Toward a computational model of expert tutoring: a first report

Barbara Di Eugenio, Trina C. Kershaw, Xin Lu, Andrew Corrigan-Halpern, Stellan Ohlsson
University of Illinois, Chicago, USA
{bdieugen,tkersh1,xlu4,ahalpe1,stellan}uic.edu

Abstract

We are exploring the differences between expert and less expert tutors with two goals: cognitive (what does tutoring tell us about learning) and applied (which features of tutoring dialogues should be included in interfaces to ITSs). We report results from human tutoring dialogues where an expert tutor was compared to less expert tutors. We also report results from a comparison among four versions of an ITS, that vary in the degree and kind of feedback they provide. Our results establish upper and lower bounds for the effectiveness of tutoring interactions in our domain.

Introduction

Intelligent Tutoring Systems (ITSs) help students master a certain topic. Research on the next generation of ITSs explores Natural Language (NL) as one of the keys to bridge the gap between current ITSs and human tutors. Part of this inquiry concerns uncovering whether the NL interaction between students and an ITS does in fact improve learning, and if yes, which features of the interaction are responsible for the improvement. Whereas very recently the first results have appeared, that show that students learn more when interacting in NL with an ITS (Di Eugenio *et al.* 2005a; Evens & Michael 2005; Graesser *et al.* 2005; Litman *et al.* 2004; Peters *et al.* 2004; Rosé *et al.* 2003), we still don't know what exactly is responsible for these results. For example, in (Di Eugenio *et al.* 2005a) we found that students learned more when given more abstract but also more directive feedback in an ITS that teaches troubleshooting; Litman et al. (2004) found that there was no difference in the learning gains of students who interacted with a mechanics ITS using typed text or speech.

It is not surprising that the community is still investigating which features of an NLP interface to an ITS are more effective for learning, since it is not yet well understood what makes human tutoring effective, notwithstanding the many findings from the literature, e.g. (Fox 1993; Graesser, Person, & Magliano 1995; Chi *et al.* 2001).

In this paper, we contribute another piece of the puzzle. Among the many issues worth of study in this area, we focus on what distinguishes expert from novice tutors. In the literature there are hardly any comparisons between expert and novice tutors, although there are observations on one or the other. In particular, there is scant if any evidence that expert tutors are actually more effective. From the point of view of computationally modeling a dialogue, the simpler the language phenomena, the simpler the computational task. Expert tutors are likely to use more complex dialogue strategies than novices (Putnam 1987; Graesser, Person, & Magliano 1995; Lepper, Drake, & O'Donnell-Johnson 1997; Glass *et al.* 1999). If expert tutors and novice tutors were equally effective, modeling novice tutors would make the computational task easier.

Our domain concerns extrapolating complex letter patterns (Kotovsky & Simon 1973). The student is given a patterned sequence of letters (e.g., MABMCDM) and is asked to extrapolate the sequence while maintaining the pattern (i.e., MEFMGHM) – here M works as a *marker*, and chunks of two letters form a progression according to the alphabet.

We collected tutoring dialogues with three tutors, one expert, one novice, and one experienced in teaching, but not in one-on-one tutoring. The expert tutor was significantly more effective than the other two tutors. We discuss an initial analysis of the differences in the dialogues between the expert tutor and the other tutors.

In the same domain, we also implemented four different versions of an ITS. In the *no feedback* version of the ITS, each letter the subject inputs turns blue, with no indication and no message regarding whether it is correct or incorrect; in the *neutral* version, the only feedback subjects receive is via color coding, green for correct, red for incorrect; in the *positive* version, they receive feedback via the same color coding, and in addition, verbal feedback on correct responses only; in the *negative* version, they receive feedback via the same color coding, and in addition, verbal feedback on incorrect responses only. The language in the *positive* and *negative* conditions was inspired by (but not closely modelled on) the expert tutor's language. We ran a between-subject experiment, in which each group of subjects interacted with one of the systems. We found that, even if subjects in the verbal conditions do perform slightly better and make fewer mistakes, these differences are not significant; in particular, there are no significant differences with the *no feedback* condition.

Our results thus establish an upper / lower bound on what type of feedback an ITS needs to provide to engender significantly more learning than simple practice. The lower bound is established by the ITSs that provide some language feedback but are not different from the *no feedback* condition, that simply draws attention to the task. The upper bound is established by the expert tutor's language. Neither the lower nor the upper bound are as tight as we would like them to be. In particular, we would like to find a much tighter upper bound than *full dialogue with an expert tutor*, since this would require solving the whole NLP problem. Our corpus analysis of the expert vs other tutors' data provides some initial answers to this question.

The paper is organized as follows. We first discuss the human tutoring data we collected and analyzed, and the comparison between expert and non expert tutors. We then present our ITS with its 4 conditions, and the results we obtained. Finally, we conclude and discuss future work.

The human tutoring data

To investigate whether expert tutors are more effective than less experienced tutors, we ran an experiment in the letter pattern domain. Subjects were individually tutored by three different tutors: the *expert*, who had years of experience as a professional tutor; the *lecturer*, who had years of experience as a lecturer but scarce experience in one-on-one tutoring; the *novice*, an undergraduate with no teaching or tutoring experience.[1] A control group was not tutored at all.

There were 11 subjects in each condition, including in the control group. In the three tutoring conditions, the subjects went through a curriculum of 13 problems of increasing difficulty, using paper and pencil, and then solved two post-test problems via a computer interface. The post-test presented subjects with 2 patterns, each 15 letters long. Subjects had to reproduce the same pattern 6 times, but each time, starting with a different letter. For example, if one post-test problem were TRPNL, they would be asked to reproduce the pattern once starting from Q (correct answer is QOMKI), once starting from J (IGECA), etc. Subjects had 1 minute for each trial and did not receive any feedback of any kind. The subjects in the control condition received no instruction, and solved the same post-test problems with the same interface.

Figure 1 illustrates performance for all conditions on Problem 2. Performance is measured, per trial, in number of letters correct out of the 15 letters each pattern is composed of.

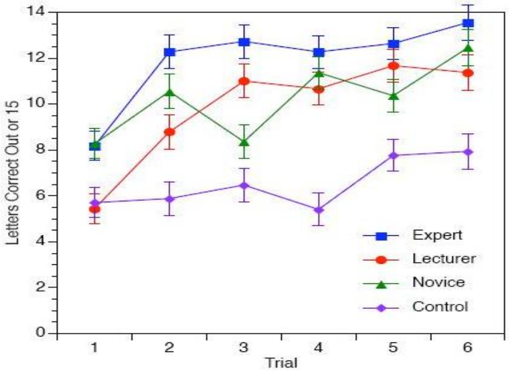

Figure 1: Performance on post-test problem 2

	Novice	Lecturer	Expert
Problem 2	**10.33**	17.00	34.83
Problem 9	**16.17**	69.50	69.83

Table 1: Tutor utterances per problem

On the whole, we found that the expert tutor is indeed more effective. Specifically (all our statistical results are based on ANOVAs; when significant, ANOVAs are followed by Tukey's tests to determine which condition is significantly different from the others):

- The expert tutor is significantly more effective than the other two tutors on both post-test problems ($p < 0.05$ in both cases)
- Collectively, the tutors are significantly better than control (no tutoring) on post-test problem 2 ($p < 0.001$)
- The expert tutor is significantly more effective than control on post-test problem 2 ($p < 0.005$)

The next question is, what does the expert tutor do that is more effective?. The tutoring dialogues were videotaped, and a selected subset has been transcribed. We choose to transcribe two specific problems, #2 and #9, to have a sample of what tutors do at the beginning of the curriculum, and what they do later on a much more complex problem. The dialogue excerpts for six subjects per tutor were transcribed and annotated, where the same subject solved problems 2 and 9, for a total of 36 dialogue excerpts. Our transcription guidelines are a small subset of the CHILDES transcription manual (MacWhinney 2000).

Table 1 illustrates the average number of tutor utterances per problem. Table 2 illustrates the average number of tutor and student words, and of tutor and student utterances, per tutor. Numbers in boldface refer to significant differences, that we will discuss below.

	Novice	Lecturer	Expert
Tu. words	107.33	369.17	419.17
St. words	55.00	209.00	83.00
St. words /Tu. words	.51	.57	**.20**
Tu. Utts.	13.25	43.25	52.33
St. Utts.	7.74	29.50	17.67
St. Utts. / Tu. Utts.	.58	.68	**.32**

Table 2: Average numbers of words and utterances, per tutor

[1] From the literature, it is unclear who exactly should qualify as an expert tutor. Here we equate tutor expertise with general tutoring experience, not with experience in tutoring in the specific domain. While the latter is certainly important (Glass *et al.* 1999), it seems less relevant here, since the letter sequence pattern problems don't require anything beyond knowledge of the alphabet.

Annotation for tutor moves

The transcribed excerpts have been annotated for tutor moves, and are being annotated for student moves. We developed our annotation scheme based on the literature, e.g. (Chi *et al.* 2001; Litman *et al.* 2004), and with simplicity in mind. Our tutor moves include four high level categories, *reaction*, *initiative* (further categorized, see below), *support*, *conversation*. *Support* is used when the tutor encourages the student in his/her work without referring to particular elements of the problem; *conversation* is used for acknowledgments, continuers, and small talk.

Tutor *reaction* – the tutor reacts to something the student says or does – is subcategorized as follows:

- Answering: answering a direct question from the student
- Evaluating: giving feedback about what the student is doing
- Summarizing: summarizing what has been done so far

Tutor *initiative* is subcategorized as follows:

- Prompting: prompting the student into some kind of activity, further subcategorized as:
  - General: laying out what to do next *Why don't you try this problem*
  - Specific: trying to get a specific response from the student *What would the next letter be?*
- Diagnosing: trying to determine what the student is doing *Why did you put a D there?*
- Instructing: providing the student with information about the problem. Further subcategorized as:
  - Declarative: providing facts about the problem *Notice the two Cs here? They are separating different parts of the problem*
  - Procedural: giving hints or tricks about how to solve problem *Start by counting the number of letters in each period*
- Demonstrating: showing the student how to solve the problem. *Watch this. First I count the number of letters between the G and J here.*

Two annotators (undergraduate psychology students) coded all the dialogues. After a first round of coding, the annotators met with a graduate student overseer and discussed their disagreements. They recoded the most problematic dialogues, and the intercoder reliability statistics were computed. In further discussions they came to an agreed upon coding for all the dialogues.

Tables 3 and 4 report the rates of intercoder agreement respectively across all categories per tutor, and for the individual categories and subcategories. We use the Kappa coefficient of agreement, as has become standard in NLP (Krippendorff 1980; Carletta 1996; Di Eugenio & Glass 2004). In both Tables 3 and 4, boldface highlights acceptable Kappa values. There is debate in the literature on what exactly Kappa values mean. Here we follow (Rietveld & van Hout 1993) in assessing that $0.60 < Kappa \leq 0.80$ denotes substantial agreement, and $0.80 < Kappa \leq 1$ almost perfect agreement.

Level	Novice	Lecturer	Expert	Overall
Full	**.688**	.553	.452	.528
High Level	**.750**	**.655**	.597	**.644**

Table 3: Kappa values by tutor

Category	Subcategory	Kappa
Answering		**0.75**
Evaluating		0.56
Summarizing		**0.60**
Prompting		**0.82**
	General	0.34
	Specific	**0.73**
Diagnosing		**0.63**
Instructing		0.55
	Declarative	0.33
	Procedural	0.37
Demonstrating		0.39
Instr-Demon		**0.63**
Support		0.39
Conversation		0.55

Table 4: Tutor moves: Kappa values

Table 3 reports two results: for the full scheme (13 categories), and with no subcategorization for *instructing* and *prompting* (high level, 9 categories). In both cases, the dialogues with the novice are the easiest to annotate, followed by those with the lecturer and then those with the expert.

In Table 4, we report the Kappa values for different categories and subcategories. Some categories are very reliable, such as *prompting*, and its subcategory *specific prompting*; some categories are acceptable, such as *diagnosing*; some categories are not, such as *support* and *instructing*. The former is not problematic for our analysis, since there are very few instances of *support* in our coded data. The latter instead is, since *instructing* is one of the categories where tutors differ. Only when we collapse *instructing* and *demonstrating* (see *Instr-Demon*), which in fact the coders reported as hard to distinguish, we obtain an acceptable Kappa value.

Table 5 reports the percentages of moves by tutor. Note that the columns don't add up to exactly 100%, because of few utterances left without any tag, and vice-versa, few utterances with more than one tag – coders were allowed to use more than one code, although they were not encouraged to do so. We'll discuss some differences between tutors below.

Annotation for student moves

The dialogues have been annotated for five student moves:

- Answering: directly answering a tutor's question
- Explaining: explaining what the student said or did, reasoning, or thinking aloud
- Reflecting: evaluating own's understanding
- Questioning: asking the tutor a question

Category	Novice	Lecturer	Expert
Answering	**10.1%**	5.4%	1.4%
Evaluating	16.4%	13.0%	8%
Summarizing	**6.9%**	16.9%	16.7%
Gen. Prompting	5.0%	3.7%	4.1%
Spec. Prompting	17.6%	**27.6%**	13.9%
Diagnosing	2.5%	3.3%	3.3%
Decl. Instructing	**22.6%**	6.2%	4.0%
Proc. Instructing	0.6%	4.3%	**17.0%**
Demonstrating	6.3%	0.0%	11.1%
Support	0.6%	0.6%	5.4%
Conversation	9.4%	17.1%	10.5%

Table 5: Percentages of tutor moves, by tutor

- Action response: performing some action (e.g., writing down a letter) in response to the tutor's question or prompt

We don't report distributional data since it is not final yet: the coding is undergoing revisions in response to the first round of intercoder reliability results. Below, we comment on some preliminary trends.

Discussion

All the tables shown so far suggest that there are some substantial differences between tutors. In particular, there is evidence that the expert tutor behaves differently from the predictions on effective tutoring from the literature. As regards Table 1, we ran ANOVAs, where the tutor and the problem are independent variables, and number of tutor utterances is the dependent variable. We found:

- a main effect of problem ($p < 0.05$): there are more utterances for problem 9 than problem 2
- a main effect of tutor ($p < 0.05$): the novice has significantly fewer utterances than the other two, i.e., both expert and lecturer have longer dialogues with subjects
- an interaction between problem and tutor ($p < 0.05$): the novice's utterances don't significantly increase, the other two tutors' do.

As regards the expert tutor's behavior, (Chi *et al.* 2001) shows that subjects learn best when they construct knowledge by themselves, and that as a consequence, the tutor should prompt and scaffold subjects, and leave most of the talking to them. In contrast, Table 2 shows that our expert tutor's subjects do not talk more: the ratio of student utterances to tutor utterances is significantly lower for the expert tutor ($p < 0.05$), and so is the ratio of student words to tutor words ($p < 0.001$).

Further looking at tutor moves, i.e., at Tables 3 and 5, we see that, first, the expert dialogues are the hardest to code. This supports the intuition that expert tutors use more sophisticated strategies, but does not bode well for computational modelling of expert tutors: if it is harder to code expert dialogues, the data on which to train the NL interface will be less reliable than for other types of tutors. As far as individual moves are concerned, we found that again the expert tutor does not behave as one would expect him to:

- the expert does not prompt his subjects more (the lecturer does, $p < 0.05$; and consistently, the student move annotation shows that his students explain more)
- the expert does not answer more questions from subjects (the novice does, $p < 0.05$; consistently, subjects ask more questions of the novice, perhaps because in fact they are more confused when interacting with her)
- the expert uses more procedural instructing ($p < 0.05$)[2]

Other findings are that the novice summarizes less than the other two ($p < 0.05$), and uses more declarative instructing ($p < 0.05$). From the student move annotation, we see that the subjects interacting with the expert reflect more, i.e., assess their understanding of the problem more often. Some of these findings agree with our informal impressions that the expert talks more than the others; seems to spend more time on problem 2 than the other two tutors, as if to lay the foundation for what will come next (partially supported: his dialogues for problem 2 are marginally significantly longer than the novice's ($p = 0.06$), but not than the lecturer's); gives subjects "tricks" on how to easily go forward and backward in the alphabet, and how to detect patterns (his usage of procedural instructing).

Four ITSs for the letter pattern problem

In parallel with the data collection and analysis, we started developing an ITS to solve the letter pattern problems. We developed four versions of the ITS, which differ in the kind of feedback they provide the student. We built our four ITSs by means of the Tutoring Development Kit (TDK) (Koedinger, Aleven, & Heffernan 2003), based on the ACT-R theory (Anderson *et al.* 1990). This theory postulates that skills involved in a complex task can be modeled as production rules: *correct* rules model the solution(s) for each problem and *buggy* rules capture possible errors. The same set of rules (correct and buggy) provide the backbone for the corresponding model-tracing tutor.

As we will see, none of the four ITSs models the expert tutor in any deep sense, although that is our ultimate goal. This development strategy is due to a series of general questions on the role of feedback in learning we are interested in, and to the need to provide one or more baselines to which to compare the final ITS that will model the expert tutor.

In previous studies of ours, subjects learned to extrapolate complex letter patterns of the kind discussed above and were provided positive and/or negative graphical feedback (Corrigan-Halpern & Ohlsson 2002). The pattern contained a hierarchical structure and could be decomposed into chunks. Feedback could be given for each letter in a chunk (local feedback), or for an entire chunk (global feedback). Feedback consisted of the color coded words *Correct* in green and *Wrong* in red – these words appeared below the corresponding letter or chunk. In one study, performance was compared as a function of feedback scope and type (positive or negative). There was an interaction between feedback type and scope. Subjects given negative feedback per-

[2] This finding should be taken with a grain of salt because of the low Kappa value for this category.

formed best when it was given locally. Subjects given positive feedback performed best when it was given globally.

These previous studies suggested a design in which subjects in one condition (*neutral*) were given graphical feedback (green or red); further, to tease apart the functions of positive and negative feedback, the graphical feedback was augmented with language feedback only for positive feedback (*positive*), or for negative feedback (*negative*), but the two kinds of messages were not mixed together. Note that the feedback is always local. Finally, a fourth condition (*no feedback*) in which each letter turns blue (i.e., no indication is given as to whether it is correct or incorrect) was added to check whether practice and drawing attention to the task are predictive of performance.

The ITSs presented the subjects with the same curriculum that had been used in the human data collection. Subjects also solved the same post-test problems as in the human data collection, via the same computer interface (separate from the ITSs).[3] In a fifth control condition, subjects solved the post-test problems without any training. Further details on the ITSs can be found in (Di Eugenio *et al.* 2005b).

Method and Results. We ran a between-subjects study in which each group of subjects (positive [N = 33], negative [N = 36], neutral [N = 37], no feedback [N=31]) interacts with one version of the system.

The ITSs collect detailed trace information for each subject as s/he works through the training: time stamps on and location of every response the subject provides, which production rule (correct or buggy) fires and when, and when the subject closes the feedback message window (a new window pops up with every new feedback message).

We first discuss how the four ITS versions fared against each other. The main result is that, surprisingly, there was no difference between conditions other than with respect to control (see Table 6). There are no differences between the *no feedback* condition and the three feedback conditions, or among the three feedback conditions. Subjects in the positive condition did slightly better than subjects in the other conditions on each post-test problem, and overall. However, none of these differences is significant. We also performed a linear regression analysis with post-test scores as the dependent variable and condition, time spent on training, and number of bug messages as the predictors. We found that the more time spent on training and the higher number of errors (bugs) made during training, the worse the performance. More details on the regression analysis can be found in (Di Eugenio *et al.* 2005b).

Finally, we compared the performance of the subjects in the ITS conditions with the subjects in the human tutoring condition – as control, we used the control group from the ITS experiments, because it is larger.[4] Table 6 reports the performance on the two problems for every condition.

No differences were found in the correct answers of the human tutors and the ITSs in problem 1. However, the expert tutor and all the verbal ITS conditions scored better than control for problem 1. For problem 2, subjects in the expert tutor condition answered correctly more often than subjects in any other condition. All subjects in the human tutor and ITS conditions scored higher than control subjects on problem 2. Overall, all subjects in all human tutor conditions answered more questions correctly than did subjects in the control condition. Additionally, subjects in the expert tutor condition had more correct answers than subjects in the negative, neutral and no feedback ITS conditions.

	Post-test problem 1	Post-test problem 2	Total
Expert	50.45	71.64	122.09
Lecturer	33.45	58.00	91.45
Novice	30.27	54.82	85.09
Positive	42.21	45.30	87.52
Negative	40.06	37.58	77.64
Neutral	39.11	37.51	76.62
No Feedback	33.58	42.19	75.77
Control	18.16	18.78	36.94

Table 6: Performance for all conditions

Discussion and future work

Through our collection of tutoring dialogues with human tutors of different expertise, and our experimentation with using simple feedback in the different versions of the ITS we built, we have established lower and upper bounds for the effectiveness of verbal feedback, in our domain. Clearly, there is a vast space between the two bounds, and our current and future work is trying to bring the bounds closer.

Although verbal feedback did not make a difference in our ITS experiment, it would obviously be premature to conclude that it does not help in general. First, the nature of the task may be such that feedback does not make too much of a difference with respect to simple practice, since the *no feedback* condition did not perform differently from the feedback conditions. However, the fact that the expert tutor is more effective than any other condition, at least on post-test problem 2, seems to rule this out. Second, subjects may have not really read the feedback, especially since it may sound too repetitive after a while – indeed students don't read long or repetitive feedback (Heift 2001).

The real reason why verbal feedback may have not been effective is that it is not sophisticated enough. Again, the effectiveness of the expert tutor seems to suggest this. Moreover, in (Di Eugenio *et al.* 2005a) we compared three different versions of an ITS that teaches troubleshooting. We found a significant difference in learning between two versions where the feedback is very detailed and a third version that highlights the functions played by the subparts of the systems by using language that abstracts away from individual parts.

The next step in the letter pattern project is indeed to build a more sophisticated version of the ITS that provides feedback based on expert tutor dialogue patterns. This requires

[3] The only difference is that the subjects in the human data collection had 6 trials per problem, the subjects who used the ITSs 10 trials. In the comparison below, we use only the first 6 trials per problem from the ITSs and control conditions.

[4] Note that the cardinality of the conditions are different, being 11 each for the human tutors, and above 30 for the ITS conditions.

the annotation of student moves which is under way, and the transcription and annotation of more dialogues, which we have recently started. We will then use machine learning techniques to extract dialogue patterns from the annotated data, and embody the patterns we uncover in the sixth and final version of the letter pattern ITS.

Finally, our findings on the effectiveness of the expert tutor, and on the somewhat unusual features of his tutoring, are based on a small dataset, and on one single tutor. They clearly need to be repeated with different tutors and / or in different domains. We are halfway through a different tutoring dialogue collection that again compares expert and non-expert tutors. The domain is introductory Computer Science, i.e., basic data structures and algorithms. In this setting, subjects take a pretest, then interact with one of two tutors, then take the post-test. One tutor, the expert, is a retired Math and Computer Science college professor with many years of experience in one-on-one tutoring; the other, the novice, is a senior in Computer Science, with just a few hours under his belt as a volunteer tutor for some introductory classes. As for the letter pattern domain, we intend to extract successful dialogue patterns, to be used in a NL interface to an ITS that tutors basic Computer Science.

Acknowledgments. This work is supported by awards CRB S02 and CRB S03 from the University of Illinois at Chicago, and by grant N00014-00-1-0640 from the Office of Naval Research.

References

Anderson, J. R.; Boyle, C. F.; Corbett, A. T.; and Lewis, M. W. 1990. Cognitive modeling and intelligent tutoring. *Artificial Intelligence* 42:7–49.

Carletta, J. 1996. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics* 22(2):249–254.

Chi, M. T. H.; Siler, S. A.; Yamauchi, T.; and Hausmann, R. G. 2001. Learning from human tutoring. *Cognitive Science* 25:471–533.

Corrigan-Halpern, A., and Ohlsson, S. 2002. Feedback effects in the acquisition of a hierarchical skill. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*.

Di Eugenio, B., and Glass, M. 2004. The Kappa statistic: a second look. *Computational Linguistics* 30(1):95–101.

Di Eugenio, B.; Fossati, D.; Yu, D.; Haller, S.; and Glass, M. 2005a. Aggregation improves learning: experiments in natural language generation for intelligent tutoring systems. In *ACL05, Proceedings of the 42nd Meeting of the Association for Computational Linguistics*.

Di Eugenio, B.; Lu, X.; Kershaw, T. C.; Corrigan-Halpern, A.; and Ohlsson, S. 2005b. Positive and negative verbal feedback for intelligent tutoring systems. In *AIED 2005, the 12th International Conference on Artificial Intelligence in Education*. (Poster).

Evens, M., and Michael, J. 2005. *One-on-one Tutoring by Humans and Machines*. Mahwah, NJ: Lawrence Erlbaum Associates.

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

Glass, M.; Kim, J. H.; Evens, M. W.; Michael, J. A.; and Rovick, A. A. 1999. Novice vs. expert tutors: A comparison of style. In *MAICS-99, Proceedings of the Tenth Midwest AI and Cognitive Science Conference*, 43–49.

Graesser, A.; Person, N.; Lu, Z.; Jeon, M.; and McDaniel, B. 2005. Learning while holding a conversation with a computer. In PytlikZillig, L.; Bodvarsson, M.; and Brunin, R., eds., *Technology-based education: Bringing researchers and practitioners together*. Information Age Publishing.

Graesser, A. C.; Person, N. K.; and Magliano, J. P. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology* 9:495–522.

Heift, T. 2001. Error-specific and individualized feedback in a web-based language tutoring system: Do they read it? *ReCALL Journal* 13(2):129–142.

Koedinger, K. R.; Aleven, V.; and Heffernan, N. T. 2003. Toward a rapid development environment for cognitive tutors. In *12th Annual Conference on Behavior Representation in Modeling and Simulation*.

Kotovsky, K., and Simon, H. 1973. Empirical tests of a theory of human acquisition of information-processing analysis. *British Journal of Psychology* 61:243–257.

Krippendorff, K. 1980. *Content Analysis: an Introduction to its Methodology*. Beverly Hills, CA: Sage Publications.

Lepper, M. R.; Drake, M. F.; and O'Donnell-Johnson, T. 1997. Scaffolding techniques of expert human tutors. In Hogan, K., and Pressley, M., eds., *Scaffolding student learning: Instructional approaches and issues*. Cambridge, MA: Brookline.

Litman, D. J.; Rosé, C. P.; Forbes-Riley, K.; VanLehn, K.; Bhembe, D.; and Silliman, S. 2004. Spoken versus typed human and computer dialogue tutoring. In *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems*.

MacWhinney, B. 2000. *The CHILDES project. Tools for analyzing talk: Transcription Format and Programs*, volume 1. Lawrence Erlbaum, Mahwah, NJ, third edition.

Peters, S.; Bratt, E. O.; Clark, B.; Pon-Barry, H.; and Schultz, K. 2004. Intelligent systems for training damage control assistants. In *Proceedings of I/ITSEC 2004, Interservice/Industry Training, Simulation, and Education Conference*.

Putnam, R. 1987. Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal* 24:13–48.

Rietveld, T., and van Hout, R. 1993. *Statistical Techniques for the Study of Language and Language Behaviour*. Berlin - New York: Mouton de Gruyter.

Rosé, C. P.; Bhembe, D.; Siler, S.; Srivastava, R.; and VanLehn, K. 2003. Exploring the effectiveness of knowledge construction dialogues. In *AIED03, Proceedings of AI in Education*.