Advances in Cognitive Systems 1 (2012) 93-108 Submitted 11/2011; published 12/2012

The Role of Knowledge and Certainty in Understanding for Dialogue

Susan L. Epstein SUSAN.EPSTEIN@HUNTER.CUNY.EDU
Department of Computer Science, Hunter College and The Graduate School of The City University of New York, New York, NY 10016 USA

Rebecca J. Passonneau BECKY@CS.COLUMBIA.EDU
Center for Computational Learning Systems, Columbia University, New York, NY 10115 USA

Joshua Gordon JOSHUA@CS.COLUMBIA.EDU
Department of Computer Science, Columbia University, New York, NY 10115 USA

Tiziana Ligorio TLIGORIO@HUNTER.CUNY.EDU
Department of Computer Science, Hunter College of The City University of New York, NY 10016 USA

Abstract

As people engage in increasingly complex conversations with computers, the need for generality and flexibility in spoken dialogue systems becomes more apparent. This paper describes how three different spoken dialogue systems for the same task reason with knowledge and certainty as they seek to understand what people want. It advocates systems that exploit partial understanding, that consider credibility, and that are aware both of what they know and of their certainty that it matches the user's intent.

1. Introduction

In *human-machine* dialogue a person (the *user*) and a spoken dialogue system communicate with speech to address a common task. Although the participants seek to *understand* one another (i.e., to perceive each other's intent), human-machine dialogue is often fraught with frustration for the human and uncertainty for the system. Our thesis is that a proficient system requires knowledge about how to agree with its user on exactly which objects are under discussion and what is to be done with them. This paper reports on three spoken dialogue systems with different approaches to these challenges. The most promising one has a clearinghouse for knowledge about what the system knows, hypothesizes, and expects, and an extensive variety of rationales that it learns how to use. This rich cognitive structure supports flexible reasoning and interaction during dialogue.

Understanding benefits from a shared context, knowledge that allows speakers to focus upon the same objects (*targets*), and ways to talk about them. Because a description may not identify a unique object, people consistently assure one another about their understanding, including which targets are in their common ground. This behavior, known as *grounding*, uses vocal gestures (e.g., "uh-huh"), speech, and non-verbal cues to confirm mutual understanding (Clark &

Schaefer, 1989). Confronted by partial or inaccurate information about a target, a spoken dialogue system also may be able to use knowledge to collaborate on the common ground.

To understand its users, a spoken dialogue system must advance well beyond *speech recognition* (translation from audio signal to text string). Dialogue for a complex task may include multiple subtasks and targets of different kinds. Moreover, if a spoken dialogue system cedes to the user some of its control over the path dialogue may take (*mixed initiative*), the system must determine both the targets and how they relate to one another.

A spoken dialogue system *misunderstands* when it misinterprets what it has heard and incorrectly selects a value for the identity of a target or partial information about it. Misunderstanding is common in human-machine dialogue, but difficult to detect and recover from. A *non-understanding* occurs when the system cannot go from an input audio signal to a useful representation of what has been said. A typical commercial system's response to a non-understanding is to ask the user to repeat. Repeated non-understandings can drive the system to end the dialogue.

To guard against misunderstanding, a commercial spoken dialogue system often grounds *explicitly*: it repeats what it believes the user has said and insists upon confirmation before it proceeds. That drive for accuracy (e.g., "I heard you say 17Z946-AQ347R. Is that correct?") annoys many users. Moreover, to prevent the user from saying the unexpected, the system often maintains *system initiative*, that is, determines what is under discussion, and even what may be said.

This paper describes three spoken dialogue systems for the same task: book orders at the Heiskell Braille and Talking Book Library, part of the New York Public Library system. Heiskell's patrons order their books by telephone and receive them by mail. Book requests are by title or author as often as by catalog number. All three systems have ample procedural knowledge about communication and dialogue. They know that speakers should take turns, and that listening provides a continuous audio signal. They know that speech signals can be mapped to phonemes, and that only some sequences of phonemes are meaningful (*known words*). They know too that relevant word sequences provide possible bindings for targets or for *indicators* (e.g., yes, no).

To understand and respond to spoken input, commercial spoken dialogue systems and applications where dialogue is subsidiary often rely on a *pipeline* architecture. A pipeline does best with well-recognized utterances about one class of objects over a limited vocabulary. In contrast, our library task is noteworthy for its confusability (e.g., the same name could be a patron, a title, or an author), its unusually long and complex responses (e.g., average title length of 6.4 words), and its scale: 5,000 patrons plus a vocabulary of 54,448 words drawn from 71,166 books by 28,031 authors. Moreover, book titles are more similar to unrestricted language than to structured data, and more difficult to understand. This task also challenges automated speech recognition with users' diverse native languages, and with transmission noise and background noise in the audio signal.

In response to these challenges, we advocate novel approaches to partial information and certainty for spoken dialogue systems. We recently introduced *partial understanding* to describe situations where the system has a confident interpretation of only some part of the user's intent, one that engenders a question whose answer could support and enhance that interpretation (Gordon, Epstein and Passonneau, 2011). This paper elaborates on how partial understanding can avoid both non-understandings and misunderstandings. For example, given a person's full name as a target, with recognition confidence high for the last name but low for the first, a traditional spoken dialogue system might re-prompt for the full name or signal non-understanding. In contrast, partial understanding could lead the system to engage the user in a subdialogue to ground the last name, and then to elicit a first name consistent with the database and similar to the poorly-recognized first name.

The next section of this paper describes the Olympus/RavenClaw pipeline. Subsequent sections describe three spoken dialogue systems for the library task and how they differ in their use of knowledge and heuristics. Finally, we discuss experimental results and directions for future work.

2. Knowledge Representation and Error in a Pipeline

The traditional approach to spoken dialogue systems is represented in Figure 1 by *Olympus/RavenClaw*, an architecture that has supported the development of more than a dozen spoken dialogue systems (Bohus & Rudnicky, 2009). When it detects voice activity in the incoming signal, such a system's audio manager labels and segments it into *frames* (short intervals of signal analysis output), and judges when the user began and stopped speaking (*endpointing*). It forwards each frame for what it presumes to be a complete utterance to an interaction manager that determines whether the user intends to continue speaking despite a pause. The interaction manager also supervises while text strings from the speech recognizer are mapped to concepts by the subsequent natural language understanding process. If the initial endpointing is not semantically coherent, the interaction manager can override it (e.g., can combine two speech segments into one utterance) (Raux & Eskenazi, 2007).

To transcribe the speech signal into text, automated speech recognition relies on an *acoustic model* that maps speech sounds to phonemes, a *lexicon* that maps phoneme sequences to words, and a *language model* that indicates the probabilities of sequences of *n* words. Automated speech recognition forwards its output to a semantic parser, where a *concept* is an attribute of an object (e.g., a book's title). The parser tries to associate a given automated speech recognition text string with one or more concepts, and it can skip words that cannot be parsed. A confidence annotator then selects at most one best parse that meets a preset confidence threshold.

The pipeline forwards that parse with its confidence to the RavenClaw *dialogue manager* (Bohus & Rudnicky, 2009). Given a confident parse, this module performs one or more optional

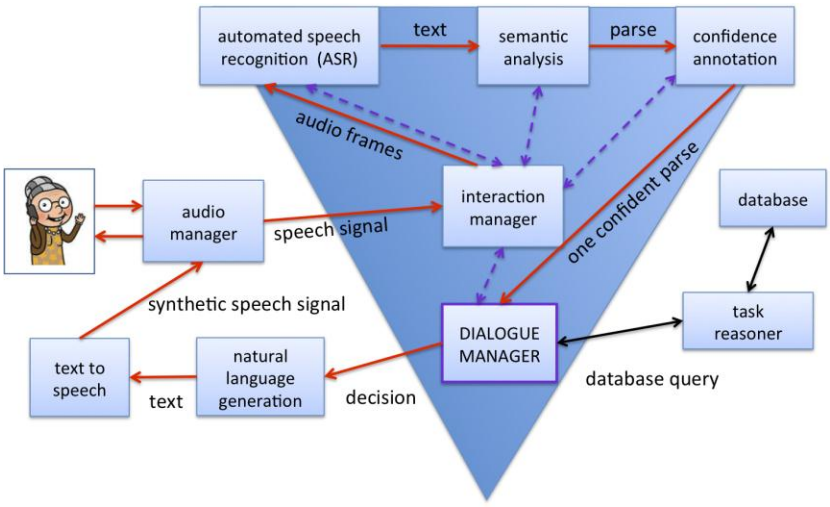

Figure 1: The pipeline (red arrows) in the Olympus/RavenClaw architecture.

queries to backend databases, followed by a command to the natural language generator. For example, if the best parse indicated that the utterance was a book title, the dialogue manager could query the book database for similar titles, and then direct the natural language generator to formulate text to confirm the most similar title (e.g., "Did you want *Jane Eyre*?"). The natural language generator forwards that text to the text-to-speech module, which in turn forwards the speech it generates to the interaction manager for transmission to the audio manager and then to the user. Given an unconfident parse or none at all, the dialogue manager invokes error-handling subdialogues appropriate to a misunderstanding or a non-understanding.

Errors may arise at many points in this pipeline. The audio manager might improperly endpoint the speech signal, and the interaction manager might be unable to correct it. The user's words might not be in the system's lexicon. Disfluencies might disrupt the structure of spoken dialogue (Jurafsky & Martin, 2008). Automated speech recognition might mismatch signal to phoneme, or phoneme sequence to words. Finally, even with perfect speech recognition, a string might have a best (or first) parse that is incorrect, or even no parse at all.

Thus a pipelined spoken dialogue system confronts substantial challenges. Its input is human speech, and therefore inherently disfluent and imprecise. The modality shift from sound to text may introduce segmentation errors that fragment a single request, mistake phonemes, misidentify words, find an incorrect concept or none at all, or otherwise introduce errors and uncertainties about what has been said. We have built three systems that confront such input for the library task. None of them can be quite certain about exactly what the user has said or intended, but they are all expected to respond quickly and intuitively to what the user wants.

3. CheckItOut

CheckItOut is a spoken dialogue system that accepts telephoned orders for up to four books. During a call, each dialogue addresses a sequence of subtasks: identify the user as a known patron, accept book requests, and offer an order summary. The focus of this paper is book requests, the most difficult subtask.

3.1 Knowledge and Representation

At varying points in the pipeline described above, a single book request is represented as a continuous speech signal, a segmented sound signal, a sequence of phonemes, a text string, and some number of parses or possibly none at all. Confidence levels represent the system's certainty that words in the text string and the parses are correct. Despite increasingly accurate automated speech recognition, deployed spoken dialogue systems sometimes contend with word error rates as high as 68% (Raux et al., 2005). The work reported here has a similar word error rate to support research on strategies robust to poor automated speech recognition.

In addition to Heiskell's entire book and (sanitized) patron databases, CheckItOut has extensive declarative and procedural knowledge available to it, many of which are Olympus/RavenClaw modules. This knowledge includes the language and acoustic models used by the PocketSphinx recognizer (Huggins-Daines et al., 2008), and task-independent error-handling mechanisms provided by a domain-dependent dialogue task tree (described below). It also includes the semantic grammar productions in its *Phoenix* parser (Ward & Issar, 1994), and productions derived from MICA dependency parses of book titles (Bangalore et al., 2009; Gordon & Passonneau, 2010). It is important to note, however, that interaction, segmentation, speech recognition, and parsing all

rely on heuristics embedded within the boxes of Figure 1. These heuristics are taken with their default values; our three systems vary in the way they make decisions based on those modules.

As required by RavenClaw, CheckItOut's dialogue manager is a *task tree*, a hierarchy of pre-specified dialogue procedures (e.g., *Login*, in Figure 2). Some leaf nodes (e.g., *Get area code*) issue prompts to determine values for concepts. The task tree is executed depth-first, but preconditions on nodes can redirect it. For example, *Inform lookup error* will return control to *Login* if there is no match on the telephone number. The task tree effectively preprograms dialogue flow. (RavenClaw's support for limited mixed initiative was not used here.) Although there may be multiple ways to endpoint the speech signal, to transform it into a text string, and to parse that text string, ultimately the pipeline produces at most one hypothesis about what the user just said and, therefore, about what the user wants.

3.2 Decision Making

Even among many choices, people can ferret out an object that corresponds to a speaker's intent. A pilot study used noisy automated speech recognition for book titles (e.g., SOONER SHEEP MOST DIE). Three subjects were given 50 such titles along with a plain text file of the library's 71,166 titles and unlimited time offline (Passonneau et al., 2009). The subjects correctly matched 74% of the automated speech recognition strings to a listed title.

CheckItOut matches such noisy automated speech recognition against its database with the Ratcliff/Obershelp similarity metric between two strings (*R/O score)*: the ratio of the number of correct characters to the number of characters (Ratcliff & Metzener, 1988). (The R/O score for *Robert Lowell* and ROLL DWELL, for example, is 0.61.) CheckItOut's best matches for SOONER SHEEP MOST DIE are *Soon She Must Die, Why Someone Had to Die*, and *The Messenger Must Die.* Clearly, the first is the intended book. Indeed, for a book title with a single confident parse from noisy automated speech recognition with a word error rate of about 70%, the search return with the top R/O score is the correct match about 65% of the time (Ligorio et al., 2010). An example of a successful CheckItOut subdialogue appears in Table 1. A skilled human, however, achieves 85.5% accuracy at this task. The next system seeks to exploit that skill.

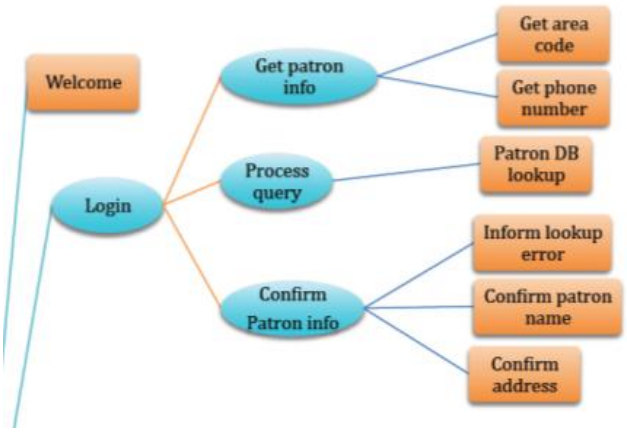

Figure 2: Part of the task tree for CheckItOut.

Table 1: A CheckItOut dialogue fragment, where unconfident automated speech recognition output is delimited by periods.

Automated speech recognition: DIDN'T I AM THE .GOLDEN. ARM
CheckItOut: Is the title *The Man with the Golden Arm*?
Automated speech recognition: NO
CheckItOut: Sorry, I misunderstood. Can you repeat the title please?
Automated speech recognition: .A. .AMBLING. .THE. .GAME. .EDELMAN. STORY
CheckItOut: Is the title *Up and Running: The Jami Goldman Story*?
Automated speech recognition: YES

4. CheckItOut+

The second spoken dialogue system, *CheckItOut+*, has all of CheckItOut's knowledge, plus three models of how people behave when they have only a partial understanding of what the user just said. When CheckItOut+ understands (i.e., has a single confident parse whose slots match a known concept), its dialogue manager behaves exactly the way CheckItOut would, that is, it uses the task tree in Figure 2. Without a single confident parse, however, CheckItOut+ uses three learned models of how people would address the problem to advance the dialogue. These models rely on information from all stages of spoken language understanding. Thus CheckItOut+ either has a single hypothesis about what the user just said, or it evaluates its own internal state to determine how to elicit information that would enable it to produce a single reliable hypothesis.

The models in CheckItOut+ address what in CheckItOut would have been non-understandings. They were derived from an elaborate experiment that replaced CheckItOut's dialogue manager with an *ablated wizard*, a person given the same input and query facility as the spoken dialogue system, and restricted to a limited set of dialogue acts. These models were learned by logistic regression on 163 features, including both system features and wizard actions from data collected at runtime during 913 dialogues (Ligorio, 2011).

The first CheckItOut+ model considers *voice search*, which bypasses parsing and confidence annotation to query with the full text string produced by automated speech recognition. Table 2 is an example of voice search. If CheckItOut+ has at least one automated speech recognition text string but no confident parse, *Model S* (for search) decides whether the most confident text string is good enough to use in voice search. If so, CheckItOut+ searches three times, because the utterance might refer to a book by title, by author, or by catalogue number. The second model, *Model O* (for offer), determines whether to offer the return with the highest R/O score, or to request additional information from the user. Finally, if Model S chose not do voice search, *Model Q* (for question) decides whether to question the user about another way to identify the book or merely to signal non-understanding. Table 3 is a request for further description.

CheckItOut+'s models use 24 features. A *move on* asks the user to request another book and return to this one later, and an *adjacency pair* is the portion of the dialogue from one system prompt to just before the next one, possibly with multiple user utterances. Model S determines whether to search on the recognition text string. Its features describe the current book request (number of adjacency pairs, number of database or title queries), the dialogue (number of questions so far), the context (whether this adjacency pair was initiated by an explicit confirmation, whether a non-understanding had just occurred), the degree to which the utterance was understood (average

Table 2: A CheckItOut+ dialogue fragment, where unconfident automated speech recognition output is delimited by periods. Here voice search harnesses partial understanding.

CheckItOut+: What's the next book?
Automated speech recognition: .FOR. .NONDRIVERS. .SCHULLER. CHAPMAN
CheckItOut+: Is the author Fern Schumer Chapman?
Automated speech recognition: YES
CheckItOut+: Would you like *Motherland beyond the Holocaust: A Mother-Daughter Journey to Reclaim the Past?*
Automated speech recognition: YES

speech recognition word confidence, number of words covered, number of parses for this utterance, whether this is the top grammar slot in the best parse tree), and the number of author queries in this request. Model O decides whether to offer a book based on whether or not this is a new request, the total number of move ons, mean acoustic score, and mean and standard deviation of the returns' R/O scores. Finally, Model Q decides whether to ask the user a question given the number of user utterances in this adjacency pair and features describing the current request: number of author queries for it, number of partial explicit confirmations for it, maximum word-level recognition confidence, overall acoustic model score, whether there was a title slot in the best parse, the number of words not covered by the best parse, and the confidence in the best parse. Models Q and S share three features: whether the current adjacency pair was initiated by an explicit confirmation, average speech recognition confidence, and number of author queries in a request.

5. The FX2 System

The third spoken dialogue system, *FX2*, conducts simple dialogues for a single book request, without patron identification or order summaries. The thesis of FX2 is that dialogue is a dynamic, cognitive process in which participants formulate and evaluate hypotheses to reach a mutual goal. Therefore FX2 represents a dialogue with a dynamic graph (as in Figure 3), rather than a static tree (as in Figure 2). This section describes the representations upon which FX2's components depend and explains its underlying architecture.

5.1 Knowledge and Representation

FX2 has two kinds of knowledge that the earlier systems lacked. Because it views dialogue as a sequence of linked adjacency pairs (rather than a single user signal that now requires a response), FX2 knows how close it believes it is to the goal It also knows how its database sounds, that is, it has *DoubleMetaphone* (phonetic) representations for its authors, titles, and catalogue numbers.

The most crucial representational difference, however, is the way FX2 entertains possibilities for what the user wants. Recall that grounding is the behavior that dialogue participants use to confirm their mutual agreement. In CheckItOut and CheckItOut+, grounding is managed by RavenClaw's error-handling routines. When Helios identifies a sufficiently confident parse, those systems search on it, and then offer the best query result to the user. Both systems retain at most one binding for a target between user turns. In contrast, FX2 refines its interpretation of what it has heard gradually, based on subsequent user utterances. In this sense, FX2 treats understanding in dialogue as a set of related utterances that contribute to a single task, rather than as a sequence of separate reactions to user utterances.

Table 3: CheckItOut+ requests further description.

CheckItOut+: What's the next book?
Automated speech recognition: DUST .TO. DUNE
CheckItOut+: Is the title *Dust to Dust* ?
Automated speech recognition: NO
CheckItOut+: Sorry, I misunderstood. Can you repeat the title please?
Automated speech recognition: GOING THE .OTHER. ARETHA THAT
CheckItOut+: Can you please give me the catalogue number?
Automated speech recognition: THREE ONE NINE NINE EIGHT
CheckItOut+: *Gorbachev: Heretic of the Kremlin*. Got it!

The mechanism that allows this more cognitively-plausible approach is the agreement graph. An *agreement* in FX2 is a subdialogue to bind a target (e.g., a book in an order), and a *hypothesis* is its belief about a possible value for an agreement node. FX2 maintains an *agreement graph*, a dynamic structure that captures hypotheses, their relationships, and their respective *merits* (FX2's confidence in each of them). An example for author appears in Figure 3. Initially, an agreement graph node represents a target or an attribute of a target as its child. Each node also records progress toward its grounding, as described below. Thus, from one user utterance to the next, the graph retains partial understandings (e.g., a patron's perfectly recognized first name). This also allows FX2 to entertain multiple inconsistent hypotheses about the target, and the likelihood that each of them is correct. FX2 entertains multiple hypotheses for agreement nodes, and retains some hypotheses until a node is bound.

FX2 has access to the same interaction management, automatic speech recognition, parser, natural language generation, and text-to-speech modules used by CheckItOut and CheckItOut+. Thus it knows the segmented speech signals, the text string produced by speech recognition from it and the confidence on each word, the parses and the confidence levels associated with them, and the query returns with their R/O scores. The principal difference is that FX2 treats these modules as collaborating heuristics with which to construct reasonable hypotheses for the dialogue manager, rather than as a pipeline that generates at most a single hypothesis. This required a new spoken dialogue system architecture, FORRSooth.

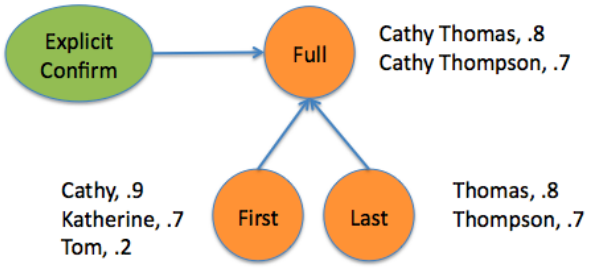

Figure 3: Part of an FX2 agreement graph for author name. First and last are attribute children; hypotheses appear with their respective merits. The node on the left shows a decision to ground.

5.2 The FORRSooth Architecture

FORR (FOr the Right Reasons) is a cognitive architecture for learning and problem solving (Epstein, 1994). A FORR-based decision maker has *Advisors*, resource-bounded procedures that produce any number of *comments*. Each comment supports or opposes an action with a *strength* that reflects that Advisor's underlying rationale. For example, a FORR-based system to match automatically-recognized speech to a book title might have one Advisor whose comment strength reflects how similar they are in length, and another whose comment strength reflects how similar they sound. To make a decision, a FORR-based system *votes*, that is, it solicits comments about possible actions from its Advisors, tabulates a weighted combination of comment strengths that address each action, and identifies the action with highest support.

FORRSooth is a new cognitive architecture for spoken dialogue systems. It provides six *services*: SATISFACTION, INTERPRETATION, GROUNDING, INTERACTION, GENERATION, and DISCOURSE. Each is a FORR-based decision maker with its own set of heuristic dialogue Advisors. SATISFACTION represents dialogue goals and progress toward them, INTERPRETATION formulates hypotheses about what the user has said, and GROUNDING monitors the system's confidence in its interpretation of user utterances. FX2 is an experimental system constructed with FORRSooth.

FORRSooth is intended to learn rapidly to tailor a spoken dialogue system's responses to its task. To select an action, a FORR-based system uses weights learned from labeled training examples; a FORRSooth-based system learns one set of weights for each service and each concept. We used FORR's domain-independent Relative Support Weight Learning algorithm to learn weights for FX2's GROUNDING and INTERPRETATION Advisors. *Relative support* for an action is the normalized difference between the comment strength with which an Advisor supports an action and the strength with which it supports other actions. Relative Support Weight Learning reinforces Advisors' weights in proportion to their comment strengths (Petrovic & Epstein, 2007). Further details on learning in FX2 are available in Epstein et al. (2011).

5.3 Rationales for Understanding in Dialogue

A FORR-based system deliberately incorporates many individual Advisors to capture the varied rationales that people might use in a domain. The Advisors in any service are likely to disagree over how to reason about the partial understandings in the agreement graph; they vote to resolve these disagreements. For example, a hypothesis similar to a book title in both length and sound will be preferred to another that is similar only in length. Weight learning in each service effectively calculates an appropriate balance. While the agreement graph has any unbound node, FX2 runs an interpret-bind-ground loop: INTERPRETATION'S matching Advisors generate hypotheses and attach them to nodes in the agreement graph, and GROUNDING determines whether any hypothesis is strong enough to bind its value to the node without additional confirmation from the user. If FX2 commits a binding, then INTERPRETATION's merging Advisors hypothesize further. Otherwise GROUNDING selects one or more hypotheses to confirm with the user, and when the user replies, the loop begins again.

FX2 has 14 INTERPRETATION Advisors: nine *matching Advisors* that propose hypotheses and five *merging Advisors* that combine them. Four of the matching Advisors are parse-oriented. One proposes the top return as CheckItOut would, one proposes hypotheses for all parses that meet the confidence threshold, another parses both confident and unconfident words, and the fourth proposes any concept consistent with the best parse. Comment strengths for these Advisors are based

on overall and word-level recognition confidence, R/O score, relative position and edit distance between query and return, and the fraction of words in the text string covered by the parse. The other matching Advisors rely on the returns from three voice searches: one each for author, title, and catalogue number. One Advisor proposes the best parse based on DoubleMetaphone similarity, another the best parse based on R/O score. The remaining matching Advisors propose the concepts identified by every parse, with comment strengths proportional to the number of confident words, to DoubleMetaphone similarity, or to R/O score. Based on the matching Advisors' comments, SATISFACTION records hypotheses with their merits on the corresponding agreement nodes. The *merit* of a hypothesis is the extent to which the Advisors' comments support it over alternatives for that node. Merit is the percentile into which the (normalized) strengths of the comments that support a hypothesis fall, relative to others for the same node.

Merging Advisors propose hypotheses for a target based on the degree to which existing hypotheses for its children match a known concept. Two propose hypotheses for full names from hypotheses for first and last names, one for author names and another for patron names. One proposes hypotheses for full titles from hypotheses for the main title and the subtitle; another proposes telephone numbers (for patron identification) from the confidence scores of the area code and seven-digit phone number. The fifth raises or lowers the merits of existing hypotheses based on grounding actions.

GROUNDING determines when and where to append a grounding agreement to the graph. GROUNDING monitors merit values on the agreement graph, relying on learned Advisor weights (Gordon, Epstein, & Passonneau, 2011). GROUNDING proposes node values to the user, and elicits corroboration, further information, or tacit agreement when it determines that a hypothesis requires no confirmation from the user (e.g., "Got it. Next book?"). Its Advisors seek to advance the dialogue with fewer questions and little chance of error. To do so, they reference information such as word confidence, how long the dialogue has progressed, and whether a hypothesis is for a target node or for its child.

GROUNDING's 23 *commit-bindings Advisors* seek to commit to values for nodes in the agreement graph without further user input. Some Advisors advocate binding high-confidence targets, nodes that have just been grounded, child nodes, or a node that has just received a positive outcome. Others prefer easy binding early in a dialogue, or in proportion to confidence, or in proportion to confidence relative to competing values for that node. One prefers nodes at the top of the agreement graph, another at the bottom, and a third comments at random. One prefers to bind in the order in which nodes have been clarified during the dialogue. One Advisor prefers indicator targets, and another lowers the threshold for them. Others address unconfident words, or seek to finish the request. Finally, some oppose binding a value to a node because there has just been a negative response from the user with respect to it, or hypotheses for its children do not appear promising or conflict, or it is a target with conflicting hypotheses, or the node requires clarification first. One Advisor forbids commitment on any low confidence value, and another votes for hypotheses with exceptionally high merits.

If GROUNDING cannot commit to a binding, it considers how to discuss its uncertainty with the user. A *grounding agreement* either elicits confirmation for a particular hypothesis from the user or seeks to disambiguate between competing hypotheses. A grounding agreement is attached to a target or to an attribute node whose hypothesis it addresses. It has an *expectation* for its anticipated user response, and it specifies a grounding action. Grounding actions in FX2 include explicit confirmation (e.g., "Is the title *Dust to Dust*?"), implicit confirmation (e.g., "By John Wooden"), and disambiguation (e.g., "Was that 'Jane' or 'John'?"). In Table 4, for example, GROUNDING

Table 4: Grounding behavior in an FX2 dialogue fragment, with what the user actually said as well as what the system "heard."

FX2: What title would you like?
User: *Family and Friends*
Automated speech recognition: FAMILY .FRIENDS.
FX2: I have two guesses. The first is *Family and Friends.* The second is *Family Happiness.* Is it either of these?
User: The first one
Automated speech recognition: .NEXT. .FIRST.
FX2: Let's try something else. Is the full title *Family and Friends?*
User: Yes
Automated speech recognition: YES

finds two promising but competing hypotheses for the same target, and offers them both to the user. Although the response matches the grounding agreement's expectation, INTERPRETATION fails to understand the response, and FX2 tries a different grounding action.

GROUNDING has 23 *next-grounding-action Advisors* that ask the user for further information. Some of these Advisors ask the user to repeat because there is no clarification hypothesis, there are no satisfactory bindings, or there was just a negative user response. Three more seek clarification on whether a particular grounded child node is correct, whether any grounded child node is correct, or on any agreement not yet discussed with the user. Two seek to confirm a binding based purely on confidence, or on confidence in comparison to competing bindings for that node. One Advisor prefers to ground nodes lower in the tree, another prefers nodes that are higher – note that once bound, nodes higher in the agreement graph advance the dialogue further, but they may also be more difficult to understand than their children. One seeks to bind the root quickly, while others prefer to discuss the full names, full titles, or main (i.e., no subtitle) titles. One Advisor simply selects a random grounding action; another seeks to discuss the unbound target with the highest confidence. One Advisor tries to disambiguate a just-elicited clarification; another requests explicit confirmation. One persists in an attempt to clarify the same agreement. Finally, several Advisors comment negatively on some grounding actions; they dislike repeated requests for clarification or on the same agreement, and oppose repetition of the last clarification action.

6. Empirical Results

The hypothesis investigated here is that partial understanding is an essential component of human reasoning, and that it can support a spoken dialogue system. We test this premise in two ways. CheckItOut+ acknowledges partial understanding and addresses it as people have been observed to do, on one user utterance at a time. FX2 is less reactive; it explicitly represents the degree to which it understands the user, and postulates possible bindings for the targets of interest across a sequence of user utterances. This holistic approach, we argue, is more psychologically plausible.

To investigate this hypothesis we had human subjects call each system. Before each call to CheckItOut or CheckItOut+, the user retrieved a randomly-generated assignment from our website: a patron identity and the author, title, and catalogue number for each of four books. The user was told to request one book by author, one by title, one by catalogue number, and one by any method of her choice. (For a request by author, a database query returns the three books by that

author with the highest circulation.) Each experiment asked 10 subjects to make 50 calls each to the system. In the FX2 experiment, users interacted with the system by microphone rather than telephone, and interactions were subdialogues for a concept, such as author identity.

A spoken dialogue system should respond appropriately, effectively, and in real time to user speech. Spoken dialogue system performance is gauged not only by *success* (task achievement) and *cost* to the user (e.g., elapsed time), but also by user satisfaction, a non-trivial metric where faster and more accurate is not always better (Walker et al., 1997). All differences reported below are significant at $p < 0.05$ under a one-tailed *t*-test.

CheckItOut serves as the baseline here. When CheckItOut produces a single confident parse for a title or an author, its dialogue manager searches for it in the database with the parsed text string, and offers the user the return with the top R/O score, as in Table 1. In 562 dialogues between CheckItOut and 10 users, this approach was accurate on 65% of all book requests, and another 6% of the time the correct match was elsewhere in the top 10 returns (Ligorio et al., 2010).

In 505 dialogues, again with 10 users, CheckItOut+ improved task success. Throughput rose — the number of ordered books increased from 3.22 with CheckItOut to 4.00 per call, while the elapsed time per ordered book decreased from 65.57 to 56.01 seconds. Costs rose too — the system spoke more, and the user had to speak more often. Total elapsed time per call rose from 210.93 to 223.96 seconds, while the elapsed time per correct book decreased from 87.89 to 82.95. CheckItOut+ identified more books correctly on every call (2.70 instead of 2.40), but it also got more wrong, which forced the user to correct it more often.

To gauge user satisfaction, each subject completed the same questionnaire about her experience with CheckItOut or CheckItOut+ three times in the course of her 50 calls. Although answers were consistent (Cronbach's $\alpha = .97$), there was only one significant difference: CheckItOut+ users more often indicated that they had to pay close attention while using that system. Users of CheckItOut+ had to correct the system more often (2.4 times per call, but 1.7 for CheckItOut). There was no statistically significant user preference, however, for one system over the other, even though CheckItOut+ identified more books correctly and processed individual book requests faster. CheckItOut+ also asked questions when it had only a partial understanding.

FX2, however, virtually eliminated misunderstandings. Its INTERPRETATION service produced relatively reliable hypotheses for patron names; their quality degraded gracefully as automated speech recognition performance declined (Gordon et al., 2011). When FX2 could bind a target, it was always correct. Its learned weights led it to prefer to ground by disambiguation (52%) and repetition (32%), with occasional recourse to confirmation (15%) and other strategies (1%). While FX2 is less comparable with the other systems (its task differs and there were no user surveys), its real-time performance is an empirical demonstration of the power a system can derive from partial understanding.

7. Discussion

The differences in performance among these systems are in large measure attributable to the role that understanding, or partial understanding, plays in their decisions. CheckItOut and CheckItOut+ behave identically, unless they fail to produce a single confident parse. When CheckItOut does not understand, it discards what it has heard and computed, and simply asks the user to repeat, or eventually hangs up. CheckItOut+ has richer options, however. Without a single confident parse, it may perform voice search, which allows it to seek a match without knowing

whether what it has heard is a title, an author, or a catalogue number. CheckItOut+ also filters voice search returns for reasonableness, and may ask the user for different identifying information (e.g., the author instead of the title) rather than signal non-understanding. Our experiments demonstrate that, even without the traditional parse, spoken input often provides information that improves task success. As implemented here, however, the wizard models did not improve user satisfaction. The only subject who had called both CheckItOut and then, months later, called CheckItOut+, commented on the change: "This new system just doesn't let up on you." CheckItOut+ is indeed persistent, resulting in dialogue like Table 3.

FX2's agreement graph is a more cognitive approach to the same skills that CheckItOut and CheckItOut+ already share. FX2 uses its ability to transform signal to text, to parse, and to search as a way to represent what it expects, what it hypothesizes, and what it confirms as true. This allows FX2 to be more resourceful in the face of partial understanding. Among the host of rationales FX2 uses to reason about its agreement graph, many are drawn from the same features that underlie CheckItOut+'s models, that is, the Advisors advocate decisions based on the same premises that model expert humans. (FX2 also harnesses the additional modality of sound with DoubleMetaphone support for several of its Advisors.) In addition, the agreement graph represents the conversational state, that is, what dialogue utterances have contributed to the current common ground with respect to task objects. FX2 allows a new utterance to change an agreement graph for a target already addressed by an earlier utterance. It periodically removes weak hypotheses, and makes decisions based on the merits of those that remain.

FX2 involves more cognitive processing in its retention and use of knowledge in the agreement graph from one user utterance to the next. Both CheckItOut and CheckItOut+ are reactive: they process the most recent user utterance only, and discard knowledge computed from earlier utterances. Reactivity works well in spoken dialogue when input is sufficiently accurate; here it proved successful 65% of the time. When reactivity fails, however, such a system becomes repetitive and brittle. CheckItOut+ sought to compensate with models of human resourcefulness. In contrast, FX2 uses subsequent user utterances to refine what it believes. FX2 also makes a clear distinction between what is plausible (as evidenced by hypotheses and their merits) and what is certain (i.e., what has been grounded).

Certainty is an essential aspect of cognitive systems, one that CheckItOut+ begins to address for decisions in spoken dialogue system. Its three models reference system-component confidence values and other metrics on performance accuracy (e.g., number of questions) to select its actions. These models recognize when CheckItOut+ has a partial understanding, when it has a reasonable guess, and when it should seek another way to identify a target. This procedural metaknowledge is learned from features for components where developers of spoken dialogue systems know that errors are likely to arise. We argue that FX2's approach to certainty is more cognitive, and more direct: it scales certainty as merit, and represents partial information explicitly. It links targets in its agreement graph with plausible values, and formulates grounding behaviors for strong hypotheses. The agreement graph is a clearinghouse for commentary on what may or may not have been intended by the user, as construed by FX2's Advisors. Thus, FX2 harnesses partial understanding and multiple perspectives to match spoken input and domain knowledge to targets. FX2 recognizes that the common ground must be carefully and appropriately determined based on hypotheses and their merits, and employs rationales people use to reason about it.

Our experiments made clear that people want a spoken dialogue system that is not only fast and effective, but also easy to converse with. Users also need confirmation, so that they know what the system believes. For example, even when a wizard was both certain and correct, several users

complained that they were surprised at the end of the call to hear that the order summary they had demanded actually included the correct books. FX2's fine-grained grounding provides more transparency about how the common ground evolves.

Some of this work has appeared in venues for natural language processing, human cognition, or system design. Here, we have sought to compare and analyze it, primarily to clarify the role of knowledge and certainty in understanding during dialogue. Task-specific knowledge about objects often provides contextual data against which to match accurate input. Two spoken dialogue systems here, however, use contextual data to generate plausible hypotheses from imperfect input. One learns models of human decision making from thousands of instances. The other learns to combine many rationales that were effectively gleaned from a few hundred instances of human behavior. FX2's rationales propose hypotheses, gauge their accuracy, and may confirm them with the user. They use knowledge about how to match and how to work toward common ground.

Human expertise inspires and supports FORRSooth, and therefore FX2, in a variety of ways. To create Advisors and devise strengths for their comments, we mined both commentaries from subjects in our pilot study, and the features that drive CheckItOut+'s models. Subjects' comments have also led to some Advisors that oppose actions (e.g., do not ground) as well as others that support them. There is even an INTERPRETATION Advisor that simulates an expert CheckItOut wizard. Other cognitive architectures have also begun to address dialogue. CogX (Lison & Kruijff, 2009) retains the pipeline; its perceptron learns to discriminate among the many more parses its relaxed grammar rules produce. SOAR uses written subdialogues to teach an apprentice goal-oriented plans using an extendible, but thus far small, vocabulary (Assanie & Laird, 2011).

A machine's context, however, is not a human one; a spoken dialogue system lacks people's world and social knowledge, and should not be restricted to human reasoning mechanisms and behaviors. Therefore, FORRSooth's Advisors also capture the perspective of the system. For example, one INTERPRETATION Advisor, before any database query, relies on a learned classifier to remove from the automated speech recognition tokens likely to correspond to noise. Whether or not people do this, a spoken dialogue system certainly should.

FORRSooth extends FORR with the ability to propose hypotheses, but it remains a work in progress. FX2 is its first application, and SATISFACTION requires further development. Some of its services (an INTERACTION manager, GENERATION of natural language, and DISCOURSE to focus attention and manage objects) are not yet implemented; FX2 uses the modules from CheckItOut instead. A FORR-based system traditionally uses a three-tiered hierarchy of Advisors; some are always correct, and others heuristically formulate behavior sequences. Both kinds of Advisors are a focus of current work for every service. Eventually we expect all six services to operate in parallel, so that a FORRSooth-based system would listen and think at the same time.

Much of this work is task-independent, including merit, the weight learning algorithm, and the agreement graph. Among the 24 features found in CheckItOut+'s models, only two (author and title queries) are library-specific. As a result, 52 of FX2's 60 INTERPRETATION and GROUNDING Advisors are domain-independent as well. The other eight, intended only for names, apply important ideas about the way attributes identify an object uniquely. Future work generalizes them for other concepts and other identifiers. Our best wizards' problem-solving behaviors are also task-independent, and therefore likely to pertain to other cognitive systems as well: search before you reply, disambiguate among likely search returns, and notice when no match looks reasonable.

8. Conclusion

As we demand more of spoken dialogue systems, they will find it increasingly challenging to understand their users. Future systems will have to detect and address subtasks, and consider how speech about attributes of objects can be exploited to identify those objects with certainty. People, meanwhile, will continue to expect the efficient, virtually error-free performance traditional spoken dialogue systems now produce when they receive short utterances from a limited vocabulary.

Our three spoken dialogue systems all rely on the same speech recognizer and databases, but the cognitively-oriented FX2 uses them with considerably more success. CheckItOut+ monitors its pipeline, and behaves differently when it believes an error has arisen. FX2 employs a variety of rationales observed in human behavior during our pilot study and our wizard experiment, and learns to balance them, instead of pre-specifying their interaction. CheckItOut+ models, to some extent, how people make the kinds of decisions a spoken dialogue system must make. Some of its models' features are about dialogue history, but the system retains no partial information from one adjacency pair to the next. Making decisions like a person constrained within a pipelined dialogue system proves to be less effective than collaboration with the user on the common ground to minimize misunderstandings. Nonetheless, the features behind human decisions are a rich, task-independent resource for dialogue decision rationales, one that FX2 exploits to its advantage.

FX2's agreement graph is a dynamic representation of what it believes the user meant across multiple utterances, and its certainty in that information. It begins as a model of the task (a set of targets to be bound), but rapidly becomes a representation of what the system suspects, what it has confirmed, and what remains to be determined. The agreement graph makes it possible to tell the user what the system "thinks" (as in Figure 3), and FORRSooth's Advisors can explain why it thinks so (e.g., "this sounds like the first name and is similar to the last name"). FORRSooth's services and most FX2 Advisors are task-independent procedures that capture a broad range of reasons to consider something a good match or worthy of consideration for binding. Together they use knowledge and certainty to support understanding with precision as good as, or better than, the best of our human wizards.

Acknowledgements

The National Science Foundation supported this work under awards IIS-084966, IIS-0745369, and IIS-0744904.

References

Assanie, M., & Laird, J. (2011). Learning by instruction using a constrained natural language interface. http://www.eecs.umich.edu/~soar/sitemaker/workshop/19/assanie-FinalWorkshop.pdf

Bangalore, S., Bouillier, P., Nasr, A., Rambow, O., & Sagot, B. (2009). MICA: A probabilistic dependency parser based on tree insertion grammars. *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*. Boulder, CO.

Bohus, D., & Rudnicky, A. I. (2009). The RavenClaw dialog management framework: Architecture and systems. *Computer Speech and Language, 23*, 332-361.

Clark, H. H., & Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science, 13*, 259-294.

Epstein, S. L. (1994). For the right reasons: The FORR architecture for learning in a skill domain. *Cognitive Science, 18*, 479-511.

Epstein, S. L., Passonneau, R. J., Gordon, J., & Ligorio, T. (2011). The role of knowledge and certainty in understanding for dialogue. *Proceedings of the AAAI Fall Symposium on Advances in Cognitive Systems*. Arlington, VA.

Gordon, J., Epstein, S. L., & Passonneau, R. (2011). Learning to balance grounding rationales for dialogue systems. *Proceedings of the SIGDIAL 2011 Conference*. Portland, OR.

Gordon, J., & Passonneau, R. J. (2010). An evaluation framework for natural language understanding in spoken dialogue systems. *Proceedings of the Seventh International Conference on International Language Resources and Evaluation*. Valleta, Malta.

Gordon, J., Passonneau, R. J., & Epstein, S. L. (2011). Helping agents help their users despite imperfect speech recognition. *Proceedings of the AAAI Symposium on Help Me Help You: Bridging the Gaps in Human-Agent Collaboration*. Stanford, CA: AAAI Press.

Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., & Rudnicky, A. (2008). PocketSphinx: A free, real-time continuous speech recognition system for hand-held device. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. Las Vegas, NV.

Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing* (2nd Ed.). New Brunswick, NJ: Prentice Hall.

Ligorio, T. (2011). *Feature selection for error detection and recovery in spoken dialogue systems.* Doctoral dissertation, The Graduate Center of The City University of New York, New York, NY.

Ligorio, T., Epstein, S. L., Passonneau, R., & Gordon, J. (2010). What you did and didn't mean: Noise, context, and human skill. *Proceedings of the Annual Meeting of the Cognitive Science Society*. Portland, OR.

Lison, P., & Kruijff, G.-J. M. (2009). Robust processing of situated spoken dialogue. *Proceedings of the 32nd German Conference on Artificial Intelligence*. Paderborn, Germany.

Passonneau, R. J., Epstein, S. L., Gordon, J. B., & Ligorio, T. (2009). Seeing what you said: How wizards use voice search results. *Proceedings of the IJCAI-09 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. Pasadena, CA.

Petrovic, S., & Epstein, S. L. (2007). Random subsets support learning a mixture of heuristics. *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference*. Key West, FL.

Ratcliff, J. W., & Metzener, D. (1988, July). Pattern matching: The Gestalt approach. *Dr. Dobb's journal*.

Raux, A., & Eskenazi, M. (2007). A multi-layer architecture for semi-synchronous event-driven dialogue management. *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*. Kyoto, Japan.

Raux, A., Langner, B., Black, A., & Eskenazi, M. (2005). Let's go public! Taking a spoken dialog system to the real world. *Proceedings of the Ninth Biennial Conference of the International Speech Communication Association*. Lisbon, Portugal.

Walker, M. A., Litman, D., J., Kamm, C. A., & Abella, A. (1997). Paradise: A framework for evaluating spoken dialogue agents. *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics.* Madrid, Spain.

Ward, W., & Issar, S. (1994). Recent improvements in the CMU spoken language understanding system. *Proceedings of the ARPA Human Language Technology Workshop*. Plainsboro, NJ.