On the Utility of Decision-Theoretic Hidden Subdialog

Tim Paek & Eric Horvitz

Microsoft Research
One Microsoft Way
Redmond, WA 98052
{timpaek|horvitz}@microsoft.com

Abstract

A spoken dialog system typically characterizes a domain task with multiple states interconnected by actions or thresholds as transitions between states. As the system attempts to solicit a piece of information from the user, it may have to engage in a *hidden subdialog*, or error handling within a particular state, before transitioning to a new state. Hidden subdialogs generally center on illocutionary repairs such as a request for repetition or confirmation of a heard utterance. We summarize what we believe to be the distinct advantages of representing error handling in a hidden subdialog as decision making under uncertainty. We motivate the discussion with examples drawn from dialog systems built within the Conversational Architectures Project at Microsoft Research.

1. Introduction

A task-oriented spoken dialog system typically considers multiple states represented in a graph, automata, or matrix with actions or thresholds as transitions between states. For any state, the system must solicit a piece of information from the user before transitioning to a new state. Error handling in any particular state has been called a *hidden subdialog*, as the dialog is self-contained and does not occur at the transitional level. Hidden subdialogs generally center on illocutionary repairs relating to the act of communication itself, and can occur anywhere. For example, if a dialog system receives a poor speech recognition result, it may request that the user repeat or confirm the heard utterance. Such requests concern only that particular utterance and constitute what sociolinguists have called a "side sequence" ([13]).

We have been exploring decision-theoretic approaches to managing a hidden subdialog in the Conversational Architectures project. We summarize what we believe to be the distinct advantages of representing error handling in a hidden subdialog as a type of "decision making under uncertainty;" that is, representing in an explicit manner preferences about the costs and benefits of alternate repair actions.

1.1. Applying decision theory

Before outlining the distinct advantages of applying decision theory to managing a hidden subdialog, we begin with a brief primer on decision theory and its application to spoken dialog.

The application of decision theory to any problem requires two components, each of which must be explicitly modeled: uncertainty and utility. With a hidden subdialog, critical uncertainties include the inputs a dialog system receives and its beliefs about their illocutionary and perlocutionary force ([1]). Since speech recognition is the primary mode of input, dialog systems can access credible confidence scores or "perplexity" metrics from a speech engine. These measures can serve as or inform one or more other measures of uncertainty about recognition accuracy or reliability. In decision theory, the level of uncertainty is represented as a probability distribution over states of recognition.

Although most spoken dialog systems do not take for granted the reliability of recognition results, not all systems exploit the degree of uncertainty afforded by confidence metrics. When they do, confidence is typically handled in an ad-hoc fashion ([10]). For example, it is not uncommon for systems to use rules of the form:

- Rule: If the recognized utterance falls below some confidence threshold *n*, and the system has not asked a particular repair question more than *m* times, then ask the repair question.

Such ad-hoc rules and approaches have poorly characterized performance; it is unclear how to determine what constituents, such as *n* and *m*, belong in these rules and how to tune them. While there is no easy solution to this problem, decision theory helps to mitigate the heuristic nature of leveraging measures of recognition uncertainty by setting decisions about what repair action to take within a principled mathematical framework, one that is guided by *expected utility*. Decision-theoretic approaches center on representations and procedures for identifying real-world actions under uncertainty that have maximal expected utility.

The process of performing a decision analysis includes expending effort to construct models that consider (1) feasible actions, (2) uncertainties about the current state of the world, and (3) preferences about taking each action given the truth of different states of the world. Scalar measures of preference that are consistent with the "Axioms of Utility Theory" are referred to as *utilities*. Utility Theory is comprised of compelling assertions about the nature of preferences under uncertainty ([18]).

Over the past five years, we have worked to apply the power of decision theory to the challenge of building and refining dialog repair strategies ([4], [5], [10]). For managing a hidden subdialog, the actions to consider are the plausible repair strategies a dialog system can take, given its uncertainties about its inputs, internal beliefs, and the state of the dialog. For example, the utility of the "outcome" of asking the user to repeat an utterance, given a poor recognition result, is typically relatively high; although, that utility tends to drop with the number of times the system has previously engaged in the same repair within the hidden subdialog.

In order to determine what actions constitute good actions, decisions are made according to the "Principle of Maximum Expected Utility" (MEU), a unifying policy on the choice of best action, following logically from the Axioms of Utility Theory. According to the MEU Principle, the principal agent should select the action $A = a$ that maximizes expected utility, $EU(a|\xi)$. If ξ denotes all background information and H represents all possible states of the world, then we select actions guided by the following optimization:

$$\arg\max_a EU(a \mid \xi) = \arg\max_a \sum_h P(H = h \mid \xi) u(a, h) \quad (1)$$

where $u(a,h)$ expresses the utility of taking action a when the state of the world is h. Note that a cost or loss function can also serve as the utility.

1.2. Building systems

Although the MEU Principle can be implemented in a number of different computational architectures, we have found it valuable to employ an expressive representation called an *influence diagram* ([6],[14]). An influence diagram is a generalization of the more familiar Bayesian network representation, where the arcs and random variables in a Bayesian network are extended with utility and decision nodes ([11]). The influence-diagram representation allows a system builder to explicitly define actions, key uncertainties, and preferences about repair actions in a hidden subdialog, and to capture how these key factors are related. Figure 1 displays a high-level influence diagram capturing some of the more detailed computations in the DeepListener system (see [5] for an extended treatment). The DeepListener research project has explored decision-theoretic models for guiding hidden subdialog in a command-and-control speech recognition setting.

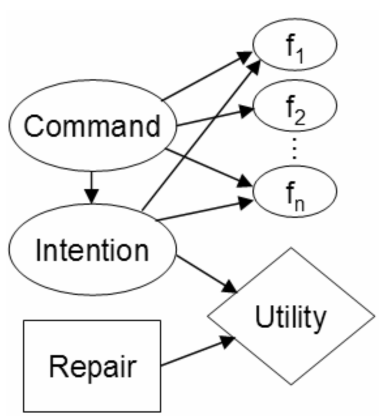

Figure 1: An influence diagram for dialog repair

The *Intention* node in the above influence diagram contains a probability distribution of over possible intentions for the production of a sound input, such as:

- User is issuing a command
- User is signaling the need for more time
- Speech has been overheard

These are explicitly represented as discrete states in the probability distribution for the *Intention* node. Likewise, the *Repair* node contains the set of all possible repair actions, including the following:

- Confirm the most likely command
- Confirm between the top two most likely commands
- Ask for a repetition
- Ask for a rephrase

Since the *Repair* node is a decision node, no probability distribution over these states is required. Finally, the *Utility* node, whose parents are the *Intention* and *Repair* nodes as shown in the arcs, relates the utility of taking any repair action in the context of a given user intention, such as the utility of asking for a repetition if the user is issuing a command, or:

- u(Ask for a repetition, User is issuing a command)

The utilities for the cross product of all repair actions by user intentions are expressed numerically. As far as scale is concerned, the numeric scale of the utilities is mathematically unique up to a positive affine transformation such that if $u(x)$ is the utility, then $a\,u(x) + b$ is equivalent for any constant $a > 0$ and any constant b ([18]).

The influence diagram in Figure 1 graphically depicts the notion that what the user intends is dependent on how likely it is that the system heard any particular command; this is expressed in the arc from *Intention* node to *Command* node, which maintains a probability distribution over all commands. The *Command* node also depends on any number of features f_1 through f_n, such as the confidence score and acoustic events exposed in the speech interface. The features can be anything, including unobserved features that depend on other features. The *Command* node in this influence diagram is effectively acting as a Naïve Bayes classifier over the feature set ([3]). Note that some of the features may also be dependent on the *Intention* node as shown in the arcs.

As acoustic events are observed, the *Command* node infers a distribution over possible commands, and propagates its probabilities to the *Intention* node. Inferring a marginal distribution over all possible intentions is the same as determining $P(H=h|\xi)$ in equation (1) above, which along with the utilities, facilitates the calculation of the expected utilities for all actions or repair strategies, the optimal action being that which has the highest expected utility.

Instead of authoring an intractable list of ad-hoc rules that specify what thresholds must be met before any particular repair strategy can be engaged, the MEU Principle utilized by an influence diagram allows optimal actions to be selected in a mathematically principled manner. Rules are not explicitly stated, but they are implicit in the probability distributions that make up the influence diagram. In fact, if desired, influence diagrams can be converted into decision trees ([14]), which subsequently, can be read off as production rules ([12]), though some behavioral robustness may be lost in the conversion. At run-time, procedures that maximize expected utility effectively compute just the "right" rules in a dynamic manner.

1.3. Making successive decisions

In a hidden subdialog, multiple repair sequences may be required in order to obtain requisite information from the user, in which case, an influence diagram can be used to make successive decisions. When an influence diagram carries over probability distributions from one time slice, such as a conversational turn, into the next time slice, the MEU Principle is applied successively to optimize each local decision. In Figure 2, the likelihood of hearing a particular command is influenced by its own previous likelihood, as shown in the arc from the Command node at time *t-1* to time *t*. Decision making in a temporal influence diagram constitutes a Markov Decision Process (MDP), where the state space is constrained by the structure of the influence diagram and policies are evaluated locally. Since the Intention node is never observed but only inferred, the influence diagram in Figure 2 is considered to be partially observable (POMDP).

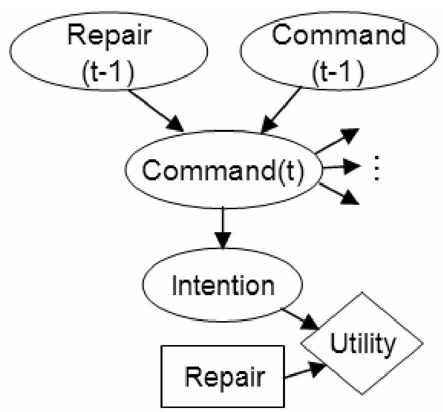

Figure 2: Likelihood of a command depending on its likelihood in the previous turn.

1.4. Reinforcement learning

A related approach using notions of probability and utility in optimizing repair strategies is *reinforcement learning* ([16]). Reinforcement learning also construes decision making in spoken dialog as a MDP. Treating utterances as observations and applying a reward or utility function for reaching various outcome states, the objective is to derive a sequence of actions or policy that maximizes the expected reward in the long run ([7], [9], [15]). Unlike temporal influence diagrams, which optimize actions locally, model based reinforcement learning techniques typically seek to optimize globally over the entire search space, which is a space proportional to size $|H| \times |H| \times |A|$ ([16]), though efforts have been made to approximate and compact that space ([9]). For an influence diagram with only one decision and one utility node, such as Figure 2, the space is proportional to $|H| \times |A|$, where $|H|$ comprises any uncertainty nodes that are parents of the utility node.

1.5. Challenges

While decision-theoretic approaches may be optimizing successive decisions about what repairs actions to take in a hidden subdialog, there is no guarantee that the optimal decision is the most ”natural” one. For both reinforcement learning and influence diagrams, the main challenges revolve around assessing uncertainties and utilities in a spoken dialog setting.

In reinforcement learning, assessing uncertainties revolves around learning transition probabilities between dialog states, where each state comprises summary features for the dialog such as matched utterances and their confidence scores ([15]). Rewards are assigned to dialog outcomes such as termination using experimentally derived measures such as user satisfaction, and the expected cumulative reward or *Q-value* is computed for dialog policies ([15]).

In an influence diagram, assessing uncertainties involves more local modeling of dialog dependencies; instead of summarizing features in a dialog state, the features themselves become uncertainty nodes, some of which may be observable such as confidence scores, and some of which are unobservable such as user intention. These uncertainty nodes are related to one another through conditional probability tables, which can be estimated using either decision analysis techniques for quantifying human expertise ([6]), or from collected data ([2], [3]). Utilities are not assigned to dialog outcomes, but rather to specific dialog actions taken in the context of a generalized dialog state, such as the *Intention* node in Figure 2 or “levels of grounding” ([10]). Currently, utilities can be assessed through decision analysis or user interface tools ([5]), though new techniques are being developed to learn utilities through operant conditioning. Despite the challenges facing any decision-theoretic approach to dialog, efforts have been made in both reinforcement learning and influence diagrams to improve uncertainty and utility assessment

2. Advantages of a Decision-Theoretic Approach

Despite the challenges, decision-theoretic approaches to dialog offer distinct advantages for managing a hidden subdialog. In this section, we describe three advantages: (1) the propagation of uncertainties over time to assist recognition, (2) the ability to leverage key contextual dependencies, such as the acoustic environment, and (3) the consideration of the stakes involved in taking real-world actions. These advantages are discussed with reference to real-time systems and authoring tools that utilize influence diagrams. Although other computational systems may be capable of reproducing all or some of the advantages, we maintain that influence diagrams are ideally suited for handling the specific issues that commonly arise in managing a hidden subdialog.

2.1. Assisting recognition

It goes without saying that in spoken dialog systems, hidden subdialogs often center on repairs caused by poor speech recognition. Most systems utilize the recognizer as a black-box incapable of exploiting dialog features in its internal processing. However, they also unnecessarily leave the recognition results as is without any post-processing. As such, dialog systems treat each utterance of a hidden subdialog as if it was heard for the first time; that is, adjacent utterances, even in a clarification-dialog setting are considered to be probabilistically independent. This runs counter to common sense. For example, if an utterance is heard at a confidence level below a certain threshold, the system may ask for repetition. If the user produces the utterance again—or a different, but conceptually equivalent utterance with the same intention—at the same confidence level, the system may ask for a repetition again, despite the fact that the second utterance came in response to a request for repetition.

With a decision-theoretic approach, the responses to clarifications would not be considered as independent utterances since uncertainties are propagated over time through Markovian dependencies. For example, consider the temporal influence diagram in Figure 2. Here, the likelihood of any command at time *t* depends on the likelihood of that command at *t-1* as well as the repair action taken by the system at *t-1*. Hence, with respect to the previous example, even if the second utterance by the user had the same confidence level as the first, its likelihood could increase because it is conditioned on both its previous likelihood and the repair of asking for a repetition. In particular, the value of this temporal dependency rests on the following condition for the case where a user is following a request to clarify an utterance:

$$p(c_i^t \mid c_i^{t-1}, r^{t-1}) \geq \underset{j \neq i}{\forall}\, p(c_j^t \mid c_i^{t-1}, r^{t-1}) \qquad (2)$$

which states that the likelihood of the same command *c* is at least as great as that of all other commands given that *c* was the most likely command in the previous turn, and the system took a particular repair action *r*. These kinds of conditions, where the likelihood of a command increases or decreases based on events in the previous time slice, appear frequently in a hidden subdialog context because users often repeat the same information to a system given a speech recognition failure ([5]).

Dialog systems without temporal modeling typically resort to ad-hoc rules specifying how often to engage in a repair if confidence scores do not reach a particular threshold. As we discussed earlier, this approach suffers from several disadvantages. Another approach is to try to predict when misrecognitions might occur and adjust dialog strategy accordingly ([8]). This approach is actually complementary to temporal modeling in that the predictive factors of a misrecognition can be incorporated in the influence diagram as key contextual dependencies. For example, the DeepListener system ([5]) incorporates features in the acoustic environment that tend to cause misrecognitions, as we discuss in this next section.

2.2. Leveraging contextual dependencies

Another advantage to applying decision theory to hidden subdialogs is the ability to model and leverage key contextual dependencies. These dependencies can relate to features of the domain task, the acoustic environment, the dialog in progress, and just about anything that allows for better estimates of the uncertainties in the model. Relevant contextual dependencies can be determined by collecting labeled feature data and learning predictive features ([3]), or by simply modeling noted dependencies in the research literature. We will discuss a key contextual dependency which we have found in our experience with the DeepListener project to be useful in managing a hidden subdialog: namely, consideration of the type of background acoustic environment.

As researchers have long noted, the reliability of the speech recognizer is often the bottleneck in dialog success and user satisfaction ([17]). In attempting to extract the most amount of information from each speech recognition result, we found that classifying the type of background acoustic environment that speech was being produced in and incorporating that into the influence diagram allowed for more appropriate repairs. Basic types of background acoustic environments include the following:

- No noise detected
- Constant static or white noise detected
- Voices in the background

The types included depend on the granularity of distinction needed for the application domain. Inferring the type of background acoustic environment in real-time allows the dialog system to anticipate problems that may occur in recognition. For example, before DeepListener considers the command a user may have uttered, it first considers whether the sound produced was intended for the system or for someone else. If the most likely background acoustic environment is "Voices in the background," then the likelihood of "Speech has been overheard" increases. The system can then use this likelihood in a repair by suggesting that the user move to a more quiet area where there are less people and a better chance for a more accurate recognition.

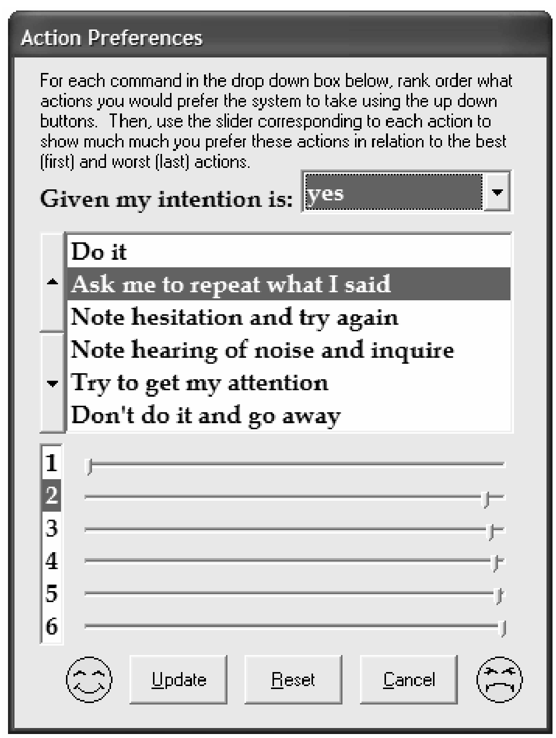

Figure 3: Utility assessment tool in DeepListener for setting the stakes of repair actions.

2.3. Considering the stakes

Every action has consequences, and in a hidden subdialog, depending on the stakes involved in those consequences, less uncertainty may be required for a repair. For example, if a dialog system is walking a user through a help document on reinstalling the operating system, the consequences of a false recognition to the question, “Would you like to view the next email message?” is quite different from that of “Would you like a clean install?” The consequences in the latter are almost irreversible and of high cost, whereas making a mistake in the former is correctable and of low cost. Figure 3 displays one of several pages of the utility assessment tool in DeepListener for specifying the cost of various repair actions. The tool displays pages for considering outcomes in the context of particular user intentions for a given dialog state. For example, if the prompt for the dialog state is, “Would you like to view the next email message?” the user may want the action to be executed without engaging in repair sequences. In that case, the user specifies that for the intention of “Yes” as a response to the prompt, the best action to take is to “Do it.” All other actions, including repairs such as “Ask me for a repeat what I said” is very undesirable for the user, as expressed by the pointers in the slider bars are set to the right. By rank ordering actions and specifying relative preferences between actions using the slider bars, designers can automatically construct a utility function for the underlying influence diagram.

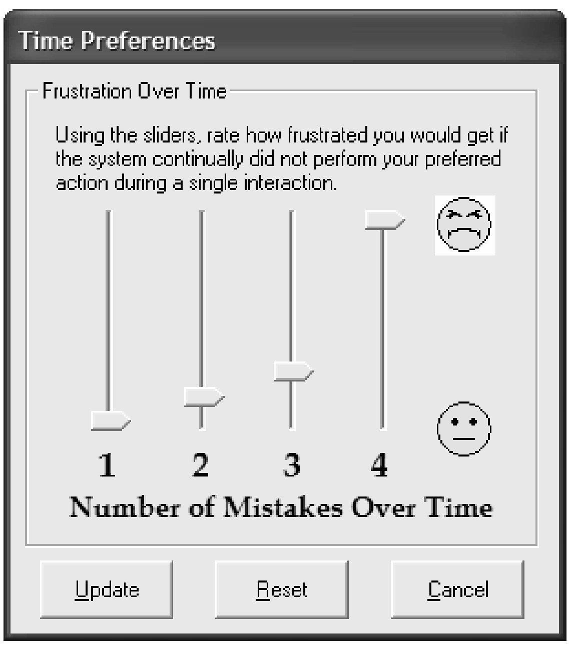

Figure 4: Assessment tool for specifying how utilities may change over time.

In an influence diagram, the stakes are represented in the utility node. So if the utility or cost of taking the wrong action is considerably great, the system is more likely to ask for confirmations before engaging in that action for that state. Rather than assigning a reward function to the entire dialog outcome, as is the case for reinforcement learning, the utilities for each state of a dialog may be changed if the stakes of making a mistake in one part of the dialog exceed that of other states.

In addition to the stakes involved in taking real-world actions, even the stakes of engaging in repairs change over time in a hidden subdialog. For example, asking for a repetition costs very little the first time, but by the second or third time, the cost seems to grow nonlinearly as users become more frustrated. With an influence diagram, changing utilities can be specified by having a measure of time, such as the number of turns, be a parent of the utility node, or by specifying a transformation function for the utilities. In the DeepListener system, we created an interface that allowed users to specify the shape of the transformation function, as shown in Figure 4. Here, the user specifies that while the first three repairs for this particular state do not bother the user all that much, after the third repair however, frustration intensifies considerably. DeepListener maps the shape of this user preference into a nonlinear transformation function for the utilities.

2.4. Real-time processing

Putting all three advantages together, decision-theoretic approaches to hidden subdialog endow systems with the capability to fuse together in real-time all available evidence to infer the likelihoods of different intentions. The methods update probabilities as a dialog evolves over the interplay of turns. Such an inferred probability distribution over different intentions is combined with the values of different outcomes under uncertainty, to identify the action with the highest expected utility.

Figures 5 and 6 (drawn from [5]), display inferences, over a dialog session, of the likelihoods of intentions and expected utilities of alternate repair actions, respectively. Figure 5 shows a trace of probabilities over the course of an interactive session with DeepListener, given recognitions of a user’s response to a recent offer of an automated email scheduling service [5] that the system is attempting to confirm. The inferred intention of confirming the service dominates at turn 3 after briefly competing with the likelihood that the system had overheard a conversation at turn 1.

Figure 6 displays the expected utilities of alternate actions, again computed at each turn. The expected utilities are computed based on a consideration of utilities about different outcomes and the updated probabilities as displayed in Figure 5. At each dialog turn, the action with the highest

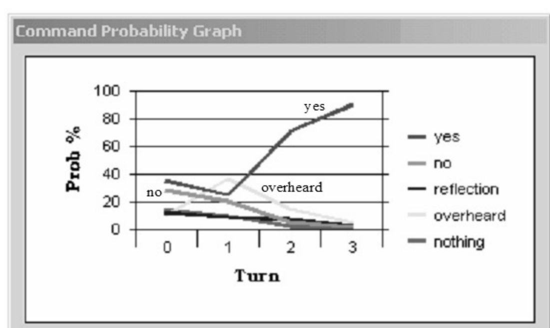

Figure 5: Probabilities of different intentions computed by DeepListener as a dialog progresses.

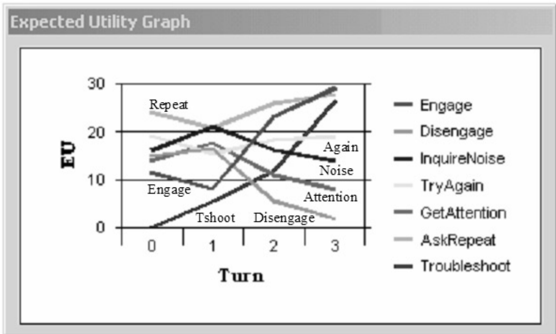

Figure 6: Expected utilities of repair actions as dialog progresses, considering dynamic probabilities displayed in Figure 5.

expected utility is selected. Given the initial acoustical signals that the system has analyzed at the outset of this interactive session, the best initial action to take is to relay to a user who may be present (in feedback encoded graphically as a "thought cloud" positioned over an animated agent's head), that the system has likely heard noise or utterances directed elsewhere. After listening for a new attempt by a user to communicate, the system is still not confident enough to take action, given the inferred uncertainties and the preferences that have been encoded about alternate actions. The best action then is to actively seek clarification. Hence, the user is asked to repeat the utterance. After a repetition of the intention, now said in a different way, the likelihood that the user wishes to confirm a desire for an automated activity rises sufficiently to make performing the service the optimal action. The service is executed and the dialog session ends.

Note that the expected utility of troubleshooting has continued to rise with additional repair actions undertaken during the session. If convergence had not been reached soon, the system would have initiated troubleshooting behaviors such as pausing to reflect with the user about the problem they have been experiencing during the communication, and revealing to the user its current uncertainties and some of its history of beliefs over the interaction. The system also relays to a user how to best communicate intentions by sharing out the phrases it recognizes well for the maximally likelihood intentions.

3. Conclusion and Future Directions

In managing a hidden subdialog, we have outlined three advantages to taking a decision-theoretic approach: (1) the propagation of uncertainties over time to assist recognition, (2) the ability to leverage key contextual dependencies, such as the acoustic environment, and (3) the consideration of the stakes involved in taking real-world actions. In discussing these advantages, we have argued that influence diagrams are ideally suited to handling the specific problems that come up in the context of a hidden subdialog.

As far as future directions, the big challenge in decision-theoretic approaches, as discussed in the primer on decision theory, is to automatically learn the model in a tractable fashion, while at the same time trying to achieve a "naturalistic" interaction with users. To accomplish this, one direction we are exploring is to try reducing the state space for assessment by collapsing multiple utterances into intention *classes*. Another direction is to learn key probabilities and preferences by watching users and to update predefined or "seed" models trained "at the factory" as starting points for a system. Seed models can be refined with usage with an explicit learning phase and/or in the stream of dialog operations; for instance, via user interface affordances that allow feedback for success and failures.

4. References

[1] Austin, J.L. 1976. How *to do Things with Words*. Oxford University Press.

[2] Chickering, D.M., Heckerman, D., & Meek, C. 1997. A Bayesian approach to learning Bayesian networks with local structure. In *UAI-97*: 80-89

[3] Heckerman, D., Geiger, D., & Chickering, D. 1995. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20:197-243.

[4] Horvitz, E. 1999. Principles of mixed-initiative user interfaces, In *CHI-99*: 159-166.

[5] Horvitz, E. & Paek, T. 2001. Harnessing models of users' goals to mediate clarification dialog in spoken language systems. In *User Modeling 2001*: 3-13.

[6] Howard, R. A., & Matheson, J. (eds.) 1983. *The Principles and Applications of Decision Analysis.* Strategic Decisions Group.

[7] Levin, E., Pieraccini, R. & Eckert, W. 1998. Using Markov decision process for learning dialogue strategies. In *ICASSP-98*: 201-204.

[8] Litman, D.J. & Pan, S. 2000. Predicting and adapting to poor speech recognition in a spoken dialogue system. In *AAAI-2000*: 722-728.

[9] Litman, D.J., Kearns, M.S., Singh, S., & Walker, M.A. 2000. Automatic optimization of dialogue management. In *COLING-2000*.

[10] Paek, T. & Horvitz, E. 2000. Conversation as action under uncertainty. *UAI 2000*: 455-464.

[11] Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

[12] Quinlan, J.R., 1987. Generating production rules from decision trees. In *IJCAI-87*: 304-307.

[13] Sacks, H., Schegloff, E., & Jefferson, G. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language* 50: 696-730.

[14] Shachter, R.D. 1986. Evaluating influence diagrams. *Operations Research* 34(6): 871-882.

[15] Singh, S., Litman, D.J., Kearns, M.S., & Walker, M.A. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artificial Intelligence Research* 16:105-133.

[16] Sutton, R.S. & Barto, A.G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

[17] Walker, M.A., Litman, D.J., Kamm, C.A., & Abella, A. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *ACL-97*: 271-280.

[18] Von Neumann, J. & Morgenstern, O. 1947. *Theory of Games and Economic Behavior* (2nd edition). Princeton University Press.