Integrated Interpretation and Generation of Task-Oriented Dialogue

Alfredo Gabaldon[1]
Ben Meadows[2]
Pat Langley[1,2]

[1] Carnegie Mellon Silicon Valley
[2] University of Auckland

IWSDS 2014

The Problem: Conversational Assistance

We want systems that interact with humans in natural language to help them carry out complex tasks. Such a system should:

- *Interpret* users' utterances and other environmental input;
- Infer *common ground* (Clark, 1996) about joint beliefs/goals;
- *Generate* utterances to help users achieve the joint task.

In this talk, we report on two systems for task-oriented dialogue and the architecture that supports them.

Target Domains

We have focused on task-oriented dialogue settings in which:

- Two cooperating agents adopt and pursue a shared goal;
- The agents construct models of each others' mental states;
- One agent (the advisor) has expertise to solve the problem but cannot act directly; and
- Another agent (the novice) lacks expertise but can act when provided with instruction.

However, the architecture is not limited to these settings and should apply to other areas that involve social cognition.

A Medic Assistant

One application focuses on scenarios in which a human medic helps injured teammates with system assistance:

- The medic has limited training but can provide situational information and affect the environment;
- The system has medical expertise, but cannot sense or alter the environment directly; it can only offer instructions;
- The medic and system collaborate to achieve their shared goal of helping the injured person.

The system uses a Web interface similar to a messaging app; we hope to replace with a spoken-language interface in the future.

Sample Dialogue* for Medic Assistant

M: We have a man injured!

A: Ok. What type of injury?

M: He's bleeding.

A: How bad is the bleeding?

M: Pretty bad. I think it is the artery.

A: Ok. Where is the injury?

M: It's on the left leg.

A: Apply pressure on the leg's pressure point.

M: Roger that.

A: Has the bleeding stopped?

M: No. He's still bleeding.

A: Ok. Apply a tourniquet.

M: Where do I put the tourniquet?

A: Just below the joint above the wound.

M: Ok. The bleeding stopped.

A: Good job.

M: medic A: advisor

* We provide the system with the meaning for each of M's utterances.

A System for Meeting Support

A second system supports cyber-physical meetings in which people and devices interact to carry out a joint task.

One scenario is a physical therapy session in which a therapist and a patient:

- Participate in the meeting from remote locations;
- Share the goal of completing a physical therapy task;
- Contribute to a dialogue that includes our system;
- Interact with sensors (e.g., motion detector) and effectors (e.g., a video player).

Users communicate by sending controlled English sentences via a menu-based phone interface (also to be replaced).

Sample Dialogue for Meeting Support

T: John, please do leg lifts.

P: Okay. I will do leg lifts.

TV: [starts playing video tutorial]

MS: [detects movement; sends signal]

Sys: Great start John!

MS: [sending more movement signals]

Sys: You are doing great, John!

P: I can't do any more leg lifts.

T: It's okay, John. That was good.

Sys: John, you did eight repetitions.

P: How many did I do in the previous session?

Sys: In the previous session you did five repetitions.

T: therapist
P: patient
MS: motion sensor
Sys: meeting support system

Observations about Sample Dialogues

These sample dialogues, although simple, raise several issues:

- Behavior is *goal-directed* and involves communicative actions about *joint activity*;
- Participants develop *common ground* for the situation that includes each other's *beliefs* and *goals;*
- Many beliefs and goals are not explicitly stated but rather *inferred* by participating agents; and
- The overall process alternates between drawing inferences and executing goal-directed activities.

We have focused on these issues in our work, but not on other important challenges, such as speech recognition.

System Requirements

These observations suggest that a dialogue system should have the ability to:

- Incrementally interpret both the physical situation and others' goals and beliefs;
- Carry out complex goal-directed activities, including utterance generation, in response to this understanding;
- Utilize both domain and dialogue-level knowledge to support such inference and execution.

We have developed an integrated architecture that responds to these requirements.

Dialogue Architecture

Both systems are based on a novel architecture for task-oriented dialogue that combines:

- Content specific to a particular domain;
- Generic knowledge about dialogues.

The new dialogue architecture integrates:

- Dialogue-level and domain-level knowledge;
- Their use for both understanding and execution.

We will describe in turn the architecture's representations and the mechanisms that operate over them.

Representing Beliefs and Goals

Domain content: reified relations/events in the form of triples:

[inj1, type, injury] [inj1, location, left_leg]

Speech acts: inform, acknowledge, question, accept, etc.

inform(medic, computer, [inj1, location, left_leg])

Dialogue-level predicates:

belief(medic, [inj1, type, injury], ts, te)

goal(medic, [inj1, status, stable], ts, te)

belief(medic, goal(computer, [inj1, status, stable], t1, t2), ts, te)

goal(medic, belief(computer, [inj1, location, torso], t1, t2), ts, te)

belief(computer, belief_if(medic, [inj1, location, torso], t1, t2), ts, te)

belief(computer, belief_wh(medic, [inj1, location], t1, t2), ts, te)

belief(computer,inform(medic,computer,[inj1,location,left_leg]),ts,te)

Representing Domain Knowledge

Our framework assumes that domain-level knowledge includes:

- Conceptual rules that associate situational patterns with predicates; and
- Skills that associate conditions, effects, and subskills with predicates.

Both concepts and skills are organized into hierarchies, with complex structures defined in terms of simpler ones.

Representing Dialogue Knowledge

Our framework assumes three kinds of dialogue knowledge:

- Speech-act rules that link belief/goal patterns with act type;
- Skills that specify domain-independent conditions, effects, subskills (e.g., a skill to communicate a command); and
- A dialogue grammar that states relations among speech acts (e.g., 'question' followed by 'inform' with suitable content).

Together, these provide the background content needed to carry out high-level dialogues about joint activities.

Example of Dialogue Knowledge

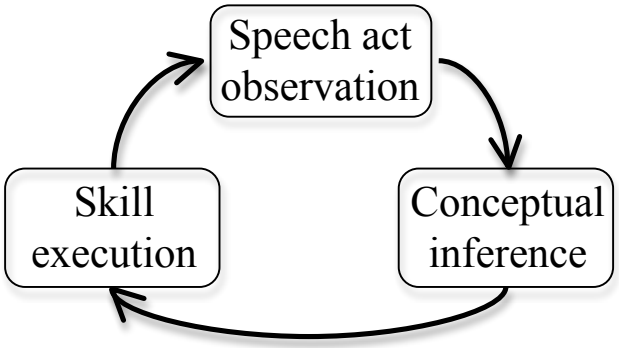
Speech-act rules associate a speech act with a pattern of beliefs and goals, as in:

$$\begin{aligned} \text{inform}(S, L, X) \leftarrow\ & \text{belief}(S, X, T0, T4), \\ & \text{goal}(S, \text{belief}(L, X, T3, T7), T1, T5), \\ & \text{belief}(S, \text{inform}*(S, L, X), T2, T6), \\ & \text{belief}(S, \text{belief}(L, X, T3, T7), T2, T8), \\ & T0 < T1,\ T1 < T2,\ T2 < T3,\ T1 < T4,\ T6 < T5. \end{aligned}$$

This rule refers to the content, X, of the speech act, that occurs with the given pattern of beliefs and goals.

The content is a variable and the pattern of beliefs and goals is domain independent.

Architectural Overview



Our architecture operates in discrete cycles during which it:

- Observes new speech acts, including ones it generates itself;
- Uses inference to update beliefs/goals in working memory;
- Executes hierarchical skills to produce utterances based on this memory state.

At a high level, it operates in a manner similar to production-system architectures like Soar and ACT-R.

Dialogue Interpretation

Our inference module accepts environmental input (speech acts and sensor values) and incrementally:

- Retrieves rule instances connected to working memory elements;
- Uses coherence and other factors to select a rule to apply; and
- Makes default assumptions about agent beliefs/goals as needed.

This abductive process carries out heuristic search for a coherent explanation of observed events.

The resulting inferences form a *situation model* that influence system behavior.

Dialogue Generation

On each architectural cycle, the hierarchical execution module:

- Selects a top-level goal that is currently unsatisfied;
- Finds a skill that should achieve the goal whose conditions match elements in the situation model;
- Selects a path downward through the skill hierarchy that ends in a primitive skill; and
- Executes this primitive skill in the external environment.

In the current setting, execution involves producing speech acts.

The system generates utterances by filling in templates for the selected type of speech act.

Claims about Approach

In our architecture, integration occurs along two dimensions:

- *Knowledge* integration at the domain and dialogue levels;
- *Processing* integration of understanding and execution.

We make two claims about this integrated architecture:

- Dialogue-level knowledge is useful across distinct, domain-specific knowledge bases;
- The architecture's mechanisms operate effectively over both forms of content.

We have tested these claims by running our dialogue systems over sample conversations in different domains.

Evaluation of Dialogue Architecture

We have tested our dialogue architecture on three domains:

- Medic scenario: 30 domain predicates and 10 skills;
- Elder assistance: six domain predicates and 16 skills;
- Emergency calls: 16 domain predicates and 12 skills.

Dialogue knowledge consists of about 60 rules that we held constant across domains.

Domain knowledge is held constant across test sequences of speech acts within each domain.

Test Protocol

In each test, we provide the system with a file that contains the speech acts of the person who needs assistance.

Speech acts are interleaved with the special speech acts *over* and *out*, which simplify turn taking.

In a test run, the system operates in discrete cycles that:

- Read speech acts from the file until the next *over;*
- Add these speech acts to working memory; and
- Invoke the inference and execution modules.

The speech acts from the file, together with speech acts that the system generates, should form a coherent dialogue.

Test Results

Using this testing regimen, the integrated system successfully:

- Produces coherent task-oriented dialogues; and
- Infers the participants' mental states at each stage.

We believe these tests support our claims and suggest that:

- Separating generic dialogue knowledge from domain content supports switching domains with relative ease;
- Integration of inference and execution in the architecture operates successfully over both kinds of knowledge.

The results are encouraging but we should extend the system to handle a broader range of speech acts and dialogues.

Related Research

A number of earlier efforts have addressed similar research issues:

- Dialogue systems:
  - TRIPS (Ferguson & Allen, 1998)
  - Collagen (Rich, Sidner, & Lesh, 2001)
  - WITAS dialogue system (Lemon et al., 2002)
  - RavenClaw (Bohus & Rudnicky, 2009)
- Integration of inference and execution:
  - Many robotic architectures
  - ICARUS (Langley, Choi, & Rogers, 2009)

Our approach incorporates many of their ideas, but incorporates a uniform representation for dialogue and domain knowledge.

Plans for Future Work

In future research, we plan to extend our architecture to address:

- Dialogue processing with more types of speech acts;
- Interpretation and generation of subdialogues;
- Mechanisms for recovering from misunderstandings; and
- Belief revision to recover from faulty assumptions.

These steps will further integrate cognitive processes and increase the architecture's generality.

Summary Remarks

We have developed an architecture for task-oriented dialogue that:

- Cleanly separates domain-level from dialogue-level content;
- Integrates inference for situation understanding with execution for goal achievement;
- Utilizes these mechanisms to process both forms of content.

Our experimental runs with the architecture demonstrate that:

- Dialogue-level content works with different domain content;
- Inference and execution operate over both types of knowledge.

These encouraging results suggest our approach is worth pursuing.

End of Presentation