Back to the Future for Dialogue Research

Philip R. Cohen
Laboratory for Dialogue Research
Faculty of Information Technology
Monash University
philip.cohen@monash.edu

Abstract
This “blue sky” paper argues that future conversational systems that can engage in multiparty, collaborative dialogues will require a more fundamental approach than existing technology. This paper identifies significant limitations of the state of the art, and argues that our returning to the plan-based approach to dialogue will provide a stronger foundation. Finally, I suggest a research strategy that couples neural network-based semantic parsing with plan-based reasoning in order to build a collaborative dialogue manager.

Introduction
Imagine a not-too-distant future in which you have a household robot or conversational assistant that is designed to help your family. Of course, as with any family discussing their daily activities, you have the usual heated conversations, agreements, disagreements, etc. What can a household assistant do to help the family and its members achieve their goals if it cannot even represent the differences much less track their resolution and accommodate them? In this paper, I argue that current technologies will not by themselves enable us to build such collaborative multi-party dialogue systems. Instead, we should revisit a previous approach, namely that of plan-based dialogue systems.

A critical requirement of such a system is that it actually collaborate with its users. People have learned to be helpful at a very young age and are strongly expected to collaborate as part of ordinary social interaction (Warneken and Tomasello, 2006). Collaborative interaction involves agents’ being jointly committed to their partners’ success (Grosz and Sidner, 1990; Cohen and Levesque, 1991). In doing so, a collaborator recognizes its partner’s plan to achieve a joint goal, and then performs actions to facilitate them. In general, people’s plans involve physical (and now digital) acts, as well as speech acts, such as requests, questions, confirmations, etc. When the process of collaboration is applied to communication, people infer the reasons behind their interlocutor’s utterances and attempt to ensure their success by (at least) telling them what they need to know to be successful, and by potentially volunteering to perform actions on their behalf. Such reasoning is apparent when a system responds to the user’s asking “Where is Dunkirk playing tonight?” with “It’s playing at the Roxy theater at 7:30pm, however it is sold out.” Here the literal and truthful answer (shown here in plain font) would be uncooperative if the respondent knew that the theater was sold out. On the other hand, we would want a collaborator to go beyond inferring the user’s plan by attempting to debug it. If the plan is expected to fail, the collaborator may develop and suggest (or execute) an alternative plan to achieve the user’s higher-level goal. To continue the example, a collaborative assistant system might then say “It’s also showing at the Forum theater tomorrow at 8pm, and tickets are available. Would you like me to purchase them?” In order to provide such responses, an assistant needs to infer a plan in which people want to know where an entity is (the location where the movie is showing), in order to go there (the theater), in order to perform a normal activity done on that entity at that location (watch a movie). The assistant checks the plan’s preconditions (watching a movie requires that the person has a ticket), and also the applicability conditions (tickets must be available). If the latter fails, the intention is impossible, so the system must drop it and attempt to find another plan to achieve the higher level goal (of having seen the movie). This collaborative process underlies many conversations.

Overall, except for one-off examples, current assistant systems are not typically engaging in collaborative behavior. In order to build collaborative systems, research is needed on joint action, planning, plan recognition, and reasoning about people’s mental and social states (beliefs, desires, goals, intentions, permissions, obligations, etc.). Plan-based interaction acknowledges that communication is a special case of purposeful behavior (Allen and Perrault, 1980; Cohen and Perrault, 1979). Plan recognition involves observing actions and inferring the (structure of) reasons why those actions were performed, often to enable

1 Please see https://www.youtube.com/watch?v=Z-eU5xZW7cU
the actor to perform still other actions (Allen and Perrault, 1980; Sukthankar et al. 2014).

Regarding multi-party conversation, another requirement is that the system must represent the participants’ different mental states (e.g., beliefs, desires, goals, intentions) and their rational balance (Cohen and Levesque, 1990a; Icard, Pacuit, and Shoham 2010). Such a system should in principle track the family’s agreements and disagreements, in order to respond helpfully. However, we argue that present approaches cannot represent key properties of mental states.

The next section examines the current state of the art and argues that a more fundamental approach is needed.

Limitations of Current Dialogue Technologies

Chatbots. One might imagine that end-to-end trained chatbot technology could be useful for participating in multi-party collaborative dialogue because such systems are built to talk about any topic for which human-human conversational training data is available. However, such systems are not now able to track differences in the participants’ mental states nor track agreements and disagreements in running multi-party conversation. Furthermore, a huge multi-party domain-independent training corpus would be needed.

Task-oriented dialogue systems. A more domain-limited approach of current research and industrial interest is to build so-called “task-oriented” dialogue systems (TODS), whose restricted objective is to get an agent to perform actions, often termed “intents”, such as to book a hotel or restaurant reservation. These systems are designed to obtain required and optional atomic values to fill in argument positions (termed “slots”) in an action schema or “frame” (Bobrow et al. 1977). For example, a TODS would obtain the date, time, and number of people for a restaurant reservation. If an argument is missing, it would ask the user to supply it. Can this type of system as currently conceived support multi-party collaborative dialogue? Again I claim it cannot. To see why not, let us examine TODS in more detail through the Dialogue State Tracking Challenge (DSTC) (Henderson, 2015).

Limitations of Slot-Filling Dialogue Systems

The DSTC, and much research based on it, has defined the term dialog state as [emphasis mine] “loosely denoting a full representation of what the user wants at any point from the dialog system. The dialog state comprises all that is used when the system makes its decision about what to say next.” (Henderson, 2015). The DSTC has collected a number of relatively simple slot-filling dialogue corpora, which have led many groups to build such systems. Slot-filling TODS (called “intent+slots” or I+S systems) designed from the DSTC are limited in at least four ways that prevent expansion to multi-party collaborative dialogues:

1. Restricted meaning representations
2. Restricted dialogue state representation
3. Rigid dialogue initiative
4. Lack of collaboration (which we already discussed)

1. Restricted meaning representations. First, the current approach to building these I+S TOD systems limits the set of meaning representations that the dialogue system can consider by assuming that the user will provide an atomic value to fill a slot. For example, I+S systems can be trained to process simple atomic responses like “7pm” to the question “what time do you want to eat?” However, the systems typically will not accept such reasonable but complex responses as “not before 7pm,” “between 7 and 8 pm,” or “the earliest time available,” which do not supply atomic values but rather state constraints, whose meaning involves shared variables. What’s missing from these systems are true logical forms (LFs) that employ a variety of operators (e.g., and, or, not, all, equals, if-then-else, some, every, before, after, count, superlatives, comparatives, etc.) rather than only a flat attribute=value representation. Many utterances have a scoped compositional representation. For example, “What is the closest parking to the Japanese restaurant nearest to the Empire State building?” will have two superlative expressions, which are scoped one within the other. The meaning of “What are the three best Chinese or Japanese restaurants within walking distance of Madison Square Garden?” will have a superlative, count, and a disjunction. However, complex LFs representing the meanings of the above sentences can now be produced robustly from competent neural network semantic parsers (e.g., Duong et al. 2018; Wang, Berant and Liang 2015).

2. Restricted dialogue state representation. The I+S approach to TODS, as exemplified in the DSTC represents dialogue state in terms of the user’s desires as applied to actions (the “intents”) whose attribute-values are to be obtained. However, this representation of dialogue state is overly restrictive. For example, the I+S approach does not explicitly represent the user’s desire, but rather assumes it to be the content of the system’s so-called “belief state.” In our scenario, for a system to serve a family, there may be different desires that need to be considered so they will need to be made explicit. For example, when asked the slot-filling question “what time do you want to eat?”, a multi-party system should be able to handle the response “whenever Mary wants.”

The concept of “belief state” as a database that encodes a distribution of implicitly desired actions with possible slot values (Young et al., 2013) is itself an overly simple representation that cannot support many of the important characteristics of belief, especially the representation of vague beliefs. For example, I+S and database systems cannot currently represent “John knows Mary’s phone number” because one cannot simply put an expression like phone-number(mary,X) in a database of John’s be-
liefs/knowledge. That essentially says John believes an existential statement, that Mary has a phone number. Likewise, one cannot put in a constant for the phone number, because then the system already knows what John thinks it is. The solution will involve the famous philosophical problem of “quantifying-in” (Kaplan 1968; Kripke 1967), specifically, quantifying a variable into a modal operator, as in:

$$\exists X \text{bel}(john, \text{phone-number}(mary, X))$$

This means (in a possible-worlds semantics) that there is some value for X such that in all possible worlds compatible with John’s beliefs, X is Mary’s phone number. The system does not happen to know what X is, but it is the same value in all worlds compatible with John’s beliefs, so he knows what it is.² Various early plan-based dialogue systems represented and reasoned with quantifying-in via the knowref operator³:

$$\text{knowref}(\text{agent} \ <\text{var}> \ <\text{description}>)$$

meaning the agent knows the value of the variable such that description is true of it (Allen 1979; Cohen and Levesque 1990b; Cohen and Perrault 1979; Perrault and Allen 1980; Sadek, Bretier, and Panaget 1997). This knowref expression appears in the preconditions and effects of informref and wh-question speech acts, which are used during planning. Once dialogue systems have to deal with multi-party interaction, they will need to represent such vague beliefs, for example to decide whom to ask – the person whom the system asks should be someone whom it believes knows the answer.⁴ The system should also be able to acquire such information about someone’s beliefs from dialogue. For example, if the system asks: “what is Mary’s phone number?” it should be able to handle the response “I don’t know but Mary does” and plan to ask Mary. Dialogue state for task-oriented dialogue systems is thus considerably more complex than envisioned by I+S approaches. Extensions to I+S to allow goals from multiple domains (Budzianowski et al. 2018) should consider constraints (such as temporal ones) across those goals, e.g., to have dinner before the movie.

3. Rigid dialogue initiative. The dialogue structure of I+S TODS is overly prescriptive. Essentially, the user makes a request, the system asks for the missing information, the user supplies that information, the system (eventually) confirms the action to be performed, the user agrees or disconfirms, etc. However, real dialogues can have many shifts of initiative in which the parties collaboratively work to accomplish goals. Below is an example that shows the need for complex logical forms and mixed initiative:

1. U: Please book the closest good restaurant to the Orpheum Theater on Monday for four people.
2. S: OK, I recommend Guillaume. What time would you like to eat?
3. U: what’s the earliest time available?
4. S: 6 pm
5. U: too early
6. S: how about 7 pm?
7. U: OK

Here the system has responded to the user’s complex request in (1) with a slot-filling question (2). Rather than answer the question, the user replies with another question (3), a not infrequent occurrence though it violates the typical assumptions of simple slot-filling dialogue systems. Notice that Question (3) starts a subdialogue (3-7) by establishing a constraint on the desired time in (2) (Litman and Allen, 1987). The times specified by the system in (4) and (6) are not times the user wants to eat. Only when the user accepts the system’s proposal in (7) do we learn when the user wants to eat. However, the slot-filling approach assumes that it is the user who fills the slots. System and user are thus collaborating to achieve the user’s goals (Clark and Wilkes-Gibbs 1986; Cohen et al. 1990; Grosz and Sidner 1990; Rich and Sidner 1998). How then can we build multi-party, collaborative dialogue systems?

Back to the Plan-based Model of Dialogue

Over the years, many researchers have advocated a plan-based model of dialogue (Allen and Perrault 1980; Allen et al 1995; Breen et al., 2014; Cohen and Perrault 1979; Galescu et al. 2018; Perrault and Allen 1980; Sadek, Bretier, and Panaget 1997) in which the same planning and plan recognition algorithms are applied to physical, digital, and communicative acts. When applied to physical or digital acts, the system is planning over physical or digital states. When applied to communicative acts, the system plans to alter other agents’ mental states, such as beliefs, goals, and intentions, sometimes to cause them to perform actions.

The above early speech act planning work used a hierarchical variant of STRIPS (Fikes and Nilsson, 1972), and employed forward and backward chaining rules as applied to plan operators that represented physical and communicative acts. To formalize this, Cohen and Levesque (1990a) provided a multimodal logic of mental states and action, analyzing intention in terms of a persistent goal (pgoal) to perform an action⁵. We then showed (Cohen and Levesque 1990b) how to describe various speech acts in the logical language. Sadek, Bretier, and Panaget (1997) then built dialogue systems reasoning with a more restric-

² Note that quantifier’s scope may include multiple modal operators.
³ An example of an inference: knowref(agent, D’(p(D) & q(D))) implies knowref(agent, D’q(D)), but not the converse.
⁴ The same issue arises with “knowing whether P”, which is defined as (bel X P) V (bel X not(P)). A speaker can plan a yes/no question that P to the agent whom it believes knows whether P (Cohen and Perrault 1979; Perrault and Allen, 1980; Sadek, Bretier and Panaget 1997).
⁵ A persistent goal is one the agent is committed to keep until the agent believes it is achieved, impossible, or irrelevant.
tive modal logic, which they deployed in the France Télé-
com network.

To show the feasibility of using the aforementioned logic
to drive a collaborative dialogue system, we recently built
a plan-based dialogue manager (DM) prototype that rea-
sons about actions and mental states in that logic. The
DM: Asks yes/no and wh-questions when the addressee is
believed to know the answers; requests actions when the
system wants the effect and the addressee is believed to be
able to perform it; informs that a proposition is true when
the addressee believes it and does not believe s/he already does; suggests actions that the addressee may
want in order to achieve his/her goals. The DM collabora-
tes by inferring and debugging the user’s plans, as dis-
cussed above. Slot-filling occurs in virtue of reasoning
about what people want, believe, and need to believe in or-
der to perform actions. A DSTD “slot,” which involves the
user’s desire (Henderson 2015), can be expressed by quanti-
fying an action’s arguments into the user’s pgoal that the
action be done (Cohen 2019). For example, the slot para-
phrased by “the day Joe wants to me to reserve XYZ for
him” can be expressed as:

\[
\exists Day\ pgoal[joe, \exists[T,N]
\]
\[
done(sys, reserve([patron\text{-}joe, restaurant\text{-}xyz, day\text{-}Day,
\text{time}\text{-}T, num\text{-}diners\text{-}N]))\]
\]

Notice the Day variable is quantified into the pgoal, which
means there is a Day on which Joe is committed (i.e., has a
pgoal) to there being a Time, and number of diners N such
that the system reserves XYZ restaurant on that Day at that
Time for N diners. The system has thus represented there
being a particular day that Joe wants the system to reserve
XYZ, but the system does not know what day that is. Now
assume the system also has this belief (1):

\[
\text{knorref[Agt, Day}^*pgoal[Agt,}
\]
\[
\exists[\text{Time,Rest}]\ done(sys, reserve([Agt, xyz, Time,N]))\]
\]
i.e., the system has a belief that Agt knows what day s/he
wants the system to reserve. If the system adopts pgoal (2):

\[
\text{knorref[sys, Day}^*pgoal[Agt,}
\]
\[
\exists[\text{Time,Rest}]\ done(sys, reserve([Agt, xyz, Time,N]))\]
\]
it then wants to come to know what Day that is. It can
therefore plan the slot-filling question “what day do you
want to me to reserve XYZ restaurant?” because it believes
formula (1), i.e., that Joe knows the answer. If the agent of
the knorref in (1) were Mary, the system would plan to ask her.

Expanding the Scope of Dialogue Systems

In order to expand today’s limited dialogue systems to
multi-party collaborative ones, I argue that we should re-
visit the foundations of dialogue and build scalable collabor-
ativa dialogue components based on joint action, epis-
temic reasoning, planning and plan recognition. To do so,
I suggest we investigate dialogue systems that are hybrids

of semantic parsing, and planning/reasoning systems, aug-
mented with machine learning of various flavors. We have
found to be effective a process of building a semantic par-
sser using the crowd-sourced “overnight” approach (Duong
et al., 2018; Wang, Berant, and Liang 2015), which maps
crowd-paraphrased utterances onto LFs derived from a
backend API or data/knowledge base. This methodology
involves: 1) Creating a grammar of LFs whose predicates
are chosen from the backend application/data base, 2) us-
ing that grammar to generate a large number of LFs, 3)
generating a “clunky” paraphrase of an LF, and 4) collect-
ing enough crowd-sourced natural paraphrases of those
clunky paraphrases/LFs. A neural network semantic par-
sser trained over such a corpus can handle considerable ut-
terance variability, including the creation of logical forms
both for I+S utterances, and for complex utterances not
supportable by I+S approaches. In the past, we have used
this method to generate a corpus of utterances and logical
forms that supported the semantic parsing/understanding
of the complex utterances discussed previously (Duong
et al., 2018).

Planning and plan recognition are vibrant literatures but
their approaches will need to be extended to reason about
mental states and communication. Current planning and
automated reasoning subsystems will no doubt be formally
incomplete, but of course, current machine-learned I+S
task-oriented DMs are themselves incomplete reasoners.
On the other hand, automated reasoning systems cannot
easily handle the uncertainty for which neural networks
(with sufficient data) excel, but there are a variety of prob-
abilistic plan recognition approaches that could be investi-
gated (e.g., Albrecht, Zukerman, and Nicholson 1998;
Charniak and Goldman 1993; Sukthankar et al. 2014).

In combining these technologies, it is not obvious that
the current dialogue research practice of learning both the
relevant semantic parser and the dialogue policy jointly is
advantageous. Because there is far more variability in natu-
ral language than there is in the goal lifecycle (Galescu
et al. 2018, Johnson et al. 2018), by separating semantic pars-
ing from dialogue, a system can avoid having to relearn
how to converse for each domain. Instead, a dialogue man-
ger that operates at the level of plans and goals as applied
to physical, digital, and communicative acts, can be do-
main independent. We can perhaps acquire the probabilis-
tic information (facts and domain actions) that a plann-
er/plan recognizer operates over by crowd-sourcing and
text mining (Fast et al. 2016; Jiang and Riloff 2018). A
plan-based DM could be trained to play both sides of a col-
laborative conversation by planning and interpreting
speech acts and their propositional content, giving the par-
ties’ differing beliefs, goals and intentions, in a given situa-
tion. In this way, the DM could generate possible response
plans that then could be used to train a dialogue manage-

---

6 This might take longer than overnight (cf. Wang, Berant and Liang 2015).
ment component, similar to the “dialogue self-play” approach of (Shah et al. 2018). Such a system could thus learn how to reason, plan, and converse.

Concluding Remarks

I have argued that to build a multi-party collaborative dialogue manager, we should revisit the foundations of dialogue, and base dialogue systems on joint action, epistemic reasoning, planning and plan recognition. It may be “blue sky” to think we can do so because that would require solutions to long-standing problems. However, I suggest it is time we return to such an approach, as the benefits could be substantial.

References


