Invited Talk

Shared Experiences, Shared Representations, and the Implications for Applied Natural Language Processing

Amanda J. Stent
AT&T Labs – Research
180 Park Ave., Building 103
Florham Park, NJ 07932

Abstract

When people interact with language-producing agents (other people or computers), they assume that the shared experience leads to shared representations – of the world, the interaction, and the language used in the interaction. This phenomenon occurs even during interaction with systems that give no evidence of building shared representations. The absence of shared representations leads to errors and delays; alternatively, even simple shared representations can lead to reduced error rates and more efficient interaction. In this talk, we present three case studies: a mobile local business search application that builds no interaction representations; a telephone-based recommendation and review system that builds limited representations of the shared language in the interaction; and computer models of coreference that use shared representations to permit both coreference resolution and referring expression generation. We lay out a range of possibilities for shared representations, show that they can be built incrementally as an interaction progresses, and point to possibilities for future work in probabilistic shared representations for interactive systems.

Introduction

It is now well-known that people conversing together exhibit convergent behaviors that make the interaction more efficient and successful. For example, controlled experiments have shown that conversational partners converge in their choice of referring expressions (Brennan and Clark 1996; Garrod and Anderson 1987), words and syntactic options (Bock 1986b; 1986a; Branigan, Pickering, and Cleland 2000; Reitter, Keller, and Moore 2006), prosody and speaking rate (Jungers, Speer, and Palmer 2002). These findings have been confirmed through corpus studies (Dubey, Sturt, and Keller 2006; Reitter, Keller, and Moore 2006; Reitter and Keller 2007; Stenchikova and Stent 2007).

Of course, spoken dialog with a computer partner is different from dialog with a human partner (Pierraccini and Huerta 2005). Nonetheless, people exhibit the same convergent behaviors in human-computer dialog contexts that they do when talking with other humans. For example, they converge on choice of words and syntactic structures (Brennan 1996; Levow 2003; Parent and Eskenazi 2010). Furthermore, research has found that lexical and syntactic alignment in dialog are correlated with perceived task success (Reitter and Moore 2007; Nenkova, Gravano, and Hirschberg 2008). If researchers can build on the adaptive behaviors people naturally exhibit, they can potentially improve the performance of understanding components of dialog systems (e.g. speech recognizers, parsers, pronoun resolution modules). In addition, if dialog systems can model adaptive behaviors themselves, natural language generation for dialog systems will improve.

There is currently a key issue preventing modeling of adaptive language behaviors in dialog systems: current deployed spoken dialog systems use restricted internal representations, consisting mainly of the dialog state the system is in and/or the system-related concepts the user referred to in current and previous utterances (Williams and Young 2007). Since systems do not build shared linguistic representations, it is almost impossible for them to reproduce the adaptive behaviors humans exhibit in conversation. In this paper, we argue that the lack of these shared representations: (a) can and does have a negative impact on dialog systems performance; (b) can and does constrain the types of tasks attempted by deployed dialog systems. However, relatively lightweight shared representations (not requiring much processing) can be incorporated into dialog systems; we will give several examples of how simple shared representations can lead to richer dialog interactions. Finally, we will identify architectures for dialog systems that permit construction and use of rich shared representations.

Shared Representations for Dialog

Cognitive psychologists disagree on the mechanisms by which shared representations are built and manipulated in dialog: for example, the extent to which apparent adaptations are egocentric and recency-based vs. partner-directed (e.g.(Pickering and Garrod 2004; Brennan, Galati, and Kuhlen 2010; Pickering and Garrod 2006)). In fact, multiple processes may give rise to the appearance of convergent behavior (Bard and others 2000; Lieberman 1963; Stenchikova and Stent 2007). One is simple *alignment* (Pickering and Garrod 2004; 2006): when the understanding and production systems process language, the seman-

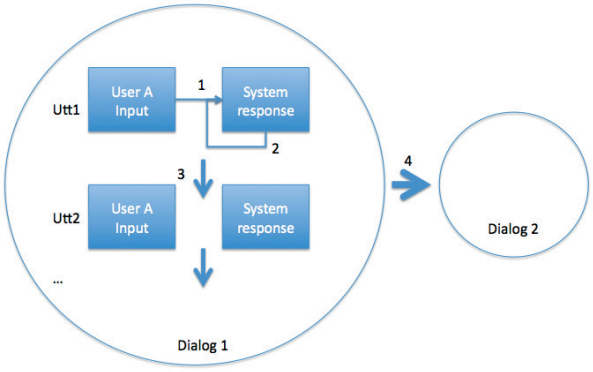

Figure 1: Possibilities for shared representations in dialog. Arrows (1) and (2) indicate representations to support alignment. Arrow (3) indicates representations to support partner adaptation. Arrow (4) indicates representations to support long-term partner adaptation (linguistic user models).

tic, lexical, syntactic and other elements constructed become more accessible, or primed, for use in the near future. Another is *partner adaptation* (Brennan and Clark 1996; Brennan, Galati, and Kuhlen 2010): a speaker may make production choices (e.g. in articulation, choice of words and syntactic structures, realization of gestures) in order to facilitate the listener's understanding.

A computational perspective is not limited to the cognitive mechanisms employed by humans. From the perspective of a dialog system designer, in order to permit rapid understanding and production of convergent behaviors, the shared representations depicted in Figure 1 must be in place:

1. Shared lexical, syntactic and semantic representations between the system's own understanding and production systems facilitate reuse of words and syntactic structures.
2. Shared discourse representations between the system's model of the world and its model of the user's knowledge facilitate referring expression resolution and production.
3. Sharing of these representations across dialogs permits extended adaptive behaviors and modeling of the language and interaction styles of particular users.

Note that 'shared' here does not mean the system exposes its internal representations to the user, but rather that it shares its internal model of the user's interactions with its internal model of its own state, over time.

While this seems like a large representational burden, most of these representations already exist in dialog systems – hidden away in individual modules. However, several dialog system architectures currently in use are capable of exposing and manipulating these representations. We will review two of these towards the end of this paper.

Lack of Shared Representations Limit Power

In this section we will briefly present two spoken language interfaces with minimal shared representations. We will show how the lack of shared representations limits important and useful capabilities in these systems.

Category	Query pair type	Count
Invalid	Invalid query	319
	Incomplete/Empty	22
Not related	Not related	731
Likely ASR error	Exact repetition	551
Paraphrases	Query expansions	154
	Query abbreviations	153
	Location modification	19
	Query term spelled out	11
	Other paraphrases	130
Semantically related	Instance/Category	59
	Listing/Container	18
	Product/Listing	12
	Other	210

Table 1: Analysis of Speak4It query pairs

Speak4It

The Speak4It dialog system (www.speak4it.com) is a speech-driven local search system for smart phones. Spoken interactions with the system are designed to be one-exchange only: the user specifies query terms (the name or business category of a business) and optionally location terms (city and state in which to search). Example Speak4It queries include *Starbucks* and *hotel in Los Angeles, California*. The system returns a set of matching business listings, which are displayed on a map or in a list. The user interacts with the resulting listings on a graphical user interface.

Speak4It builds almost no shared linguistic representations: the system does not track the words or concepts of the user across exchanges, and does not even provide error handling other than to indicate whether a failure occurred in the recognition or search modules of the system. The entire internal representation is the set of listing results, which is exposed to the user in the GUI.

Nonetheless, users interacting with Speak4It sometimes assume the system is tracking what they have said. For example, we analyzed 2389 Speak4It *query pairs*: pairs of queries from the same user separated by no more than a 60 second interval in time. These query pairs were extracted from the first three months of Speak4It logs (excluding queries from Speak4It developers) and manually labeled. Of these 2389 query pairs, 766 query pairs (32%) involved related queries. 467 were paraphrases of some kind, while another 299 were semantically related. For query pairs involving location modification and query abbreviation, shared representations across searches by single users could fill in missing values. For query pairs involving 'other' semantic relationships (e.g. *Burger King* followed by *McDonalds*, the addition of some taxonomic information from the listing database would permit intelligent query expansion (e.g. to *fast food*) likely to be useful to the user.

In Speak4It there is only one language action (request listings), and it requires only two pieces of information (location and listing name/category). Storing the words and concepts associated with these two pieces of user input would permit many interactions not currently possible, including: automatic propagation of user-provided location across multiple queries (e.g. *Starbucks in Houston* followed by *hotels*

S	*Were* the *exams*: too hard, hard, about right, easy, or too easy?
U	too hard
S	Feel free to take as long as you like. Why do you think the *exams were too hard*?
U	[um] they were too hard because he would just ask very detailed questions to them and [um] most of us weren't prepared to answer such detailed questions we just studied the basics and didn't realizae that we had to go into such depth

Figure 2: Example subdialog for course aspect *exams*. Words and phrases from the system's shared representations appear in italics.

near there); automatic propagation of user corrections and additional user constraints on search (e.g. *fish* followed by *tackle* or *restaurant*); use of anaphora (e.g. *Sam's club here*); and help for users (e.g. *Dallas, Oregon or Dallas, Texas?*). In short, the shared experience of spoken interaction with minimal shared representations leads to self-limiting system functionalities.

Rate-a-Course

The Rate-a-Course system is a telephone-based survey system developed at Stony Brook University (Stent, Stenchikova, and Marge 2006). The system permits users to review their courses along five dimensions (instructor, teaching assistant, exams, assignments and class size) and to hear summaries of ratings provided by other users. To collect a review for an aspect of a course, the system first asks the user to give a quality rating (e.g. excellent, very good, okay, bad, terrible), then asks the user to explain their rating (see Figure 2 for an example subdialog). This means that the user and system each have at least two opportunities to refer to each aspect of the course, and that the system always gets to go first.

The Rate-a-Course system also has minimal shared representations; in addition to dialog state, it tracks only the following features: the rating assigned by the user to the course aspect currently under discussion (so that it can use that phrase in the follow-on prompt to the user); the verb tense being used in this dialog; and the realizations of each of the course aspects for this dialog. For each dialog, the system chooses from up to five possible realizations of each course aspect (e.g. the professor might be referred to as *the professor, the lecturer, the instructor* or *the teacher*). The system also chooses for each dialog whether to refer to the course in the past or present tense (e.g. *Why was the teacher excellent?* or *Why is the teacher excellent?*).

Even though the Rate-a-Course system has minimal shared representations, we see convergent behavior in this system too. In an analysis of user responses to course aspect questions by 48 users across 96 course reviews, we found that 53% of user responses to system questions about a course aspect refer explicitly to that aspect, and of those 53%, 64% use the same term for the course aspect as used by the system. In addition, 39% of user responses to system questions about a course aspect contain at least one verb, and of those 39%, 74% use the same tense as used by the system.

Of course, the only reason the Rate-a-Course system needs to remember the verb tense is that it has no semantic representation of the date the user took the course, even though the user provides this information as part of the course survey. In addition, it does not recognize the free-flowing responses from users in answer to the *Why do you think the <aspect> is/was <rating>?* question. Instead, these are transcribed by hand and then fed back into the system's database at a later date. This lack of lexical shared representations means that the system cannot ask additional follow-up questions, even if the user's review is off-topic (e.g. consists of providing a rating for the next course aspect) or non-existent (silence). Also, once a user has rated one or more courses, a good review system would provide the user with reviews of other courses tailored to the user's preferences. However, the Rate-a-Course system does not currently share representations of user preferences across interactions. Again, a lack of shared representations leads to significant limitations in system usability.

Shared Representations are Useful in Interpretation

We have illustrated ways in which the lack of shared representations is disadvantageous; now we turn to ways in which even limited shared representations can be useful. We will briefly present two sets of results: one in which acoustic, lexical and semantic information are stored for for task-related concepts only, and one in which lexical, syntactic, semantic and dialog history features are used to improve dialog modeling and coreference resolution.

Let's Go

The *Let's Go!* dialog system is a telephone-based bus information system hosted at Carnegie Mellon University (Raux and others 2005). Unlike Speak4It, users can have multi-turn dialogs with *Let's Go!*. The shared language experience is longer, and there are more shared representations. In particular, the system tracks which system-related concepts the user has referred to in this dialog (departure location, arrival location, departure time, arrival time, bus route number). However, the system does not build sophisticated models of language used by itself or the user; i.e. the shared representations are at the task and concept level only. Furthermore, the system has a fairly high speech recognition error rate (64.3% in 2006 (Raux and others 2006)); in other words, it does not give evidence of human-like language processing ability. Nonetheless, through fairly simple manipulations we showed that users adapt to the system's choices of verbs, prepositions and concept-related phrases (e.g. *nine*

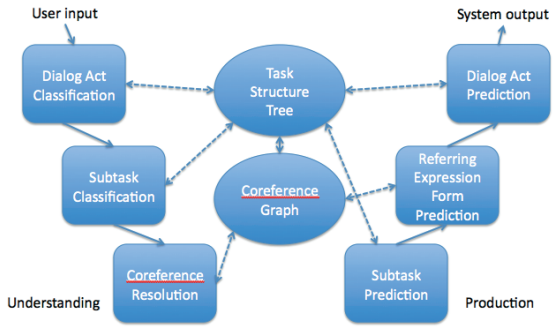

Figure 3: A statistical parsing-based model for task-oriented dialog

in the morning, nine a.m.) (Stoyanchev and Stent 2009b)[1]. Furthermore, if the system appears to be adapting to the user's realization of task-related concepts, then users are more likely to adapt (Stoyanchev and Stent 2009b).

If a dialog system can prime users' choice of words for task-related concepts, then it can guide users to choose forms that are more likely to be correctly recognized. In a separate experiment with *Let's Go!* data, we showed that it is possible to predict the user's choice of task-related concepts given acoustic, lexical, semantic and discourse information, and that these predictions can be used to modify the behavior of the system's speech recognizer, leading to improved speech recognition performance (Stoyanchev and Stent 2009a).

Parsing-Based Dialog Models

The core tasks of a dialog manager are: to identify user intentions from user input; to determine which task-related concepts a user is referring to; and to predict the system's next actions. For several years, we have been pursuing an incremental parsing-based model for cooperative task-oriented dialog. Figure 3 shows the architecture of our model. Utterances are processed one-by-one and used to build up the task stucture for the dialog. Each input user utterance is assigned a dialog act and subtask based on the lexical, syntactic and semantic features of the utterance, features of recent utterances, and the task tree for the dialog. The subtask and dialog act of each system utterance are predicted based on the lexical, syntactic and semantic features of the most recent utterances in the dialog and on the task tree for the dialog. Possible user and system intentions are restricted to those on the frontier of the task tree (Bangalore and Stent 2009). An interesting feature of this model is that all decisions are made based on classifiers trained over a large corpus of dialogs.

Recently we have applied this same approach to the coreference task. After a user utterance has been processed, we extract the noun phrases from the utterance and pass them through a coreference resolution classifier. The output of this classifier is used to build a graph expressing coreference links between noun phrases in the dialog. After the goal for the system's next utterance has been predicted and the objects to be mentioned have been collected, we pass each object through a referring expression predictor, which decides what form the output noun phrase should take (e.g. pronoun, definite NP with modifiers, bare indefinite NP).

In experiments with various parts of this architecture, we have shown that only a small amount of dialog history (1 utterance) is necessary for user dialog act classification, while more history is necessary for system dialog act prediction (Bangalore and Stent 2009). Also, the use of dialog act and task/subtask information from the task tree improves the performance of coreference resolution (Stent and Bangalore 2010). We are currently evaluating the performance of referring expression form prediction.

Shared Representations are Useful in Generation

As the CHILD experiments show, rich shared linguistic representations are useful not only for understanding user input, but also for generation in dialog. In this section, we summarize experimental results showing that shared representations are also useful for referring expression generation.

Partner-Specific Referring Expression Generation

If a speaker, looking at a room of different colored blocks, says *take the blue block*, and the listener responds with *I have the large block*, the speaker is likely to think that either there are two blue blocks, or that the listener took the wrong block on purpose. This is because the listener failed to adapt to the speaker's choice of referring expression, and is exactly the situation in many dialog systems today.

If a dialog system tracks mentions of task-related concepts by the user, this information can be used to improve the production of referring expressions. The shared representation in this case consists of the surface forms of referring expressions previously seen in this interaction, and of a speaker-specific priority list of attributes (e.g. size, color, type) that can be used to describe task-related objects, updated as the dialog progresses. In a series of experiments (Gupta and Stent 2005; Fabbrizio, Stent, and Bangalore 2008), we showed that if a system can observe a human's preferred descriptors for objects and build the shared representations just described, the system can produce referring expressions more like those produced by a human dialog partner. Referring expressions produced in this way also enable a human reader to more quickly identify the object being described (Gatt, Belz, and Kow 2008).

Shared Representations and Dialog Systems

The examples in the previous sections are intended to illustrate possible uses of shared representations in dialog systems. These shared representations are not particularly heavy (i.e. they do not require large amounts of computation). They include the following:

- Lexical and syntactic information – about choices made by the user and by the system. This supports alignment and partner adaptation.

[1]This finding was recently confirmed and extended by (Parent and Eskenazi 2010).

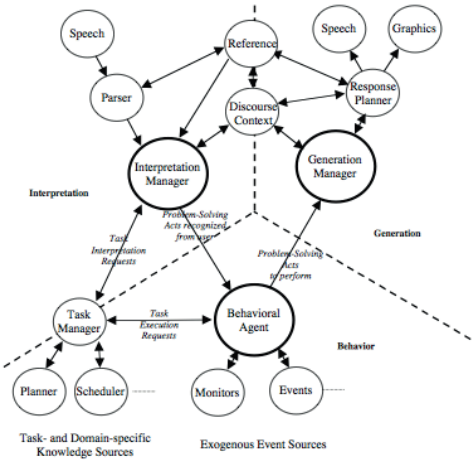

Figure 4: Dialog system architecture supporting shared representations; figure taken from (Allen, Ferguson, and Stent 2001)

.

- Speaker information – to distinguish choices made by the user from those made by the system.
- Dialog history information – to distinguish choices made recently from those made long ago.
- "One bit model" features (Brennan, Galati, and Kuhlen 2010) – these may include the task-related concept currently under discussion, and information about which task-related objects the user can be presumed to know about, which the user has previously discussed, etc.

Probabilities or confidence scores may be associated with any of these types of information, representing for example the system's degree of certainty about output from the speech recognizer, parser, reference resolution module, etc.

Shared representations of the kinds we are discussing impose constraints on the architecture of dialog systems. In particular, the dialog system architecture must support making shared representations accessible (a) as early as possible, and (b) as widely as possible. The dialog system processing can be as modular as desired, but the dialog system's knowledge must be readable by all.

Several existing dialog system architectures support the use of rich linguistic shared representations. We will highlight two here. The first is the architecture proposed in (Allen, Ferguson, and Stent 2001), and illustrated in Figure 4. This architecture stores representations often hidden inside a dialog manager in a separate discourse context accessible to the understanding and production components. A careful separation is maintained between task information (below the dotted lines) and dialog information (above the dotted lines), which permits language-related components to interpret or produce interactions (such as turn-taking and grounding behaviors) that rely on linguistic context. The discourse context can be updated by any of the language components at any time. Instatiations of this architecture typically include rich lexical, syntactic and semantic representations (e.g. (Allen et al. 2007)). The parsing-based dialog model described earlier matches the parts of this architecture that lie above the dotted lines, except that the dialog task structure and coreference links are stored separately.

A second architecture that supports rich shared representations is the incremental architecture proposed in (Schlangen and Skantze 2009). In contrast to the architecture of Allen et al. (2001), in this architecture shared representations are distributed about the system. Each component in this architecture has an input buffer and an output buffer, which are accessible to other components. A component reads input from its input buffer and posts hypotheses on its output buffer. It may withdraw a hypothesis (causing changes to the input buffers of other components) or commit to a hypothesis (permitting other components to commit to the consequences of the hypothesis). Instantiations of this architecture are quickly developing rich shared representations (e.g. (Schlangen and others 2010)).

Conclusions

In this paper, we have presented the need for dialog systems to incorporate richer shared linguistic representations. We have motivated the discussion with case studies from deployed dialog systems and with experimental data. We have described desiderata for shared representations, and highlighted two dialog system architectures that support these representations in different ways.

References

Allen, J. F.; Dzikovska, M.; Manshadi, M.; and Swift, M. 2007. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*.

Allen, J. F.; Ferguson, G.; and Stent, A. 2001. An architecture for more realistic conversational systems. In *Proceedings of the Intelligent User Interfaces Conference*.

Bangalore, S., and Stent, A. 2009. Incremental parsing models for dialog task structure. In *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics*.

Bard, E., et al. 2000. Controlling the intelligibility of referring expressions in dialogue. *Journal of Memory and Language* 42:1–22.

Bock, J. K. 1986a. Meaning, sound, and syntax: Lexical priming in sentence production. *Journal of Experimental Psychology: Learning, Memory, & Cognition* 12:575–586.

Bock, J. K. 1986b. Syntactic persistence in language production. *Cognitive Psychology* 18:355–387.

Branigan, H. P.; Pickering, M. J.; and Cleland, A. A. 2000. Syntactic co-ordination in dialogue. *Cognition* 75:B13–25.

Brennan, S. E., and Clark, H. H. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory and Cognition* 22:1482–1493.

Brennan, S. E.; Galati, A.; and Kuhlen, A. K. 2010. Two minds, one dialog: Coordinating speaking and understanding. In Ross, B. H., ed., *The Psychology of Learning and*

Motivation, volume 53. Burlington: Academic Press. 301–344.

Brennan, S. E. 1996. Lexical entrainment in spontaneous dialog. In *Proceedings of the 199th Internatioanl Symposium on Spoken Dialogue*.

Dubey, A.; Sturt, P.; and Keller, F. 2006. Parallelism in coordination as an instance of syntactic priming: evidence from corpus-based modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Fabbrizio, G. D.; Stent, A.; and Bangalore, S. 2008. Trainable speaker-based referring expression generation. In *Proceedings of the Conference on Natural Language Learning*.

Garrod, S., and Anderson, A. 1987. Saying what you mean in dialogue: A study in conceptual and semantic coordination. *Cognition* 27:181–218.

Gatt, A.; Belz, A.; and Kow, E. 2008. The TUNA challenge 2008: Overview and evaluation results. In *Proceedings of the Fifth International Natural Language Generation Conference*.

Gupta, S., and Stent, A. 2005. Automatic evaluation of referring expression generation using corpora. In *Proceedings of the Workshop on Using Corpora for Natural Language Generation*.

Jungers, M. K.; Speer, S. R.; and Palmer, C. 2002. Prosodic persistence in speech production and music performance. In *Abstracts of the Psychonomic Society, 43rd Annual Meeting*.

Levow, G.-A. 2003. Learning to speak to a spoken language system: Vocabulary convergence in novice users. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*.

Lieberman, P. 1963. Some effects of semantic and grammatical context on the production and perception of speech. *Language and Speech* 6:172–187.

Nenkova, A.; Gravano, A.; and Hirschberg, J. 2008. High frequency word entrainment in spoken dialogue. In *Proceedings of the Meeting ofthe Association for Computational Linguistics*.

Parent, G., and Eskenazi, M. 2010. Lexical entrainment of real users in the Let's Go spoken dialog system. In *Proceedings of Interspeech*.

Pickering, M. J., and Garrod, S. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences* 27.

Pickering, M. J., and Garrod, S. 2006. Alignment as the basis for successful communication. *Research on Language and Communication*.

Pierraccini, R., and Huerta, J. 2005. Where do we go from here? research and commercial spoken dialog systems. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*.

Raux, A., et al. 2005. Let's Go Public! Taking a spoken dialog system to the real world. In *Proceedings of Interspeech*.

Raux, A., et al. 2006. Doing research on a deployed spoken dialogue system: One year of Let's Go! experience. In *Proceedings of Interspeech*.

Reitter, D., and Keller, F. 2007. Against sequence priming: Evidence from constituents and distituents in corpus data. In *Proceedings of the Annual Conference of the Cognitive Science Society*.

Reitter, D., and Moore, J. 2007. Predicting success in dialogue. In *Proceedings of the Meeting ofthe Association for Computational Linguistics*.

Reitter, D.; Keller, F.; and Moore, J. D. 2006. Computational modelling of structural priming in dialogue. In *Proceedings of the Human Language Technology Conference ofthe North American Chapter of the ACL*.

Schlangen, D., et al. 2010. Middleware for incremental processing in conversational agents. In *Proceedings of SIGdial*.

Schlangen, D., and Skantze, G. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics*.

Stenchikova, S., and Stent, A. 2007. Measuring adaptation between dialogs. In *Proceedings of the7th SIGdial Workshop on Discourse and Dialogue*.

Stent, A., and Bangalore, S. 2010. Interaction between dialog structure and coreference resolution. In *Proceedings of the Spoken Language Technology Workshop*.

Stent, A. J.; Stenchikova, S.; and Marge, M. 2006. Dialog systems for surveys: The Rate-a-Course system. In *Proceedings ofthe 1st IEEE/ACL Workshop on Spoken Language Technology*.

Stoyanchev, S., and Stent, A. 2009a. Predicting concept types in user corrections in dialog. In *Proceedings of the EACL Workshop on Semantic Representation of Spoken Language*.

Stoyanchev, S., and Stent, A. J. 2009b. Lexical and syntactic priming and their impact in deployed spoken dialog systems. In *Proceedings of the Meeting of the North American Chapter of the Association for Computational Linguistics*.

Williams, J. D., and Young, S. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language* 21(2):393–422.