Towards explaining effective tutorial dialogues

Barbara Di Eugenio and **Davide Fossati**
{bdieugen, dfossa1}@uic.edu
Department of Computer Science, University of Illinois at Chicago, USA

Stellan Ohlsson and **David Cosejo**
{stellan, dcosej1}@uic.edu
Department of Psychology, University of Illinois at Chicago, USA

Abstract

We present a study of human tutorial dialogues in a core Computer Science domain that: focuses on individual tutoring sessions, rather than on contrasting different types of tutors; uses multiple regression analysis to correlate features of those sessions with learning outcomes; and highlights the effects of two types of tutor moves that have not been studied in depth so far, direct instruction and positive feedback.

Keywords: Artificial Intelligence; Education; Discourse; Interactive Behavior; Human Experimentation; Intelligent Tutoring Systems.

Introduction

One-on-one tutoring has been shown to be a very effective form of instruction compared to other educational settings. For more than twenty years, researchers have worked on uncovering features of tutorial interaction that engender learning in students (Fox, 1993; Graesser, Person, & Magliano, 1995; Lepper, Drake, & O'Donnell-Johnson, 1997; Chi, 2001; Moore, Porayska-Pomsta, Varges, & Zinn, 2004; Evens & Michael, 2006; Litman et al., 2006; Di Eugenio, Fossati, Haller, Yu, & Glass, 2008). This research is partly motivated by the search for computational models[1] of effective tutoring strategies, which are an essential component of dialogue interfaces to Intelligent Tutoring Systems, or ITSs.

From such an extensive body of work, many findings have arisen, but a unifying explanation has yet to emerge, partly because tutorial interactions are so rich that it is hard to characterize them fully. One line of attack, pursued by us as well as by others in the past, has tried to understand what *expert* tutors do (Lepper et al., 1997; Evens & Michael, 2006; Di Eugenio, Kershaw, Lu, Corrigan-Halpern, & Ohlsson, 2006; Cade, Copeland, Person, & D'Mello, 2008). However, it is not clear what *expert* tutors are to start with; *expertise* does not necessarily mean, 'with extensive *experience*'. Another approach is to imbue the notion of *expertise* with what we know from pedagogy, and equate *expert* tutors with idealized, Socratic human tutors. However, as Graesser and collaborators pointed out in a series of papers (Person, Graesser, Magliano, & Kreuz, 1994; Graesser et al., 1995), the idealized Socratic human tutor is difficult to come by. In the course of our own tutorial dialogues collection and analysis (Di Eugenio et al., 2006, 2008), we have come to believe that even many experienced tutors do not behave socratically most of the times. Still, students learn with them, sometimes but not necessarily more than what they learn with inexperienced tutors. As we discuss in Ohlsson et al. (2007), this brought us to advocate a different approach to tutorial dialogue analysis. We eschew defining who an expert tutor is a priori, or casting one type of tutor against another, and analyzing differences between them. Rather, we pool the data from all the tutors together, and we focus on the *effectiveness* of the *individual* tutoring session, by running regression analyses that correlate features of those sessions with learning outcomes.

The next point concerns which *features* of the tutoring session should be entered in the regression analysis. The body of work we described above has proposed a variety of coding schemes to annotate tutorial dialogues. In Ohlsson et al. (2007), we noted that most often such coding schemes are motivated by pedagogical tenets, by linguistic theories of speech acts, and often by what happens in the tutoring dialogues themselves. What many schemes are missing is being informed by cognitive insights into how people learn.

In this paper, we present a corpus analysis which embodies those earlier proposals of ours: it pools together tutoring sessions collected in a study that had originally started as a contrast between a more experienced and a less experienced tutor; because it is motivated by what is cognitively plausible for learning, it focuses on two features that have not been studied in depth so far, direct instruction and positive feedback; and uses regression analysis. In particular, we show that positive feedback correlates with learning, and that our methodology allows us to further annotate the data in a controlled, principled way. We also very briefly discuss the ITS we have built, partly based on these results.

Human tutoring in Computer Science

Our domain of interest is Computer Science (CS), specifically, introductory data structures such as *linked lists, stacks* and *binary search trees*, and the algorithms that manipulate them. This domain is partly motivated by the interests of some of us, as educators in CS. Basic

[1] We mean *computational* in the *algorithmic* sense coming from computer science, rather than in the sense of *mental computation* coming from cognitive psychology.

data structures and algorithms are in the core of the CS undergraduate curricula, and have been identified as difficult concepts for students to master (Katz, Allbritton, Aronis, Wilson, & Soffa, 2003). At the same time, basic CS has received little attention from the educational and ITS research communities. The few ITSs that concern CS cover a smattering of topics, including computer literacy (Graesser, Person, Lu, Jeon, & McDaniel, 2005), programming languages such as Lisp (Corbett & Anderson, 1990), general programming and design skills (Lane & VanLehn, 2003), and databases (Mitrović, Suraweera, Martin, & Weerasinghe, 2004). We believe iList, the system we have developed partly on the basis of our corpus study, and which we will briefly discuss at the end of the paper, fulfills an important need.

Methods

We collected a corpus of 54 one-on-one tutoring sessions on *linked lists*, *stacks*, and *binary search trees* (for brevity, we will refer to the first topic as *lists*, and to the third, as *trees*). Each individual student participated in only one tutoring session, with a tutor randomly assigned from a pool of two tutors. One of the tutors is an experienced CS professor, with more than 30 years of teaching experience, including one-on-one tutoring. The other tutor is a senior undergraduate student in CS, with only one semester of previous tutoring experience.

Students took a pre-test right before the tutoring session, and an identical post-test immediately after. The test had two problems on lists, two problems on stacks, and four problems on trees. Each problem was graded out of 5 points, for a possible maximum score of 10 points each for lists and stacks, and 20 points for trees. Pre- and post-test scores for each topic were later normalized to the [0..1] interval.

The tutors did not know the problems on the pre-test, since we did not want them to tutor to the test, but they were presented with a high level summary of the pre-test results. Additionally, they had a predefined set of problems they could use during tutoring, and were alerted when the tutoring session had lasted 45 minutes. Even so, tutors had considerable latitude in how long they tutored students: in some cases they decided to continue beyond 45 minutes, while in other cases they stopped tutoring around 35 minutes, hence the variability in the length of the tutoring sessions (see below).

An additional group of 53 students (control group) took the pre- and post-tests, but instead of participating in a tutoring session they attended a 40 minute lecture about an unrelated CS topic. The rationale for such a control condition was to assess whether by simply taking the pre-test students would learn about data-structures, and hence, to tease out whether any learning we would see in the tutored conditions would be indeed due to tutoring. We did not run more usual control conditions, such as reading relevant material in a textbook, since it is well established that human tutors are more effective than those other conditions (Evens & Michael, 2006; VanLehn et al., 2007).

Table 1: Learning gains and t-test statistics

Topic	Tutor	N	Mean	Std	t	p
List	Novice	24	.09	.22	-2.00	.057
	Experienced	30	.18	.26	-3.85	$< .01$
	Combined	54	.14	.25	-4.24	$< .01$
	Control	53	.01	.15	-0.56	ns
Stack	Novice	24	.35	.25	-6.90	$< .01$
	Experienced	24	.27	.22	-6.15	$< .01$
	Combined	48	.31	.24	-9.20	$< .01$
	Control	53	.05	.17	-2.15	$< .05$
Tree	Novice	24	.33	.26	-6.13	$< .01$
	Experienced	30	.29	.23	-6.84	$< .01$
	Combined	54	.30	.24	-9.23	$< .01$
	Control	53	.04	.16	-1.78	ns

In both conditions, the vast majority of students were recruited from our undergraduate introductory CS classes, before they studied the data structures of interest; there were also few graduate students from other engineering departments, but none of them was familiar with the data structures in the pre-test, as the learning results show.

Learning outcomes If there were no learning among our tutored subjects, it would obviously not make sense to annotate the corpus to discover how students learn. Hence, the first finding we report is that the tutored students did learn, whereas those in the control group did not. Paired samples t-tests revealed that post-test scores are *significantly higher* than pre-test scores in the two tutored conditions for all the topics, except for lists with the less experienced tutor, where the difference is only marginally significant. If the two tutored groups are aggregated, there is significant difference for all the topics. Students in the control group show significant learning only for stacks, but not for lists and trees. Means, standard deviations, and t-test statistic values are reported in Table 1. N represents the number of subjects in that specific group (for stacks, N is lower for the Experienced tutor, because the tests administered to 6 subjects did not include problems on stacks).

The learning gain, expressed as the difference between post-score and pre-score, of students that received tutoring is *significantly higher* than the learning gain of the students in the control group, for all the topics. This is showed by ANOVA between the aggregated group of tutored students and the control group. For lists, $F(1, 106) = 11.0$, $p < 0.01$. For stacks, $F(1, 100) = 41.4$, $p < 0.01$. For trees, $F(1, 106) = 43.9$, $p < 0.01$.

There is *no significant difference* between the two tutored conditions in terms of learning gain. The fact that

Table 2: Turns, utterances and words: mean (std)

Unit	Tutors	Students
Turns	46.6 (37.4)	45.5 (37.3)
Utterances	186 (107.6)	48.6 (40)
Words	1971.8 (1072)	155.7 (151.3)

students did not learn more with the experienced tutor was an important finding that led us to question the approach of comparing and contrasting more and less experienced tutors. Whereas these specific learning outcomes may be idiosyncratic to our two tutors, and should be confirmed by further experiments, the research direction this finding led us to is independently fruitful.

Corpus analysis

The 54 tutoring sessions were videotaped, and amount to a total of 33 hours and 52 minutes. The videotaped material was subsequently transcribed, following a subset of the conventions described in the transcription manual of the CHILDES project (MacWhinney, 2000). An utterance is a natural unit of speech bounded by breaths or pauses. Figure 1 shows excerpts from two of our dialogues that include some of the transcription conventions. For example, `'+...'` marks trailing; angle brackets mark abandoned speech, or overlap when they occur in two adjacent utterances by two different speakers; `'xxx'` marks unintelligible speech.

Descriptive statistics Table 2 shows the average number of turns, utterances, and words, per tutor (across the two tutors), and per student. Tutors are shown as an aggregate since there are no significant differences between these two tutors along those dimensions. These numbers show that tutors talk much more than students, producing more than 10 times as many words. Tutors produce longer turns, consisting of 4 utterances on average, while students' turns contain 1 utterance on average; and tutors' utterances are longer, consisting of 10.6 words on average, while students' utterances contain 3.2 words on average. Note that the numbers of tutor and student turns are equivalent by construction, since a turn is an uninterrupted sequence of utterances by one of the speakers.

The verbosity of our tutors may surprise the reader. They talk a lot, to the tune of producing 93.5% of the total words! Certainly, they do not resemble the idealized Socratic tutor, who is supposed to prompt the student to construct knowledge by themselves (Chi, 1994): in that scenario, students should do most of the talking. In our view, this is an idealized view of the tutoring process. Whereas our tutors may be extreme in their verbosity, even tutors who appear to come closer to the ideal do talk more and with longer sentences than students. In the CIRCSIM face-to-face studies, tutors produced 63% of the words (Evens & Michael, 2006); their utterances were also longer than students' utterances, with 8.2 words per sentence, as opposed to 5.8 words from students. Person's expert tutors (Cade et al., 2008) also talk more than their students, producing 77% of total words (p.c.).

Another observation from Table 2 is that standard deviations are large. The variability in the length of dialogues is another reason why a regression analysis that can take into account the length of individual sessions, is appropriate.

Coding categories As we noted above, whereas a number of different coding schemes for tutorial dialogues exist, we tried to go back to (some of) the basics: we want the system for categorizing tutoring behavior to be informed by what we know about learning (Ohlsson et al., 2007). We postulated that, given current theories of skill acquisition (Anderson, 1986; Sun, Slusarz, & Terry, 2005; Ohlsson, 2008), the following types of tutorial intervention can be explained in terms of why and how they might support learning:

1. A tutor can tell the student how to perform the task (*direct procedural instruction*).
2. A tutor can provide *feedback*:
   (a) positive, to confirm that a correct but tentative student step is in fact correct;
   (b) negative, to help a student detect and correct an error.
3. A tutor can state declarative information about the domain.

This should not be taken as a full inventory of what a tutor may do, but as a core set of moves that are theoretically motivated (the reader may be surprised not to see *prompts*; please see below for further comments). This inventory may be augmented as necessary, if the regression reveals that these moves do not account for enough of the variance. Hence, we set out to code our dialogues for categories corresponding to these conceptualizations. So far, we have coded for what we call *direct procedural instruction (DPI)*, namely, telling the student what to do, and for *positive* and *negative* feedback. We will come back to the specifics of our coding for DPI and for feedback below.

Results

We present the results of our analysis, performed via linear regression models. Table 3 summarizes significant correlations, separately by topic and cumulatively. We found three models to be significant: Model 1 accounts for prior knowledge via the pre-test, Model 2 accounts for session length as well, Model 3 includes both positive and negative feedback (since the variables were entered one at a time, there exists a significant Model 2.5 that

```
           T:  do you see a problem?
           T:  I have found the node a@l, see here I found the node b@l, and then I put g@l in
               after it.
Begin +    T:  here I have found the node a@l and now the link I have to change is +...
           S:  ++ you have to link e@l <over xxx.> [>]
End +      T:  [<] <yeah> I have to go back to this one.
           S:  *mmhm
           T:  so I *uh once I'm here, this key is here, I can't go backwards.
```

```
Begin -    S:  <so you> [>] <you won't get the same> [//] would you get the same point out of
               writing t@l close to c@l at the top?
           T:  oh, t@l equals c@l.
           T:  no because you would have a type mismatch.
End -      T:  t@l <is a pointer> [//] is an address, and this is contents.
```

Figure 1: Positive and negative feedback (T = tutor, S = student)

adds only positive feedback to Model 2. We don't show it in Table 3 because of space constraints). We note that a measure of student activity we devised did not result in significant correlations.

Prior knowledge (Model 1)

First of all, we need to factor out the effect of *prior knowledge*, measured by the pre-test score. A linear regression model reveals a strong effect of pre-test scores on learning gains (Table 3). However, the R^2 values show that there is a lot of variance left to be explained, especially for lists and stacks, although not so much for trees. Note that the β weights are negative, namely, students with higher pre-test scores learn *less* than students with lower pre-test scores. This is an example of the well-known *ceiling effect*: students with more previous knowledge have less *learning opportunity.*

Time on task (Model 2)

Another variable that is recognized as important by the educational research community is *time on task*, and we can approximate it with the length of the tutoring session. Surprisingly, session length has a significant effect only on lists (Table 3).

Tutor moves (Model 3)

Feedback has been extensively studied in one form or the other. The focus has often been on *negative* feedback (even if indirect, because human tutors avoid too much negativity (Fox, 1993; Lepper et al., 1997)). The role of *positive* feedback, i.e., feedback provided in response to something *correct* the student did or said, has not been studied as much. However, it turns out that positive feedback is much more abundant than negative feedback: in our CS domain, positive feedback occurs eight times as often as negative, and in a previous study of ours, in the domain of solving sequence patterns, it occurred four times as often (Lu, 2007).

The dataset was manually annotated for *feedback episodes*, where either positive or negative feedback is delivered, with acceptable intercoder agreement (κ = 0.67). Examples of feedback episodes are shown in Figure 1.

The number of positive feedback episodes and the number of negative feedback episodes have been introduced in the regression model (Model 3, Table 3). The model showed a significant effect of feedback for lists and stacks, but no significant effect on trees. Interestingly, the effect of positive feedback is *positive*, but the effect of negative feedback is *negative*, as can be seen by the sign of the β values.

Table 3: Linear regression – human tutoring

Topic	Model	Predictor	β	R^2	p
List	1	Pre-test	-.45	.18	< .05
	2	Pre-test	-.40	.28	< .05
		Session length	.35		< .05
	3	Pre-test	-.35	.36	< .05
		Session length	.33		.05
		+ feedback	.46		.05
		- feedback	-.53		< .05
Stack	1	Pre-test	-.53	.26	< .01
	2	Pre-test	-.52	.24	< .01
		Session length	.05		ns
	3	Pre-test	-.58	.33	< .01
		Session length	.01		ns
		+ feedback	.61		< .05
		- feedback	-.55		< .05
Tree	1	Pre-test	-.79	.61	< .01
	2	Pre-test	-.78	.60	< .01
		Session length	.03		ns
	3	Pre-test	-.77	.59	< .01
		Session length	.04		ns
		+ feedback	.06		ns
		- feedback	-.12		ns
All	1	Pre-test	-.52	.26	< .01
	2	Pre-test	-.54	.29	< .01
		Session length	.20		< .05
	3	Pre-test	-.57	.32	< .01
		Session length	.16		.06
		+ feedback	.30		< .05
		- feedback	-.23		.05

We additionally annotated the episodes of positive and negative feedback for *initiative*. An episode can be ini-

Table 4: Feedback initiative: mean (std)

	Student initiative	Tutor initiative
Positive feedback	3.9 (3.8)	10.2 (9.1)
Negative feedback	1.7 (1.2)	2.0 (1.2)

tiated either by the *student* or by the *tutor*. In the first case, the student volunteers some information without being asked or prompted by the tutor, and the tutor replies with some feedback (as in the bottom part of Figure 1). In the second case, the tutor first asks or prompts the student (not necessarily verbally), then the student replies, and finally the tutor provides feedback on the student's answer (as in the top part of Figure 1). The distribution of initiative labels is reported in Table 4. The numbers in the table are aggregated across the three topics, but splitting the three topics apart revealed similar patterns.

ANOVA revealed overall significant differences among the four groups ($F(3, 325) = 43.27$, $p < 0.01$). Tukey post-hoc test revealed significant differences ($p < 0.01$) between positive-tutor and positive-student; positive-tutor and negative-tutor; and positive-tutor and negative-student. Namely, there are significantly more feedback episodes, both positive and negative, initiated by the tutor, rather than by the student; and the tutors themselves initiate significantly more positive than negative feeback episodes. Note that tutor initiative as coded here is a subset of a more general *prompt* category, that includes prompts that did not result in subsequent feedback; the latter are rare in our data.

Direct procedural instruction (DPI) In our coding manual, we define DPI as the tutor directly telling the student what to do. This includes: correct steps that lead to the solution of a problem (e.g., "and there is nothing there, so we put six right there"); high-level steps or subgoals (e.g., "it wants us to put the new node that contains G in it, after the node that contains B"); and tactics and strategies (e.g., "so with these kind of problems, the first thing I have to say is always draw pictures"). We initially tried to distinguish goals and subgoals from the rest, but failed; when we collapsed all these categories, coders managed to reach an acceptable level of intercoder agreement, $\kappa = 0.71$.

Linear regression showed a significant positive correlation between DPI and learning gain for lists ($\beta = 0.0038$, $t(49) = 2.69$, $R^2 = 0.11$, $p < 0.01$) and trees ($\beta = 0.0024$, $t(50) = 3.07$, $R^2 = 0.14$, $p < 0.01$). Currently, we are incorporating DPI in the multiple regression models.

Discussion and Conclusions

The results of our analyses are complicated in their details, but relatively straightforward in principle: Prior knowledge has a strong effect on learning outcomes, time on task some, although not as much. This is not surprising. More interesting is that several aspects of the tutor's behavior impact learning gains, as evidenced by variables accounting for variance in the learning outcomes, over and above the variance accounted for by prior knowledge and time on task. To our surprise, the evidence is mounting that tutors engage in direct instruction, and that this may be effective. Exactly why direct instruction may be effective in the one-on-one context, when it is generally believed not to be effective in a classroom setting remains to be explained. The data also indicate that tutorial feedback is important, but the evidence for positive feedback is as strong if not stronger than the evidence for negative feedback. This is interesting, given that negative feedback directly informs the student about what he or she needs to change, while positive feedback requires that the student already got the answer or the problem solving step correct. Much work in ITS research, including our own past work, has assumed that negative feedback is pedagogically more powerful than positive feedback; the latter has been interpreted mostly in motivational terms. Our results indicate that this view might need to be reconsidered. Methodologically, the multiple regression analysis allows us and other ITS researchers to apply a stringent criterion for empirical support for a particular aspect of tutor behavior: that it accounts for unique variance in learning outcomes.

The next steps are to understand whether two additional tutoring modes, providing declarative knowledge and prompting, contribute to learning – prompting consists of both the tutor initiative in feedback episodes we already coded for, and other cases we have recently coded for. With the multiple regression analysis, the field can gradually sort out what works and what does not, and so assemble an effective toolkit of tutoring modes that allows routine design and implementation of effective intelligent tutoring systems.

Finally, from the point of view of CS pedagogy, our models are stronger and more interesting for lists, in that prior knowledge has the least explanatory power. This agrees with our beliefs as CS educators that lists are among the most difficult concepts that new students need to master. Hence, partly on the basis of the results of our corpus analysis, we have built various versions of *iList*, a tutoring system that focuses on tutoring lists. *iList* has been already successfully deployed in classrooms, both at our institution and at the US Naval Academy in Annapolis (Fossati, Di Eugenio, Brown, & Ohlsson, 2008). We are currently endowing the latest version with the ability of providing both positive and negative feedback and direct instruction.

Acknowledgments

This work is supported by ONR (Award N00014–07–1–0040), and by the UIC Graduate College (2008/2009

Dean's Scholar Award).

References

Anderson, J. R. (1986). Knowledge compilation: The general learning mechanism. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning* (Vol. 5, pp. 289–310). Los Altos, CA: Kaufmann.

Cade, W. L., Copeland, J. L., Person, N. K., & D'Mello, S. K. (2008). Dialogue modes in expert tutoring. In *Intelligent tutoring systems* (Vol. 5091, p. 470-479). Springer Berlin / Heidelberg.

Chi, M. T. H., de Leeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*(3), 439-477.

Chi, M. T. H., Siler, S. A., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, *25*, 471-533.

Corbett, A. T., & Anderson, J. R. (1990). The effect of feedback control on learning to program with the Lisp tutor. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 796–803). Cambridge, Ma.

Di Eugenio, B., Fossati, D., Haller, S., Yu, D., & Glass, M. (2008). Be brief, and they shall learn: Generating concise language feedback for a computer tutor. *International Journal of AI in Education*, *18*(4), 317-345.

Di Eugenio, B., Kershaw, T. C., Lu, X., Corrigan-Halpern, A., & Ohlsson, S. (2006). Toward a computational model of expert tutoring: a first report. In *FLAIRS06, the 19th International Florida AI Research Symposium.* Melbourne Beach, FL.

Evens, M. W., & Michael, J. A. (2006). *One-on-one tutoring by humans and machines.* Mahwah, NJ: Lawrence Erlbaum Associates.

Fossati, D., Di Eugenio, B., Brown, C., & Ohlsson, S. (2008). Learning Linked Lists: Experiments with the iList System. In *ITS 2008, the 9th International Conference on Intelligent Tutoring Systems.*

Fox, B. A. (1993). *The human tutorial dialogue project: Issues in the design of instructional systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Graesser, A., Person, N., & Magliano, J. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, *9*, 495-522.

Katz, S., Allbritton, D., Aronis, J. M., Wilson, C., & Soffa, M. L. (2003). Gender and race in predicting achievement in computer science. *IEEE Technology and Society, Special Issue on Women and Minorities in Information Technology*, *22*(3), 20–27.

Lane, H. C., & VanLehn, K. (2003). Coached program planning: Dialogue-based support for novice program design. In *Proceedings of the Thirty-Fourth Technical Symposium on Computer Science Education (SIGCSE '03)* (pp. 148–152). ACM Press.

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

Litman, D. J., Rosé, C. P., Forbes-Riley, K., VanLehn, K., Bhembe, D., & Silliman, S. (2006). Spoken versus typed human and computer dialogue tutoring. *International Journal of Artificial Intelligence in Education*, *16*, 145–170.

Lu, X. (2007). *Expert tutoring and natural language feedback in intelligent tutoring systems.* Unpublished doctoral dissertation, University of Illinois - Chicago.

MacWhinney, B. (2000). *The CHILDES project. tools for analyzing talk: Transcription format and programs* (Third ed., Vol. 1). Lawrence Erlbaum, Mahwah, NJ.

Mitrović, A., Suraweera, P., Martin, B., & Weerasinghe, A. (2004). DB-suite: Experiences with three intelligent, web-based database tutors. *Journal of Interactive Learning Research*, *15*(4), 409–432.

Moore, J. D., Porayska-Pomsta, K., Varges, S., & Zinn, C. (2004). Generating Tutorial Feedback with Affect. In *FLAIRS04, Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference.*

Ohlsson, S. (2008). Computational models of skill acquisition. In R. Sun (Ed.), *The Cambridge Handbook of Computational Psychology* (p. 359-395). Cambridge, UK: Cambridge University Press.

Ohlsson, S., Di Eugenio, B., Chow, B., Fossati, D., Lu, X., & Kershaw, T. (2007). Beyond the code-and-count analysis of tutoring dialogues. In *AIED 2007, the 13th International Conference on Artificial Intelligence in Education.*

Person, N., Graesser, A., Magliano, J., & Kreuz, R. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. *Learning and Individual Differences*, *6*(2), 205–229.

Sun, R., Slusarz, P., & Terry, C. (2005). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review, 112.*

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P. W., Olney, A., & Rosé, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, *31*(1), 3–62.