Spoken Natural Language Dialog Systems: A Practical Approach

Ronnie W. Smith and D. Richard Hipp
(East Carolina University and Hipp, Wyrick and Company, Inc.)

New York: Oxford University Press,
1994, xiv+299 pp; hardbound, ISBN
0-19-509187-6, $49.95

Reviewed by
David R. Traum
Université de Genève

This book gives a detailed yet readable account of the design, implementation, testing, and analysis of a spoken task-oriented dialogue system. Although there is a thorough reporting of related work, the plural in the main title is a bit misleading, as the book focuses on only one system, built by the authors at Duke University. The subtitle is well warranted, as Smith and Hipp put a premium on achieving useful interaction rather than adherence to any particular psychological or linguistic theory of language processing.

The writing is generally clear, even when discussing details of the sometimes complex algorithms. The index is also very helpful. In addition, for those who want more specifics, an appendix gives instructions for electronically obtaining the dialogue system code and transcripts of user interactions with the system. The code is also commented with pointers to aspects of the book, including sections, figures, and pages. Smith and Hipp carefully point out not only the successes of their system, but also its shortcomings and some prospects for improvement.

Smith and Hipp do a good job of describing and analyzing related work and comparing it to their own system. Chapter 2, in particular, reviews foundational work on dialogue processing. The main focus is on work in the sub-areas of speech act processing, user modeling, use of expectations, and mixed initiative. Also included are several other proposals for integrated dialogue models. The highlight of the chapter is a comparison of 26 other dialogue systems, mostly from the 1980's, listing the application domain and output modality, as well as the types of processing performed and how the systems were evaluated. Other related work is introduced as appropriate in subsequent chapters. The authors are perhaps not quite as successful in recasting the innovations from their own implementation in a more general conceptual framework that could be useful for designers of systems with very different architectures.

A task-oriented spoken language dialogue system is a large piece of software, requiring successful integration of a number of heterogeneous sub-components to perform at least the following functions: speech recognition, language understanding, dialogue reasoning, task reasoning, language generation, and speech generation. The thrust of the authors' work is on the language understanding and dialogue reasoning components, as well as the overall architecture that ties these subsystems together. These aspects are also presented more briefly by Smith, Hipp, and Biermann (1995), who also include a more detailed discussion of the relationship to Grosz and Sidner's (1986) theory of discourse structure.

The authors' system used independent components for speech recognition (speaker-dependent connected speech) and output, while the main dialogue processing took place on a Sun 4 workstation, running Prolog and C. The system understood a vocabulary of 125 words, and used 560 grammar rules. Users of the system were instructed to perform single sentence utterances of 3–6 words in length, waiting for system response before uttering a new sentence. Normal response time for the whole system was 6–8 seconds for utterances of this length. All the words were correctly recognized in only 50% of 2840 utterances, although the error-correcting parser managed to achieve the correct meaning in 81% of the utterances.

As this book demonstrates, a task-oriented dialogue system must be more than just an interface or front-end. For coherent dialogue, it is crucial that there be close cooperation between the task and language reasoning modules. In order to interpret utterances in context, respond appropriately, and track user focus, the dialogue system must have some idea of what the task reasoner is doing, at least at an abstract level. The authors adopt a strong hypothesis here that dialogue structure *is* task structure, following the intentional discourse structure of Grosz and Sidner (1986). Each task step is related to a (potential) subdialogue, and expectations and meaning interpretation are always computed relative to the focused or likely subdialogues.

The domain processor uses Prolog-style theorem-proving on a set of plan libraries in the form of circuit-diagnosis proofs. The authors connect task reasoning to language by what they call the *Missing Axiom Theory*. Some of the axioms in the diagnosis proofs refer to knowledge of states of the world or the performance of actions. When these axioms are *missing* (i.e., the system knows that they would help in the proof, but can't prove them), actions or observations must be performed. In particular, since this system has no way to directly sense or act upon the elements of the circuit, it must use language to instruct the user to perform the desired actions. Thus, language is viewed as a way of providing *missing axioms* for the circuit diagnosis.

Being embedded within an interactive dialogue system provides constraints on the domain task reasoning itself. In particular, the system must be able to reason about action and plans from both directions: from world state to best solution path—in order to output requests for action and observation to the user, and from arbitrary action or observation and world state to likely plan step—for interpreting user utterances in context. As well, the system reasoning must be prompt (for quick response, maintaining fluid interactivity) and interruptible—full proofs must be interspersed with actions and observations. For this purpose, the authors have implemented a modified Prolog system that allows a user (or, in this case, the dialogue system) access to an ongoing proof, with the ability to notice and provide missing axioms, or modify the proving process.

The system starts the process of understanding user inputs by reasoning about context before actually interpreting the language input; the system calculates a set of *expectations* of what the user will say in the current circumstance. There are a number of possible user responses at any given point, and the authors classify the expectations into those coming from the task processor (e.g., descriptions of a component) or the dialogue controller (descriptions of subgoals or reference to ancestor subdialogues), and in each of these categories, whether they are specifically about the situation or task itself, or merely related.

These expectations are ordered and weighted as to likelihood and used to help disambiguate parses, as well as provide necessary contextual information for providing full interpretations of pronouns, elided material, and short answers such as "yes". Some of the expectations will have variables included (e.g., for a numerical value when expecting the report of a measurement). These underspecified expectations are unified with similar forms that result from parsing the user's input. When completely unexpected inputs occur that the system cannot relate to the current task structure, it tries to teach the user how to perform the current goal (making the assumption that the user utterance was a misunderstood attempt to discuss this goal).

The main input to the parser is a lattice of the word possibilities produced by the speech recognizer. Also provided is the set of expectations calculated by the dialogue module. The parser computes the most likely parses using an n^3m dynamic programming algorithm (where n is the size of the input string, and m is the size of the grammar), assigning a cost function based on string distance between the actual input and the right-hand side of a grammar rule. The least expensive parses are compared to the expectations produced by the dialogue system, for final selection of the interpretation. For the dialogues collected in their experiments, the authors determined that the dialogue expectations help only to disambiguate parses with equal cost. Thus, in fact, the way expectations are used turns out to be equivalent to the "traditional" pipe-lined processing model, in which context is consulted after producing an initial parse. What is interesting, though, is the authors' *practical approach*, which determined this fact empirically by examining the performance on the actual input.

Another important aspect of dialogue is the degree of *initiative* taken by the system (Whittaker and Stenton, 1988). This amounts to a determination of which participant will guide the direction of the dialogue and to what degree. Smith and Hipp include four levels of initiative ranging from *directive*, where user focus is ignored in formulating the computer's dialogue goals, to *passive*, where the computer will only process and confirm understanding of the user's utterance and provide information only in response to direct questions. In the intermediate levels, the system will try to find a common relationship between the user's focus and its own goal. Although there is some flexibility allowed (the system will try to take more control when it suspects that the user does not know what to do next), generally the initiative level must be preset for each dialogue. The system was tested in both declarative and directive modes (after an initial training session in directive mode), with results showing both shorter dialogues in the declarative mode, as well as more users stating that the system had too much control when in directive mode. The authors provide a detailed analysis of the numbers of utterances and completion times for subjects using the directive or declarative styles.

Smith and Hipp also include a chapter on verifying potentially mistaken inputs as an adjunct to the error-tolerant processing. Although this was added after the experiments were carried out, there is some analysis of the collected dialogues to see how the system would have performed with such strategies. One of the major problems is deciding when and when not to verify; too much verification results in an unwieldy system, while too little can result in a higher rate of misinterpretations. Smith and Hipp propose several measures for confidence in an interpretation, relating to the cost of the parse (amount of errors), the distance of the result from expectations, and the relative ambiguity (closeness of the best parse to other possible parses). Several estimate functions are proposed on the basis of combinations of these measures and are compared as to how they would have performed in the corpus from the experiments. When the confidence drops below a threshold, then a verification is performed. Assuming that the verification subdialogues would eventually produce the correct answer, they calculate that engaging in the verification procedures would raise the percentage of correct interpretations from 83% to 97%.

In general, Smith and Hipp's system emphasizes the task-related *intentional* structure too much at the expense of the *linguistic* or *social* structure. Both are important for fluent dialogues. While it is certainly interesting to see how far an approach based solely on task structure can get, there are some issues that would need expanding in a more general system. For example, the way that pronoun resolution is performed is by matching the input to the lowest cost expectation. The pronoun is then identified as the corresponding object in the expectation. While this will work well for many cases, it may have problems if an unexpected utterance is made about a focused object. In this case, traditional pronoun resolution techniques (e.g., centering (Grosz, Joshi, and Weinstein, 1995) or focus-matching (Grosz, 1977)) would find the referent, while presumably Smith and Hipp's approach would not be able to match the unexpected input at all. Also, while the dialogue model allows for clarification subdialogues, it does not encode many general non task-related patterns of linguistic interaction such as the linguistic expectations used by McRoy and Hirst (1995).

While the dialogue system is fairly successful at interacting with a user to fix the circuits, there are still some aspects of the interaction that diverge from natural dialogue. First, a rigid turn-taking system was imposed, which allowed the user to only say a single sentence before waiting for a system response. While this kind of limitation is fairly standard for written dialogue systems involved in query-answering, it detracts from the flexibility of spoken dialogue by not allowing followup elaborations, clarifications, or shifts in initiative. Secondly, the *directive* initiative level mentioned in the previous paragraph is *too* inflexible. When the system is in control, it tends to just repeat the previous query if it gets a reply that it cannot understand as directly satisfying it, while ignoring the reply itself. A real dialogue participant should respond to what was said, if even just to rephrase the request or chastise the other for going off topic. This kind of repetition in the face of non-compliance is also likely to be misinterpreted by the user (Suchman, 1987). Finally, the experimental set-up allowed the experimenter to intervene in specified ways to counteract specific system limitations, such as words not in the vocabulary, occasional slow system response time, or deviating from the strict single-sentence turn-taking conventions.

Although some of the specific devices employed in this system will not be used in future dialogue systems, due to, for example, rapid developments in speech processing technology, this book and the system described will continue to be interesting for the practical approach to dialogue, and the careful analysis of the interactions of specific dialogue phenomena. In particular, the method of general, parameterized system design, with parameters set by analyzing performance on a particular corpus, should allow general dialogue architectures to be customized to particular domains. This system also serves, for the present, as a demonstration that building such a system is possible even with off-the-shelf technology and limited resources.

References

Grosz, Barbara J. 1977. The representation and use of focus in a system for understanding dialogues. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA, pages 67–76.

Grosz, Barbara J., Aravind K Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226.

Grosz, Barbara J. and Candace L. Sidner. 1986. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

McRoy, Susan W. and Graeme Hirst. 1995. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics*, 21(4):435–478.

Smith, Ronnie W., D. Richard Hipp, and Alan W. Biermann. 1995. An architecture for voice dialogue systems based on Prolog-style theorem proving. *Computational Linguistics*, 21(3):281–320.

Suchman, Lucy A. 1987. *Plans and Situated Actions.* Cambridge University Press.

Whittaker, Steve and Phil Stenton. 1988. Cues and control in expert–client dialogues. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, pages 123–130.

David R. Traum is a post-doctoral researcher in the educational technologies group (TECFA) of the Psychology and Education Department at the University of Geneva. He received his PhD in Computer Science from the University of Rochester, while working on dialogue management issues in the TRAINS natural language system. Traum's address is: TECFA, FPSE, Université de Genève, 9 Route de Drize, Bat D, CH-1227 Carouge, Switzerland; e-mail: David.Traum@tecfa.unige.ch