Teaching Dialogue to Interdisciplinary Teams through Toolkits

Justine Cassell
Technology and Social Behavior
Northwestern University
justine@northwestern.edu

Matthew Stone
Computer Science and Cognitive Science
Rutgers University
matthew.stone@rutgers.edu

Abstract

We present some lessons we have learned from using software infrastructure to support coursework in natural language dialogue and embodied conversational agents. We have a new appreciation for the differences between coursework and research infrastructure—supporting teaching may be harder, because students require a broader spectrum of implementation, a faster learning curve and the ability to explore mistaken ideas as well as promising ones. We outline the collaborative discussion and effort we think is required to create better teaching infrastructure in the future.

1 Introduction

Hands-on interaction with dialogue systems is a necessary component of a course on computational linguistics and natural language technology. And yet, it is clearly impracticable to have students in a quarter-long or semester-long course build a dialogue system from scratch. For this reason, instructors of these courses have experimented with various options to allow students to view the code of a working dialogue system, tweak code, or build their own application using a dialogue system toolkit. Some popular options include the NLTK (Loper and Bird, 2002), CSLU (Cole, 1999), Trindi (Larsson and Traum, 2000) and Regulus (Rayner et al., 2003) toolkits. However, each of these options has turned out to have disadvantages. Some of the toolkits require too much knowledge of linguistics for the average computer science student, and vice-versa, others require too much programming for the average linguist. What is needed is an extensible dialogue toolkit that allows easy application building for beginning students, and more sophisticated access to, and tweakability of, the models of discourse for advanced students.

In addition, as computational linguists become increasingly interested in the role of non-verbal behavior in discourse and dialogue, more of us would like to give our students exposure to models of the interaction between language and nonverbal behaviors such as eye gaze, head nods and hand gestures. However, the available dialogue system toolkits either have no graphical body or if they do have (part of) a body—as in the case of the CSLU toolkit—the toolkit does not allow the implementation of alternative models of body–language interaction.

We feel, therefore, that there is a need for a toolkit that allows the beginning graduate student—who may have some computer science or some linguistics background, but not both—to implement a working embodied dialogue system, as a way to experiment with models of discourse, dialogue, collaborative conversation and the interaction between verbal and nonverbal behavior in conversation. We believe the community as a whole must be engaged in the design, implementation and fielding of this kind of educational software. In this paper, we survey the experience that has led us to these conclusions and frame the broader discussion we hope the TNLP workshop will help to further.

Proceedings of the Second ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL, pages 9–14,
Ann Arbor, June 2005.

2 Our Courses

Our perspective in this paper draws on more than fifteen course offerings at the graduate level in discourse and dialogue over the years. Justine Cassell's course *Theories and Technologies of Human Communication* is documented on the web here:

http://www.soc.northwestern.edu/justine/discourse

Matthew Stone's courses *Natural Language Processing* and *Meaning Machines*[1] are documented here:

http://www.cs.rutgers.edu/~mdstone/class/533-spring-03/
http://www.cs.rutgers.edu/~mdstone/class/672

These courses are similar in perspective. All address an extremely diverse and interdisciplinary audience of students from computer science, linguistics, cognitive science, information science, communication, and education. The typical student is a first or second-year PhD student with a serious interest in doing a dissertation on human-computer communication or in enriching their dissertation research with results from the theory or practice of discourse and dialogue. All are project courses, but no programming is required; projects may involve evaluation of existing implementations or the prospective design of new implementations based on ongoing empirical research. Nevertheless, the courses retain the dual goals that students should not only understand discourse and the theory of pragmatics, but should also understand how the theory is implemented, either well enough to talk intelligently about the implementation or, if they are computer scientists, to actually carry it out.

As befits our dual goals, our courses all involve a mix of instruction in human-human dialogue and human-computer dialogue. For example, Cassell begins her course with a homework where students collect, transcribe and analyze their own recordings of face-to-face conversation. Students are asked to discuss what constitutes a sufficient record of discourse, and to speculate on what the most challenging processing issues would be to allow a computer to replace one of the participants. Computer scientists definitely have difficulty with this aspect of the course—only fair, since they are at the advantage when it comes to implementation. But computer scientists see the value in the exercise: even if they do not believe that interfaces should be designed to act like people, they still recognize that well-designed interactive systems must be ready to handle the kinds of behaviors people actually carry out. And hands-on experience convinces them that behavior in human conversation is both rich and surprising. The computer scientists agree—after turning in impoverished and uninformed "analyses" of their discourse for a brutal critique—that they will never look at conversation the same way again.

Our experience suggests that we should be trying to give students outside computer science the same kind of eye-opening hands-on experience with technology. For example, we have found that linguists are just as challenged and excited by the discipline of technology as computer scientists are by the discipline of empirical observations. Linguists in our classes typically report that successful engagement with technology "exposes a lot of details that were missing from my theoretical understanding that I never would have considered without working through the code". Nothing is better at bringing out the assumptions you bring to an analysis of human-human conversation than the thought experiment of replacing one of the participants by something that has to struggle consciously to understand it—a space alien, perhaps, or, more realistically, an AI system. We are frustrated that no succinct assignment, comparable to our transcription homework, yet exists that can reliably deliver this insight to students outside computer science.

3 Framing the Problem

Our courses are not typical NLP classes. Our treatment of parsing is marginal, and for the most part we ignore the mainstays of statistical language processing courses: the low-level technology such as finite-state methods; the specific language processing challenges for machine learning methods; and "applied" subproblems like named entity extraction, or phrase chunking. Our focus is almost exclusively on high-level and interactional issues, such as the structure of discourse and dialogue, information structure, intentions, turn-taking, collaboration,

[1] The catchy title is the inspiration of Deb Roy at MIT.

reference and clarification. Context is central, and under that umbrella we explicitly discuss both the perceptual environment in which conversation takes place and the non-verbal actions that contribute to the management of conversation and participants' real-world collaborations.

Our unusual focus means that we can not readily take advantage of software toolkits such as NLTK (Loper and Bird, 2002) or Regulus (Rayner et al., 2003). These toolkits are great at helping students implement and visualize the fundamentals of natural language processing—lexicon, morphology, syntax. They make it easy to experiment with machine learning or with specific models for a small scale, short course assignment in a specific NLP module. You can think of this as a "horizontal" approach, allowing students to systematically develop a comprehensive approach to a single processing task. But what we need is a "vertical" approach, which allows students to follow a specific choice about the representation of communicative behaviors or communicative functions all the way through an end-to-end dialogue system. We have not succeeded in conceptualizing how a carefully modularized toolkit would support this kind of student experience.

Still, we have not met with success with alternative approaches, either. As we describe in Section 3.1, our own research systems may allow the kinds of experiments we want students to carry out. But they demand too much expertise of students for a one-semester course. In fact, as we describe in Section 3.2, even broad research systems that come with specific support for students to carry out a range of tasks may not enable the specific directions that really turn students on to the challenge of discourse and dialogue. However, our experience with implementing dedicated modules for teaching, as described in Section 3.3, is that the lack of synergy with ongoing research can result in impoverished tools that fail to engage students. We don't have the tools we want—but our experience argues that we think the tools we really want will be developed only through a collaborative effort shared across multiple sites and broadly engaged with a range of research issues as well as with pedagogical challenges.

3.1 Difficulties with REA and BEAT

Cassell has experimented with the use of her research platforms REA (Cassell et al., 1999) and BEAT (Cassell et al., 2001) for course projects in discourse and dialogue. REA is an embodied conversational agent that interacts with a user in a real estate agent domain. It includes an end-to-end dialogue architecture; it supports speech input, stereo vision input, conversational process including presence and turn-taking, content planning, the context-sensitive generation of communicative action and the animated realization of multimodal communicative actions. BEAT (the behavior expression animation toolkit), on the other hand, is a module that fits into animation systems. It marks up text to describe appropriate synchronized nonverbal behaviors and speech to realize on a humanoid talking character.

In teaching dialogue at MIT, Cassell invited students to adapt her existing REA and BEAT system to explore aspects of the theory and practice of discourse and dialogue. This led to a range of interesting projects. For example, students were able to explore hypothetical differences among characters—from virtual "Italians" with profuse gesture, to virtual children whose marked use of a large gesture space contrasted with typical adults, to characters who showed new and interesting behavior such as the repeated foot-tap of frustrated condescension. However, we think we can serve students much better. Many of these projects were accomplished only with substantial help from the instructor and TAs, who were already extremely familiar with the overall system. Students did not have time to learn how to make these changes entirely on their own.

The foot-tapping agent is a good example of this. To add foot-tapping is a paradigmatic "vertical" modification. It requires adding suitable context to the discourse state to represent uncooperative user behavior; it requires extending the process for generating communicative actions to detect this new state and schedule an appropriate behavioral response; and then it requires extending the animation platform to be able to show this behavior. BEAT makes the second step easy—as it should be—even for linguistics students. To handle the first and third steps, you would hope that an interdisciplinary team containing a communication student and a computer sci-

ence student would be able to bring the expertise to design the new dialogue state and the new animated behavior. But that wasn't exactly true. In order to add the behavior to REA, students needed not only background in the relevant technology—like what a computer scientist would learn in a general human animation class. To add the behavior, students also needed to know how this technology was realized in our particular research platform. This proved too much for one semester.

We think this is a general problem with new research systems. For example, we think many of the same issues would arise in asking students to build a dialogue system on top of the Trindi toolkit in a one semester course.

3.2 Difficulties with the CSLU toolkit

In Fall 2004, Cassell experimented with using the CSLU dialogue toolkit (Cole, 1999) as a resource for class projects. This is a broad toolkit to support research and teaching in spoken language technology. A particular strength of the toolkit is its support for the design of finite-state dialogue models. Even students outside computer science appreciated the toolkit's drag-and-drop interface for scripting dialogue flow. For example, with this interface, you can add a repair sequence to a dialogue flow in one easy step. However, the indirection the toolkit places between students and the actual constructs of dialogue theory can by quite challenging. For example, the finite-state architecture of the CSLU toolkit allows students to look at floor management and at dialogue initiative only indirectly: specific transition networks encode specific strategies for taking turns or managing problem solving by scheduling specific communicative functions and behaviors.

The way we see it, the CSLU toolkit is more heavily geared towards the rapid construction of particular kinds of research prototypes than we would like in a teaching toolkit. Its dialogue models provide an instructive perspective on actions in discourse, one that nicely complements the perspective of DAMSL (Core and Allen, 1997) in seeing utterances as the combined realization of a specific, constrained range of communicative functions. But we would like to be able to explore a range of other metaphors for organizing the information in dialogue. We would like students to be able to realize models of face-to-face dialogue (Cassell et al., 2000), the information-state approach to domain-independent practical dialogue (Larsson and Traum, 2000), or approaches that emphasize the grounding of conversation in the specifics of a particular ongoing collaboration (Rich et al., 2001). The integration of a talking head into the CSLU toolkit epitomizes these limitations with the platform. The toolkit allows for the automatic realization of text with an animated spoken delivery, but does not expose the model to programmers, making it impossible for programmers adapt or control the behavior of the face and head.

We think this is a general problem with platforms that are primarily designed to streamline a particular research methodology. For example, we think many of the same issues would arise in asking students to build a multimodal behavior realization system on top of a general-purpose speech synthesis platform like Festival (Black and Taylor, 1997).

3.3 Difficulties with TAGLET

At this point, the right solution might seem to be to devise resources explicitly for teaching. In fact, Stone advocated more or less this at the 2002 TNLP workshop (2002). There, Stone motivated the potential role for a simple lexicalized formalism for natural language syntax, semantics and pragmatics in a broad NLP class whose emphasis is to introduce topics of current research.

The system, TAGLET, is a context-free tree-rewriting formalism, defined by the usual complementation operation and the simplest imaginable modification operation. This formalism may in fact be a good way to present computational linguistics to technically-minded cognitive science students—those rare students who come with interest and experience in the science of language as well as a solid ability to program. By implementing a strong competence TAGLET parser and generator students simultaneously get experience with central computer science ideas—data structures, unification, recursion and abstraction—and develop an effective starting point for their own subsequent projects.

However, in retrospect, TAGLET does not serve to introduce students outside computer science to the distinctive insights that come from a computational approach to language use. For one thing, to reach a broad audience, it is a mistake to focus on repre-

sentations that programmers can easily build at the expense of representations that other students can easily understand. These other students need visualization; they need to be able to see what the system computes and how it computes it. Moreover, these other students can tolerate substantial complexity in the underlying algorithms if the system can be understood clearly and mechanistically in abstract terms. You wouldn't ask a computer scientist to implement a parser for full tree-adjoining grammar but that doesn't change the fact that it's still a perfectly natural, and comprehensible, algorithmic abstraction for characterizing linguistic structure.

Another set of representations and algorithms might avoid some of these problems. But a new approach could not avoid another problem that we think applies generally to platforms that are designed exclusively for teaching: there is no synergy with ongoing research efforts. Rich resources are so crucial to any computational treatment of dialogue: annotated corpora, wide-coverage grammars, plan-recognizers, context models, and the rest. We can't afford to start from scratch. We have found this concretely in our work. What got linguists involved in the computational exploration of dialogue semantics at Rutgers was not the special teaching resources Stone created. It was hooking students up with the systems that were being actively developed in ongoing research (DeVault et al., 2005). These research efforts made it practical to provide students with the visualizations, task and context models, and interactive architecture they needed to explore substantive issues in dialogue semantics. Whatever we do will have to closely connect teaching and our ongoing research.

4 Looking ahead

Our experience teaching dialogue to interdisciplinary teams through toolkits has been humbling. We have a new appreciation for the differences between coursework and research infrastructure—supporting teaching may be harder, because students require a broader spectrum of implementation, a faster learning curve and the ability to explore mistaken ideas as well as promising ones. But we increasingly think the community can and should come together to foster more broadly useful resources for teaching.

We have reframed our ongoing activities so that we can find new synergies between research and teaching. For example, we are currently working to expand the repertoire of animated action in our freely-available talking head RUTH (DeCarlo et al., 2004). In our next release, we expect to make different kinds of resources available than in the initial release. Originally, we distributed only the model we created. The next version will again provide that model, along with a broader and more useful inventory of facial expressions for it, but we also want the new RUTH to be more easily extensible than the last one. To do that, we have ported our model to a general-purpose animation environment (Alias Research's Maya) and created software tools that can output edited models into the collection of files that RUTH needs to run. This helps achieve our objective of quickly-learned extensibility. We expect that students with a background in human animation will bring experience with Maya to a dialogue course. (Anyway, learning Maya is much more general than learning RUTH!) Computer science students will thus find it easier to assist a team of communication and linguistics students in adding new expressions to an animated character.

Creating such resources to span a general system for face-to-face dialogue would be an enormous undertaking. It could happen only with broad input from those who teach discourse and dialogue, as we do, through a mix of theory and practice. We hope the TNLP workshop will spark this kind of process. We close with the questions we'd like to consider further. What kinds of classes on dialogue and discourse pragmatics are currently being offered? What kinds of audiences do others reach, what goals do they bring, and what do they teach them? What are the scientific and technological principles that others would use toolkits to teach and illustrate? In short, what would your dialogue toolkit make possible? And how can we work together to realize both our visions?

5 Acknowledgments

Thanks to Doug DeCarlo, NSF HLC 0308121.

References

Alan Black and Paul Taylor. 1997. Festival speech synthesis system. Technical Report HCRC/TR-83, Human Communication Research Center. http://www.cstr.ed.ac.uk/projects/festival/.

J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. 1999. Embodiment in conversational characters: Rea. In *CHI 99*, pages 520–527.

Justine Cassell, Tim Bickmore, Lee Campbell, Hannes Vilhjalmsson, and Hao Yan. 2000. Human conversation as a system framework. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 29–63. MIT Press, Cambridge, MA.

Justine Cassell, Hannes Vilhjálmsson, and Tim Bickmore. 2001. BEAT: the behavioral expression animation toolkit. In *SIGGRAPH*, pages 477–486.

Ron Cole. 1999. Tools for research and education in speech science. In *Proceedings of the International Conference of Phonetic Sciences*. http://cslu.cse.ogi.edu/toolkit/.

Mark G. Core and James F. Allen. 1997. Coding dialogs with the DAMSL annotation scheme. In *Working Notes of AAAI Fall Symposium on Communicative Action in Humans and Machines*. http://www.cs.rochester.edu/research/cisd/resources/damsl/.

Douglas DeCarlo, Corey Revilla, Matthew Stone, and Jennifer Venditti. 2004. Specifying and animating facial signals for discourse in embodied conversational agents. *Journal of Visualization and Computer Animation*. http://www.cs.rutgers.edu/~village/ruth/.

David DeVault, Anubha Kothari, Natalia Kariaeva, Iris Oved, and Matthew Stone. 2005. An information-state approach to collaborative reference. In *ACL Proceedings Companion Volume (interactive poster and demonstration track)*. http://www.cs.rutgers.edu/~mdstone/pointers/collabref.html.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340. http://www.ling.gu.se/projekt/trindi/.

Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. http://nltk.sourceforge.net.

Manny Rayner, Beth Ann Hockey, and John Dowding. 2003. An open source environment for compiling typed unification grammars into speech recognisers. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computation Linguistics (interactive poster and demo track)*. http://sourceforge.net/projects/regulus.

C. Rich, C. L. Sidner, and N. Lesh. 2001. COLLAGEN: applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 22:15–25.

Matthew Stone. 2002. Lexicalized grammar 101. In *ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL*, pages 76–83. http://www.cs.rutgers.edu/~mdstone/class/taglet/.