Naturalistic Dialogue Management for Noisy Speech Recognition

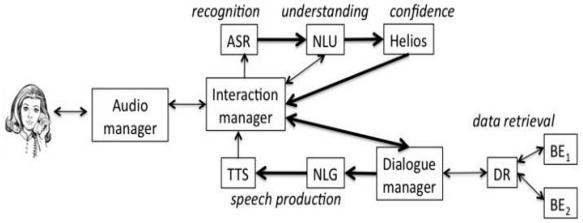
Rebecca J. Passonneau, Susan L. Epstein, Tiziana Ligorio

***Abstract*—With naturalistic dialogue management, a spoken dialogue system will behave as a human would under similar conditions. This paper reports on an experiment to develop more naturalistic clarification strategies for noisy speech recognition in the context of spoken dialogue systems. We collected a wizard-of-Oz corpus in which human wizards with access to a rich set of clarification actions made clarification decisions online, based on human-readable versions of system data. The experiment compares an evaluation of calls to a baseline system in a library domain with calls to an enhanced version of the system. The new system has a clarification module that consists of a suite of three machine-learned models organized in a decision tree. The enhanced system has a significantly higher rate of task completion, greater task success and improved efficiency, and relies on naturalistic dialogue management to achieve this.**

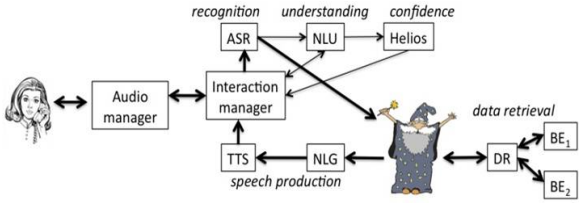
***Index Terms*—Human Computer Interaction, Robustness, Speech, System Performance, Machine Learning**

I. Introduction

Despite the increasing prevalence of speech-enabled devices, human-machine dialogue remains far less fluent than human-human dialogue. There is a vast disparity between human facility with speech and the brittle language skills of spoken dialogue systems. Automated speech recognition performs very well for non-interactive applications, such as search or transcription of broadcast news. It is particularly challenged, however, by spoken language characteristics associated with spontaneous dialogue and turn taking, such as disfluencies and overlapping turns [1]. In response, much work on dialogue strategy for automated systems has modeled decision making as a stochastic optimization problem for a specific range of conditions, such as recognizer performance for a given target population on a particular dialogue task [2]–[7]. This approach requires pre-specification of dialogue acts and relevant features of the dialogue state, both of which become increasingly difficult as dialogue tasks become more complex. The thesis of this paper is that deeper investigation of how humans rely on context, interaction skills, and linguistic knowledge is necessary to develop more flexible dialogue systems. In particular, identification of the specific strategies that people adopt for interpretation of noisy speech recognition is useful for stochastic, knowledge-based or hybrid approaches. This paper recounts experiments that elicited and applied humans' natural problem-solving abilities in the face of speech recognition errors. There are two principal results. First, people drew upon many clarification strategies rarely used in dialogue with other people, and unavailable in existing systems. Second, when augmented with only a few of these strategies, a dialogue system significantly improved its task performance and reduced its dialogue costs.



(a) Baseline CheckItOut



(b) Embedded Wizard

Fig. 1. Information pipelines for the baseline and wizarded systems

To investigate the impact of human strategies, we began with a baseline spoken dialogue system with consistently poor speech recognition, shown schematically in Figure 1(a). Users followed scenarios to order library books from the baseline system. Then, to investigate how a human would respond to the same quality of recognition strings processed by the system, we inserted a wizard module to operate side-by-side with the dialogue manager, as shown in Figure 1(b). With the new module, a human wizard could interpret the user's intentions based on the transcription output of an automatic speech recognizer (ASR), and select the next dialogue action or system action. Because we intended to model human choices, the large set of clarification actions available to the wizards were derived from an earlier experiment with human subjects. Because we intended to apply models of wizard behavior in an automated system, we collected a host of system features to represent the system dialogue states concurrent with the wizard's runtime decisions.

Two of our wizards performed much better than the others, and their strategies were quite distinct. Figure 2 (an excerpt of Figure 5), illustrates how one of them avoided misunderstandings and non-understandings through constant grounding

Rebecca Passonneau is with the Center for Computational Learning Systems of Columbia University, New York, NY 10115 USA e-mail: (see http://www.michaelshell.org/contact.html).

Susan L. Epstein is with Hunter College and The Graduate Center of The City University of New York.

Tiziana Ligorio completed this work while a graduate student at The Graduate Center of the City University of New York

Manuscript received DATE.

16 S: Did you ask for an author?
17 U: YES
18 S: What is the author's name?
19 U: .OF. .NOPE. .NO. .YOU. .GO. SAY THAT
20 S: Did you ask for a title?
21 U: NO
22 S: I'm still having trouble. Let's try the next book . . .

Fig. 2. Abbreviated example of wizards' dialogue strategies

of partial information. The user has already responded to a prompt for the next book, but the wizard cannot interpret the ASR output. The wizard prompts the user to find out if the utterance specified an author (line 16), which the user confirms. When the wizard prompts for the author's name and cannot interpret the response, she tries another attribute (title), and when that fails, the wizard suggests they temporarily move on to another book. Expressing a failure to understand, or re-prompting for the same information, are common behaviors in dialogue systems. Our wizards had these options, but rarely used them. Instead, this wizard typically sought new information that would move the dialogue closer to resolution of the user's goal. Our other high-performing wizard more often resorted to the dialogue action shown at line 22: to temporarily suspend the current goal for another one.

The work presented here seeks to learn how a spoken dialogue system could address poor speech recognition well enough to nip misunderstandings in the bud, and achieve respectable task success. We enhanced the baseline system with strategies learned from the data provided by the two best wizards. The learned strategies apply in situations where the baseline system does not produce a semantic interpretation of the user's speech. We refer to the resulting dialogue strategies as naturalistic because the system's behavior is derived from people who made decisions under similar conditions of noise in the communication channel. The learned strategies are not natural, in the sense that it is not natural for one dialogue participant to have access only to ASR transcriptions of the other's speech. They are, however, naturalistic in that they have been learned from people who apply their natural ability to handle a noisy channel through reliance on context, linguistic knowledge, and problem solving skills.

The remainder of the paper has the following structure. Section III presents related work. Sections IV-VII provide an overview of our experimental design and describe the baseline dialogue system, the wizard version, and the enhanced one. Section VIII compares the enhanced system to the baseline, and shows that the improved system is more effective and efficient. Section IX discusses the results, and the benefits of dialogue management that acknowledges the key differences between human and machine language capabilities at the same time that it relies on human adaptability to the communication constraints faced by a a spoken dialogue system.

II. Motivation

A. Aptness of an Embedded WOz Corpus

The dialogue manager chooses the communicative and task actions for a spoken dialogue system to execute. Dialogue management design is typically informed by one of several classes of corpora: human-human, human-wizard, or simulated human-machine. Machine learning has been applied to such corpora with various goals, for example, to learn dialogue acts from human-human corpora [8], to learn error-handling parameters for frame-based dialogue management from wizard-of-Oz (WOz) corpora [9], or to learn an optimal policy for a predefined set of dialogue actions through reinforcement learning with simulated users [2] or real ones [10]. We collected a corpus of dialogues between users and wizards embedded in our dialogue system. This provided data on actual recognition errors across a wide range of speakers, and on a wide range of system features from all phases of spoken language understanding. Our corpus permits offline learning of what actions to take and in what dialogue states.

Spontaneous human-human dialogues produced in a natural setting illustrate how conversational participants address a range of conversational and real-world goals through language. This exemplification function is important for dialogue management design, because the mapping between utterances and intentions is indirect, and depends on inference and contextual knowledge. The design of our baseline system is informed by a human-human corpus we illustrate below. However, human-human dialogue is not an ideal model for what a spoken dialogue system should do when there are speech recognition errors. Misunderstandings and non-understandings are much less frequent in human-human dialogue than in human-machine dialogue [11]. When they do occur, it is less often because a dialogue participant has misheard an utterance and more often that she has confusions about her conversational partner's underlying intent [12].

B. Intentions, Grounding Actions, and Domain Knowledge

Under ordinary conditions, human-human dialogue exhibits few speech channel confusions, yet it is well known that people can interpret a noisy signal or incomplete acoustic channel. For example, the cost of telephone service is lowered by transmitting less than the full frequency bandwidth of human speech; people are largely unaware of the missing frequencies. Here we consider an example of a human clarification strategy pertaining to the speech channel that illustrates two communicative skills relied on by our human wizards: to infer the speaker's intentions, and to continuously ground one's understanding of the speaker.

The first skill we consider, the ability to identify and respond to a speaker's intention, is part of but distinct from the interpretation of a speaker's current utterance. An utterance within an ongoing dialogue can introduce an entirely new intention, but the majority of intra-dialogue utterances address an existing intention. For example, if a library patron speaks with a librarian to borrow a particular book, the intended book must be identified, and this might take several utterances. We assume that each dialogue participant has her own communicative intentions, that the intentions of both participants evolve in coordination with each other, and that a particular intention continues to evolve until addressed or abandoned. A speaker infers the other's intentions by considering how her words relate to the current context. We assume that a dialogue

1.0	L	okay
2.0	L	do you have the title
3.0	P	I not really
4.1	P	[no]
4.2	L	[author]
5.0	P	excuse me
6.0	L	wh- wha- do you have the author
7.0	P	Cesar Millan
8.0	L	-M- -I- -L- -A- -N-
9.0	P	yes

Fig. 3. Excerpt from a transcript of a phone call placed by a patron to a librarian representing a single *intentional segment*. The patron mistakenly confirms the spelling of an author name at line 9.0.

system should behave as if it can understand the evolving intentional context, whether it has an explicit representation of user intention, as in agenda-based systems [9], or an implicit one, as in stochastic approaches that track belief state [13].

The second dialogue skill we consider is collaboration to establish the *common ground* [14]. The listener collaborates with her conversational partner to indicate how confident she is that she understands the speaker's utterance and its relation to the inferred intentional context. This *conversational grounding* covers a broad range of verbal and non-verbal behaviors. Through backchannel actions, for example, a conversational participant indicates to her partner that she is attending to the discourse [15]. Manifestations of continued attention include physical orientation (e.g., eye contact), gestures (e.g., head nods), and vocalizations (e.g., *uh huh* or *okay*). Most dialogue systems neither engage in nor monitor backchannels, but they do rely on grounding actions such as requests for clarification, and implicit and explicit confirmations. Our wizards had a large number of clarification actions to support continuous grounding in a manner inspired by the example in Figure 3, and derived from an earlier study [16].

Figure 3 shows a sequence of turns from a phone conversation between a library patron (P) and a librarian (L) taken from calls we recorded at the Andrew Heiskell Braille and Talking Book Library of New York City. This library provides materials in a proprietary audio format and in braille to qualified patrons.[1] Most library transactions are handled by phone. The excerpt in Figure 3 illustrates a rare case of confusion about the speech channel. The first field of the line number indicates sequence in time (e.g., 2.0 follows 1.0). A non-zero in the second field is an arbitrary number to distinguish simultaneous speech, thus 4.1 represents the patron reiterating a previous negative response (*no*) at the same time in 4.2 that the librarian prompts for the author (with question intonation). When the librarian takes the turn at 6.0, she produces two false starts followed by a self-repair. Such disfluencies are particularly difficult for speech recognizers.

The sequence of utterances shown in Figure 3 constitutes a single *dialogue segment*. A dialogue segment is the observable reflection of single intention. This segment was identified in the corpus through a reliable manual annotation procedure described in [17], [18]. The librarian initiates the segment with the intention of helping the patron identify the book she wants. Because the librarian asks a sequence of questions at 2.0, 4.2, 6.0 and 8.0, and the patron responds in turn, the librarian maintains the dialogue initiative (i.e., control of turn sequencing) throughout the segment, except where the patron prompts for a clarification at turn 5.0. The patron's cooperative responses reflect her intention to coordinate with the librarian on this task. Note that the segment begins with the discourse cue word *okay* [19], a frequent marker of a new discourse segment, in combination with other features [20].

Figure 3 illustrates several types of conversational grounding. The librarian implicitly confirms her understanding that the patron wants a book when she requests values of book attributes at lines 2.0, 4.2 and 6.0. There are three clarification requests: one from the patron at 5.0 after the librarian's elliptical question, one from the librarian at 6.0 where she poses her question in a more explicit form, and another from the librarian at 8.0 where she spells out the author surname with a question intonation. The patron confirms the spelling at 9.0, which turns out to be a mistake that leads to a sequence of 40 speaker turns (80 utterances, 13 discourse segments; cf. [17]) before the librarian finally corrects the error. We have no way of knowing the cause for the incorrect confirmation of the spelling in 9.0. The patron possibly thought she heard -L- -L- , or may not have known the correct spelling, or perhaps did not perceive the difference between one or two Ls as consequential. In any case, the patron exhibited no irritation. If we could determine exactly when humans can tolerate such high dialogue costs from each other, we could get them to tolerate them from systems as well.

The noisy channel model of human language accounts for the observation that uncertainty of individual linguistic units, such as phonemes or words, varies from high to low. Units with relatively higher probability can be omitted from a message without severe loss in the information transmitted, independent of the intentional context. When a relevant context is provided, still higher degrees of signal degradation can be tolerated. Voice search exploits this fact. *Voice search* involves fuzzy matches of imperfect automated speech recognition (ASR) transcriptions against relevant database fields, with similarity metrics to rank candidate matches. The assumed intentional context determines which database fields to query. If the librarian in Figure 3 had searched the catalog for books by *Cesar Milan* using a voice search query, she would have found Cesar Millan.

III. Related Work

This section first situates our study in the context of current research on stochastic learning of dialogue strategies. It then reviews previous work on voice search for spoken dialogue systems, grounding actions to avoid and recover from misunderstandings, ablated wizard studies that address noisy speech recognition, and features to represent the status of spoken language understanding in dialogue management.

An important class of dialogue managers model dialogue as a Markov Decision Process (MDP) [2], a Partially Observable MDP (POMDP) [21], or a Hidden Information State POMDP

[1]Calls were transcribed and aligned utterance-by-utterance with the speech signal; the corpus of 82 transcripts and corresponding audio files will be released at the end of the project.

(HIS-POMDP) [13]. These approaches specify a set of actions $a_j \in A$ to perform, depending on the current state $s_i \in S$, and each state-action pair (s_i, a_j) has a transition probability T_{ij}. A reinforcement signal is associated with transitions. Reinforcement learning seeks an optimal policy to map states to actions that maximizes a global reward function. The reward function can be based on user satisfaction [22], a combination of task success and efficiency [23], or a model of user satisfaction as a linear combination of task success and efficiency [3]. The learned policy is sensitive to the reward function [3].

Typically, only a partial policy can be learned from a human-human corpus, because the corpus will not represent all possible states. More often a policy is learned through interaction with users [10], or with simulated users [4], [5], [23]. With simulation it is possible in principle to explore the full state space, but simulation becomes impractical as the state space grows larger. It has been shown early on that reinforcement learning can learn an optimal ordering of information requests [24]. Work that produces ASR errors and confidence levels as part of the user simulation learns a strategy that orders requests earlier in the dialogue for attribute values that have better ASR results (e.g., numbers) [6].

In POMDP, where the partially observable states include the users' intentions, more robust strategies have been learned than with MDP [4]. In this work, eligibility traces boosted the relevance of state-action pairs that were visited more recently. More recent work to extend MDP approaches includes belief-state tracking, where a learned strategy is based on a stochastic representation of the dialogue state at each turn [25], or on a partition over states where each subset consists of similar states representing a single belief [13], [26]. Our work addresses two types of prior knowledge that MDP approaches depend on: specification of appropriate dialogue actions, and identification of relevant features of the state representation.

Voice search is particularly suited to applications in which users of a dialogue system seek information from large databases, especially where the values are unstructured text. Voice search has been used for personal names in directory assistance [27], [28], technical paper titles in conference information systems [29], recordings for automobile music systems [30], businesses and product reviews [31] and ebooks [32]. Voice-rate [31] provides ratings for more than a million products, 200,000 restaurants, and 3,000 businesses. Let's Buy Books [32] provides access to 15,000 ebooks. No information is provided on the sizes of the databases for the systems in the other systems we have referred to here. Our library database has nearly 72,000 holdings by more than 28,000 authors. In contrast to the directory assistance applications, our book titles and authors make use of a large vocabulary ($>$ 54,000 words), and exhibit great variation in length ([1,40], $\mu = 4.89, \sigma = 3.22$) and syntactic structure.

Spoken dialogue system strategies include dialogue actions to implicitly or explicitly confirm understandings, to avoid non-understandings, and to correct misunderstandings. However, corrections made to systems tend to be more poorly recognized than non-correction utterances [33]. When a system has no understanding, it often re-prompts the user for the same information, which can lead to hyperarticulation and concomitant degradation in recognizer performance. Users seem to prefer systems that minimize non-understandings and misunderstandings, even at the expense of dialogue efficiency. Users of the TOOT train information system preferred system-initiative to mixed- or user-initiative, and preferred explicit confirmation to implicit confirmation or none at all [34]. This was true even though a mixed-initiative, implicit confirmation strategy led to fewer turns for the same task.

Ablated WOz studies in which wizards interpret real or simulated ASR explore the strategies humans produce to handle misunderstandings or non-understandings due to ASR errors. Such studies include the use of real [35], [36] or simulated ASR errors [37]–[39]. Word Error Rate (WER) for speech recognition compares the output string with a reference transcription, and normalizes the number of insertions, deletions and substitutions by the total length. Simulation makes it possible to control for WER to observe how wizards' strategies change as WER increases [39]. In a study where simulation yielded low, medium or high WER, wizards followed a non-understanding or misunderstanding more often with a task related question than a clarification under low or medium WER [39]. Under high WER, however, misunderstandings significantly increased when wizards followed non-understandings or misunderstandings with a task related question instead of a clarification. In another study with simulated ASR, wizards simulated a multimodal MP3 player application with access to a database of 150K music albums [38]. In the noisy transcription condition, wizards made clarification requests about twice as often as people did in similar human-human dialogue. In a study with real ASR at 30% WER, wizard utterances indicated a failure to understand in only 35% of cases with incorrect ASR [36]. Wizards relied on phonetic similarities of ASR words to words salient in the domain. A large study with 43% WER also found that wizards signaled misunderstanding very rarely (5% of all their turns) [40]. For example, for 20% of non-understandings, wizards continued a route description, asked a task related question, or requested a clarification. Our study uses real ASR for two reasons. Our goal was to present wizards with the types of transcription errors systems actually make, including any trends from utterance to utterance arising from changes in speaker state that would be difficult to simulate (e.g., drop in intensity). This makes it possible for wizards to rely on similarities of the ASR output string to domain words (through voice search results) in a way that extends the finding that wizards can infer user intent by paying attention to the relation of the transcription to the domain under discussion [36]. We therefore provided wizards with a large range of clarification actions that include questions about the ASR output string.

MDP approaches to learn dialogue strategy make use of disparate features, and sometimes learn the state representation offline [4]. In other approaches to dialogue management, such as frame-based or agenda-based, early work that addressed confidence annotation of input to the dialogue manager illustrates how low level ASR features interact with features from natural language understanding and dialogue management. Confidence annotation has been done through linear regression

models whose predictors include parse and dialogue state, relying on at most a dozen features to learn a binary threshold [41], [42]. When learning was done over multiple days for nine recovery strategies, system performance improved [42]. Confidence scores can also be continuous. They can be based on a normalized sum of confidence over components of the semantic representation of user utterances [43], or a probabilistic combination of acoustic, semantic and discourse features [44].

In our experiments, we learn a suite of strategies for a specific choice point: cases where the baseline system fails to arrive at a semantic interpretation. Previous work has also focused on specific strategies [10] or choice points [45] in the context of a larger system. For the dialogue actions of whether to use a directive versus an open-ended prompt, and whether to explicitly confirm understanding, a training corpus was simulated using a random policy in [10]. Their learned policies depended on seven features: dialogue state, history, a semantic confidence score combining acoustic model (AM) score with whether the user confirmed an attribute's value, and the use of a restricted versus unrestricted grammar. System performance improved for some objective measures (e.g., task completion), but not task success. A much larger set of 53 features from all phases of understanding was used in [45]. They learned to identify problematic dialogues within a few utterances. Most dialogues were 5 exchanges or less, compared with 25 to 30 in our data.

IV. Experimental Design

The application investigated here is to identify books to check out from the Andrew Heiskell Braille and Talking Book Library. Heiskell patrons request library books by telephone, and receive and return books by mail. Patrons receive monthly newsletters listing new holdings, and typically know the full title, author or call number of books they request. Librarians take turns at the telephone help desk throughout the day, and often answer multiple calls at once. To study the domain and the characteristics typical of patron-librarian calls, we recorded and transcribed eighty-two calls. In these calls, patrons made 375 distinct book requests, sometimes for the same book, 320 (85%) of which were for specific books. Of the requests for specific books, 57.5% were by call number, 21.6% by title alone, 19.1% by author alone or in combination with title, and 1.9% by other means. The overwhelming majority of requests by title were a full title or subtitle. Thus we did not address here the problem of inferring the intended book given an approximate version of the title (but cf. [32]).

This work addresses the choice point where the initial dialogue system, CheckItOut, has no sufficiently confident semantic interpretation to send to its dialogue manager. Machine learning applied to data from our embedded WOz corpus learned dialogue strategies for this choice point. To focus specifically on the issue of how the dialogue manager can infer user intentions when ASR is very poor, the dialogue strategy here is limited to system initiative, apart from an initial open-ended prompt to get the user started. We implemented a clarification module with the learned strategies to produce an enhanced version: CheckItOut+. Otherwise, CheckItOut and CheckItOut+ had identical functionality.

There were three data collections: a baseline corpus of calls to CheckItOut, calls to the wizard version, and calls to CheckItOut+. For each data collection, we randomly selected 3,000 books to support author and title data for the recognizer's language model, the semantic grammar rules, and on-demand generation of book-borrowing scenarios. From experiments with the ASR framework, we determined that a language model and a semantic grammar constructed from a random selection of 3,000 titles yielded the target WER of approximately 50%. Language data for everything but book titles and authors was identical across data collections. Backend queries accessed a table containing the library book data for 50,000 books. To avoid issues of information presentation, we included at most three books per author, and eliminated books with one-word titles, which had relatively few ASR errors.

Identical data collection protocols were used to evaluate CheckItOut and CheckItOut+. Five female and five male users were asked to make at least 50 calls each to CheckItOut; 562 calls were completed in July, 2010. A different set of 10 users (apart from one), again balanced for gender, performed the same task 8 months later with CheckItOut+, and 502 calls were collected. All subjects were recruited from the student bodies of New York City colleges and universities.

Prior to each call, the user accessed a website that generated a scenario on demand, including three attributes (the author, title, and call number) for each of four books. Dialogues averaged 24 turn exchanges for CheckItOut and 31 for CheckItOut+ (Table V). Users were told to use a single attribute each time they requested a book, and to use each attribute in at least one request per call. The books for each scenario were randomly selected from the 3,000 titles used for the language model and Phoenix grammar. In summary, all conditions for both data collections were held constant, other than the dialogue management for cases where the baseline system would have a non-understanding of a user utterance. Therefore, any differences in performance between the two systems must be attributed to the learned strategies.

V. CheckItOut Baseline

CheckItOut is an Olympus/RavenClaw system with an information pipeline [9], as illustrated in Figure 1a). An audio manager performs an initial segmentation of the audio stream. To a limited degree, the initial segmentation can be modified by an interaction manager [46] that considers the pipeline from the speech recognizer through the RavenClaw dialogue manager [47]. RavenClaw controls queries to the backend database and formulates prompts that are mapped to text, sent to the Kalliope text-to-speech synthesizer, and ultimately back to the user. Interaction with users uses VOIP telephony.

The PocketSphinx recognizer passes its output (orthographic transcription of user speech, plus various ASR scores) to Phoenix, a robust semantic parser [48]. A Phoenix parse consists of one or more semantic frames (e.g., `BookRequest`) with slots (e.g., `Title`) and fillers (e.g., *Bully: A True Story of High School Revenge*). Phoenix achieves robustness by allowing the parse to skip tokens in the input string either between slots within a frame, or between frames. In most

ASR .BODIE. A TRUE STORY OF HIGH SCHOOL
Parse [[Title] ([NN_phr] ([DT] (A) [JJ] (TRUE) [NN] (STORY) ([IN_phr] [IN] (OF) [JJ] (HIGH) [NN] (SCHOOL))))]
Term A TRUE STORY OF HIGH SCHOOL

Mod VS	*A TRUE STORY OF HIGH SCHOOL*
R/O	Candidate in Modified VS return
0.78	bully: a true story of high school revenge
0.72	true story of the novel
0.71	the true history of chocolate

Full VS	*.BODIE. A TRUE STORY OF HIGH SCHOOL*
R/O	Candidate in Full VS return
0.77	bully: a true story of high school revenge
0.72	the new rules of high school
0.68	the true history of chocolate

Fig. 4. Recognized title (ASR), parse, terminals from the parse (Term), and voice search (VS) return from CheckItOut

Olympus/RavenClaw applications, Phoenix grammars are often hand-generated, and productions typically map slots to combinations of strings, pointers to strings, and wild cards. To construct a Phoenix grammar rich enough for book titles, we automatically transduced Phoenix productions from dependency parses of the book titles in our database. This preserved robustness while adding recursive syntactic structure and ordering constraints.

Figure 4 shows a sample parse. Parses that consume more words with fewer slots, and fewer semantic frames per utterance, have higher scores. The parser can return multiple parses with tied scores, but ties are rare. The parse is passed to the Helios confidence annotator, along with features such as number of tokens not consumed by the parse, and the utterance-level ASR confidence.

In turn, Helios passes the parse and a binary confidence score to the dialogue manager [41]. Based on the Helios confidence score and the semantic parse, the dialogue manager determines what level of understanding to convey. For example, the dialogue manager can confirm a concept value explicitly (*Did you say 'John Grisham'*), confirm one implicitly (*'An Accidental Woman' is available*), or confirm a misunderstanding (*I'm sorry, I must have misunderstood you*) or a lack of understanding (*Sorry, I didn't understand you*).

In a previous offline pilot study [16], we explored the utility of voice search. Subjects found correct items quite successfully when given a large text file of book titles, noisy ASR transcriptions of user requests, and unlimited time. In a subsequent online task to resolve a single book request by title, embedded wizards who read noisy ASR transcriptions to handle user requests were provided with a voice search query that returned a short list of likely matches (without the numeric scores shown in Figure 4) [49]. Wizards could identify a correct match even if it was not the most highly ranked candidate. Only the most expert wizards were able to tell when a voice search return did not contain a match.

In an Olympus/RavenClaw system, only the dialogue manager can query the backend, and only when it has a confident semantic parse. For CheckItOut, we incorporated a modified voice search for the backend query, based on the parsed portions of the ASR output string. Returns were ranked by Ratcliff/Obershelp (R/O) similarity, the ratio of the number of characters two strings have in common to the total number of characters in both strings [50]. Figure 4 illustrates how voice search against known domain entities can resolve the literal content of an utterance. It shows an actual ASR output string, where periods delimit a low confidence word that goes unparsed (`.BODIE.`), the parse, and the terminal words extracted from the parse (Term). The modified voice search (MOD VS) used in CheckItOut queries with the Term string, because the unconfident ASR word `BODIE` does not appear in the parse. Here the highest scoring return is correct, despite the omission of words from the parse string. Figure 4 also shows the voice search used in CheckItOut+ (FULL VS), which uses the full ASR; note the difference in the three top-ranked candidates and their R/O scores. Here, the top ranked title by R/O score is the correct title, but this is not always the case. For example, in our earlier study where wizards addressed a single book request without a full dialogue, the top ranked return from full voice search was correct for 65.24% out of 4,172 book requests by title, and was missing from the top nine candidates 28.43% of the time.

VI. CheckItOut with Embedded Wizards

Olympus/RavenClaw systems rely on the Galaxy message passing architecture, which permits arbitrary modules to be inserted in the hub [51]. We inserted a wizard module to allow wizards to interact with users without the ability to hear them or speak to them. The pipeline information flow in the architecture is illustrated in Figure 1b): ASR transcriptions go directly to wizards, and they select all response prompts. Prompts are then mapped to words (natural language generation), sent to the speech synthesizer, and finally to the user. Through a graphical user interface (GUI), wizards saw the transcriptions in real time, and selected actions from a finite set described below. The natural language understanding and dialogue manager modules in Figure 1b) produced output in parallel with the wizard. This allowed us to collect run-time features from the recognizer, the Phoenix parser, the Helios confidence annotator, the dialogue manager, and voice search, along with features from the wizard GUI to represent the wizards' dialogue states and their actions.

Nine wizard candidates were trained in use of the GUI, including use of voice search. Three males and three females were selected. Each of ten users, evenly balanced for gender, made at least fifteen calls to each of the wizards, for a total of 913 calls. Wizards had up to six minutes per call, compared with an average of 3.5 minutes per call to the baseline CheckItOut.

A. Automated speech recognition quality

Given the expense and overhead of human transcription, we do not have an exact WER for the full corpus. We estimate WER to be about 50%, based on an actual measure of average WER for 315 utterances collected during the experiment from a similar range of speakers, and on the WER measured for an earlier experiment with similar recognition settings [49]. While quite poor, WER this high is not unheard of for a research

1	S:	OK, let's order some books. You can ask for a title, author or book number. And if I get something wrong you can say 'That's not what I said'
2	U:	.VAN. .LOON. WELL
3	S:	Did you ask for an author?
4	U:	NOPE
5	S:	Did you ask for a title?
6	U:	YES
7	S:	Can you repeat the book request please?
8	U:	FOUR FOUR FOUR FOUR EIGHT
9	S:	Is the title the blue wall?
10	U:	YES
11	S:	Next book?
12	U:	.CATHERINE. AIRD
13	S:	Did you ask for an author?
14	U:	YES
15	S:	What is the author's name?
16	U:	.OF. .NOPE. .NO. .YOU. .GO. SAY THAT
17	S:	Did you ask for a title?
18	U:	NO
19	S:	I'm still having trouble. Let's try the next book and come back to this later
20	U:	.CATHLEEN. .MAGNETIC.
21	S:	Did you ask for an author?
22	U:	YES
23	S:	Is the book by Cathleen Medwick?
24	U:	YES
25	S:	Is the title 'Teresa of Avila, the Progress of a Soul'?
26	U:	YES

Fig. 5. Example of poor ASR performance and WA's strategy

spoken dialogue system handling calls in real-world conditions (cf. [52]). WER varies widely across speakers and calls.

Because words differ in semantic importance, WER is a coarse measure of understanding. Previous work has used concept accuracy [33]. Here we use task success to illustrate the challenges presented by poor ASR, and the wide variation across speakers. Average task success in ordering four books ranged from a low of 25% to a high of 82% for the ten users ($\mu = 0.57, sd = 0.18$). The high performance for the most successful user is doubtless due to the superior ASR performance for this speaker, evident from the readability of the transcripts.

Figure 5 illustrates how clues about content can be inferred from very poor ASR. In lines 2 through 10, wizard WA addressed the unintelligible transcription at line 2 by asking what attribute value was provided (title or author). Note that when WA ultimately identified the book (by call number) the last title word was phonetically similar to the last word in line 2 (*wall* versus `WELL`). In the same dialogue, WA was able to identify the intended author at line 23; first she confirmed that the utterance provided an author value, then she noticed the similarity to the top ranked voice search return (*Cathleen Medwick* versus `.CATHLEEN. MAGNETIC`). A dialogue system with strategies to build on partial information in this way would have a greater variety of clarification strategies relevant for poor ASR. Incorporating only a few of these leads to significant performance improvements for our system.

B. Wizard GUI and Dialogue Actions

Embedded wizards made all decisions through their GUI. Figure 6 is a screenshot of the GUI for book request subdialogues. Scrollable ASR output appears at the upper left, with low confidence words delimited by periods. Wizards could make any number of queries before selecting a prompt, and could continue to prompt the user before performing a query. Voice search results appear at the upper right as a simple ranked list (without R/O scores). Wizard decisions and actions were achieved by mouse clicks on dialogue action buttons on the middle right and at the bottom of the GUI.

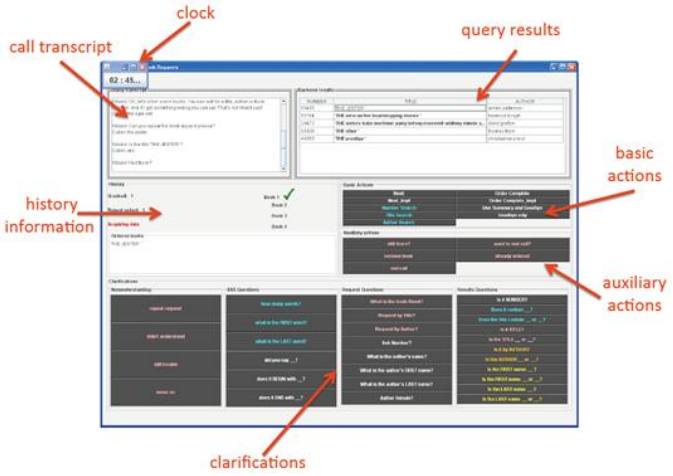


Fig. 6. Graphical User Interface for Embedded Wizards

For book request subdialogues, the middle right frame of the GUI provided wizards with five prompts for basic actions and six for auxiliary actions. The basic actions were to prompt for a book, to explicitly confirm a requested book combined with a prompt for the next book, to offer to summarize the book order, to provide such a summary, and to thank the user. The auxiliary actions were to ask if the user were still on the line, to ask whether the user wanted to hang up, to indicate that the requested book had been ordered, to have the system hang up, to ask the user to call back later, and to re-prompt for the next book. Of these 11 actions, librarians in 82 calls we recorded and transcribed used variants of all 5 basic actions, and none of the auxiliary actions. A librarian might ask if a user was still on the line; this never occurred in the calls we transcribed, but it occurred 34 times in the 913 wizard dialogues. None of the other auxiliary prompts would be likely to occur in human-human dialogue.

The bottom of the GUI provided additional wizard actions, categorized as non-understandings, or as clarification requests about the ASR, the user's request, or the voice search results. The GUI offered three non-understanding prompts to elicit a repetition or rephrasing from the user, and a fourth to suggest moving on to the next book. The remaining actions consisted of 25 clarification questions. These were informed by our previous embedded wizard study of a single turn exchange in which users requested a book by title and wizards performed voice search [49]. In that study, if wizards chose not to offer a candidate from the query return, they could ask a free-form question. Table I lists 18 request prompts that occurred 0.01 times per call or more, in decreasing order of frequency per call (from 2.69 to 0.01), classified into the 9 categories shown

Freq.	Prompt	Category
2.69	Is the title __ ?	1) resolve query return
0.76	Did you ask for an author?	2) clarify book attribute
0.69	Did you ask for a title?	2) clarify book attribute
0.68	What is the author's name?	3a) request attribute value
0.68	Did you speak the word _?	4a) confirm ASR word
0.61	Can I have the catalog number?	3a) request attribute value
0.22	What is the author's last name?	3b) request partial attribute value
0.13	Is the author's last name __ ?	4b) confirm partial attribute value
0.13	Is the author's first name __ ?	4b) confirm partial attribute value
0.11	Does the title contain the word __ ?	4) confirm ASR word
0.10	What is the author's first name?	3b) request partial attribute value
0.09	What is the book about?	5) request subject matter
0.07	Does it begin with _?	4a) confirm ASR word
0.05	What's the first word?	6) request ASR info
0.04	Does it end with __ ?	4) confirm ASR word
0.01	How many words are there?	6) request ASR info
0.02	What's the last word?	6) request ASR info
0.01	Is the author female?	3a) request attribute value

TABLE I
CLARIFICATION PROMPTS AVAILABLE TO THE WIZARDS ORDERED BY FREQUENCY PER CALL

in the third column. The 7 other prompts occurred less than 0.01 times per call.

Table I illustrates how this data collection supports investigation of clarification questions that can resolve noisy ASR. Apart from the category *request subject matter*, none of the question types in the table would occur with any frequency in a human-human corpus. Nonetheless, these questions provide a spoken dialogue system with ways to address noisy transcriptions beyond those that it might typically use.

C. Contrasting Wizard Strategies

Our six wizards chose different proportions of GUI actions per call, and had different completion and success rates. As shown in Table II, the two wizards identified as WA and WB had the highest proportion of correct books per call, 2.69 and 2.53 respectively. Wizards differed in their overall strategies; this is especially noticeable with the two most successful wizards. WA asked the most questions per book request and performed the most voice search (cf. values in boldface Table II). She often performed multiple searches within a book request to gather more information, and asked questions after each search, as illustrated in Figure 5. In contrast, WB executed searches at the average rate for all wizards, and asked the fewest questions. The most distinctive aspect of WB's strategy is that, more than any other wizard, WB would suspend a troublesome book request and move on to a new or previously suspended request. The prompt for the action, *Let's try the next book and come back to this later*, occurred from 0.36 to 0.65 times per dialogue for other wizards, compared with 1.07 times per dialogue for WB. Also, WB's dialogues had an average of 4.4 book requests each. Because only 4 distinct books were provided per scenario, this indicates that in dialogues with WB, users often returned to a previously unsuccessful request. Our next challenge was to exploit these two different but equally successful approaches.

	Task Success						
Per Scenario (4 books)	Avg.	WA	WB	WC	WD	WE	WF
Book requests	3.72	3.64	4.44	3.75	3.57	3.19	3.69
Correct books	2.26	2.69	2.53	2.19	2.07	1.89	2.28
Actions per book request							
Searches	1.73	**2.10**	1.72	1.73	1.70	1.70	1.67
Questions	3.41	**4.09**	**2.28**	3.53	3.68	3.90	3.28
Confirmations	1.74	2.05	1.10	1.66	1.56	2.37	1.93

TABLE II
BREAKDOWN OF WIZARD BEHAVIOR

Learned model	Features	Instances	Accuracy	F
VoiceSearch	11	3161	84.81	0.88
OfferResults	11	714	69.37	0.75
NonUndersting	5	1159	67.70	0.71

TABLE III
PERFORMANCE OF THREE LOGISTIC REGRESSION MODELS WITH RFW+ FEATURE SELECTION

VII. CHECKITOUT+: LEARNED DM STRATEGIES

To extend CheckItOut with new behaviors for ASR too noisy to yield a confident semantic parse, we performed machine learning on wizard data. The input to the machine learning algorithms was data from the calls of the two most successful wizards (WA and WB), who had distinct, complementary strategies. Their dialogues provided training examples for three decision points. Each learned model is a component of the overall dialogue strategy. This section describes the data, the learned models, and how they were combined in a single module to produce CheckItOut+.

The learning task was to predict the wizard's action at three decision points: whether to use voice search to resolve noisy ASR, whether to offer a voice search result to the caller or to prompt for additional information, and whether to move on from the current book request or to indicate a non-understanding. One model was learned for each decision point. Each training instance was a feature vector that represented an adjacency pair [53] consisting of the wizard's prompt, a query if she made one, and the caller's response. Labels on the instances represented the wizard's next action. Some features represented input available to the wizard (e.g., whether the adjacency pair was the initial request in a given subdialogue). Additional features came from the semantic parse and the confidence annotator. In all, 163 features were assembled and grouped into categories to facilitate feature selection.

Learning was conducted in Weka [54] using C4.5 decision trees, support vector machines and logistic regression. From the 913 dialogues for all wizards, we assembled 16,956 training instances; each had a single caller utterance and no more than one database query within the adjacency pair.

Given so many features, feature selection was essential. We extended Weka with XFF, an experimental framework for feature selection methods [55]. There are two major approaches to the interaction between feature selection and learning algorithms [56]. In a *filter* method, features are selected prior to training through a learner-independent metric (e.g., correlation). In contrast, a *wrapper* iteratively tests feature subsets for a particular learner. We compared several feature selection

Index	Description	Group	Coefficient
	VoiceSearch		
0	Intercept		-1.9050
1	number of database searches in this book request	(A) query	13.9992
2	number of words covered by the best parse	(B) parse	5.2468
3	average helios confidence of second recognition hypotheses for this request	(C) confidence	1.9949
4	**average word level confidence in hypothesis 1**	(D) recognition	1.7821
5	whether there was no parse for this user utterance	(B) parse	0.5909
6	number of questions asked in this call	(E) adjacency pair history	-0.3150
7	number of top grammar slots in the best parse	(B) parse	-0.6991
8	**number of database searches by title in this book request**	(F) query history	-1.3243
9	number of parses for first recognition hypothesis	(B) parse	-2.1087
10	**total number of explicit confirmations in the call**	(E) adjacency pair history	-2.8164
11	number of adjacency pairs in this book request	(E) adjacency pair history	-9.7340
	OfferResults		
0	Intercept		-2.5626
1	standard deviation of the average R/O score of search return	(A) query	3.3481
2	average R/O score of the search return	(A) query	3.1356
3	number of move ons in this call	(E) adjacency pair history	0.3711
4	average acoustic model score of the best recognition hypotheses in this call	(D) recognition	-0.0428
5	whether a new book request was initiated	(E) adjacency pair history	-0.2551
	NonUnderstanding		
0	Intercept		10.7225
1	**number of database searches by title in this book request**	(F) query history	2.8356
2	number of user utterances in the call	(E) adjacency pair history	1.7930
3	**average word level confidence in hypothesis 1**	(D) recognition	1.6260
4	maximum word level confidence in the first recognition hypothesis	(D) recognition	1.4682
5	**total number of explicit confirmations in the call**	(E) adjacency pair history	1.0495
6	number of partial explicit confirmations in this book request	(E) adjacency pair history	0.5904
7	number of title slots for the parse in the first recognition hypothesis	(B) parse	-0.4353
8	helios confidence of the first recognition hypothesis	(C) confidence	-0.7359
9	number of words not covered by the best parse	(B) parse	-0.8826
10	number of database searches by author in this book request	(F) query history	-0.9554
11	acoustic model score of the first recognition hypothesis	(D) recognition	-12.4924

TABLE IV
REGRESSION MODELS

methods, including the Randomized Feature Weighting (RFW) wrapper developed for this work, and its extension RFW+ [55]. In these experiments and on seven datasets from the UCI repository [57], we found RFW to be at least as accurate as other wrappers (e.g., [58]), but often orders of magnitude faster. RFW+ extends RFW to preserve diversity across user-supplied feature categories. After preliminary feature selection, RFW+ adds features from underrepresented categories as long as learning improves.

The models learned for CheckItOut+ rely on RFW+, which outperformed other feature selection methods (including RFW) on the wizard data. Thus the feature selection results alone support our premise that it is beneficial to preserve diversity of features across all phases of spoken language understanding. Under 10-fold cross-validation, accuracy on the full dataset with RFW+ ranged from 88% to 98%, depending on the sizes of subsets of features tested during learning, and on the grouping of features into categories. Learning performance on the reduced sets that focus on the behavior of WA and WB were somewhat lower (Table III), probably due to the smaller training sets.

The three learned models are organized in a decision tree. The first two decisions take advantage of WA's ability to pursue a partial interpretation through a sequence of actions. The first learned model, VoiceSearch, determines whether CheckItOut+ will perform a voice search given ASR that cannot be given a confident semantic parse. The second learned model, OfferResults, to be used only after a voice search, determines whether CheckItOut+ offers a high ranked search return or prompts the user for more information. The third learned model, NonUnderstanding, decides what to do if the VoiceSearch model results in no voice search. NonUnderstanding is based on WB's skillful suspension of the current book request in order to move on to a new one; a suspended request is potentially resumed later in the dialogue.

Table III shows the number of instances and features selected by RFW+ in three logistic regression models, along with their accuracy and F measure. VoiceSearch was learned from 3,161 training instances from WA's dialogues. OfferResults was learned from 714 instances where WA had performed a voice search. NonUnderstanding was learned from 1,159 training instances.

Given a binary classification where the positive class has a probability p, the odds of the positive class is the ratio of p to $1-p$. Logistic regression models the logit of the odds ratio, or log odds, $(log(\frac{p}{1-p}))$ as the sum $\alpha + \beta_i\, x_i + \varepsilon$ for the intercept α, the predictor variables x_i and the residual error ε. The intercept thus indicates the log odds of the positive class $(log(\frac{p}{1-p}))$ independent of any predictors. The coefficient of each predictor indicates the difference in log odds for each unit increase in the predictor, thus a predictor is more influential the higher the absolute value of its coefficient, and the sign on the coefficient indicates whether the predictor increases or decreases the log odds. Table IV lists the intercept, predictors

and coefficients for each logistic regression (all features were normalized in [0,1]). As shown, features from all phases of spoken language understanding, as well as query and history features, played a role. Further, the feature sets for each decision are largely distinct: three in boldface for VoiceSearch also occur for NonUnderstanding, for a total of 24 distinct features in these models.

The VoiceSearch intercept in Table IV shows low log odds of performing voice search. Three features that change the log odds the most pertain to the query, the parse, or the adjacency pair history. Log odds of voice search increase the most with each increase in number of preceding queries, and also with each next ASR word that is consumed by the semantic parse. Log odds of voice search decrease the most for each prior adjacency pair in the book request. For OfferResults, the log odds are negative, and are increased the most by features pertaining to the query. They are slightly increased by the number of times in the call that the move-on prompt was used, decreased by whether the current book request is new, and slightly decreased by increases in average acoustic model (AM) score of the best recognition hypotheses in the call. The NonUnderstanding model has high log odds that are further increased by the number of voice searches by title in the current book request. The feature that has the most impact in this model is the AM score of the best recognition hypothesis, which reduces the odds of indicating a non-understanding.

When CheckItOut fails to get a confident semantic parse, it signals a non-understanding. It provides different degrees of direction about what the user can do, depending on the number of consecutive non-understandings. After three consecutive non-understandings, CheckItOut tells the user to call back.

When CheckitOut+ fails to get a confident semantic parse, however, the new clarification module chooses a response with its three learned models. The clarification module organizes the three learned models in a decision tree, represented in the pseudo-code in Figure 7. At each branch point, one of four things happens: the next model is evoked (in lines 1, 3 and 6, possibly after a query, common to both systems; a prompt informed by the wizard data is selected (in lines 5, 9 and 10, such as a request for a different attribute); an action informed by the wizard data is selected (in line 8, where the system moves on to the next book); or the subdialogue continues in the usual way (in lines 4 and 7, where a book is identified, or the system signals non-understanding).

```
 1 IF VoiceSearch(ASR) [Model 1]
 2  THEN ExecuteVoiceSearch(ASR,Result)
 3       IF OfferResults(Result) [Model 2]
 4         THEN OfferTopRanked(Result)
 5       ELSE RequestAlternateAttribute
 6  ELSIF NonUnderstanding [Model 3]
       THEN
         IF NumNonUnderstanding < 3
 7          THEN SignalNonUnderstanding
         ELSIF NumNonUnderstanding >2
 8          THEN MoveOn
 9       ELSE OR(RequestCatalogNumber,
10          QuestionAuthorRequest)
```

Fig. 7. Pseudo-code for the error handling component of the new DM

In CheckItOut+, if the VoiceSearch model determines that CheckItOut+ should perform voice search, three queries use R/O scores to compare the ASR transcription against the author, title and catalog number fields of the backend. After voice search, the OfferResults model determines whether to offer the return with the highest R/O score to the caller. If it decides not to offer the highest ranked candidate, it will still build on information from the query return by prompting for an attribute other than the one with the highest R/O score (e.g., *Can you please give me the catalog number?*). When there has been no voice search and two consecutive non-understandings, the NonUnderstanding model determines whether to move on to the next book request. If the decision is not to move on, then CheckItOut+ continues with the current book request, either with a request for the catalog number or by asking the user whether she just asked for an author.

Because the learned models depend on features from most or many phases of spoken language understanding, including recognition and parse features, as well as features from voice search queries, CheckItOut's information pipeline cannot accommodate them. RavenClaw is not designed to examine a multiplicity of confidence values for distinct phases of spoken language understanding, and the Olympus/RavenClaw architecture requires backend searches to be executed with the results of a semantic interpretation, rather than with the ASR. Rather than re-build parts of RavenClaw and restructure the information pipeline, we implemented a new dialogue manager in CheckItOut+ with exactly the same functionality as the CheckItOut dialogue manager, apart from the clarification module. Other modules remained the same.

VIII. Evaluation: Performance Gains

The new module in CheckItOut+ adds the capability to reduce non-understandings; it is invoked only when CheckItOut would have failed to arrive at confident semantic interpretation. It played a role in 91.5% of calls, and was triggered 4.7 (+/- 3.9) times per call on average. As discussed in this section, the improved error-handling leads to significantly higher task completion and success rates, significantly shorter durations per subtask, and significant changes in the rates at which users request books using call numbers versus author names. Despite the improved task success, user satisfaction remains the same across both systems. While users of CheckItOut+ always order all 4 books in the task compared to 3.2 for CheckItOut users, the greater success required increased effort on the part of users. PARADISE models to account for user satisfaction in terms of task performance and dialogue costs show that user satisfaction depends on very distinct system properties.

As noted above, for CheckItOut 562 calls were collected, and 502 were collected for CheckItOut+ (Section IV).

Table V presents 21 dialogue task and cost metrics used to compare the two systems. We first discuss whether the metrics show significant differences between the two systems. Analysis of variance (ANOVA) measures whether values for a given measurement, such as task completion, have the same or different means and variance across groups, such as two systems. On all measures other than two, differences in the system

Id	Description	CIO	CIO+	p (ANOVA)
1	Ordered books	3.2	4.0	0
2	Offered books	4.1	6.4	0
3	Completion rate	0.80	0.90	0
4	Correct books	2.4	2.7	7.3×10^{-5}
5	Incorrect books (↓ CIO+)	0.8	1.3	1.9×10^{-12}
6	Task success	0.60	0.68	7.3×10^{-5}
7	Call duration (↓ CIO)	210.9	224.4	3.2×10^{-2}
8	Time per book	64.6	56.1	4.2×10^{-4}
9	S(ystem) turns (↑ CIO; ↑ CIO+)	25.1	32.9	0
10	U(ser) turns	24.3	31.9	0
11	Time per S turn (↑ CIO; ↑ CIO+)	8.3	6.8	0
12	Time per U turn (↓ CIO)	8.6	7.0	0
13	User utterances (↓ CIO)	25.2	31.9	0
14	Help messages (↓ CIO+)	1.8	0.1	0
15	Given data messages	4.1	8.3	0
16	Relevant data messages	0.8	1.3	8.5×10^{-12}
17	Corrections from U (↑ CIO)	1.5	2.4	8.9×10^{-10}
18	Book requests	8.1	8.0	0.89
19	Book requests by num	2.7	2.1	1.2×10^{-7}
20	Book requests by author (↑ CIO; ↓ CIO+)	1.6	2.1	1.2×10^{-4}
21	Book requests by title (↑ CIO)	3.75	3.83	0.65
	Computed user satisfaction	3.1	3.0	
	Summary user satisfaction	2.6	3.4	

TABLE V
MEANS AND ANOVA P VALUES FOR DIALOGUE TASK AND COST METRICS. METRICS THAT ARE PREDICTIVE IN PARADISE MODELS FOR CHECKITOUT (CIO) OR CHECKITOUT+ (CIO+) ARE IN PARENTHESES; ARROWS INDICATE DIRECTION OF INFLUENCE.

means are highly significant, as shown by the p values. Note that in the column for p values, '0' represents probabilities so small as to effectively be zero (underflow for the R statistical package). Table V also shows two cases with no significant difference. The number of *Book request* subdialogues for callers to both systems was about the same (8.1 versus 8.0), and the difference in the number of *Book requests by title* (3.75 versus 3.83) was not statistically significant.

On task metrics, such as *Ordered books*, completion rate, success (*Correct books* and *Incorrect books*) and success rate (*Task success*), CheckItOut+ performed significantly better then CheckItOut. The longer call duration for CheckItOut+ was also statistically significant, but in the 13.5 extra seconds in calls to CheckItOut+, callers were able to complete a request for an additional book. The average durations per received book or per correct book were significantly shorter for CheckItOut+.

Our user satisfaction survey was based on [59]–[61]. It had ten questions about the caller experience, plus one about the users' overall satisfaction, all on a 5-point scale. Users completed no more than three surveys. Responses to the questions provided a reliable scale [62]. We compared user satisfaction of both sets of users using two scores shown at the bottom of Table V: computed from the average of the 10 questions (Computed), and the response to the summary question (Summary).

Overall, users of both systems were equally satisfied. While the means are nearly the same for the Computed scores of both systems, and the mean Summary score for CheckItOut+ looks higher than for CheckItOut, the differences are not statistically significant. For CheckItOut, the Computed satisfaction scores were somewhat higher than the Summary scores, and vice versa for CheckItOut+. Both types of scores lead to the same conclusion, that overall callers were neutral about their experience, which suggests that the ten questions for the computed scores do a reasonable job at capturing individual components of the caller experience.

On individual questions, there was a statistically significant difference in responses from callers on only two of the eleven questions. For the statement, *I had to play close attention while using the system*, CheckItOut+ callers agreed more strongly with the statement than did CheckItOut callers. This difference in the demand on the user's attention can perhaps be explained in terms of the much greater frequency with which CheckItOut+ users had to correct the system (measure 17). CheckItOut corrections occurred 1.5 times per call compared with 2.4 times per call for CheckItOut+. The other question showing a difference is less easy to explain, and may account for some of the discrepancy between the two types of scores. Callers were asked to indicate how strongly they agreed with the statement *I found the system voice easy to understand*. Although both systems used the same synthesized voice, CheckItOut callers agreed with the statement more than did CheckItOut+ callers. It has been observed in the literature that user ratings of spoken dialogue systems often depend more heavily on the qualities of the synthesized voice than on ease of use, but here there seems to be a reverse effect where callers' impressions of the same voice differ, depending on the quality of the rest of their experience with the system.

The PARADISE evaluation method assumes that user satisfaction can be modeled as a linear sum of measures of dialogue task performance and costs [59]. Linear regression models were explored stepwise, using all measures shown in Table V as predictors of user satisfaction, for each satisfaction score. All callers completed at least two surveys, and some did a third. Satisfaction scores derived from the surveys that occurred about half way through a caller's fifty sessions were associated with the first half of her calls, and scores derived from the survey completed at the end were associated with the second half of her calls. The models for the summary scores produced somewhat better fits, and relied on more predictors, so we discuss these models here.

The rows in Table V with CIO or CIO+ represent measures that were predictive of user satisfaction for the corresponding system, with the arrows representing whether satisfaction increased with an increase (↑) or a decrease (↓) in the predictor. The PARADISE model for CheckItOut users has eight predictors. A smaller number of predictors (N=5) model the CheckItOut+ user satisfaction for about the same adjusted R^2: 0.06 for CheckItOut versus 0.05 for CheckItOut+. Two predictors in both models increase user satisfaction: increases in the number of system turns and increases in time per system turn. These occur when the system is able to offer the user a specific book. One predictor appears in both models with an opposite direction of influence. CheckItOut users were more satisfied when there were increases in book requests by author. The table shows that CheckItOut+ users tended to request books by author one and a third times as often as CheckItOut users, but they experienced a relative increase in satisfaction when they ordered by author less often.

The models for each system each have unique predictors as well. Only users of CheckItOut had increased satisfaction with increases in corrections from the user and book requests by title. Decreases in call duration, the total number of user utterances, and the time per user turn also increased their satisfaction. CheckItOut+ users had increased satisfaction with decreases in the number of incorrect books, and in the number of help messages from the system. In summary, the more CheckItOut users were able to request books by title and author, to be prompted more often by the system and with longer system turns (with more information), and with more corrections from the user, and the less time they spent on a call, the more satisfied they were. CheckItOut+ users also experienced more satisfaction given more system turns and longer ones (more information), but in contrast to CheckItOut, decreases in the number of incorrect books in their order, fewer help prompts from the system, and less reliance on asking for books by author also increased their satisfaction.

IX. Discussion

In comparison to CheckItOut, CheckItOut+ had a higher completion rate and task success with about the same total duration per call, and therefore less time per book or per correct book. CheckItOut+'s success depended on a small, manually constructed decision tree of three learned models to address the choice point where CheckItOut would have a non-understanding. We claim that the resulting dialogue management is more naturalistic in several regards. First, requests for the caller to repeat her utterance are a last resort, which is consistent with the ablated WOz studies cited above. Second, problematic requests are dealt with over successive utterances, and through various means, to move a partial interpretation towards a resolution. Third, the clarification actions that the dialogue manager relies on most are representative of those that the wizards relied on most. Finally, the new clarification module draws on features from many dimensions of context to represent dialogue state, including the domain knowledge represented in the backend.

Reliance on the new clarification strategies enabled CheckItOut+ to interpret noisy ASR by intelligent selection among a richer set of dialogue actions than was available to CheckItOut. Of the 18 clarification actions that wizards chose with any frequency, the most frequent was explicit confirmation of the title (Table I). This dialogue action occurs in CheckItOut+ whenever the learned VoiceSearch model determines there should be voice search, followed by a decision from the OfferResults model to offer the top ranked return. The next most frequent clarification action selected by wizards was a clarification question about the attribute that the caller was providing: *Did you ask for an author?* This dialogue action occurs in CheckItOut+ when the NonUnderstanding model decides to address the current book request rather than move on to a new one. Wizards also often relied on requests for a book attribute other than the current one. Thus, this dialogue action depends on partial information about the type of attribute the caller has tried to specify.

CheckItOut+'s clarification module executed 4.72 (+/-3.94) times per call, or a little more than once per book ordered, and most often resulted in a voice search offer. The combination of VoiceSearch and OfferResults occurred 3.76 times per call (+/- 3.35). The caller confirmed these offers as correct 1.04 times per call (+/- 0.75). The fact that the offer of a voice search candidate was correct only about one time in four was not a fatal flaw. Dialogues continued constructively, and callers had the impression that CheckItOut+ would persist in its attempts to resolve their request. The clarification module elicited a different attribute 0.92 (+/- 1.20) times per call.

To develop the machine-learned models for CheckItOut's clarification module, we performed feature selection on 163 features to represent recognition confidence, acoustic features, semantic parse features, global confidence features, voice search results and confidence scores, and dialogue history features. While each model in CheckItOut+'s new clarification module relies on relatively few features, together they encompass the full range of spoken language understanding components. The usefulness of complementary sources of information for dialogue decisions suggests that a dialogue manager could profit from more subtle information about dialogue history and dialogue structure. One promising avenue for further research would be to mine the acoustic signal for prosodic cues to discourse structure [63], or for other information apart from the words produced by the speaker. A feature representing the onset of a new discourse unit played a role in our second model, OfferResults (Table IV). It lowered the log odds of offering the top-ranked voice search return. We speculate this could be because the top-ranked return might be less reliable at the beginning of a new request, possibly due to higher likelihood of recognition errors at the onset of a new discourse unit. In a comparison of two state-of-the-art recognizers on the same corpus, Goldwater et al. [1] present results on a variety of features that degrade the performance of two state-of-the-art speech recognizers. Among the features they found, several have been shown to correlate with discourse and dialogue structure. These include discourse cue words [19], extreme prosodic characteristics, which might correspond to the increase in pitch range that characterizes new discourse units [64], and various types of disfluencies, such as filled or unfilled pauses [20].

X. Conclusion

CheckItOut+ performs well despite poor ASR because it replicates the behavior of human wizards who were asked to interpret ASR output of similarly poor quality. Wizards were asked to solve a type of problem that was novel to them. They did not know in advance which aspects of the context or what actions would be most relevant, and each wizard developed distinctive dialogue strategies. The wizards relied on a much larger set of clarification actions to choose from than the original CheckItOut. When some of these actions were incorporated in CheckItOut+, there was a clear and statistically significant improvement in performance.

Given the time constraints we imposed, wizards were often forced to end a dialogue before they could address all four books in a scenario. Nevertheless, WA and WB, the two most successful wizards, respectively identified 2.69 and 2.53

correct books per call. CheckItOut's success rate of 2.40 was higher than the 2.26 average for all wizards, but not as high as WA or WB. In contrast, CheckItOut+ achieved a success rate essentially as good as WA's by relying on a strategic combination of the distinct strategies from WA and WB.

From our two previous studies, an offline pilot and online clarification of a single book request, we expected voice search would be an important component of a dialogue management approach to achieve good task success despite noisy ASR. We also expected, however, that voice search alone would be insufficient, because addressing a user's intentions involves more than the interpretation of each user utterance. Our wizards relied on their assumptions about the users' intentions, their skill at grounding their understanding in collaboration with their conversational partner, and their background knowledge about how to achieve a task in different ways. In sum, our results indicate that a spoken dialogue system should:

- exploit access to the domain entities that users will refer (e.g., perform voice search as a way to enrich context);
- adapt commonsense knowledge about user requests to build upon partial interpretations of noisy ASR output (pursue partial understanding);
- exploit a range of dialogue acts when ASR is poor, including a variety of clarifications about the task, the user utterance, or the query return (rich set of clarification actions);
- rely on a wide range of knowledge sources to characterize the dialogue state for each utterance, including features from the speech recognizer, the semantic interpreter, voice search, the dialogue manager, and dialogue history.

Voice search can either fully resolve a noisy transcription, or result in a partial understanding when a query returns competing candidates. Questions based on a partial understanding, such as identification of certain words in an utterance, can lead to an understanding of a user's intent across multiple turns. This kind of incremental understanding of a single user intention over multiple turns is a more naturalistic way to communicate than to re-prompt users when the system has a non-understanding. The results presented here indicate a need for two kinds of generalization. First, a deeper investigation of the role of commonsense knowledge in spoken dialogue could lead to a principled mechanism for a generative model of dialogue acts. Second, a more comprehensive analysis of features to represent dialogue state across application domains and systems with distinct capabilities could lead to more general representations of dialogue state.

Our wizard corpus has a rich set of actions and states, and therefore many potential uses beyond the application presented here. Due to the mismatch between the richness of the dialogue states in our learned models and the input expected by an Olympus/RavenClaw dialogue manager, our experiment did not exploit the full range of dialogue actions selected by our wizards. In addition, we restricted our attention to instances that were adjacency pairs containing a single voice search. Wizards, however, often performed multiple searches in sequence, especially WA. We are currently preparing the corpus for public release, through the Columbia University Academic Commons, a service of the Columbia Libraries' Center for Digital Research and Scholarship.

ACKNOWLEDGMENTS

The Loqui project is funded by National Science Foundation awards IIS-0745369, IIS-0744904 and IIS-084966. We thank Joshua Gordon for his dedicated efforts to produce CheckItOut and to assist in running the baseline and evaluation data collections. We thank Alex Rudnicky, Brian Langner, and others at CMU for tutorials in Olympus/RavenClaw and implementation advice. We thank the wizards for their enthusiastic participation, and the many undergraduates at Columbia and Hunter who helped at numerous stages along the way.

REFERENCES

[1] S. Goldwater, D. Jurafsky, and C. D. Manning, "Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase speech recognition error rates," *Speech Communication*, vol. 52, no. 3, pp. 181–200, 2010.

[2] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialog strategies," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 11–23, 2000.

[3] M. A. Walker, "An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email," *Journal of Artificial Intelligence Research*, vol. 12, pp. 387–416, 2000.

[4] K. Scheffler and S. Young, "Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning," in *Human Language Technology Conference*, 2002.

[5] O. Lemon, K. Georgila, and J. Henderson, "Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the TALK TownInfo evaluation," in *Spoken Language Technology Workshop, 2006. IEEE*. IEEE, 2006, pp. 178–181.

[6] O. Pietquin and S. Renals, "ASR system modeling for automatic evaluation and optimization of dialogue systems," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2002, pp. 46–49.

[7] O. Pietquin, "Optimising spoken dialogue strategies with the reinforcement learning paradigm," in *Reinforcement Learning: Theory and Applications*, C. Weber, M. Elshaw, and N. M. Mayer, Eds., 2008.

[8] K. Samuel, R. Carberry, and K. Vijay-shanker, "Dialogue act tagging with transformation-based learning," in *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 1998, pp. 1150–1156.

[9] D. Bohus and A. I. Rudnicky, "The RavenClaw dialog management framework: Architecture and systems," *Computer Speech and Language*, vol. 23, no. 3, pp. 332–361, 2009.

[10] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing dialogue management with reinforcement learning: experiments with the NFJun system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 105–133, 2002.

[11] D. Schlangen and R. Fernández, "Speaking through a noisy channel – experiments on inducing clarification behaviour in human-human dialogue," in *8th Annual Converence of the International Speech Communication Association (INTERSPEECH 2007)*, 2007, pp. 1266–1269.

[12] M. Purver, J. Ginzburg, and P. Healey, "On the means for clarification in dialogue," in *2nd SIGdial Workshop on Discourse and Dialogue*, 2001.

[13] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: A practical framework for POMDP-based spoken dialogue management," *Computer Speech & Language*, vol. 24, no. 2, pp. 150–174, 2010.

[14] H. H. Clark and E. F. Schaefer, "Contributing to discourse," *Cognitive Science*, vol. 13, pp. 259–294, 1989.

[15] V. Yngve, "On getting a word in edgewise," in *Papers from the 6th Regional Meeting of the Chicago Linguistic Society*, 1970.

[16] R. J. Passonneau, S. Epstein, J. Gordon, and T. Ligorio, "Seeing what you said: How wizards use voice search results," in *Proceedings of the 6th Workshop on Knowledge and Reasoning in Practical Dialogue Systems, International Joint Conference of Artificial Intelligence*, 2009.

[17] R. J. Passonneau, I. Alvarado, P. Crone, and S. Jerome, "PARADISE-style evaluation of a human-human library corpus," in *Proceedings of 12th SIGdial Meeting on Dialogue and Discourse*, 2010, pp. 325–331.

[18] J. Hu, R. Passonneau, and O. Rambow, "Contrasting the interaction structure of an email and a telephone corpus: A machine learning approach to annotation of dialogue function units," in *Proceedings of the SIGDIAL 2009 Conference*, London, UK, September 2009, pp. 357–366.

[19] J. Hirschberg and D. Litman, "Empirical studies on the disambiguation of cue phrases," *Computational Linguistics*, vol. 19, no. 3, pp. 501–530, 1993.

[20] R. J. Passonneau and D. J. Litman, "Discourse segmentation by human and automated means," *Computational Linguistics*, vol. 23, no. 1, pp. 104–139, 1997.

[21] J. Williams and S. Young, "Partially observable Markov Decision Processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 231–422, 2007.

[22] S. Singh, M. Kearns, D. Litman, and M. Walker, "Reinforcement learning for spoken dialogue systems," in *Proceedings of NIPs*, 1999.

[23] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, "Agenda-based user simulation for bootstrapping a POMDP dialogue system," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Association for Computational Linguistics, 2007, pp. 149–152.

[24] E. Levin and R. Pieracinni, "A stochastic model of computer-human interaction for learning dialogue strategies," in *EUROSPEECH*, 1997.

[25] J. Henderson and O. Lemon, "Mixture model POMDPs for efficient handling of uncertainty in dialogue management," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, 2008, pp. 73–76.

[26] M. Gašic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, K. Yu, and S. Young, "Training and evaluation of the HIS-POMDP dialogue system in noise," *Proc. Ninth SIGdial, Columbus, OH*, 2008.

[27] D. Yu, Y.-C. Ju, Y.-Y. Wang, G. Zweig, and A. Acero, "Automated directory assistance system - from theory to practice," in *Interspeech*, 2007.

[28] M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, B. Strope, and G. Inc, "Deploying GOOG-411: Early lessons in data, measurement, and testing," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 5260–5263.

[29] D. Bohus, S. G. Puerto, D. Huggins-Daines, V. Keri, G. Krishna, R. Kumar, A. Raux, and S. Tomko, "Conquest—an open-source dialog system for conferences," in *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*. Rochester, New York: Association for Computational Linguistics, April 2007, pp. 9–12.

[30] S. Mann, A. Berton, and U. Ehrlich, "How to access audio files of large data bases using in-car speech dialogue systems," in *Interspeech*, 2007, pp. 138–141.

[31] G. Zweig, Y. C. Ju, P. Nguyen, D. Yu, Y.-Y. Wang, and A. Acero, "Voice-rate: A dialog system for consumer ratings," in *HLT-NAACL (Demonstrations)*, 2007, pp. 31–32.

[32] C. Lee, A. Rudnicky, and G. G. Lee, "Let's buy books: Finding ebooks using voice search," in *Spoken Language Technology Workshop (IEEE-SLT 2010)*, 2010, pp. 85–90.

[33] D. Litman, J. Hirschberg, and M. Swerts, "Characterizing and predicting corrections in spoken dialogue systems," *Computational Linguistics*, vol. 32, no. 3, pp. 417–438, 2006.

[34] D. J. Litman and S. Pan, "Empirically evaluating an adaptable spoken dialogue system," in *7th International Conference on User Modeling (UM)*, 1999, pp. 55–64.

[35] G. Skantze, "Exploring human error handling strategies: Implications for spoken dialogue systems," in *Proceedings of ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, 2003, pp. 71–76.

[36] T. Zollo, "A study of human dialogue strategies in the presence of speech recognition errors," in *Proceedings of the AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, 1999, pp. 132–139.

[37] I. Kruijff-Korbayová, N. Blaylock, C. Gerstenberger, V. Rieser, T. Becker, M. Kaisser, P. Poller, and J. Schehl, "An experiment setup for collecting data for adaptive output planning in a multimodal dialogue system," in *10th ENLG*, 2005, pp. 191–196.

[38] V. Rieser, I. Kruijff-Korbayová, and O. Lemon, "A corpus collection and annotation framework for learning multimodal clarification strategies," in *6th SIGdial Workshop on Discourse and Dialogue*, 2005, pp. 97–106.

[39] J. D. Williams and S. Young, "Characterizing task-oriented dialog using a simulated ASR channel," in *ICSLP/Interspeech*, 2004, pp. 185–188.

[40] G. Skantze, "Exploring human recovery strategies: Implications for spoken dialogue systems," *Speech Communication*, vol. 45, pp. 325–41, 2005.

[41] D. Bohus and A. I. Rudnicky, "Integrating multiple knowledge sources for utterance-level confidence annotation in the CMU Communicator spoken dialog system," Carnegie Mellon University, Tech. Rep. CMU-CS-02-190, 2002.

[42] D. Bohus, B. Langner, A. Raux, A. W. Black, M. Eskenazi, and A. Rudnicky, "Online supervised learning of non-understanding recovery policies," in *Spoken Language Technology Workshop,IEEE (SLT)*, Dec 2006, pp. 170–173.

[43] K. Komatani and T. Kawahara, "Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output," in *International Conference of Computational Linguistics (COLING)*, 2000, pp. 467–473.

[44] R. Higashinaka, K. Sudoh, and M. Nakano, "Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems," *Speech Communication*, vol. 48, no. 34, pp. 417–436, 2006.

[45] M. A. Walker, I. Langkilde-Geary, H. W. Hastie, J. Wright, and A. Gorin, "Automatically training a problematic dialogue predictor for a spoken dialogue system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 293–319, 2002.

[46] M. Raux, Antoine; Eskenazi, "A multi-layer architecture for semi-synchronous event-driven dialogue management," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2007)*, 2007.

[47] D. Bohus, "Error awareness and recovery in task-oriented spoken dialogue systems," Ph.D. dissertation, Carnegie Mellon University, 2007.

[48] W. Ward and S. Issar, "Recent improvements in the CMU spoken language understanding system," in *ARPA Human Language Technology Workshop*, 1994.

[49] R. J. Passonneau, S. L. Epstein, T. Ligorio, J. Gordon, and P. Bhutada, "Learning about voice search for spoken dialogue systems," in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2010)*, 2010, pp. 840–848.

[50] J. W. Ratcliff and D. Metzener, "Pattern matching: the Gestalt approach," 1988.

[51] S. Seneff, E. Hurley, R. Lau, C. pau, P. Schmid, and V. Zue, "Galaxy II: A reference architecture for conversational system deployment," in *Fifth International Conference on Spoken Language Systems (ICLSP-98)*, 1998.

[52] A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi, "Let's go public! taking a spoken dialog system to the real world," in *Proceedings of the Sixth Annual Conference of the International Speech Communication Association (Interspeech 2005)*, 2005.

[53] H. Sacks, E. A. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, pp. 696–735, 1974.

[54] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[55] T. Ligorio, "Feature selection for error detection and recovery in spoken dialogue systems," Ph.D. dissertation, The Graduate Center of the City University of New York, 2011.

[56] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learnig Research*, vol. 3, pp. 1157–1182, Mar 2003.

[57] A. Frank and A. Asuncion. (2010) A UCI machine learning respoitory. [Online]. Available: http://arhive.ics.uci.edu/ml

[58] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proceedings of the 11th International Conference on Machine Learning (ICML 1994)*, 1994, pp. 121–129.

[59] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, "Evaluating spoken dialogue agents with PARADISE: Two case studies," *Computer Speech and Language*, vol. 12, pp. 317–348, 1998.

[60] K. S. Hone and R. Graham, "Towards a tool for the subjective assessment of speech system interfaces (sassi)," *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems*, vol. 6, no. 3-4, pp. 287–303, 2000.

[61] M. Hajdinjak and F. Miheli, "The PARADISE evaluation framework: issues and findings," *Computational Linguistics*, vol. 32, no. 2, pp. 263–272, 2006.

[62] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, 1951.

[63] L. Lucente, J. Hirschberg, and P. Barbosa, "Intonation, discourse structure and information status in spontaneous speech," in *Experimental and Theoretical Advances in Prosody 2*, 2011.
[64] J. Hirschberg and C. Nakatani, "A prosodic analysis of discourse segments in direction-giving monologues," in *Proceedings of the Association for Computational Linguistics*, 1996, p. 286293.