Goals for Future Computational Models of Conversational Implicature

Nancy Green and Jill Fain Lehman

October 31, 1995

One avenue of research on conversational implicature (CI) in non-figurative discourse which should be addressed in the immediate future is to start implementing computational models of agents who, in the course of real-time conversational interaction, generate and comprehend conversational implicatures. This paper elaborates on this goal and describes how our current work serves as a stepping stone towards such a model.

We believe that future computational models of CI should aspire to meet the following goals:

- To be descriptively adequate models of CI. That is, to derive all and only the conversationally licensed inferences which a speaker intends for the hearer to recognize as intended in the current discourse context. To model the contextual dependence of CI. To model the cancelability as well as the loss of cancelability of CI's in different portions of the same dialogue. To be capable of handling different types of CI without unwarranted conflict.
- To embed the interpretation/generation of CI in a general computational model of conversational interaction such that the model accounts for why/when a speaker is motivated to use CI, and when/how a hearer derives a CI.
- To be a cognitively plausible model. That is, to be subject to real-time performance constraints and human resource limitations.
- To model diachronically how certain CI become semi-conventional, e.g., the association of particular linguistic forms with certain CI, making it problematic to use those forms to convey just their original meaning.
- To learn from experience how to interpret novel CI. That is, new classes of CI can be interpreted without requiring the model's builder to provide new licensing rules specific to those classes.
- To lead to new predictions about and insights into linguistic systems (including discourse) and how they interact with other cognitive capabilities (e.g. planning and reasoning) of embodied agents capable of interacting with their environment through linguistic and non-linguistic means.

The first of the above goals describes what we mean by a model of conversational implicature, as opposed to other types of discourse phenomena. We claim that the second goal is necessary for a descriptively adequate *computational* model of CI. Moreover, unless the model of CI is implemented as part of a general model of conversation, it would be hard to evaluate its success or limitations. The goal of cognitive plausibility is desirable for several practical reasons. First, since human conversationalists comprehend and generate CI, searching for a cognitively plausible model is a useful research strategy. Second, if the model is to be used in human-computer interaction, it is desirable for it to meet the same real-time performance requirements as its human conversational partners. The goals of modeling how certain CI become semi-conventional and of learning to interpret novel CI are desirable since it is unclear whether human model builders can anticipate all the situations in which CI may arise. Finally, the last goal seems desirable, not only for the sake of scientific discovery, but also to enable our models to better approximate human competence in the long term. For example, for a first approximation, a model of certain CI might be built which presupposes that the agent has acquired knowledge of principles of politeness. A more powerful model is one that explains how such principles arise, e.g., by learning through social interaction with other agents.

Given the above goals, we believe that there are several advantages to building models of CI within more general models such as NL-Soar.[1] Before supporting this claim, first, we describe NL-Soar. NL-Soar is a computational model of real-time natural language generation and comprehension implemented within the Soar architecture. (Soar is an architecture which has been proposed as a candidate model of general human cognition, and which has been implemented to provide a computational system for cognitive modeling.[2]) Previous versions of NL-Soar focused on the problems of real-time parsing [Lew93] and sentence generation [RL94]. Current areas of development include speech recognition, lexical semantics, integrating mechanisms used in sentence generation with those used in sentence comprehension, and discourse modeling, with the goal of building a system which can participate in conversation with humans or with other simulated agents within a task domain.

Most linguistic (including pragmatic) knowledge is represented in NL-Soar as operators which are applied to states to achieve goals, i.e. desired states.[3] For example, NL-Soar provides an operator, *acquire-turn*, to model the acquisition of the turn by a conversational participant. Language processing is driven by the current state of the situation model, which is a representation of relevant contextual knowledge (e.g., perceptual knowledge, intermediate linguistic representations, discourse context, and world knowledge retrieved from long-term memory).[4] During the application of an operator, NL-Soar may create subgoals. Thus, a problem may be solved by decomposition into a hierarchy of subgoals. An important property of the Soar architecture is that the act of problem-solving via subgoaling has a persistent effect upon the cognitive model, i.e., the model thereby acquires knowledge ("chunks") which may be applied in future situations.[5] For example, in generating a speaker's turn over a radio channel, NL-Soar may pursue a hierarchy of subgoals resulting in the speaker's decision to begin by making a self-introduction (e.g., *This is Dakota 204*) since the addressee might otherwise not be able to identify who is speaking. In subsequent situations with similar features, NL-Soar would be able to generate a self-introduction without subgoaling. Thus, learning is a continual side-effect of problem-solving. Finally, it should be noted that operator execution time in the Soar architecture can be used to verify if a Soar model's temporal performance is consistent with data about human performance.

Given the above goals, it would be advantageous to build models of CI within a model such as NL-Soar for the following reasons. First, doing so addresses the second goal.[6] Moreover, it enables the model of CI to share mechanisms used in other processes, linguistic or non-linguistic.

[1] Note that we are *not* claiming that NL-Soar's (or Soar's) approach is the *only* approach having these properties.

[2] For more information on Soar itself, see [New90, LLR96] or the Soar home page at http://www.isi.edu/soar/soar.html.

[3] Other linguistic knowledge is represented as elaborations and search control knowledge. Elaborations specify aspects of the current situation which are contingent upon other aspects of the current situation. Search control knowledge is used to guide selection of operators.

[4] Note that Soar supports an integrated representation of linguistic and nonlinguistic behavior, i.e., all permanent knowledge is represented uniformly, and all processing is driven by the current situation.

[5] "Chunks" are encoded in the same (symbolic) format as knowledge provided by the model's builder, and are open to inspection by the builder as well.

[6] Conversely, for NL-Soar to provide an adequate model of conversational behavior, it should incorporate a model of CI.

Second, to implement a model of CI within NL-Soar imposes the constraint of cognitive plausibility, i.e., addresses the third goal. For example, a crucial problem in interpreting CI in real-time is the control of inference, i.e., not wasting too much time on inferences which do not contribute to the comprehension of the CI. The Soar architecture permits inference to be constrained by knowledge. More specifically, persistent modifications of an agent's situation model can be achieved only through the application of an operator. In other words, the application of an operator specifies all and only the inferences, i.e. updates to the agent's current situation model, licensed in that context. The architecture permits only one operator to be applied at a time. If more than one operator is eligible to apply in the current situation, then explicit knowledge can be used to select an operator, i.e., to decide which operator to apply. This search control knowledge can be encoded by the model's builder or learned by the model itself.

It may be feasible to exploit NL-Soar's learning mechanism to address the next two goals. For example, currently NL-Soar learns syntactic and semantic comprehension operators while parsing novel inputs. These operators enable NL-Soar, on encountering similar expressions, to parse them much more efficiently using the results of its past experience instead of repeating the problem-solving process. One way of modeling how certain CI become associated with certain linguistic forms would be by use of learned operators combining syntactic, semantic, and pragmatic processing. NL-Soar's bias to use these learned operators before attempting problem-solving from first principles would account for the difficulty of recovering the original meaning of such forms.

As for the goal of interpreting novel types of CI without requiring the model builder to provide licensing rules specific to that class of CI, consider how the first author's computational model for interpreting and generating indirect answers to Yes-No questions [Gre94] could be extended to address this goal. In that model, interpretation and generation of indirect answers are modeled, respectively, as recognition and construction of the responder's discourse plan specification for a full answer. A full answer consists of the requested information, i.e. the direct answer, and possibly extra relevant information. A discourse plan specification relates a speaker's beliefs and discourse goals to his program of communicative actions. An indirect answer is the result of the responder providing only part of the planned response, but expecting that her full discourse plan specification will be recognized by the hearer. An important knowledge source in that model is a shared set of discourse plan operators encoding generic programs for conveying full answers. This knowledge source is used both in generation and interpretation, and is used for modeling both direct and indirect answers. In other words, the model does not employ a licensing rule specific to indirect answers. Instead an indirect answer is comprehended as a result of attempting to recognize the responder's discourse plan specification.

However, an unfortunate limitation of that type of model is that the set of discourse plan operators must be provided by the model's builder. On the other hand, human conversationalists seem to be able to comprehend many CI in real-time, suggesting an approach that does not require reasoning from first principles in all cases, i.e. an approach that uses some type of "compiled" knowledge. Currently, we are investigating how NL-Soar's learning mechanism might be used, as an alternative to providing a hand-crafted library of discourse plan operators, to acquire discourse planning knowledge which could be used, not only for discourse generation, but also for discourse comprehension. For example, the process of planning to produce a self-introduction results in the acquisition of a new, learned discourse operator. The same operator could be reused during interpretation to recognize a speaker's intention in uttering his name. Research within the Soar architecture on use of the operators which an agent uses to generate his own behavior for tracking other agents' behavior is described in [TR95]. We are currently investigating how to extend that approach for sets of learned operators, which would enable learned discourse planning operators to be used in discourse comprehension.

References

[Gre94] Nancy L. Green. *A Computational Model for Generating and Interpreting Indirect Answers*. PhD thesis, University of Delaware, 1994. Department of Computer and Information Sciences, Technical Report No. 95-05.

[Lew93] Richard Lawrence Lewis. *An Architecturally-based Theory of Human Sentence Comprehension*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1993. Available from Computer Science Department as Technical Report CMU-CS-93-226.

[LLR96] J. F. Lehman, J. E. Laird, and P. S. Rosenbloom. A gentle introduction to Soar, an architecture for human cognition. In S. Sternberg and D. Scarborough, editors, *Invitation to Cognitive Science*, volume 4. MIT Press, 1996.

[New90] Allen Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1990.

[RL94] R. Rubinoff and J. F. Lehman. Real-time natural language generation in NL-Soar. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 199–206, Kennebunkport, Maine, 1994.

[TR95] Milind Tambe and Paul S. Rosenbloom. RESC: An approach for real-time, dynamic agent tracking. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages ??–??, Montreal, Canada, 1995.