Helping Agents Help Their Users Despite Imperfect Speech Recognition

Joshua B. Gordon[1], Rebecca J. Passonneau[3], Susan L. Epstein[3, 4]

[1]Department of Computer Science, Columbia University New York NY USA
[2]Center for Computational Learning Systems, Columbia University New York NY USA
[3]Hunter College and [4]The Graduate Center of The City University of New York, New York, NY USA
joshua@cs.columbia.edu, becky@cs.columbia.edu, susan.epstein@hunter.cuny.edu

Abstract

Spoken language is an important and natural way for people to communicate with computers. Nonetheless, habitable, reliable, and efficient human-machine dialogue remains difficult to achieve. This paper describes a multi-threaded semi-synchronous architecture for spoken dialogue systems. The focus here is on its utterance interpretation module. Unlike most architectures for spoken dialogue systems, this new one is designed to be robust to noisy speech recognition through earlier reliance on context, a mixture of rationales for interpretation, and fine-grained use of confidence measures. We report here on a pilot study that demonstrates its robust understanding of users' objectives, and we compare it with our earlier spoken dialogue system implemented in a traditional pipeline architecture. Substantial improvements appear at all tested levels of recognizer performance.

Introduction

For people, the most natural way to interact with an agent is to speak. Dialogue between a person and a machine, however, does not allow people to speak naturally. Task-oriented systems typically prompt users for short utterances, and prevent the natural give-and-take of human dialogue. A spoken dialogue system (*SDS*) works best when its automated speech recognition (*ASR*) is very accurate. The thesis of our work is that an SDS architecture should robustly accommodate noisy ASR, and should degrade gracefully as recognition errors increase. Thus the work described here helps an SDS help the user. The principal result of this paper is substantial performance improvement when a mixture of rationales is used to interpret what the speaker wants.

An SDS receives a continuous stream of acoustic data that ASR translates into discrete linguistic units, with its final output represented orthographically. Speech recognizers rely on pre-existing acoustic models to relate acoustic energy to speech sounds, and on domain-specific language models to predict which sequences of speech sounds correspond to known words. ASR output that seems unintelligible without context (e.g., "sooner sheep most die") is readily resolved in the context of candidate database matches (e.g., the book titles *Soon She Must Die*, *Why Someone Had to Die*, and *The Messenger Must Die*.)

ASR has improved dramatically for single-party applications, such as mobile search and the transcription of broadcast news. Its accuracy in dialogue, however, lags substantially, both in the transcription of interactions (Renals, Hain and Bourlard, 2008) and in real-time human-computer SDSs (Leuski et al., 2006). The relatively poor ASR quality in SDSs intended for many users is partially responsible for the frustrating telephone conversations people often experience with them. However, people who play the role of an SDS during experiments interpret noisy ASR much better than machines do. Given real or simulated ASR output instead of the user's speech during dialogue, human subjects engage in problem solving about the task, the ASR errors, or both (Rieser, Kruijff-Korbayová and Lemon, 2005; Skantze, 2003; Williams and Young, 2004; Zollo, 1999).

FORRSooth is an SDS architecture designed to incorporate multiple strategies for utterance interpretation, including greater reliance on the task context. The experiment described here demonstrates FORRSooth's robust utterance interpretation despite recognition error, and its superiority to an existing SDS implemented in a traditional SDS architecture. The next section of this paper describes related work. Subsequent sections highlight the challenges of speech recognition for human-computer dialogue, and show how skilled people successfully interpret noisy recognition hypotheses in a dialogue setting. The paper then describes FORRSooth and reports on the pilot experiment, which compares results against our earlier SDS.

Related Work

Dialogue systems must determine the semantic intent of user utterances. Although statistical methods are particularly robust for interpreting noisy ASR (Gordon and Passonneau, 2010), they are most useful in application domains where only the broad intent of an utterance is required (e.g., the virtual humans project (Leuski and Traum, 2010)). Deeper methods (e.g., semantic parsing (Ward and Issar, 1994)) extract finer-grained concepts, but are less noise tolerant. In two-stage Natural Language Understanding *(NLU)*, statistical methods may process ASR output before deeper methods, such as semantic parsing. The AT&T

Spoken Language Understanding System (Gupta et al., 2006) explicitly separates the overall intent of an utterance from the specific concepts it contains. For an overview of existing techniques, see (Bangalore, 2006).

Task-oriented dialogue systems must often search a database with terms extracted from noisy ASR. To narrow the possible interpretations before search, decision trees have been used after a shallow semantic interpretation phase that classifies the utterance by query type or specific query slot (Komatani et al., 2005). Recent approaches use *voice search* to query a database directly with noisy ASR (Passonneau et al., 2010; Wang et al., 2008).

While there are many techniques for semantic interpretation in the literature, there is no single preferred approach. Dialogue systems rarely reuse and combine existing linguistic resources and NLU approaches. FORRSooth relies on "multiple processes for interpreting utterances (e.g., structured parsing versus statistical techniques)" as in (Lemon, 2003), but uses a wider range of resources.

Architectures for human-computer dialogue have traditionally been *pipeline*-based (Raux et al., 2005). The understanding difficulties that inspired the work reported here arose in *CheckItOut*, our SDS based on the Olympus/Ravenclaw architecture (Bohus and Rudnicky, 2009), which has been the basis for a dozen SDSs. Apart from the Apollo interaction manager (Raux and Eskenazi, 2007), data flows through CheckItOut in a pipeline. It uses the *PocketSphinx* speech recognizer (Huggins-Daines et al., 2006), for which we adapted the freely available Wall Street Journal acoustic models with approximately 10 hours of spontaneous speech. Then *Phoenix*, a robust context-free grammar semantic parser processes ASR hypotheses to identify NLU concepts (Ward and Issar, 1994). Next, *Helios* (Bohus and Rudnicky, 2002) an utterance-level confidence annotator, identifies the best parse and passes it to the RavenClaw dialogue manager with a confidence score (Bohus and Rudnicky, 2009). RavenClaw then chooses the next system action or utterance.

Alternatively, an SDS can be asynchronous (Allen, Ferguson and Stent, 2001). In the past decade, asynchronous architectures have addressed incremental processing (Schlangen and Skantze, 2009), turn management (Raux and Eskenazi, 2007), a shared communication channel (Skantze and Gustafson, 2009), or a combination of them (Blaylock, Allen and Ferguson, 2002; Paek and Horvitz, 2000). FORRSooth, our new architecture, is asynchronous.

Speech Recognition Challenges in SDS

Continuous speech recognition of spontaneous speech over a large vocabulary and by diverse speakers presents a major challenge, particularly in the context of dialogue. Background noise or poor phone transmission quality also worsens recognition performance. These challenges are evident in the higher word error rate (*WER*) that deployed SDSs have compared with their WER during laboratory testing. For example, the WER reported by Carnegie Mellon University's *Let's Go Public!* went from 17% under controlled conditions to 68% in the field (Raux et al., 2005). Above a fixed, relatively low WER threshold, SDS performance typically degrades sharply (Leuski et al., 2006). Our work seeks to develop methods that support robust utterance interpretation for large-vocabulary SDSs.

Spontaneous dialogue is difficult for speech recognizers because it exhibits utterance planning in progress, not the finished product of a prior plan (Ochs, 1979). A single utterance can be started, stopped, or resumed for completion, repair, or complete reformulation. Performance phenomena such as halting speech, rephrasing, and pause fillers (*um um, er*), are frequent, and decrease recognition accuracy. Conversants indicate that they are listening with *backchannels*, short, low-energy utterances (*okay*) or pause fillers (*umhm*). Backchannel words are difficult for recognizers because they are brief and less clearly articulated. Moreover, the way human conversational partners take turns presents difficulties for recognizers. People frequently interrupt and speak over one another. Most recognizers ignore performance phenomena, and most SDSs lack explicit models of turn taking. Our long term goal is an architecture that encourages an SDS and its users to collaborate in the way that people do (Clark and Schaefer, 1989), so that they better understand one another. This requires more robust utterance interpretation.

SDS Error-Handling Strategies

When humans converse with each other, they engage in *grounding* to establish and convey the degree of mutual understanding. This consists of subtle collaborative behaviors to demonstrate continuously how well they understand each other (Clark and Schaefer, 1989). Because interpretation of imperfect speech recognition output is difficult, grounding in an SDS must often fall back on *error-handling strategies* to resolve failures in its understanding. These strategies are far from human-like. Theoretically, when human dialogue participants are well grounded, grounding behavior at any one turn in the dialogue involves little effort — a nod of the head and taking the next turn with no interruption can suffice. When there is potential confusion, they devote more effort to grounding — they may request and receive implicit or explicit confirmation. We hypothesize that robust utterance interpretation is essential if an SDS is to engage in more human-like grounding behavior.

An SDS resorts to error-handling behavior when it lacks sufficient understanding of the user's objectives, and cannot otherwise advance the dialogue. SDS error-handling strategies for *non-understanding*, where the ASR is uninterpretable, include prompting the user to repeat or rephrase her last utterance. A more pernicious consequence of noisy ASR is *misunderstanding*, where the SDS misinterprets the user's objective entirely. To avoid misunderstandings of key information, an SDS will ask for explicit confirmation ("Did you say pick up the cup?") or will confirm it implicitly ("Okay, the cup, and where would you like me to put it?"). Explicit confirmation brings the user no closer to her goals, and makes dialogue tedious. Implicit confirmation is more likely to advance the conversational goals, and is less tedious. Human-human dialogue offers a much richer range of behaviors that present evidence of at least partial understanding.

We hypothesize that SDSs should aim for high-confidence interpretation of noisy ASR in support of dialogue strategies that advance the dialogue, and should avoid explicit confirmation and requests for repetition when possible. Previous work has shown that people presented with simulated ASR and asked to simulate a dialogue system can compensate for recognition errors and address user objectives despite moderately high WER (Williams and Young, 2004). Given that people understand speech far better than machines do, we further hypothesize that good strategies for an SDS can be learned from the ways people attempt to resolve noisy ASR.

Human Dialogue Strategies for Noisy ASR

Our domain of investigation is the Andrew Heiskell Braille and Talking Book Library, a branch of The New York Public Library and part of The Library of Congress. Heiskell's patrons order their books by telephone, during conversation with a librarian.

Wizard ablation captures the strategies people use to interpret noisy ASR during dialogue (Levin and Passonneau, 2006). In earlier work, we conducted novel experiments where a human (the *wizard*) was presented with real ASR output and database query results to provide context for interpreting the ASR. In the first experiment (Passonneau, Epstein and Gordon, 2009), undergraduates were presented off-line with noisy ASR (WER = 0.69) from 50 book titles spoken by a single individual, along with a text file of Heiskell's 71,166 titles. Subjects were asked to match each ASR string to a title without any time limit, a simulation of voice search. Although only 9% of the titles had no ASR errors, the subjects' accuracy ranged from 67.7% to 71.7%.

This motivated a large-scale experiment to collect real-time data from a single-turn exchange for 4200 book title requests with poor ASR (WER = 0.71). From among the voice search returns, seven wizard subjects could all identify correct matches reasonably well. Only two, however, could recognize when there was no match among the returns (Passonneau et al., 2010; Ligorio et al., 2010a). We learned decision trees for these more proficient wizards using runtime system features, and features that represented what the wizards saw on a customized graphical user interface. The learned trees included speech recognition metrics (e.g., speech rate and acoustic model fit). Although most SDSs rely only on a single confidence score to trigger error-handling, these results suggest that an SDS architecture should incorporate more fine-grained measures of confidence (Passonneau et al., 2010).

Our most recent wizard experiment collected 913 full dialogues between 6 wizards and 10 callers, with up to 4 book requests by author, title, or catalogue number (Ligorio et al., 2010b). The databases included all 5028 active patrons, 71,166 titles and 28,031 authors. Our statistical language model was built from a pseudo-corpus of domain utterances plus 3000 randomly selected book titles, their author names, and 50 patron names. From our initial data analysis, it is clear that two of the wizards succeeded more often at identifying the requested books, and that they relied on quite different strategies to do so. One focused more on the task and rarely confirmed information explicitly, while the other focused more on the communication and often confirmed explicitly (Ligorio et al., 2010b).

FORRSooth

FORRSooth, our new architecture for task-oriented human-computer dialogue, is intended to interact effectively "without the luxury of perfect [ASR]" (Paek and Horvitz, 2000). The high redundancy in human language makes effective human-human communication possible without perfect comprehension of the audio signal. FORRSooth exploits this redundancy for human-machine communication. Here we briefly introduce the architecture, and focus on details of the interpretation service.

FORRSooth is based on *FORR* (FOr the Right Reasons), an architecture for learning and problem solving (Epstein, 1994). FORR uses diverse (often conflicting) rationales to make decisions. It is intended for domains where multiple rationales and sequences of decisions are used to solve problems. Implementations have proved robust in game learning, simulated pathfinding, and constraint solving.

FORR relies on an adaptive, hierarchical mixture of resource-bounded procedures called *Advisors*. Each Advisor embodies a decision rationale. Advisors' opinions (*comments*) are combined to arrive at a decision. Each comment pairs an action with a *strength* that indicates some degree of support for or opposition to that action. An Advisor may make multiple comments at once, and may base them upon descriptives. A *descriptive* is a shared data structure that is computed on demand and refreshed only when required. For each decision, FORR consults three tiers of Advisors one at a time until one tier decides. A learned, weighted majority produces decisions in tier 3.

FORRSooth is a parallelized version of the mixture of experts embodied in FORR. FORRSooth models task-oriented dialogue with six FORR-based *services* that operate simultaneously: Interaction, Interpretation, Satisfaction, Grounding, Generation, and Discourse. (We do not expect to exploit the Generation service fully. It converts a conceptual representation of a system's response into words. Instead, we will rely largely on template generation similar to that in CheckItOut.) These services interpret user utterances with respect to system expectations, manage the conversational floor, and consider competing hypotheses, partial understandings, and alternative courses of action simultaneously. All services have access to the same data, represented by descriptives.

FORRSooth's Interpretation service will provide a foundation for rich grounding behavior, so that an SDS can respond to each new user utterance with enough understanding to advance the dialogue. Richer grounding strategies depend on richer utterance interpretation. Therefore development of FORRSooth has begun with the Interpretation service, which relies on a mixture of shallow and deep approaches to utterance interpretation. Voice search is a shallow resource. In earlier work on CheckItOut, we improved performance with a more syntactic analysis of book titles. This used Phoenix context-free grammar productions that were automatically mapped from MICA parses (Bangalore et al., 2009) of the book title database. The pilot described here used only tier-3 Advisors, which are ideally suited to accommodate a growing collection of heuristic interpretation strategies.

FORRSooth represents its expectations about what the user will say next as *agreements* with *targets*, variables that must be bound to achieve the task. For example, a patron identity agreement has a library patron identity target. Advisors process ASR hypotheses to produce comments about the content and intent of an utterance. Comments are affixed to a *target graph* that represents dependencies among agreements and targets, such as the fact that patrons have names. The target graph facilitates reasoning over partial understandings. It collects competing hypotheses for target nodes, and represents how hypotheses can be combined for database search with partial matching. In this way the system can postulate a user's identity from its hypotheses. Multiple hypotheses for a given target can also be merged and strengthened.

Experimental Design and Results

Wizard studies demonstrate that clever strategies can successfully interpret noisy ASR. Our initial focus for the Interpretation service in FORRSooth is on robust interpretation of responses to system prompts followed by a database query. We work with a partial interpretation of the full utterance using a suite of NLU resources. FORRSooth's target graph represents prior knowledge about the information it has just requested. (FORRSooth will eventually reason deeply about targets for a wide range of activities.)

This experiment studies the robustness of the Interpretation service for author names under worst-case realistic ASR performance. We compare FORRSooth's performance with a simulation of the language-interpretation phase of CheckItOut. The pilot used the PocketSphinx speech recognizer, CheckItOut's acoustic models, the full database of 5,000 active patrons, a random selection of 4000 titles and their 2306 authors for constructing the book title grammar, and the same type of statistical language model described earlier.

To generate input, one male and one female experimenter each read the same list of 100 randomly selected patron names into the speech recognizer. The Interpretation Advisors processed the resulting ASR output to propose hypotheses for the patron name. Our pilot presented three kinds of information from ASR results to Interpretation Advisors: the highest ranked recognition hypothesis (from an *n*-best list), utterance confidence, and word-level recognition confidence. The pilot used 10 Interpretation Advisors to bind values to the target graph for the patron's identity: 8 to produce hypotheses and 2 to embellish them.

Interpretation Advisors used a suite of NLU resources. The simplest Advisors relied on voice search to bind values for the patron's name to the target graph. These Advisors retrieved database records where the similarity score between the recognition hypothesis and the name field exceeded a threshold. Similarity was computed by Ratcliff/Obershelp (*R/O*) pattern matching: the ratio of the number of matching characters in two strings to their combined length (Ratcliff and Metzener, 1988). (For example, R/O is 0.61 for ROLL DWELL against *Robert Lowell.)* Comment strength was a function of word-level recognition confidence, similarity score, and a metric on the relative position and edit distance between words in the recognition hypothesis and the database match. Some Interpretation Advisors used SoundEx and DoubleMetaphone similarity metrics to perform phonetic partial matching.

Meanwhile, the parsing Advisors used Phoenix to construct comments whose strength was a function of overall recognition confidence, word level confidence, and the proportion of words not consumed by the parse. One parsing Advisor used Phoenix directly; another used Phoenix but included the low-confidence words that are not parsed by default, and reported lower comment strength. Yet another parsing Advisor re-ranked Phoenix parses with the Helios confidence annotator, and reported the Helios score (which includes ASR confidence) as its comment strength.

One Advisor duplicated the tools available to our wizards; it combined voice search and parsing, and refined the search query after several passes with fine-grained confidence scoring that reflected the lexical and phonetic similarity of the names. This Advisor performed multiple-partial-matching database queries using different segments of the ASR and proposed its best hypotheses selected from the query returns. Another Advisor relied on a learned classifier before any database query to remove from the ASR words likely to correspond to noise. Finally, two Advisors combined first and last name concepts into full names, and retrieved corresponding database entities.

A separate run on the same input used one Advisor to simulate CheckItOut's entire language interpretation pipeline. That Advisor invoked Phoenix, Helios, and, after the parse, an R/O database query with the top-ranked parse. Note that because Phoenix skips words it cannot parse, a database query with parse results is not pure voice search.

For the ASR, the WER (computed as average normal-

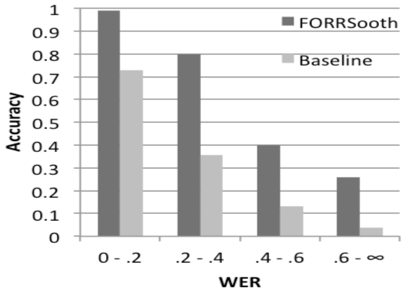

Figure 1: Accuracy at varying WERs.

Table 1: Examples of patron name interpretation with recognition and interpretation confidence.

	Name	ASR with confidence	Rank	FORRSooth with Interp. score	CheckItOut with Helios conf.
1P	Edward Martinez	edward martson 0.33	1 in 2	Edward Martinez 1.48	Edward Emerson 0.82
2P	Helen Harris	.hellen. .heiress. 0.22	1 in 6	Helen Harris 2.65	NULL NULL
3N	Mildred Bailey	mildred bouie 0.67	1 in 5	Mildred Bouie 4.61	Mildred Bouie 1.00
			2 in 5	Mildred Bailey 2.32	N/A N/A
4N	David Davis	david did 0.67	1 in 7	David Said 2.60	N/A N/A
			2 in 7	David Davis 2.44	NULL NULL

ized Levenshtein distance for words) was 0.44 (0.48 for the female speaker, 0.39 for the male). The character-level error rate was much lower, at 0.23 (0.26 female, 0.20 male)—evidence that fuzzy matches of ASR output to known words could be constructive. The accuracy of FORRSooth's top-ranked hypothesis was 0.78 (0.73 female; 0.83 male), and the accuracy of its top two hypotheses was 0.82 (0.77 female, 0.87 male). Without the 78 cases of perfect recognition (out of 200 total), WER was 0.72 (0.77 female, 0.66 male), and FORRSooth accuracy was 0.64 (0.59 female, 0.71 male).

Figure 1 compares the performance of the FORRSooth suite of Advisors with the CheckItOut simulation at 4 levels of WER. Most utterances had low WER, and the rest were roughly evenly distributed over the remaining three levels: $0 < \text{WER} \leq 0.2$ (N=119), $0.2 < \text{WER} \leq 0.4$ (N=35), $0.4 < \text{WER} \leq 0.6$ (N=30), $0.6 < \text{WER} \leq \infty$ (N=27). At all WER levels, the FORRSooth pilot outperformed CheckItOut by a large margin.

Table 1 illustrates a variety of ways in which poor ASR challenges an SDS. It shows two positive (*P*) examples where Interpretation successfully identified the patron name, and two negative (*N*) examples. Rank is the position in FORRSooth's *n*-best list of hypotheses. The Interpretation score is the weighted combination of comment strengths for the Advisors that supported the hypothesis. CheckItOut's interpretation is in the last column along with the Helios confidence score.

In example 1P, multiple rationales compensate for recognition error. FORRSooth's voice search Advisors determined that both *Edward Martinez* and *Edward Emerson* were similar to the ASR. (*Edward Emerson* was slightly closer.) When the phonetic Advisors voted in favor of *Edward Martinez*, however, FORRSooth reached the correct interpretation. In contrast, CheckItOut produced a high-confidence incorrect hypothesis based solely on R/O score. In 2P, FORRSooth's confidence was high despite low ASR confidence, because of high phonetic similarity. CheckItOut does not parse words with low confidence (indicated by a period before and after each unconfident word). This non-understanding would drive CheckItOut to ask the caller's name again. In 3N, FORRSooth and CheckItOut prefer the same (incorrect) hypothesis, but CheckItOut has no alternatives. In a full FORRSooth system, Grounding will respond to the user based on the full target graph. Given the matching first names of the top two hypotheses, and the phonetic similarity of the two last names (*Bouie* and *Bailey*), a grounding Advisor would confirm the first name implicitly and disambiguate the last name with a simple Yes/No question: "Hi, Mildred, was that Bouie?" In 4N, FORRSooth has two nearly tied hypotheses with identical first names, but dissimilar last names. This would lead to a full re-prompt for the last name ("Okay, David, and your last name again please?"). In both cases, CheckItOut would have asked the caller's name again.

FORR learns weights for tier-3 Advisors (Epstein and Petrovic, 2006). In the absence of training data, we manually determined weights for this experiment based on rough a priori estimates of Advisors' reliabilities. Our results indicate the effectiveness of using tier-3 Advisors to combine different sources of NLU and interpretation strategies, even with estimated weights. Our current work includes learned weights for Advisors, and the development of grounding Advisors to exploit partial interpretations.

Conclusion

This paper presents FORRSooth, a new SDS architecture that supports robust utterance interpretation. The pipeline architecture of our existing SDS, CheckItOut, limits its ability to interpret utterances. CheckItOut must map ASR output to concepts before it invokes the dialogue manager to initiate database queries. To avoid misunderstandings, CheckItOut rejects ASR with poor recognition confidence before semantic interpretation, and produces one confidence score on a single semantic interpretation.

FORRSooth replaces the conventional SDS pipeline with a set of utterance interpretation strategies, and provides multiple sources of information for subsequent dialogue management decisions. It also produces a graph of interpretation hypotheses, and computes finer-grained confidence results that incorporate a wide variety of interpretation resources. Our results illustrate the merits of this approach for the interpretation of noisy ASR.

The ease with which we separately reconstructed CheckItOut's NLU pipeline within a single FORRSooth Advisor demonstrates the new architecture's versatility. Our pilot results demonstrate the potential for more nuanced grounding behavior from an SDS, and the benefits of employing a mixture of strategies to help a system understand its user better.

Acknowledgements

This research was supported in part by the National Science Foundation under awards IIS-084966, IIS-0745369, and IIS-0744904.

References

Allen, J.; Ferguson, G.; and Stent, A. 2001. An architecture for more realistic conversational systems. In Proc. of the 6th International Conference on Intelligent User Interfaces, pp. 1-8.

Bangalore, S. 2006. Editorial Introduction to the Special Issue on Spoken Language Understanding in Conversational Systems. *Speech Communication* 48: 233–238.

Bangalore, S.; Boulllier, P.; Nasr, A.; Rambow, O.; and Sagot, B. 2009. MICA: a probabilistic dependency parser based on tree insertion grammars. In Proc. of the NAACL HLT 2009 Companion Volume: Short Papers, pp. 185-188.

Blaylock, N.; Allen, J.; and Ferguson, G. 2002. Synchronization in an asynchronous agent-based architecture for dialogue systems. In Proc. of SIGDIAL 2002, pp. 1-10.

Bohus, D. and Rudnicky, A. 2002. Integrating multiple knowledge sources for utterance-level confidence annotation in the CMU Communicator spoken dialog system. Technical. report CS-190, Carnegie Mellon University.

Bohus, D. and Rudnicky, A. I. 2009. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language* 23(3): 332-361.

Clark, H. H. and Schaefer, E. F. 1989. Contributing to discourse. *Cognitive Science* 13(2): *259-294*.

Epstein, S. L. 1994. For the Right Reasons: The FORR Architecture for Learning in a Skill Domain. *Cognitive Science* 18(3): 479-511.

Epstein, S. L. and Petrovic, S. 2006. Relative Support Weight Learning for Constraint Solving. In Proc. of the AAAI Workshop on Learning for Search, pp. 115-122.

Gordon, J. and Passonneau, R. J. 2010. An Evaluation Framework for Natural Language Understanding in Spoken Dialogue Systems. In Proc. of LREC 2010, pp. 72-77.

Gupta, N.; Tur, G.; Hakkani-Tur, D.; Bangalore, S.; Riccardi, G.; and Gilbert, M. 2006. The AT&T spoken language understanding system. *IEEE Transactions on Audio, Speech, and Language Processing,* 14(1): 213-222.

Huggins-Daines, D.; Kumar, M.; Chan, A.; Black, A. W.; Ravishankar, M.; and Rudnicky, A. 2006. PocketSphinx: A free, real-time continuous speech recognition system for hand-held devices. In Proc. of ICASSP 2006, pp. 185-189.

Komatani, K., Kanda, N.; Ogata, T.; and Okuno, H. G. 2005. Contextual constraints based on dialogue models in database search task for spoken dialogue systems. In Proc. of Interspeech 2005, pp. 877-880.

Lemon, O. 2003. Managing dialogue interaction: A multi-layered approach. *SIGDIAL 2003,* pp. 168-177.

Leuski, A.; Patel, R.; Traum, D.; and Kennedy, B. 2006. Building effective question answering characters. In Proc. of SIGDIAL 2006, pp. 18-27.

Leuski, A. and Traum, D. 2010. Practical Language Processing for Virtual Humans. In Proc. of IAAI-10.

Levin, E. and Passonneau, R. J. 2006. A WOz Variant with Contrastive Conditions. In Proc. of Interspeech 2006 Satelite Workshop: Dialogue on Dialogues.

Ligorio, T.; Epstein, S. L.; Passonneau, R. J.; and Gordon, J. B. 2010a. What You Did and Didn't Mean: Noise, Context, and Human Skill. In Proc. of Cognitive Science - 2010.

Ligorio, T.; Epstein S. L.; and Passonneau, R. J. 2010b. Wizards' Dialogue Strategies to Handle Noisy Speech Recognition. In Proc. of the IEEE SLT 2010. Berkeley CA.

Ochs, E. 1979. Planned and unplanned discourse. In *Syntax and semantics,* vol. 12: Discourse and syntax, ed. by T. Givon. New York: Academic Press.

Paek, T. and Horvitz, E. 2000. Conversation as action under uncertainty. In Proc. of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pp. 455-464.

Passonneau, R. J.; Epstein, S. L.; and Gordon, J. B. 2009. Help Me Understand You: Addressing the Speech Recognition Bottleneck. In Proc. of the AAAI Spring Symposium on Agents that Learn from Human Teachers.

Passonneau, R. J., Epstein, S. L.; Ligorio, T.; Gordon, J.; and Bhutada, P. 2010. Learning About Voice Search for Spoken Dialogue Systems. In Proc. of *NAACL HLT 2010*, pp. 840-848.

Ratcliff, J. W. and Metzener, D. 1988. Pattern Matching: The Gestalt Approach, *Dr. Dobb's Journal*, p.46.

Raux, A. and Eskenazi, M. 2007. A multi-layer architecture for semi-synchronous event-driven dialogue management. In Proc. of IEEE ASRU 2007*,* pp. 514-519.

Raux, A., Langner, B. A.; Black, W.; and Eskenazi, M. 2005. Let's Go Public! Taking a spoken dialog system to the real world. In Proc. of Interspeech 2005.

Renals, S., Hain, T.; and Bourlard, H. 2008. Interpretation of multiparty meetings: The AMI and AMIDA projects. In Proc. of Hands-Free Speech Communication and Microphone Arrays (HSCMA-2008), pp. 115-118.

Rieser, V.; Kruijff-Korbayová, I.; and Lemon, O. 2005. A corpus collection and annotation framework for learning multimodal clarification strategies. In Proc. of SIGDIAL 2005, pp. 97-106.

Schlangen, D. and Skantze, G. 2009. A general, abstract model of incremental dialogue processing. In Proc. of EACL 2009, pp. 710-718.

Skantze, G. 2003. Exploring human error handling strategies: Implications for Spoken Dialogue Systems. *Speech Communication* 45(3): 325-341.

Skantze, G. and Gustafson, J. 2009. Attention and interaction control in a human-human-computer dialogue setting. In Proc. of SIGDIAL 2009, pp. 310-313.

Wang, Y.-Y.; Yu, D.; Ju, Y.-C.; and Acero, A. 2008. An introduction to voice search. *IEEE Signal Processing Magazine,* 25(3): 28-38.

Ward, W. and Issar, S. 1994. Recent Improvements in the CMU Spoken Language Understanding System. In Proc. of the ARPA Workshop on Human Language Technology, pp. 213-216.

Williams, J. D. and Young, S. 2004. Characterising Task-oriented Dialog using a Simulated ASR Channel. In Proc. of ICSLP 2004, pp. 185-188.

Zollo, T. 1999. A study of human dialogue strategies in the presence of speech recognition errors. In Proc. of the AAAI Fall Symposium on Psychological Models of Communication, pp. 132-139.