Natural Language Processing for computer-supported instruction

Barbara Di Eugenio
Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA
bdieugen@cs.uic.edu

Abstract Computer-supported instruction includes many applications whose goal is instruction and education: from animated agents that execute instructions to intelligent tutoring systems, from systems that produce instructional manuals to systems that facilitate student collaboration. In this paper, I present some of the research I have undertaken in recent years to build such systems and their Natural Language interfaces. My research includes linguistic investigation, computational modeling and system evaluation. All the work presented is supported by rigorous corpus analysis.

Introduction

Many computer applications are concerned with interpreting or producing instructions and fostering education. For example, 1) animated agents that execute instructions [Webber et al.1995]; 2) systems that automatically produce instructional text like Excerpt A in Figure 1 [Paris et al.1995]; 3) intelligent tutoring systems (ITSs) that help a student master a certain subject [Anderson et al.1995; Schulze et al.2000]; 4) systems that facilitate student collaboration [Soller2001]. Of the four examples just mentioned, only (2) is a Natural Language (NL) system proper, the others are not. For example, the animated agents in (1) may be instructed via a menu; an ITS may provide feedback to the student via graphics. However, all the systems in (1), (3), and (4) potentially benefit from a Natural Language interface. For instance, consider the *learning gain*, i.e., how much a student learns in a certain setting σ. Generally, the learning gain is the difference between the student's score on the same test, before and after σ. It has been shown that the learning gain of students interacting with an ITS is halfway between the learning gain of students that were exposed to the material in the usual classroom setting (lowest) and students that interact with a human tutor (highest) [Anderson et al.1995]. The difference in learning gain between students interacting with an ITS and those interacting with a human tutor is attributed to the fact that tutor and student are engaged in a conversation [Fox1993; Graesser, Person, and Magliano1995; Hume et al.1996]. Thus, research on the next generation of ITSs explores the usage of NL as one of the keys to bridge the gap between current ITSs and their human counterparts [Evens et al.1993; Rosé and Freedman2000; Aleven2001; Graesser et al.2001].

Figure 1 presents two samples of NL instructions: the first is taken from an on-line help, the second from a tutoring dialogue. These two examples illustrate some of the problems faced by systems that have to support the interpretation and generation of instructions.

Instructions in a technical manual, on-line help or home repair manual such as those under (A) in Figure 1 [Delin et al.1994; Vander Linden and Martin1995; Di Eugenio and Webber1996] teach how to perform a task mainly by describing the steps to be performed. They often include descriptions of what will happen as the result of a certain step (e.g. *the window expands to show the Alarm options* in step 2) as a way to inform the user whether s/he is on the right track. The structure of the text closely reflects the structure of the

A. Excerpt from electronic help on the Calendar Manager (OpenWindows, ca. 1995)

To Set Up a Reminder
Calendar Manager will send a mail message at a predetermined time.

1. Choose Edit→Appointment to open the Appointment Editor window.
2. Click SELECT on Full Size. The window expands to show the Alarm options.
3. Click SELECT on Mail. Calendar Manager activates the options for hrs (hours) and Mail To.
4. Set the number of hours.
5. Type the electronic-mail addresses after Mail To.
6. Set up the rest of the appointment for which you want a mail reminder sent.
7. Calendar Manager will mail a message to the Mail To: list at the time you specify.

B. Excerpt from a tutoring dialogue on basic electricity and electronics

TUTOR: Can you understand why you would need to remove a wire from the circuit before attaching the leads of the ammeter?
STUDENT: Yes you cannot work with the power on and the power needs to be off or disconnected
TUTOR: So if the wire was not removed, you think the power would be on in the circuit eventhough the switch is open?
STUDENT: The power would not be on but it is possible for energy to still be going through the wire
TUTOR: Actually, that is not the case. Current can only flow through a circuit that is complete. So there is no way any current could still be flowing through that wire. Can you think of any other reason why you would need to remove a wire before connecting the leads of the ammeter to the circuit?

Figure 1: Two different ways of providing instruction

task, as has long been known regarding task-oriented discourse [Grosz and Sidner1986].

Tutorial dialogues such as (B) in Figure 1[1] present a completely different approach to instruction. It is apparent that the language and rhetorical structure (i.e., the extended structure of the discourse / dialogue[2]) become more complex and less structured from (A) to (B). Good tutors do not simply provide students with the correct information, e.g. the sequence of steps to be executed, they help the student build the correct knowledge by themselves [Fox1993; Graesser, Person, and Magliano1995; Hume et al.1996; Rosé et al.2001]. In fact, there is evidence that it is exactly the interaction and the collaboration between tutor and student that fosters learning [Chi et al.2001].

Some of the problems that Natural Language Processing (NLP) for computer-supported instruction has to face are common to NLP interfaces in general, for example inferring relations between sentences and solving ambiguities. In Excerpt A in Figure 1 a system will have to understand that the first 6 numbered items represent subsequent steps in a procedure, but that the seventh represents an effect that will take place in some unspecified future. Some issues are unique to instructional text. For example, both interpretation and generation of instructions are concerned with the different ways in which a specific relation between actions is expressed, and which additional meanings such a relation may carry. In [Di Eugenio1998] I showed that purpose clauses (subordinated clauses introduced by *to*, as in step 1 in Excerpt A in Fig. 1) provide constraints on the interpretation of the action described in the main clause. And finally, some issues are unique to the specific instructional application. For example, uncovering the tutoring strategies that human tutors use is specific to ITSs.

In the past few years I have worked with collaborators and students on a broad range of applications all

[1] This excerpt is taken from a dialogue in the BEESIM project corpus [Rosé, Di Eugenio, and Moore1999; Rosé et al.2001].
[2] *Discourse* is meant as a monologue, as opposed to *dialogue*, which involves two or more participants.

concerned with instruction. Specifically, I have worked on (1) animated agents executing NL instructions, on (2) systems that generate instructional text and on (3) NL interfaces to ITSs. In this paper, I will focus on NL interfaces for ITSs. I will also discuss some work on modeling collaboration in dialogue which, even if not directly applied to instructional text, is relevant to ITSs because the tutoring dialogue can be considered as a collaboration between tutor and student to build a shared understanding. Details on all the work described here, and on the other projects that I don't have space to cover can be found at `http://www.cs.uic.edu/~bdieugen/research.html`.

A dimension common to all of my work is the methodology I follow, which can be dubbed as "mark-up, mine, implement, evaluate". This general methodology has emerged for discourse / dialogue processing in the '90s [Walker and Moore1997; Chu-Carroll and Green1998; Walker1999]:

Mark-up. Because of the complexity of the phenomena of interest, data analysis cannot be based directly on raw linguistic data. Rather, it requires the development and application of appropriate coding schemes, i.e., set of labels representing features thought to correlate with the phenomenon of interest.

Mine. The second phase concerns extracting information from the annotated corpus, via statistical techniques or machine learning. The purpose is to verify hypotheses (e.g., in tutoring dialogues, does the student ask many questions?), and to find linguistic correlates of higher-level phenomena, such as proposals.

Computational Modeling. The third phase regards the development of computational frameworks based on the information extracted from the corpus.

Evaluate. Finally, the fourth phase concerns evaluating the implemented systems. The evaluation of dialogue systems is still an area of active research.

Natural Language Processing and Intelligent Tutoring Systems[3]

In the last few years I have been involved in a number of projects dealing with Natural Language interfaces for Intelligent Tutoring Systems. The main goal of these projects is to uncover how a human tutor interacts with a student, in order to inform the generation of the natural language feedback that the ITS provides (but see [Rosé, Di Eugenio, and Moore1999] for some preliminary work on interpreting the student's input). Among the many issues that need to be addressed for an ITS to generate natural language feedback are:

- Which tutoring strategy(ies) should the system adopt? If human tutoring were well understood, then the answer would be easy: choose the most effective strategies. There is evidence that just providing explanations is not effective, while *scaffolding* is. However, the definition of scaffolding is at a very high level and cannot be directly operationalized as algorithms. For instance, one definition of scaffolding is *any kind of guidance that is more than a confirmatory or negative feedback* [Chi et al.2001].

- What features of conventional dialogue carry over to tutoring dialogues? The latter reverse some of the normal conventions of conversation. For example, a question is normally asked when the person who asks doesn't know the piece of information the question is about. In tutoring dialogues, tutors ask questions to assess the student's misconceptions and to help them build the appropriate knowledge (see the two tutor's questions in Excerpt B, Figure 1).

[3]The projects described in this section have all been supported by the Office of Naval Research, Cognitive, Neural and Biomolecular S&T Division, via grants N00014-91-J-1694 to Johanna D. Moore, N00014-93-I-0812 to Johanna D. Moore and Barbara Di Eugenio, and N00014-99-1-0930 and N00014-00-1-0640 to Barbara Di Eugenio.

- Language generation systems are generally built for monologue. They rely on planners that would plan e.g. the steps of a tutoring strategy at the onset of the dialogue. In a dialogue however, some unexpected aspects in the respondent's turn may cause the original plan to be changed. This is particularly true of tutoring dialogues, in which the student's responses may cause the tutor to change tutoring strategy.
- The language should be natural and effective. It may need to include equations and drawings, as appropriate.
- Evaluation. We need to ascertain that the language interface positively impacts the students' learning.

A first project I was involved in concerned how to generate *discourse cues* (discourse markers like *now* and coordinating and subordinating conjunctions like *and* and *since*) in tutoring explanations. After annotating a corpus of tutorial monologues for features thought to affect the generation of cue phrases, we used machine learning to derive the conditions under which cue phrases are included in the explanation [Di Eugenio, Moore, and Paolucci1997].

The two later projects I will describe shortly (BEESIM and DIAG-NLP) have broader goals, from both a theoretical and a practical perspective. They both aim at building NL interfaces for VIVIDS-based tutors. VIVIDS is an authoring environment [Munro1994] to build ITSs. VIVIDS based tutors deliver instruction and practice in the context of graphical simulations. Authors build interactive graphical models of complex systems, and build lessons based on these graphical models (see Figure 2).

BEESIM `(http://www.cogsci.ed.ac.uk/~jmoore/tutoring/)`

The goal of the BEESIM project[4] is to operationalize the notion that students learn best when they construct knowledge by themselves [Di Eugenio, Rosè, and Moore1998; Rosé, Di Eugenio, and Moore1999; Core, Zinn, and Moore2000; Rosé et al.2001]. Thus, BEESIM seeks to answer the following questions:

1. What techniques are used by expert human tutors to help students construct knowledge?
2. Under what circumstances do tutors use each type of technique?
3. How are these techniques realized via natural language (especially keyboard-keyboard) dialogue?
4. How can these techniques be implemented in a computer-based tutoring environment?

To answer these questions, we collected human tutoring protocols in the context of a web-based course on basic electricity and electronics (BE&E) originally developed with the VIVIDS authoring tool [Munro1994] at the Navy Personnel Research and Development Center in San Diego, CA. The curriculum consists of four lessons and six laboratories covering basic concepts of current, voltage, resistance, and power. Each lesson consists of between 10 and 25 pages of instructional text and graphical illustrations displayed in a Netscape window. After each lesson, the student was presented with one or two laboratories designed to test and reinforce the notions discussed in the lesson. To perform the laboratory assignment, the student interacts with a simulated electronic workbench via a point-and-click interface. Figure 2 illustrates the laboratory for measuring DC voltage.

We collected keyboard-keyboard dialogues between a student going through the BE&E curriculum and a tutor. While the student interacted with the system, the video signal to the student's monitor was split so that a tutor sitting behind a partition could monitor the student's progress. The student and tutor communicated via a chat window. The dialogues were collected under two conditions, which we called socratic and didactic. In the socratic condition, the tutor was instructed to prompt the student with as

[4]Led by Johanna D. Moore, and with Carolyn Penstein Rosé and David Allbritton.

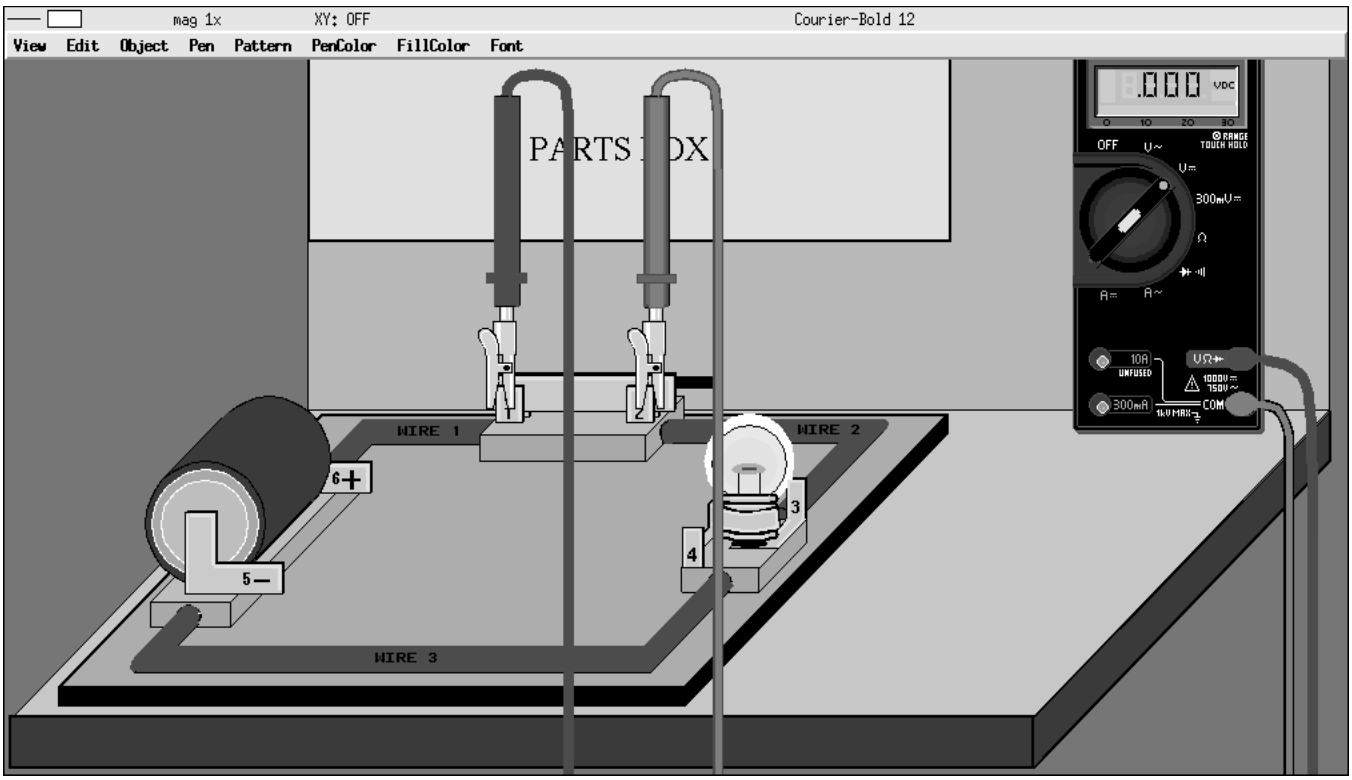


Figure 2: Simulation Window in BEESIM

little information as possible. The goal was for the student to independently construct as much knowledge as possible. In the didactic condition, the tutor was instructed to explain what she felt the student needed to know in order to proceed and then query the student in order to test for understanding. Students in the socratic condition had a higher average learning gain than students in the didactic condition, even if the average score on the test before the student goes through the curriculum is lower for students in the socratic condition than for students in the didactic condition [Rosé et al.2001].

Current work on the project includes a detailed analysis of the collected dialogues and the implementation of the dialogue architecture informed by the data collection [Core, Zinn, and Moore2000].

DIAG-NLP (http://www.cs.uic.edu/~bdieugen/tut-dial.html)

Whereas BEESIM squarely attacks the problem of uncovering the most effective tutoring strategies and their manifestation in dialogue, the project DIAG-NLP [Di Eugenio and Trolio2000; Di Eugenio et al.2001] takes a complementary approach. DIAG-NLP simplifies the problem of NL generation for an ITS in order to rapidly improve the feedback an existing ITS provides; it also aims at systematically evaluating the effectiveness of such an approach. More specifically, the goals of the project DIAG-NLP are:

1. To assess whether simple Natural Language Generation (NLG) is effective in improving an interface to an ITS.

2. To evaluate the "added value" of an NL interface to an ITS.

3. To use the results of (1) and (2), and a constrained data collection, to inform the development of a more sophisticated interface.

We took this approach for two reasons. First, we want to understand what can be accomplished by interfacing an NL generator to an ITS taken as a black box. We focus on understanding whether the ITS predefined tutoring strategies can be left as they are, or whether the dialogue strategies and the original tutoring strategies may become at odds with each other. Second, we are interested in finding out what is the "added value" of an NL interface to an ITS. One way to do so is to compare a system that does not use NL techniques to a version of the same system that uses NL.

The underlying ITS we interface to is built within DIAG [Towne1997], a shell to build ITSs that teach students to troubleshoot complex artifacts and systems, such home heating and circuitry. DIAG in turn builds on the VIVIDS authoring tool mentioned above. A typical session with a DIAG application presents the student with a series of troubleshooting problems of increasing difficulty. Figure 3 shows one of the graphical views in a DIAG application that teaches how to troubleshoot a home heating system. The subsystem being displayed is the furnace system. At any point, the student can consult the built-in tutor via the Consult menu, that pops up when the student clicks the button labelled Consult (see Figure 3).

After deciding which content to communicate, the original DIAG system (DIAG-orig) uses very simple templates to assemble the text to present to the student. The result is that the feedback that DIAG provides is repetitive, both as a sequence of replies to requests for feedback, and within each verbal feedback. In many cases, the feedback presents a single long list of many parts. This problem is compounded by the fact that most DIAG applications involve complex systems with many parts. Although there are different levels of description in the system model, and hierarchies of objects, the verbal feedback is almost always in terms of individual units. The top part of Figure 4 shows the reply originally provided by DIAG to a request of information regarding the "Visual Combustion Check".

We set out to improve on DIAG's feedback mechanism by applying aggregation rules, i.e., rules that specify how to parcel a number of propositions into sentences. For example, a long list of parts can be broken down by classifying each of these parts in to one of several smaller lists and then presenting the student with this set of lists. The bottom part of Figure 4 shows our aggregation rules at work. We interfaced DIAG to

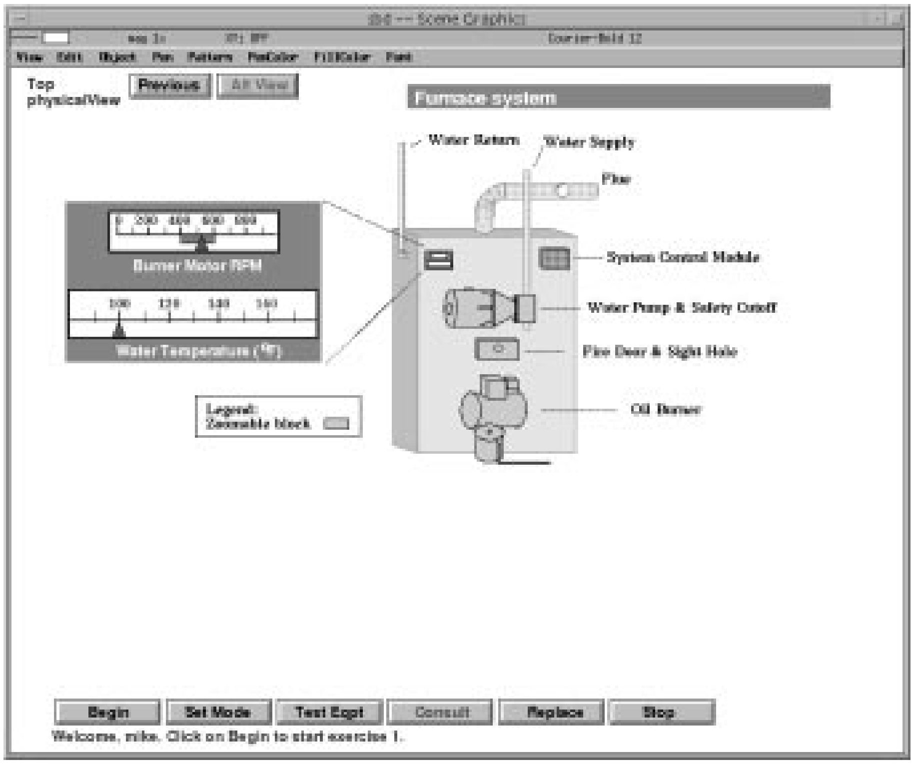
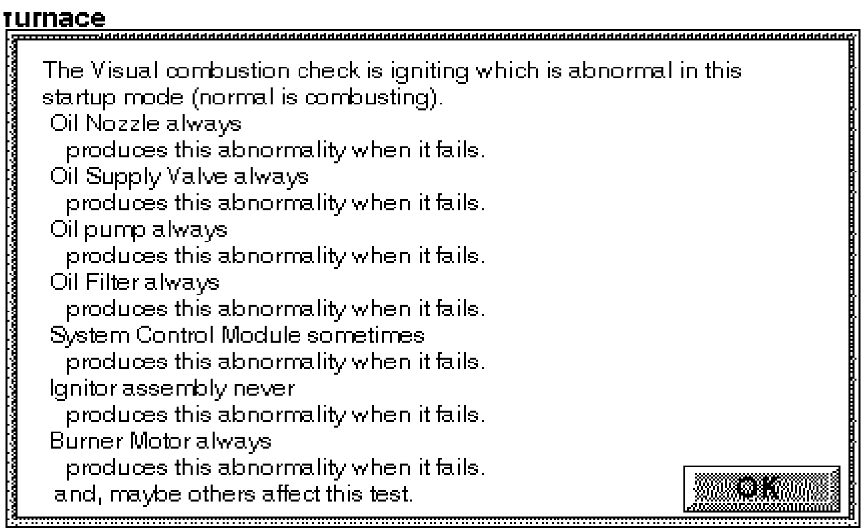


Figure 3: A screen from a DIAG application on home heating

the EXEMPLARS generator from CoGenTex [White and Caldwell1998], and implemented the aggregation rules via EXEMPLARS. The revised output groups the parts under discussion by the system modules that contain them (Oil Burner and Furnace System), and by the likelihood that a certain part causes the observed symptoms. Notice how the *Ignitor Assembly* is singled out in the revised answer. Among all mentioned units, it is the only one that cannot cause the symptom. This fact is lost in the original answer.

Intuitively, the contrast between the feedback produced by DIAG-orig and by DIAG-NLP (top and bottom in Figure 4) suggests that even simple aggregation rules dramatically improve the language feedback. To provide a real assessment of this claim, we conducted an empirical evaluation. Two groups of students interacted with DIAG-orig and with DIAG-NLP respectively. We computed several metrics, such as how many problems the student solved, time on problem, etc. We also administered the students a questionnaire, to test the student's understanding of the domain, and to ask the subject to rate the system's feedback along four dimensions such as friendliness. We found that on the whole DIAG-NLP outperforms DIAG-orig, i.e., that whereas there are almost no significant effect on individual measures, there is a cumulative effect in favor of DIAG-NLP. This cumulative effect is computed via the binomial cumulative distribution function [Di Eugenio et al.2001].

Thus, we showed that even simple aggregation rules are effective in improving an ITS's language feedback. However, while the aggregation rules we implemented appear to be plausible, they have no empirical foundation. Thus, we conducted a data collection effort to understand how a human tutor may verbalize a collection of facts, i.e., what sort of aggregation happens naturally. We collected interactions between students interacting with the same DIAG application we have previously discussed and human tutors. In this experiment the tutor and the student are in different rooms, sharing images of the same DIAG tutoring



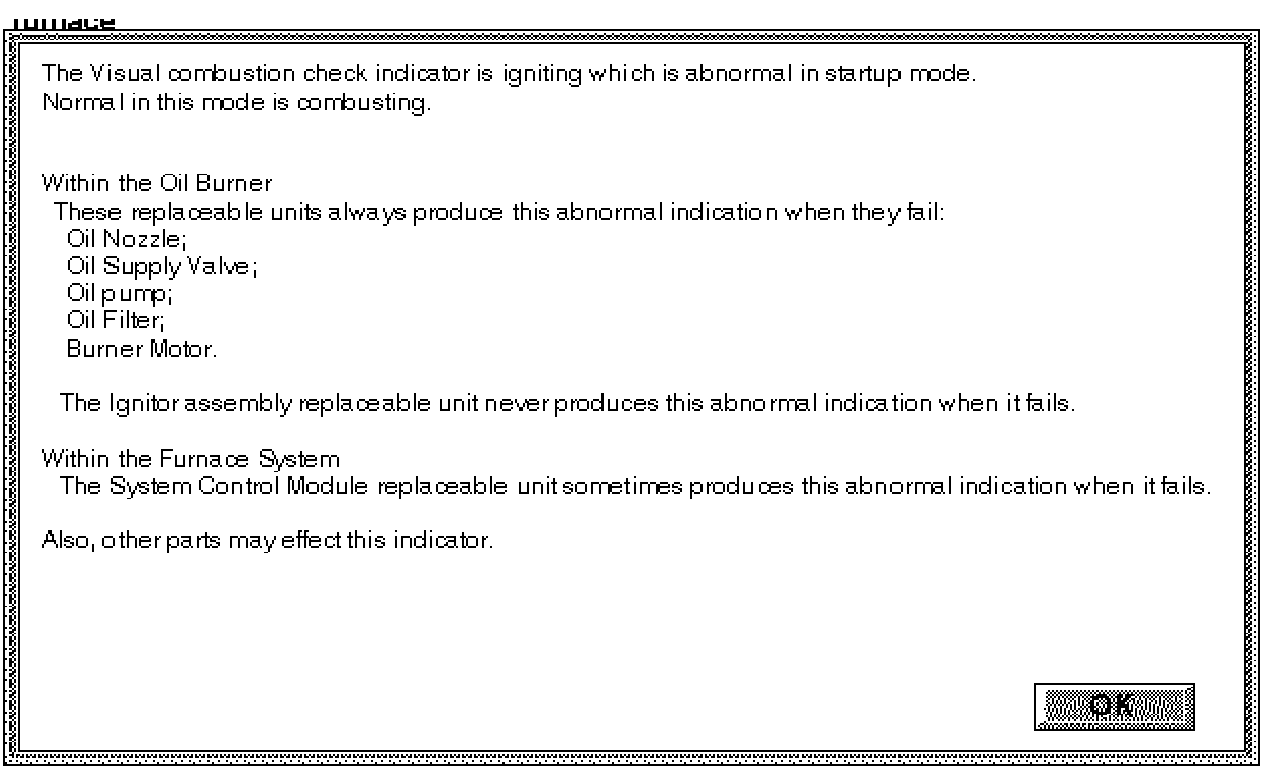


Figure 4: Original (top) and revised (bottom) answers provided by DIAG to the same *Consult* query

screen. When the student exercises the consult function the tutor sees the information that DIAG would use in generating its advice — exactly the same information that DIAG gives to EXEMPLARS in DIAG-NLP. The tutor then types a response that substitutes for DIAG's response. Although we cannot constrain the tutor to provide feedback that includes all and only the facts that DIAG would have communicated at that specific moment, we can still see the effects of how the tutor uses the information provided by DIAG.

Current work includes the mark-up of the collected language data. We will then analyze the coded corpus for the strategies the tutors use to verbalize facts, and we will implement a more sophisticated version of DIAG-NLP, that implements the strategies uncovered in the data.

The reader may have noticed that DIAG-NLP uses the *Mark-up, mine, implement, evaluate* methodology in a different way. Namely, we implemented a NL interface and evaluated it without first informing the interface with the results of corpus analysis. We followed this approach because we wanted to quickly evaluate whether simple changes to the language produced by the system affected its effectiveness. We established that this is the case. This result provides the grounds on the basis of which to collect and analyze data: if plausible but simplistic aggregation rules are effective, it is worthwhile to discover what aggregation patterns emerge from natural data, as the improvement to the interface afforded by using these patterns is bound to be greater.

COCONUT: Supporting collaboration via NL dialogue

(`http://www.isp.pitt.edu/~intgen`)

Human tutoring is a collaborative process, in which tutor and student work together to repair errors. It is a highly interactive process, with the tutor providing constant feedback to support students' problem solving. Because human tutoring is a collaboration, students are actively involved. In addition, because human tutors let their students do more of the process of recovering from impasses than ITSs, they allow students to feel more in control of the interaction than when they interact with ITSs..

A model of negotiation and collaboration in dialogue can therefore be used to model some aspects of tutoring dialogues as well. Building such a model is an area of research that I have also been pursuing in recent years [Di Eugenio et al.1998; Di Eugenio et al.2000].[5] Our model can provide the foundation to develop software tools that support people collaborating on solving a problem, including asymmetrical expertise situations such as advisor and advisee or tutor and student.

From the theoretical point of view, we propose a unified architecture for collaborative dialogue that integrates IRMA, a model of a resource-bounded rational agent [Bratman, Israel, and Pollack1988; Pollack1992] with a theory of language as collaboration [Clark1996]. IRMA (Intelligent, Resource-Bounded Machine Architecture) is especially appealing as a model of rational behavior because it brings to the fore the issue of *resource-boundedness*, i.e., the fact that agents are unable to perform arbitrarily large computations in constant time. IRMA accounts for both means-end reasoning and the need to weigh alternative options for action, and for the successful interaction of these two processes. What is missing in IRMA is an explicit link to collaboration, particularly in dialogue. Although perception is taken into account in IRMA, this architecture does not directly explain how negotiation unfolds in dialogue, how conversants come to agree on a solution, how they interpret and produce language, and the discourse strategies they use. Clark's work [Clark1992; Clark1996] provides a model of collaboration in dialogue that is an ideal candidate to bridge the gap, as it explains how the mutual belief needed for an agreement can be reached. We believe we should be able to model collaborative problem solving dialogues more effectively by integrating these two frameworks.

One issue of particular interest to us is to model collaboration under different distributions of knowledge: agents may have different types of knowledge, or rather, different instantiations of the same type of knowledge.

[5]This work is itself a collaboration with Pamela W. Jordan at the University of Pittsburgh. This work was initially supported by NSF grant IRI-9314961 to Richmond H. Thomason, Jerry Hobbs and Johanna D. Moore.

For example, a travel agent and a customer have different types of knowledge: the former of flights and hotels, the second of constraints and preferences for the trip. On the other hand, engineers who collaborate to build a circuit board may have different instantiations of the same type of knowledge (which integrated circuits are available, which constraints they impose on building the board, their cost etc). Tutors and their students have both different types of knowledge (the tutor knowledge about tutoring,[6] the student about his/her personal motivations to learn the material), and different instantiations of the domain knowledge (the student's version including wrong and/or missing items).

So far, we have used a simplified version of our model to account for the negotiation patterns we identified in computer-mediated conversations between two participants collaborating on a simple design problem, furnishing a two room apartment. We refer to these negotiation patterns as *the agreement process*. To gain insights into the agreement process, our empirical corpus study focused on how information is exchanged in order to arrive at a proposal, on what constitutes a proposal, and on its acceptance / rejection. We exploit dialogue history and the effect of the task, i.e., of the domain reasoning situation, on context, to reach the appropriate interpretation for each utterance. The results of the corpus study were used to inform a dialogue architecture based on abduction.

Conclusions

I have discussed some of the projects I have been involved in recent years on Natural Language interfaces for computer-supported instruction. I have also discussed work that more generally addresses modeling collaboration in dialogues. These projects all involve data collection and analysis, and implementation of computational models informed by the corpus analysis. Further details can be found at the URLs provided.

References

[Aleven2001] Aleven, Vincent, editor. 2001. *Workshop on Tutorial Dialogue Systems*, San Antonio, TX, May. The International Society of Artificial Intelligence in Education.

[Anderson et al.1995] Anderson, John R., Albert T. Corbett, Kenneth R. Koedinger, and R. Pelletier. 1995. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–207.

[Bratman, Israel, and Pollack1988] Bratman, Michael, David Israel, and Martha Pollack. 1988. Plan and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355.

[Chi et al.2001] Chi, Michelene T. H., Stephanie A. Siler, Takashi Yamauchi, and Robert G. Hausmann. 2001. Learning from human tutoring. *Cognitive Science*, 25:471–533.

[Chu-Carroll and Green1998] Chu-Carroll, Jennifer and Nancy Green. 1998. *Applying Machine Learning to Discourse Processing* (Papers from the 1998 AAAI Spring Symposium). Technical Report SS-98-01, American Association for Artificial Intelligence.

[Clark1992] Clark, Herbert H. 1992. *Arenas of Language Use.* The University of Chicago Press, Chicago, IL.

[Clark1996] Clark, Herbert H. 1996. *Using Language.* Cambridge University Press.

[Core, Zinn, and Moore2000] Core, Mark, Claus Zinn, and Johanna D. Moore. 2000. Supporting constructive learning with a feedback planner. In *Building Dialogue Systems for Tutorial Applications (AAAI Fall Symposium).* American Association for Artificial Intelligence, November.

[6]Even if not extensively [Graesser, Person, and Magliano1995].

[Delin et al.1994] Delin, Judy, Anthony Hartley, Cécile Paris, Donia Scott, and Keith Vander Linden. 1994. Expressing Procedural Relationships in Multilingual Instructions. In *Seventh International Workshop on Natural Language Generation*, pages 61–70, Kennebunkport, Maine.

[Di Eugenio1998] Di Eugenio, Barbara. 1998. An action representation formalism to interpret Natural Language instructions. *Computational Intelligence*, 14(1):89–133, February.

[Di Eugenio et al.2001] Di Eugenio, Barbara, Michael Glass, Michael J. Trolio, and Susan Haller. 2001. Simple natural language generation and intelligent tutoring systems. In *Workshop on Tutorial Dialogue Systems*, San Antonio, TX, May. 10th International Conference on Artificial Intelligence and Education.

[Di Eugenio et al.1998] Di Eugenio, Barbara, Pamela W. Jordan, Johanna D. Moore, and Richmond H. Thomason. 1998. An empirical investigation of collaborative dialogues. In *ACL-COLING98, Proceedings of the Thirty-sixth Conference of the Association for Computational Linguistics*, pages 325–329, Montreal, Canada, August.

[Di Eugenio et al.2000] Di Eugenio, Barbara, Pamela W. Jordan, Richmond H. Thomason, and Johanna D. Moore. 2000. The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogues. *International Journal of Human Computer Studies*, 53(6):1017–1076.

[Di Eugenio, Moore, and Paolucci1997] Di Eugenio, Barbara, Johanna D. Moore, and Massimo Paolucci. 1997. Learning features that predict cue usage. In *ACL-EACL97, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 80–87, Madrid, Spain.

[Di Eugenio, Rosè, and Moore1998] Di Eugenio, Barbara, Carolyn Penstein Rosè, and Johanna D. Moore. 1998. Towards a computational model of tutorial dialogue. Presentation at ONR Project Meeting, October. Chicago.

[Di Eugenio and Trolio2000] Di Eugenio, Barbara and Michael J. Trolio. 2000. Can simple sentence planning improve the interaction between learners and an intelligent tutoring system? In *Building Dialogue Systems for Tutorial Applications (AAAI Fall Symposium)*. American Association for Artificial Intelligence, November.

[Di Eugenio and Webber1996] Di Eugenio, Barbara and Bonnie Lynn Webber. 1996. Pragmatic Overloading in Natural Language Instructions. *International Journal of Expert Systems, Special Issue on Knowledge Representation and Reasoning for Natural Language Processing*, 9(1):53–84, March.

[Evens et al.1993] Evens, Martha W., John Spitkovsky, Patrick Boyle, Joel A. Michael, and Allen A. Rovick. 1993. Synthesizing tutorial dialogues. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 137–140, Hillsdale, New Jersey. Lawrence Erlbaum Associates.

[Fox1993] Fox, Barbara A. 1993. *The Human Tutorial Dialogue Project: Issues in the design of instructional systems.* Lawrence Erlbaum Associates, Hillsdale, NJ.

[Graesser et al.2001] Graesser, Arthur, Kurt VanLehn, Carolyn P. Rosé, Pamela W. Jordan, and Derek Harter. 2001. Intelligent tutoring systems with conversational dialogue. *AI Magazine.* To appear.

[Graesser, Person, and Magliano1995] Graesser, Arthur C., Natalie K. Person, and Joseph P. Magliano. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9:495–522.

[Grosz and Sidner1986] Grosz, Barbara and Candace Sidner. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12:175–204.

[Hume et al.1996] Hume, Gregory D., Joel A. Michael, Allen A. Rovick, and Martha W. Evens. 1996. Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences*, 5(1):23–47.

[Munro1994] Munro, Allen. 1994. Authoring interactive graphical models. In Ton de Jong, Douglas M. Towne, and Hans Spada, editors, *The Use of Computer Models for Explication, Analysis and Experiential Learning*. Springer Verlag.

[Paris et al.1995] Paris, Cécile, Keith Vander Linden, Markus Fischer, Anthony Hartley, Lyn Pemberton, Richard Power, and Donia Scott. 1995. A support tool for writing multilingual instructions. In *IJCAI95, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1398–1404. Also available as ITRI report ITRI-95-11.

[Pollack1992] Pollack, Martha E. 1992. The uses of plans. *Artificial Intelligence*, 57:43–68.

[Rosé and Freedman2000] Rosé, Carolyn P. and Reva Freedman, editors. 2000. *Building Dialogue Systems for Tutorial Applications (AAAI Fall Symposium)*. American Association for Artificial Intelligence.

[Rosé, Di Eugenio, and Moore1999] Rosé, Carolyn Penstein, Barbara Di Eugenio, and Johanna D. Moore. 1999. A dialogue based tutoring system for basic electricity and electronics. In *AI-ED 99, Proceedings of the 9th International Conference on Artificial Intelligence in Education*, Le Mans, France.

[Rosé et al.2001] Rosé, Carolyn Penstein, Johanna D. Moore, Kurt VanLehn, and David Albritton. 2001. A comparative evaluation of socratic versus didactic tutoring. In *23rd Annual Conference of the Cognitive Science Society*, Edinburgh, Scotland, August.

[Schulze et al.2000] Schulze, Kay G., Robert N. Shelby, Donald J. Treacy, and Mary C. Wintersgill. 2000. Andes: A Coached Learning Environment for Classical Newtonian Physics. In *Proceedings of the 11th International Conference on College Teaching and Learning*, Jacksonville, FL, April.

[Soller2001] Soller, Amy. 2001. Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education (in press)*, 12.

[Towne1997] Towne, Douglas M. 1997. Approximate reasoning techniques for intelligent diagnostic instruction. *International Journal of Artificial Intelligence in Education*.

[Vander Linden and Martin1995] Vander Linden, Keith and James Martin. 1995. Expressing Local Rhetorical Relations in Instructional Text. *Computational Linguistics*, 21(1):29–57.

[Walker1999] Walker, Marilyn A. 1999. Towards standards and tools for discourse tagging. ACL99 Workshop Proceedings.

[Walker and Moore1997] Walker, Marilyn A. and Johanna D. Moore. 1997. Empirical studies in discourse. *Computational Linguistics*, 23(1):1–12. Special Issue on Empirical Studies in Discourse.

[Webber et al.1995] Webber, Bonnie Lynn, Norman Badler, Barbara Di Eugenio, Christopher Geib, Libby Levison, and Michael Moore. 1995. Instructions, Intentions and Expectations. *Artificial Intelligence, Special Issue on Computational Theories of Interaction and Agency*, 73:253–269, February.

[White and Caldwell1998] White, Michael and Ted Caldwell. 1998. Exemplars: A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 266–275, Niagara-on-the-Lake, Canada.