterances. However, in this paper we are focusing only on the understanding aspect of the problem. We assume that the dialog is generated by two participants and the computational system we build is a separate observer, which interprets the dialog and makes predictions about the mental state of the speakers/listeners. Since our system does not make any commitment on who the observing agent is, our approach is also valid when the observer agent is one of the participants of the dialog.

Table 1: Example dialog between a assistant medic (M) and a computer aid (C).

```
M - We have an injured person.
C - Where is the injury?
M - He is bleeding from the left leg and
torso
C - How bad is the bleeding?
M - The leg is much worse
C - Okay, use a tourniquet to stop the
bleeding.
M - Right, where shall I put it?
C - Just below the joint above the wound.
Keep turning until it stops bleeding.
M - Okay, the bleeding has stopped.
```

The goal of the observer is to interpret the utterances by using its medical domain knowledge and general knowledge about dialogs. For instance, knowing how to stop bleeding requires medical domain knowledge, whereas detecting a shared goal when a "stop bleeding" utterance receives an acknowledgement is an an example use of dialog-level knowledge. To interpret the dialogs correctly, the observer agent should also be able to reason from the perspective of other agents, guessing their beliefs, goals and inferences, and make inferences that cannot be supported deductively based on observed dialog. For instance, from the dialog in Table 1, the observer agent can infer that the computer aid assumes that the field medic has access to a tourniquet. Next, we present our cognitive system that can form plausible new inferences with this kind of dialogs.

Dialog Understanding Framework Overview

Our dialog understanding system processes utterances incrementally. At each cycle, it reads literals that encode an utterance factual short-term memory. Next, it uses abductive inference guided by long-term knowledge (rules) and heuristics and generates new short-term memory short-term knowledge. Next, we will describe the short-term and long-term knowledge structures followed by a description of the inference mechanism.

Knowledge Representation

Since dialog and task-performance are both important cognitive abilities, it is natural to study task-oriented dialogs in the context of a cognitive architecture (Newell, 1990). The framework we describe in this paper utilizes an extension of the knowledge structures of Icarus, which is a theory of cognitive architecture that makes commitments about representation and performance mechanisms that use it. Like Soar (Laird et al., 1987) and ACT-R (Anderson, 2004). A key difference of Icarus is the differentiation of concepts and skills, the hierarchical organization of those structures and associating skills with explicitly defined goals. Our dialog understanding framework uses a knowledge organization similar to Icarus, but extends it in important ways. Our long term goal is to adapt Icarus to this new representation and incorporate dialog understanding capability into Icarus.

Short-term knowledge structures Our system uses short-term knowledge to encode factual observations (utterances) and inferences. Icarus uses grounded literals to represent its beliefs, goals and intentions. Only the beliefs, which are stored in working memory, can be accessed during the reasoning process. Icarus stores the goals and intentions in a separate memory and accesses them only during problem solving and execution. This means that it cannot access those elements to form new inferences. Since reasoning about intentions and goals deliberately is important in dialog understanding, we extended this representation and in our dialog understanding system, we allow working memory elements with belief, goal and intention modalities.

Modelling the reasoning process of other agents is an important cognitive ability. There is some work on reasoning and learning from other agents' behavior in Icarus (Nejati, Langley and Konik, 2006; Konik and Laird, 2006; Li et al., 2009), but most research in this direction, assume behavior traces presented from the perspective of only one agent and therefore these systems do not need to mark the agent under investigation explicitly. Dialog understanding requires facts represented from different agent's perpectives and beliefs of one agent about the beliefs/goals and intentions of another agent. We extended Icarus' basic short-term memory representation to contextualize facts with the agents that believe in them.

In our extended representation an agent that has the working memory structure *(believes agent1 (injured patient1))* believes that *agent1* believes that *patient1* is injured, while having the memory structure *(injured patient1)* still means believing that *patient1* is injured, as in Icarus. Similarly, an agent that knows *(goal agent1 (treated patient1))* will

Table 2: Concepts form new beliefs from other beliefs.

```
(believes ?agent
  (stabilized-injury ?injury))
<-
  (believes ?agent
    (injury-type ?injury open-wound))
  (believes ?agent
    (not (bleeding ?injury)))
```

think that *agent1* has the goal of treating *patient1*. Since we assume that the observer agent is watching the dialog between two agents, it can also have nested beliefs like *(believes agent1 (believes agent2 (injured patient1)))* or *(believes agent1 (goal agent2 (treated patient1)))*.

In general, the beliefs two agents have about each other can be arbitrarily nested (e.g. A believes that B believes that A believes ....). Reasoning systems that address this issue usually establish a common belief once the nesting is sufficiently deep and they assume that the agents have formed a common belief on that subject; When that happens, they know that they have established a common belief, and they know that the other agent knows that they have established a common belief etc. In the simplified model presented in this paper, we assume that the agents trust each other such that if an agent A believes that another agent B believes P, then A believes P unless it has stronger evidence on the contrary. Consequently, two level belief nesting is sufficient for the tasks we consider.

Table 3: Skills decompose intentions into subintentions.

```
(intention ?agent (stabilize-injury
?injury))
:relations
  (goal ?agent (stabilized-injury ?injury))
  (believes ?agent (tourniquet ?t))
  (believes ?agent (available ?t))
  (believes ?agent
    (good-tourniquet-position ?injury
                   ?position))
  (intention ?agent
    (tourniquet-positioned ?t ?position))
  (intention ?agent
    (tourniquet-tightened ?t))
```

Long term Knowledge Structures Just like Icarus, our system represents concepts and skills. The concepts resemble horn clauses and they are used to infer abstract features of the world from more basic features. For instance, the concept in Table 2 states that an open-wound type injury is stabilized when it is not bleeding. Our system uses the concepts to form new beliefs from observed utterances.

The beliefs are ground literals that describe relations among the objects in the environment. We say that the belief is an instance of a concept when the belief matches against the head of a concept. For example, the belief *(stabilized-injury patient1)* is an instance of the *stabilized-injury* concept in Table 2.

Table 4: General dialog knowledge.

(a) Inform action

```
((inform ?s ?l ?BELIEF)
:start   (speaker ?s)
         (listener ?l)
         (believes ?s ?BELIEF)
:actions (*inform ?s ?l ?id (believes ?s
?BELIEF))
         (*acknowledge ?agent2 ?agent1 ?id)
:effects (believes ?l ?BELIEF)
         (believes ?l (believes ?s
?BELIEF)))
```

(b) Request Action

```
((acknowledged-request ?agent1 ?agent2
?GOAL)
:start
  (speaker ?agent1)
  (listener ?agent2)
  (goal ?agent1 ?GOAL)
  (request-id ?id)
:action
  (*request ?agent1 ?agent2 ?id ?GOAL)
  (*acknowledge ?agent2 ?agent1 ?id)
:effects
  (goal ?agent2 ?GOAL))
```

(c) Belief adoption

```
((believes ?agent1 ?c)
:relations
  (trusts ?agent1 ?agent2)
  (believes ?agent1
      (believes ?agent2 ?c)))
```

Skills in our system represent how tasks are decomposed into subtasks. Our system uses a representation similar to Icarus skills, but it uses the skills differently and extends them in various ways. First, our system uses for reasoning about a dialog about task, while Icarus uses skills to perform a task. Next, the skills in our system encode explicit belief, intention, goal modalities and multiple agent perspectives to support the short-term working memory elements described in the previous section. Nevertheless, it is important to keep in mind that this representation is essentially an extension of Icarus representation because we believe that in future we hope to obtain an agent that understands dialogs and performs tasks using the same skill encoding.

Icarus skills describe how to achieve goals by decomposing them into subgoals or by executing actions. For instance, the first skill clause in Table 3 states that the medic can treat a bleeding injury by applying a tourniquet to the right position and then tightening it. The second skill is a primitive one that refers to actions that are executed directly in the environment. The head of a skill represents its goal , a concept, which is achieved after a successful execution of the skill. The start condition is a conjunction of concepts.

In addition to representing task-performance knowledge, our system uses skills also to encode domain independent dialog knowledge to reason about the communicative acts of the participants of the dialog. There are elaborate theoretical accounts of reasoning with dialog acts (Allen and Perrault, 1980). Our current system uses a simplified dialog model to show the feasibility of using this approach in an architectural setting and in conjunction with abductive reasoning.

Table 4 lists some of the dialog skills we use for explaining the dialog in Table 1. The *inform action* describes how beliefs are transferred from one agent to another due to primitive communicative actions like the utterance *(*inform medic computer utterance1 (believes medic (injured person1)))* and the corresponding acknowledgement response *(*acknowledge computer medic utterance1)*. The request action is similar in structure but instead of communicating a belief, it asks for adaption of a goal. The agents use these utterances to build a common performance plan. The third rule says that the agents can adapt each others beliefs, but the adaptation is not guaranteed because this rule may conflict with the inferences the agent forms based on other beliefs. If the agent has stronger reason to believe otherwise, it would not adapt the another agent's belief.

A final long term memory structures our system uses are the constraints. Our current system utilizes an important class of constraints that describe impossible situations. For example, the following constraint states that a patient cannot be at two different locations at the same time and the inference engine should not generates inferences that violate it:

(patient ?p) (location ?p ?L1) (location ?p ?L2)
(\= ?L1 ?L2) –> false

Inference Mechanism

Even though our system uses an extension of Icarus representation, its inference engine, which is based on abduction, is a significant divergence from Icarus' deductive inference engine. The inference mechanism in our dialog understanding system inputs observations of utterances, uses skills and concepts as background knowledge, and forms new inferences as output. The background knowledge it uses can encode both domain dependent task-knowledge and domain independent dialog knowledge. The system consumes utterances incrementally, and after each utterance creates a memory of observed utterances. Next, the system calls the abductive inference engine and generates new conclusions.

Our system receives utterances of a dialog encoded in a symbolic language as input, interprets them using skills and concepts and returns new predicted inferences. In general, abductive inference has a large search space. Our system utilizes the constraints to reduce that space and prioritizes its search using a heuristic that prefers inferences with higher coherence. Finally, even though abductive inference can generate incorrect conclusions, in a dialog setting, those conclusions can be fixed with further dialog and clarification. Since we have not integrated our system into a dialog generate framework yet, we are not demonstrating this hypothesis in this paper.

Our system uses an extension of an abductive inference engine described by Bridewell and Langley (accepted) in detail. Abductive system accepts general horn-clause and ground literals as input. The abduction system builds justifications, instantiated rules, matching beliefs against rules. To construct the justifications, the system iterates over the new beliefs. Each iteration starts with the selection of a belief as focus of attention. Next, the system generates a list of candidate justifications. These are scored based on a measure of local coherence, which indicates how tightly the justifications would connect to the existing explanation. The system applies the most coherent explanation. As the rules are instantiated to form justifications, the instantiated conditions in the justifications are added to the belief memory and selected as focus of attention. This system can make logically incorrect inferences like $(\alpha \wedge \beta- > \gamma), \alpha, \gamma \Rightarrow \beta$, but its coherence heuristic aims to prefer more plausible explanations.

We adapted this inference engine into our system with two changes. First, we needed to permit nesting of belief/goal/intention modalities. We achieved use of modalities by flattening model predicates into regular predicates. For example (believes a1 (believes a2 (P x))) is replaced with (B.B.P a1 a2 x). The expansion of the rules is more complicated but trivial under the assumption of fixed maximum modality nesting. Second, we changed the inference generation of Bridewell and Langley's engine to eliminate inferences that conflict with domain constraints.

Empirical Evaluation

In this section we first demonstrate how our system works an example, next we provide experimental results.

Demonstration on Medical Emergency Response

We tested our system with dialogs from a medical emergency response domain. Here, we use an example dialog to clarify the functioning of the system. We are not aiming to match human data, but we want to show that our system builds a plausible mental state of the communicating agents it observes.

Figure 1 shows the explanation our system builds after observing the dialog in Table 1. Notice that this explanation is a simplified version of the actual explanation, in that, it does not refer to the agents that form the beliefs. Instead, it shows all belief and goals that are the agents have established s common ground.

The system builds this explanation piece by piece after each utterance. For example after the utterance *(*inform*

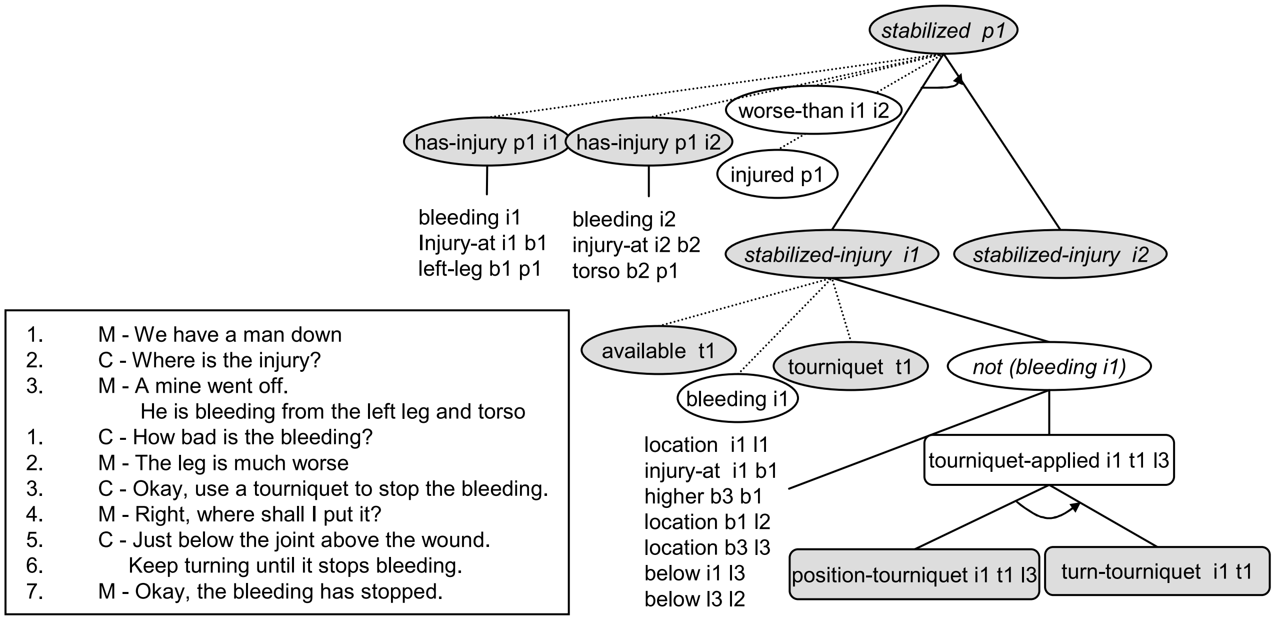

Figure 1: Explanation inferred from a dialog

medic computer utterance1 (believes medic (injured person1)) is received, the system focuses on it, justifies it using the *inform* dialog action in Table 4(a), and predicts the beliefs:

- *(believes medic (injured p1))*
- *(*acknowledge computer medic utterance1 (believes medic (injured p1))*
- *(believes computer-aid (believes medic (injured p1)))*

Focusing on the last one of those beliefs, the system uses the rule 4.c to infer *(believes computer-aid (injured p1))* because the system has no other information that could conflict with that inference.

At this moment, the observer expects that the computer-aid is going to acknowledge the medic's utterance because the observer does not have other information about the situation and this is the only inference it can generated out of the utterance, which is currently in the focus. If the computer aid says that it does not agree, the inference of the observer can be changed based on this new evidence. As new utterances are collected, the inference continues building justifications using skills and concepts. For example, the system infers *(has-injury p1 i1)* using the beliefs *(bleeding i1) (injury-at i1 b1)* and *(left-leg b1 p1)*, justifying this inference deductively using the definition of a *has-injury* concept, but it infers *(stabilize-injury i1)* . The figure shows the deductively inferred beliefs with grey background and the abductively inferred ones with white background.

As the medic makes progress in the treatment, the reasons for starting skills may change. For example, when the bleeding of the injury *i1* stops at the end of dialog, the belief *(bleeding i1)* will not appear in the observation. Consequently, the dialog will proceed toward the treatment of the second injury *i2*. Both of these reasons trigger the system not to infer *(stabilize-injury i1)* in the next utterance state and prefer *(stabilize-injury i2)* instead.

Experimental Evaluation

In previous sections, we proposed a plausible framework for understanding dialogs but we have not provided evidence that it operates as intended. In this section, we present empirical studies that support the viability of our approach. To evaluate our implementation of this framework, we conducted experiments to support the claims we discussed earlier. We compare the accuracy of the predictions of our system to a correct prediction. We will measure the accuracy with commonly used precision and recall metrics but we need to describe how we choose the target prediction.

Methodology Dialog systems are usually difficult to evaluate since they usually require human testers to verify the validity of the inferences. In this paper, we chose an evaluation, which requires very little human expertise and intervention during the experiment. We manually encode an initial dialog, but the problems that our system processes during experimentation are automatically derived from that initial dialog. Our experimentation methodology requires only little human expertise in creation of the original trace and the background knowledge.

We start with a natural language dialog, the meaning of which most people would interpret uniquely. Next, we encode each utterance with a set of literals such that each set contains enough information to infer all intended meaning of the dialog deductively. This is a much richer encoding than a direct translation of the utterances since it includes literals that a person can guess given the whole context of the dialog but cannot deductively infer from the utterances alone. We use this rich dialog encoding as the correct inference, which we will use to measure the success of our system.

Next, we separate the set of correct inferences to a input and test sets. The goal is to measure to what extend our system can retrieve the test set given the input set. We prepared problems in different groups of difficulty. In the most difficult problem set, we aim to produce 90% of the correct inferences given 10% of them as input. We generated 9 difficulty groups varying input-test ratio with 10% increments. In each difficulty group, we generated 15 distinct problems by randomly sampling the correct inference.

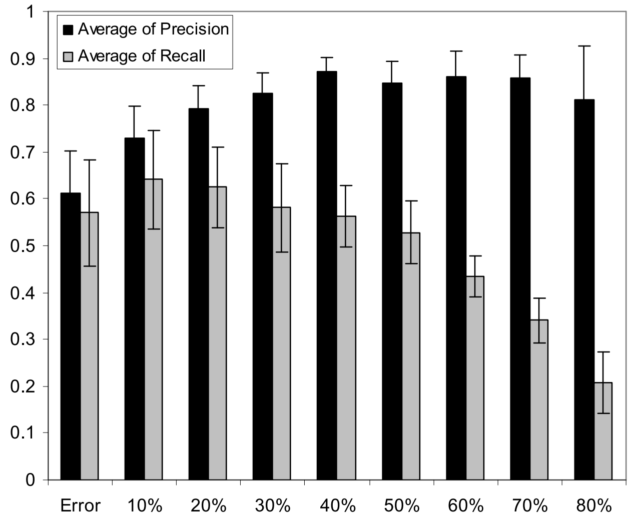

Figure 2: Independent variable: Problem Difficulty. 90% hardest problem. 10% easiest. Dependent variable (y-axis) Average Precision and Recall.

Results We ran ours system on these problems. Figure 2 report the average precision and recall we get for different difficulty groups. We observe that precision for all groups, about 0.8 for the most difficult 5 problem sets where 30%-90% of the correct data is recreated from the remaining data. We also note a surprising finding that precision decrease on simpler data when most of the correct inferences are given in the input. Closer inspection reveals that this happens because true positives are rare in those cases and our system does not make a systematic search once it can already explain the input coherently. However, this is a weakness that we should address with future research. Recall behaves more in an expected manner. It drops with problem difficulty. On the hardest problem, recall drops as low as 0.2, but this is expected since our system acts conservatively when too much information is missing, because instead of making to many default assumptions potentially reducing precision, our system relies prefers to rely on the continuation of the dialog to accumulate more knowledge. These results support our claim that our framework, which assumes incremental abductive reasoning, a coherence metric can make high-precision inferences even when encountered with incomplete information. We also see a graceful performance decline with problem, as we have claimed earlier in the paper.

In our experiments, we create new problem dialogs, by randomly removing facts from each utterance set. We expect that the difficulty of problems increase with the number of removed facts. We run our system on each problem and compare the results against the target set.

Figure 3 shows the difference between precision and recall in runs with the general dialog rules and without. This supports our second claim, which states that the dialog level rules increase performance. The precision and recall of runs with dialog level rules are consistently higher than runs without those rules in all difficulty categories and the difference can go as high as 0.5 for the simpler problem.

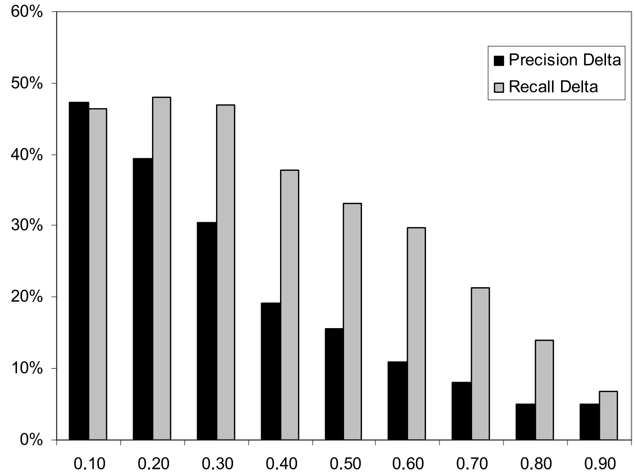

Figure 3: Comparison of with and without dialog cases. Independent variable: Problem Difficulty. 90% hardest problem. 10% easiest. Dependent variable (y-axis) Average Precision and Recall difference between with and without dialog case on the same problem. Positive means higher performance by with dialog rules.

Related Work

There are several early application of abductive inference in task-performance domains. Ng and Mooney (1992) used abduction for plan recognition and they compare alternative explanations using a global coherence, while the abductive mechanism we used in our system (Bridewell and Langley, under review) works incrementally and uses a local coherence metric. Charniak and Goldman (1991) presented a system that also worked on plan recognition but it used probabilistic inference. Josephson and Josephson (1994) presented an abductive inference mechanism, which uses a local evaluation, but their system is not incremental and it is not designed to work on dynamic environments. Abduction is also used for natural language processing. TACITUS system (Hobbs et al., 1993) reads summaries of events using abductive inference.

Austin (1962) provides one of the earliest accounts that treat utterances as actions. Cohen and Perrault (1979) presented the first work that integrates planning with STIPS-like actions (Fikes and Nilsson, 1971) with speech act theory (Austin, 1962; Gordon and Lakoff, 1971; Searle 1975) but their focus was generation of speech acts. Perrault and Allen (1980) applied a similar approach to interpretation of speech acts. Our work uses a simpler speech act theory but integrates it with a cognitive architecture and abductive reasoning. Our system also makes assumptions that treat dialog as collaboration with shared intentions (Levesque et al., 1990) or shared plans (Grosz and Sidner, 1980).

Concluding Remarks

In this paper we presented a system for understanding dialogs about task performance activities. Our system extends the representational structures of Icarus architecture to support reasoning from the perspective of other agents. Our system utilizes an abductive inference engine to be able to make predictions that cannot be deductively supported by the dialog. Our system uses the skills and concepts to represent both domain specific knowledge and domain independent dialog knowledge, both of which are important for dialog understanding. Our experiments support the feasibility of our framework. The precision of our system is high and dialog level rules improve performance.

In this work, we take the first step to add dialog capability to a cognitive architecture. Our future work will extend Icarus skill representation and execution mechanism to add dialog generation capability to the architecture. We also want to extend the current framework with episodic memory, so that abductive inference retrains its results even when the world changes. We also want to adapt a more general dialog theory to handle more complicated interactions between agents like negotiating disagreements. We believe our architectural theory holds promise to incorporate these changes and that a more complete account of task-performance dialogs can be achieved in this direction.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*, 1036–1060.

Austin, J. L. (1962) *How to do things with words*. Harvard University Press, Cambridge, MA.

Bridewell and Langley (under review) A Computational Account of Everyday Abductive Inference.

Charniak, E., & Goldman, R. (1991). A probabilistic model of plan recognition. *Proceedings of the Conference of the American Association for Artificial Intelligence* (pp. 160-165). Anaheim, CA: AAAI Press.

Cohen, P. R. & Perrault, C. R. (1979) Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3), 177-212.

Fikes, R. E & Nilsson, N. J. (1971) Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2, 189-208.

Gordon, D. and Lakoff, G. (1971). Conversational postulates. In CLS-71, pp. 200-213. University of Chicago.

Grosz, B. J. and Sidner, C. L. (1980). Plans for discourse. In Cohen, P. R., Morgan, J., and Pollack, M. E. (eds.), *Intentions in Communication*, pp. 417-444. MIT Press, Cambridge, MA.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., & Martin, P. (1993). Interpretation as abduction. Artificial Intelligence, 63, 69-142.

Josephson, J. R. and S. G. Josephson, Eds. (1994). *Abductive Inference: Computation,Philosophy, Technology*, CambridgeUniversity Press.

Konik, T., & Laird, J. (2006). Learning goal hierarchies from structured observations and expert annotations, Machine Learning 64(1-3), 263-287.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.

Langley, P., & Choi, D. (2006). A unified cognitive architecture for physical agents. *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (pp. 1469–1474). Boston: AAAI Press.

Levesque, H. J., Cohen, P. R., and Nunes, J. H. T. (1990) On Acting Together. In AAAI-90, Boston, MA, pp. 94-99. Morgan Kaufmann.

Li, N. Stracuzzi, D., Cleveland, G., Konik, T., Shapiro, D., Molineaux, M., & Aha, D. (2009). Constructing game agents from video of human behavior. Proceedings of the Fifth Conference on Artificial Intelligence and Interactive Digital Entertainment. Stanford, California.

Nejati, N., Langley, P., & Konik, T. (2006). Learning hierarchical task networks by observation, Proceedings of the Twenty-third International Conference on Machine Learning (pp. 665-672). ACM Press.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Ng, H. T., & Mooney, R. J. (1992). Abductive plan recognition and diagnosis: A comprehensive empirical evaluation. *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning* (pp. 499-508). Cambridge, MA.

Perrault, C. R. and Allen, J. (1980) A plan-based analysis of indirect speech acts. *American Jornal of Computational Linguistics 6*(3-4), 167-182.

Searle, J. R. (1975). Indirect speech acts. In Cole, P. and Morgan, J. L. (eds.), *Speech acts: syntax and semantics Volume 3*, pp. 59-82, Academic Press, New York.