Unifying language modeling capabilities for flexible interaction

Deryle Lonsdale, Rebecca Madsen

Brigham Young University
lonz@byu.edu, rmadsen@byu.net

Abstract

Dialogue processing has taken several different forms in recent years; in this paper we address a cognitive modeling approach to the problem. We begin by sketching some of our work in this area and how it derives from prior research in cognition, modeling, natural language processing, and discourse recipe compilation. We then note that, whereas the complexities of natural language—especially spoken language—have been primarily addressed by technologies that target lower-level linguistic processing, higher-level processing has been less well studied. We introduce our approach, which is to use an agent-based cognitive modeling architecture for dialogue processing. We mention some of the advantages to using such a framework for dialogue, particularly the perspicuity of a unified architecture. We then explore the difficulties and challenges of using a cognitive modeling foundation. Particularly relevant are the issues dealing with evaluation: resources, procedures, and methodological considerations.

1. Background and previous work

Discourse and dialogue drive much of the interactive communication that takes place in daily life among humans as well as between humans and machines. Studying how communication unfolds involves several disciplines, among them cognition, linguistics, pragmatics, and human-computer interaction. When such communication extends beyond traditional keyboard/mouse/icon input to spoken language modalities, the areas of speech recognition, speech understanding, and speech synthesis additionally become involved. In this paper we sketch work we have done in these areas and mention some of the tensions, tradeoffs, and outstanding problems that, to our knowledge, still confront developers of dialogue-based applications.

Traditionally, tracking how a dialogue unfolds has been possible using several available techniques. Finite-state machines have been most popular and successful when the domain and scope of a dialogue are limited and well-defined. Toolkits are available that allow a developer to specify, either explicitly or underlyingly, successive states in a dialogue and possible directions the conversation may take based on various criteria. Not surprisingly, such systems work best in largely deterministic contexts; realistic and complex scenarios are rarely this clear-cut, though.

Template-based models introduce more flexibility in a computer-human conversation, but are still suited mainly for small-scope, task-oriented problems [9, 23]. Belief-desire-intention (BDI) architectures [8, 6] and dialogue planning systems [1] have been more recently introduced to help explain why a computer system might react in a certain way; these models are generally intended for larger-scope dialogues.

We have experimented with variations on the previously mentioned architectures in our work which has involved scripting military training scenarios [18], modeling simultaneous interpretation [11], robotics applications [2], and speech-based language pedagogy [14]. Our current dialogue systems development has primarily involved a method of dialogue planning termed discourse "recipes" [5] which extends earlier agent-based task modeling situations such as flight readiness testing [16] and intelligent forces in combat situations [19]. This approach allows us to take advantage of semantic information, context, and machine learning as the dialogue unfolds. This dialogue management technique leverages a plan-based approach, managing models of discourse referents and participants. It maintains a model of given information (a common ground), and new information. The system uses a model of conversational strategies, or plans, as well as speech or dialogue acts. Crucially, the system can also learn dialogue recipes from previously compiled plans.

The overall system is agent-based and integrates the discourse recipe functionality with other types of processing, in particular various modalities of natural language (NL) processing. For examples, the agent's NL functions include recognition and generation, both of which involve dialogue-based components.

In recognition mode, word-based utterances are input to the system one-by-one in an incremental fashion over an input link from the "outside world." Their presentation rate can be controlled and they can be buffered and even subject to decay if so desired.

The agent attends to each word in succession, processing each individually by performing the following operations:

- lexical access (which retrieves phonological, morphological, syntactic, semantic, and lexical information for each word from its lexicon) [13, 20]; relevant properties are supplied from system-internal resources and external repositories (e.g. WordNet [3]).
- syntactic model construction (linking together pieces of an incrementally projected X-bar parse tree) [10]
- semantic model construction (fusing together pieces of a meaningful, appropriate, and consistent Lexical-Conceptual Structure [7] incrementally derived from the syntactic representation) [15]
- discourse model construction (extracting global coherence from individual utterances) [4, 5]

Utterance comprehension thus involves processing an input stream of words through lexical, syntactic, and semantic levels of processing. On the other hand, utterance generation constitutes the reverse of this process: the agent formulates a sentence from semantic content using the same type of linguistic structures. Once a semantic LCS model has been selected or formulated, the agent traverses the network, incrementally selecting nodes to process. Processing each concept involves converting it to a lexical form (usually a word) and then adding the word to an ongoing X-bar

syntactic model. Crucially, the same syntactic construction operators can be used for generation as for comprehension, so bootstrapping is possible across these modalities [12].

One possible strategy—a conservative one—for generating the output utterance sentence is to traverse the syntactic model when it is completed, collecting the lexical content from all the leaf nodes and linearizing them into a sentence. This is the default strategy for generation, assuring that the model is grammatically correct and complete before the agent "says" its content. Of course other, more incremental, strategies are also possible; the risk is that, if structures must be reformulated, dysfluencies will arise in the output utterance as repairs are carried out.

Once the agent has comprehended an utterance in the linguistic sense, dialogue processing must occur in order to generate a response. Dialogue processing is the step between comprehending an utterance, and formulating a response to that utterance. Different approaches to managing this step—processing dialogue in computer/human interactions—have been explored, each revealing advantages and disadvantages [18].

To recognize a dialogue plan, the dialogue component combines the syntactic and semantic features of the utterance (from comprehension) with the agent's conversational record to create the hearer's model of the speaker (HMOS). Figure 1(a) illustrates this discourse comprehension process. The language agent takes the HMOS and attempts to match it against possible dialogue moves (dialogue acts) and dialogue plans (including learned recipes) that the speaker may be intending to accomplish with the interaction. This creates the language agent's model of the discourse context (including the human user) and updates its conversational record.

Using the context—private beliefs, private desires, the agent's model of the speaker's goals, and the updated conversational record—the agent also attempts to generate a dialogue plan 1(b). Based on the context, the system then tries to determine if there is a discourse recipe, or a previously learned plan, that matches the current context. If there is not a match already, the agent compiles a new dialogue plan to generate a response. This new plan can then be learned through a compilation procedure to create a discourse recipe for future use. The discourse recipe is what the system uses to create dialogue acts, which will then go through the utterance generation process to send a response to the user.

The most recent extension in our system's capabilities allows for speech-based interaction. The human interlocutor can thus escape mouse-based and keyboard input, using instead the more natural spoken modality for interaction. For speech input we use the SPHINX automatic speech recognition (ASR) system (version 4)[1]. It can reside on the same machine as the agent or reside elsewhere and communicate via sockets.

When the human speaks into a high-quality microphone, the agent "hears" the signal which Sphinx processes and outputs a word-based textual transcription of the utterance. This utterance is then passed word-by-word (over a socket if necessary) to the agent which performs comprehension processing as outlined above.

The system also includes a text-to-speech (TTS) synthesis system built on the Festival toolkit[2]. A client/server architecture was developed so that the agent can control a Festival server, connected with the system's speakers. As utterances are generated by the language agent as explained above, they are sent—again over sockets if necessary—to the TTS server along with a few parameters to specify voice gender, intonation, etc. The utterance is then output over the onboard speaker.

As described above, the current implementation uses a textual, word-based representation of speech input and output for communication between agents. This is a temporary simplification; the ASR engine is capable of delivering a speech lattice representing several recognition hypotheses, and we are in the process of allowing it to be directly represented in the language agent's hierarchically structured input memory. This will eventually allow a closer modeling of speech perception, which is not crucial at this stage of the research, though it will eventually open several avenues of exploration including speech perception errors, prosody recognition, and so on.

2. Issues

Natural language constitutes the most natural human interface medium, along with (to varying degrees) gestures and deixis. Furthermore, the most natural language interface is speech. This immediately introduces two issues that dialogue systems developers must cope with.

The first issue is that natural language processing is difficult, far from being a solved problem. Various low-level techniques in NL processing (NLP)—such as part-of-speech tagging, named entity recognition, and word-sense disambiguation—have been well studied, documented, and tested using standardized resources, metrics, and evaluations. Even some aspects of higher-level processing, such as syntactic parsing and semantic role identification, are quickly reaching maturity as resources, techniques, and results improve. Generally this progress reflects the fact that much groundwork has been done on these lower-level aspects of language analysis by linguistics and NLP developers alike. However, processing utterances at the syntactic, semantic, and pragmatic levels remains a challenging problem. Complexity, intractibility, and a lack of theoretical consensus combine to make higher-level linguistic processing a daunting problem for most dialogue system developers. Avoiding such complexities by sidestepping these stages of processing is possible, but naturalness and spontaneity suffer. This tradeoff may in fact be acceptable for small domains or highly predictable scenarios, but it also proves problematic for naïve users or where robustness and portability to other domains are required.

The second issue is that speech processing—both speech recognition and speech synthesis—are also still emerging technologies. Speech recognition as a standalone process is becoming viable, again primarily for small domains and when enrollment and training can improve system performance. However, most ASR systems involve little or no higher-level linguistic processing (e.g. at syntactic, semantic, or pragmatic levels), principally for the reasons mentioned above.

Dialogue processing lies at the crux of both of these issues. On the one hand, it arguably involves the highest level of linguistic processing and hence relies on results, not yet entirely feasible to obtain, from prior (or at least lower) levels of processing. Unfortunately, pragmatics in general (which also involves discourse, dialogue, and conversation) is generally viewed as the least developed of all the areas of linguistic exploration, especially in the computational realm.

To the skeptic, the issues raised in this section could make convincing arguments that more progress must be made before meaningful implementations in this area can succeed. The AI-

[1] See cmusphinx.sourceforge.net for more information.

[2] See www.cstr.ed.ac.uk/projects/festival/ for more information.

related problems inherent in higher-level linguistic processing and the ASR-related problems in processing spoken language conspire to leave too much of the problem unsolved or at least impractical for all but the most circumscribed applications. Perhaps the issues should only be revisited when the language and speech technologies are more germane and enabling.

To the enthusiast, these open frontiers represent interesting but challenging research opportunities. If the shortcomings and limitations of the current state of the art are recognized and admitted, further progress, however incremental, might be possible. In fact, the very point that these still-developing technologies lie at the crux of dialogue-based issues is what makes research in this area so compelling. The optimist's point is that we need to be engaged in research in all of these areas in order to make progress, and dialogue-based R&D could in the long run prove timely and valuable to these other related questions.

3. Integrating versus unifying

Our position relative to research in dialogue processing lies more toward the enthisiast's. We are interested in dialogue as a phenomenon of human cognition, and as such view it on the same level as any other cognitive processing modality, linguistically-based or otherwise. Viewed in the larger context of cognition, the issue of dialogue processing presents both difficulties and opportunities.

Much research is unfolding on human cognition in general, and on modeling human cognition in particular. Theoretical descriptions of human cognition (e.g. [17]) have been instantiated as cognitive modeling (CM) systems. We view the latter as excellent and promising environments for implementing, modeling, and studying dialogue processing. Several CM systems exist, though ironically most previous work in dialogue modeling does not appear to take CM architectures into account.

An alternative to the CM approach is a more engineering-oriented one in which developers may focus less (or not at all) on the human factors that result in or otherwise exhibit learning, memory, expertise, and other features; cognitive plausibility is not a major concern in such work. Engineering approaches rather depend on integration of a large number of specialized subsystems or modules, each having its own compartmentalized functionality and input/output requirements. Creating such a system typically involves pipelining or blackboarding several different engines (e.g. for speech, tokenization, parsing, and semantic interpretation) and assuring that the results are in some measure compatible; this can quickly become a complex systems engineering problem. Large-scale speech translation systems (e.g. Verbmobil) are examples of such systems, where dialogue processing components (such as dialogue move engines) often constitute one of the many modules.

Our approach uses the Soar cognitive modeling system[3] for the work described above. The system is a rule-based symbolic intelligent agent architecture that uses a goal-directed, operator-based approach to problem solving which provides a unified architecture for implementing various aspects of cognition.

There are some benefits to using a CM architecture for dialogue research. One is flexibility: since the agent models human behavior to some level of granularity, we are able to use the same agent either in human-agent interaction or in agent-agent interaction, for the very reason that the agent models the human. Figure 2 shows human-agent and agent-agent speech scenarios that we have implemented with robotic agents.

[3]For more information see http://sitemaker.umich.edu/soar.

A CM architecture, because of its unified character, obviates the need for the engineering approach's large number of specialized subsystems, integration points, interface specifications, and so on. Instead, the unified architecture and its theoretical underpinnings should allow folding of dialogue processes and requisite knowledge sources (agendas, background, plans, goals) into the general cognitive framework.

If, however, CM is to be used for dialogue processing, several CM-relevant development issues also arise when developing and assessing the resultant dialogue engines. For example, it is important that CM developers follow closely the specifications of the architecture; they must implement the system in a way that is theoretically correct, or else system behavior will be unmotivated and unsupported. This is problematic, given that most of the current work in algorithmic descriptions of dialogue processing (e.g. the most popular dialogue move engines) have not taken cognitive factors directly into consideration.

We have observed that CM approaches are well suited to implementing mixed-initiative systems. This is due to the fact that, given the agent's role of modeling a human, it is therefore better able to plan and participate in behaviors more closely identifiable with human-human interaction. In fact, CM systems including our own are "glass box" rather than "black box" systems, meaning that their processing is open to inspection at any time. This means that if dialogue handling fails at any stage, the system should be able to recover either partially or completely via dynamically-activated coping strategies such as entering into clarification dialogues.

One commitment that CM in language processing entails, or at least suggests, is that discourse processing is a higher-level process derivable in a (partly) compositional fashion. We are ourselves unclear to what extent dialogue is compositional versus holistic, and how to determine this.

Among the theoretical and methodological issues inherent in CM-based dialogue processing, perhaps the most perplexing is evaluation. We can only hint here at the issues and possible solutions, but clearly this area warrants much further attention.

Evaluating discourse, as has been pointed out by others, requires both quantitative and qualitative evaluation metrics. What, exactly, should be measured, though? Using a CM architecture for dialogue research raises issues not germane to engineering-approach developers. How do we evaluate the human factor in dialogue? Or more generally, how do we evaluate cognition? This presents perplexing difficulties; evaluating the cognitive plausibility of human linguistic processing is relatively new and unexplored. Literature is growing on topics involving lower levels of human linguistic processing including lexical access, word recognition, parsing strategies, and speech errors (both in recognition and in generation). How do we quantify the contributions of each stage of processing to overall cognition in dialogue processing, and how commensurable would these measures be?

We sense acutely, therefore, one current shortcoming of a CM-based approach to dialogue modeling: the paucity of human-factor annotated dialogue corpora. Whereas other areas of active NLP research have produced standardized evaluation techniques (e.g. PARSEVAL), shared or competitive evaluations (e.g. SENSEVAL), corpora (e.g. the Wall Street Journal), and annotation standards (e.g. WSJ Treebank or TRACTOR), evaluation remains less developed within the dialogue research community. Certainly the Map Task, PARADISE [21], Communicator [22], RST, and DATE efforts have contributed much to resource development and scenario annotation, but much more remains to be done in this area.

This is especially true where cognitive strategies are involved; such annotations may only be possible after post-hoc introspection by participants. When human strategies and processing are taken into consideration, how do we evaluate and score dialogues (be they successful or not)? How closely can we track and score putative processing mechanisms for a target as complicated as dialogue modeling?

Unsurprisingly, much work remains in all of these areas. Still, we are confident that ongoing research in cognitive modeling, language and speech processing, and discourse representation will drive further progress in dialogue modeling. Crucial to these advancements will be the development of corpora and other resources with extensive annotation that reflect the depth and breadth of our knowledge about processing strategies, errors, planning, and the general participation of cognitive abilities in the communicative process.

4. References

[1] L. Ardissono, G. Boella, and R. Damiano. A plan-based model of misunderstandings in cooperative dialogue. *International Journal of Human-Computer Studies*, 48:649–679, 1998.

[2] P. Benjamin, D. Lonsdale, and D. Lyons. Designing a robot cognitive architecture with concurrency and active perception. In *Fall Symposium: The Intersection of Cognitive Science and Robotics*, pages 1–8. American Association for Artificial Intelligence, 2004.

[3] C. Fellbaum. *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA, 1998.

[4] N. Green and J. F. Lehman. Compiling knowledge for dialogue generation and interpretation. Technical Report CMU-CS-96-175, School of Computer Science, Carnegie Mellon University, 1996.

[5] N. Green and J. F. Lehman. An integrated discourse recipe-based model for task-oriented dialogue. *Discourse Processes*, 33(2), 2002.

[6] J. Harland and M. Winikoff. Agents via mixed-mode computation in linear logic: A proposal. In *Proceedings of the ICLP'01 Workshop on Computational Logic in Multi-Agent Systems (CLIMA-01)*, Paphos, 2001.

[7] R. Jackendoff. *Semantic Structures*. MIT Press, Cambridge, MA, 1990.

[8] S. Larsson, P. Ljunglof, R. Cooper, E. Engdahl, and S. Ericsson. GoDiS: an accomodating dialogue system. In *Proceedings of ANLP/NAACL-2000 Workshop on Conversational Systems*, Seattle, 2000.

[9] E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, and M. Walker. The ATT-Darpa communicator mixed-initiative spoken dialog system, 2000. Retrieved from citeseer.ist.psu.edu/576288.html.

[10] R. Lewis. *An Architecturally-based Theory of Human Sentence Comprehension*. PhD thesis, Carnegie Mellon University, 1993.

[11] D. Lonsdale. Modeling cognition in SI: Methodological issues. *International journal of research and practice in interpreting*, 2(1/2):91–117, 1997.

[12] D. Lonsdale. Leveraging analysis operators in incremental generation. In *Analysis for Generation: Proceedings of a Workshop at the First International Natural Language Generation Conference*, pages 9–13. Association for Computational Linguistics, 2000.

[13] D. Lonsdale. An operator-based integration of comprehension and production. In *LACUS Forum XXVII*, pages 123–132. Linguistic Association of Canada and the United States, 2001.

[14] D. Lonsdale, C. R. Graham, and R. Madsen. Learner-centered language programs: Integrating disparate resources for task-based interaction. In P. Zaphiris and G. Zacharia, editors, *User-centered computer aided language learning*. Information Science Publishing, Hershey, PA, 2006.

[15] D. Lonsdale and C. A. Rytting. An operator-based account of semantic processing. In *The Acquisition and Representation of Word Meaning*, pages 84–92. European Summer School for Logic, Language, and Information, 2001.

[16] G. Nelson, J. F. Lehman, and B. E. John. Experiences in interruptible language processing. In *Proceedings of the 1994 AAAI Spring Symposium of Active NLP*, 1994.

[17] A. Newell. *Unified Theories of Cognition*. Harvard University Press, 1990.

[18] R. D. Rees. Investigating dialogue managers: building and comparing FSA models to BDI architectures, and the advantages to modeling human cognition in dialogue. BYU Department of Physics: Honors Thesis, 2002.

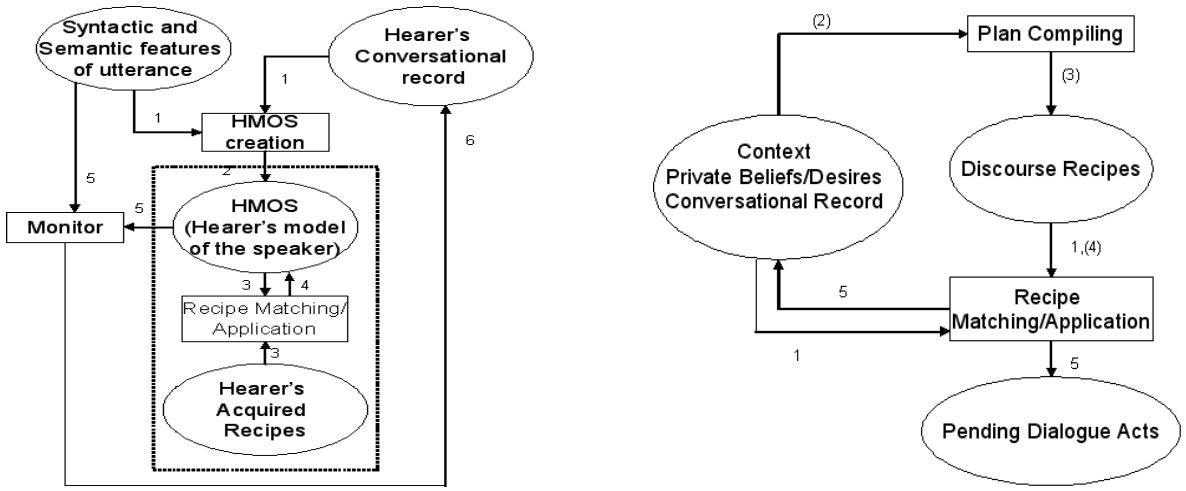
[19] R. Rubinoff and J. F. Lehman. Natural language processing in an IFOR pilot. In *Collected Papers of the Soar/IFOR Project: Spring 1994*. CMU, 1994.

[20] C. A. Rytting and D. Lonsdale. Integrating WordNet with NL-Soar. In *WordNet and other lexical resources: Applications, extensions, and customizations*, pages 162–164. North American Association for Computational Linguistics, 2001.

[21] M. Walker, D. Litman, C. Kamm, and A. Abella. PARADISE: A general framework for evaluating spoken dialogue agents. In *Proc. 35th Annual Meeting of the Association for Computational Linguistics and 8th Conf. of the European Chapter of the Association for Computational Linguistics*, pages 271–280, 1997.

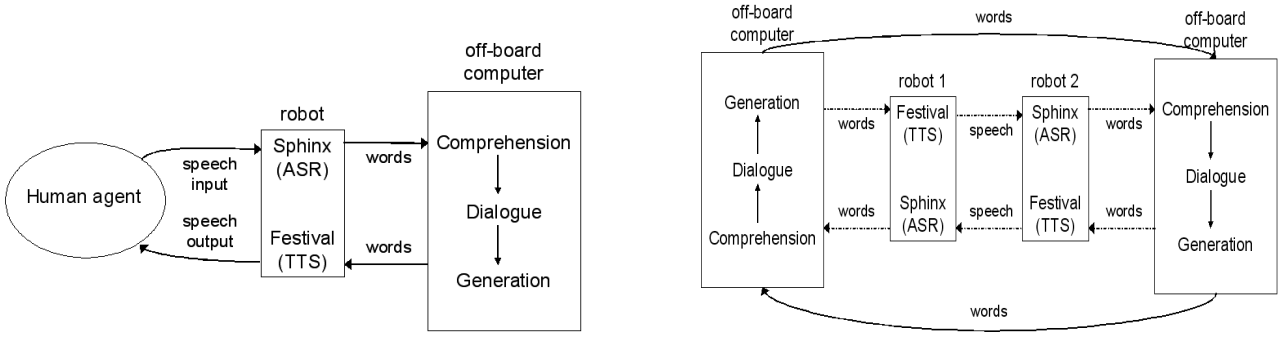
[22] M. Walker, A. Rudnicky, R. Prasad, J. Aberdeen, E. Bratt, J. Garofolo, H. Hastie, A. Le, B. Pellom, A. Potamianos, R. Passonneau, S. Roukos, G. Sanders, S. Seneff, and D. Stallard. Darpa communicator: Cross-system results for the 2001 evaluation. In *Proc. of ICSLP 2002*, 2001.

[23] W. Xu and A. Rudnicky. Task-based dialog management using an agenda. In *ANLP/NAACL 2000 Workshop on Conversational Systems*, pages 42–47, 2000.



(a) (b)

Figure 1: Processing for dialogue plan recognition 1(a) and plan generation 1(b), based on [5].



(a) (b)

Figure 2: Processing during interaction in a human-computer scenario 2(a), and during a two-agent scenario 2(b).