Exploring Effective Dialogue Act Sequences in One-on-one Computer Science Tutoring Dialogues

Lin Chen, Barbara Di Eugenio
Computer Science
U. of Illinois at Chicago
lchen43,bdieugen@uic.edu

Davide Fossati
Computer Science
Carnegie Mellon U. in Qatar
davide@fossati.us

Stellan Ohlsson, David Cosejo
Psychology
U. of Illinois at Chicago
stellan,dcosej1@uic.edu

Abstract

We present an empirical study of one-on-one human tutoring dialogues in the domain of Computer Science data structures. We are interested in discovering effective tutoring strategies, that we frame as discovering which Dialogue Act (DA) sequences correlate with learning. We employ multiple linear regression, to discover the strongest models that explain why students learn during one-on-one tutoring. Importantly, we define "flexible" DA sequence, in which extraneous DAs can easily be discounted. Our experiments reveal several cognitively plausible DA sequences which significantly correlate with learning outcomes.

1 Introduction

One-on-one tutoring has been shown to be a very effective form of instruction compared to other educational settings. Much research on discovering why this is the case has focused on the analysis of the interaction between tutor and students (Fox, 1993; Graesser et al., 1995; Lepper et al., 1997; Chi et al., 2001). In the last fifteen years, many such analyses have been approached from a Natural Language Processing (NLP) perspective, with the goal of building interfaces that allow students to naturally interact with Intelligent Tutoring Systems (ITSs) (Moore et al., 2004; Cade et al., 2008; Chi et al., 2010). There have been two main types of approaches to the analysis of tutoring dialogues. The first kind of approach compares groups of subjects interacting with different tutors (Graesser et al., 2004; VanLehn et al., 2007), in some instances contrasting the number of occurrences of relevant features between the groups (Evens and Michael, 2006; Chi et al., 2010). However, as we already argued in (Ohlsson et al., 2007), this code-and-count methodology only focuses on what a certain type of tutor (assumed to be better according to certain criteria) does *differently* from another tutor, rather than on strategies that may be effective independently from their frequencies of usage by different types of tutor. Indeed we had followed this same methodology in previous work (Di Eugenio et al., 2006), but a key turning point for our work was to discover that our expert and novice tutors were equally effective (please see below).

The other kind of approach uses linear regression analysis to find correlations between dialogue features and learning gains (Litman and Forbes-Riley, 2006; Di Eugenio et al., 2009). Whereas linear regression is broadly used to analyze experimental data, only few analyses of tutorial data or tutoring experiments use it. In this paper, we follow Litman and Forbes-Riley (2006) in correlating sequences of Dialogue Acts (DAs) with learning gains. We extend that work in that our bigram and trigram DAs are not limited to tutor-student DA bigrams – Litman and Forbes-Riley (2006) only considers bigrams where one DA comes from the tutor's turn and one from the student's turn, in either order. Importantly, we further relax constraints on how these sequences are built, in particular, we are able to model DA sequences that include gaps. This allows us to discount the noise resulting from intervening DAs that do not contribute to the effectiveness of the specific sequence. For example, if we want to

Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, pages 65–75,
Portland, Oregon, 24 June 2011.

explore sequences in which the tutor first provides some knowledge to solve the problem (DPI) and then knowledge about the problem (DDI) (DPI and DDI will be explained later), an exchange such as the one in Figure 1 should be taken into account (JAC and later LOW are the tutors, students are indicated with a numeric code, such as 113 in Figure 1). However, if we just use adjacent utterances, the *ok* from the student (113) interrupts the sequence, and we could not take this example into account. By allowing gaps in our sequences, we test a large number of linear regression models, some of which result in significant models that can be used as guidelines to design an ITS. Specifically, these guidelines will be used for further improvement of iList, an ITS that provides feedback on linked list problems and that we have developed over the last few years. Five different versions of iList have been evaluated with 220 users (Fossati et al., 2009; Fossati et al., 2010). iList is available at http://www.digitaltutor.net, and has been used by more than 550 additional users at 15 different institutions.

JAC: so we would set k equal to e and then delete. [*DPI*]
113: ok.
JAC: so we've inserted this whole list in here.[*DDI*]
113: yeah.

Figure 1: {DPI, DDI} Sequence Excerpt

The rest of the paper is organized as follows. In Section 2, we describe the CS-Tutoring corpus, including data collection, transcription, and annotation. In Section 3, we introduce our methodology that combines multiple linear regression with n-grams of DAs that allow for gaps. We discuss our experiments and results in Section 4.

2 The CS Tutoring Corpus

2.1 Data Collection

During the time span of 3 semesters, we collected a corpus of 54 one-on-one tutoring sessions on Computer Science data structures: *linked list*, *stack* and *binary search tree*. (In the following context, we will refer them as *Lists*, *Stacks* and *Trees*). Each student only participated in one session, and was randomly assigned to one of two tutors: LOW, an experienced Computer Science professor, with more than 30 years of teaching experience; or JAC, a senior undergraduate student in Computer Science, with only one semester of previous tutoring experience. In the end 30 students interacted with LOW and 24 with JAC.

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, and learning gains were computed.

Table 1 includes information on session length. Note that for each topic, the number of sessions is lower than 54. The tutor was free to tutor on what he felt was more appropriate, after he was given an informal assessment of the student's performance on the pre-test (tutors were not shown pre-tests to avoid that they'd tutor to the pre-test only). Hence, not every student was tutored on every topic.

Topic	N	Session length (minutes)				
		Min	Max	Total	μ	σ
Lists	52	3.4	41.4	750.4	14.4	5.8
Stacks	46	0.3	9.4	264.5	5.8	1.8
Trees	53	9.1	40.0	1017.6	19.2	6.6
Sessions	54	12.8	61.1	2032.5	37.6	6.1

Table 1: CS Tutoring Corpus - Descriptives

Each tutoring session was videotaped. The camera was pointing at the sheets of paper on which tutors and students were writing during the session. The videos were all transcribed. The transcripts were produced according to the rules and conventions described in the transcription manual of the CHILDES project (MacWhinney, 2000). Dialogue excerpts included in this paper show some of the transcription conventions. For example, `'+...'` denotes trailing, `'xxx'` unintelligible speech and `'#'` a short pause (see Figure 2). The CHILDES transcription manual also provides directions on utterance segmentation.

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

LOW: what's the if? [*Prompt*]
LOW: well of course, don't do this if t two is null so if t two isn't null we can do that and xxx properly # thinking I put it in here. [*DPI*]
LOW: or else if t two is null that's telling us that this is the +... [*Prompt,FB*]

Figure 2: {Prompt,DPI,FB} sequence excerpt

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.

The learning gain, expressed as the difference between post-score and pre-score, of students that received tutoring was *significantly higher* than the learning gain of the students in the control group, for all the topics. This was showed by ANOVA between the aggregated group of tutored students and the control group, and was significant at the $p < 0.01$ for each topic. There was *no significant difference* between the two tutored conditions in terms of learning gain. The fact that 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.

Please refer to (Di Eugenio et al., 2009) for further descriptive measurements of the corpus.

2.2 Dialogue Act Annotation

Many theories have been proposed as concerns DAs, and there are many plausible inventories of DAs, including for tutorial dialogue (Evens and Michael, 2006; Litman and Forbes-Riley, 2006; Boyer et al., 2010). We start from a minimalist point of view, postulating that, according to current theories of skill acquisition (Anderson, 1986; Sun et al., 2005; Ohlsson, 2008), at least 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.
2. A tutor can state declarative information about the domain.
3. A tutor can provide feedback:
(a) positive, to confirm that a correct but tentative step is in fact correct;
(b) negative, to help a student detect and correct an error.

We first read through the entire corpus and examined it for impressions and trends, as suggested by (Chi, 1997). Our informal assessment convinced us that our minimalist set of tutoring moves was an appropriate starting point. For example, contrary to much that has been written about an idealized socratic type of tutoring where students build knowledge by themselves (Chi et al., 1994), our tutors are rather directive in style, namely, they do a lot of *telling* and *stating*. Indeed our tutors talk a lot, to the tune of producing 93.5% of the total words! We translated the four types above into the following DAs: Direct Procedural Instruction (DPI), Direct Declarative Instruction (DDI), Positive Feedback (+FB), and Negative Feedback (-FB). Besides those 4 categories, we additionally annotated the corpus for Prompt (PT), since our tutors did explicitly invite students to be active in the interaction. We also annotated for Student Initiative (SI), to capture active participation on the part of the student's. SI occurs when the student proactively produces a meaningful utterance, by providing unsolicited explanation (see Figures 6 and 4), or by asking questions. As we had expected, SIs are not as frequent as other moves (see below). However, this is precisely the kind of move that a regression analysis would tease out from others, if it correlates with learning, even if it occurs relatively infrequently. This indeed happens in two models, see Table 8.

Direct Procedural Instruction(DPI) occurs when the tutor directly tells the student what task to perform. More specifically:

- Utterances containing correct steps that lead to the solution of a problem, e.g. see Figure 1.
- Utterances containing high-level steps or subgoals (*it wants us to put the new node that contains G in it, after the node that contains B*).
- Utterances containing tactics and strategies (*so with these kinds of problems, the first thing I have to say is always draw pictures*).
- Utterances where the tutor talked in the first-person but in reality the tutor instructed the student on what to do (*So I'm pushing this value onto a stack. So I'm pushing G back on*).

Direct Declarative Instruction (DDI) occurred when the tutor provided facts about the domain or

a specific problem. The key to determine if an utterance is DDI is that the tutor is telling the student something that he or she ostensibly does not already know. Common sense knowledge is not DDI (*ten is less than eleven*). Utterances annotated as DDI include:

- Providing general knowledge about data structures (*the standard format is right child is always greater than the parent, left child is always less than the parent*).
- Telling the student information about a specific problem (*this is not a binary search tree*).
- Conveying the results of a given action (*so now since we've eliminated nine, it's gone*).
- Describing pictures of data structures (*and then there is a link to the next node*).

Prompts (PT) occur when the tutor attempts to elicit a meaningful contribution from the student. We code for six types of tutor prompts, including:

- Specific prompt: An attempt to get a specific response from the student (*that's not b so what do we want to do?*).
- Diagnosing: The tutor attempts to determine the student's knowledge state (*why did you put a D there?*).
- Confirm-OK: The tutor attempts to determine if the student understood or if the student is paying attention (*okay, got that idea?*).
- Fill-in-the-blank: The tutor does not complete an utterance thereby inviting the student to complete the utterance, e.g. see Figure 2.

Up to now we have discussed annotations for utterances that do not explicitly address what the student has said or done. However, many tutoring moves concern providing feedback to the student. Indeed as already known but not often acted upon in ITS interfaces, tutors do not just point out mistakes, but also confirm that the student is making correct steps. While the DAs discussed so far label single utterances, our positive and negative feedback (+FB and -FB) annotations comprise a sequence of consecutive utterances, that starts where the tutor starts providing feedback. We opted for a sequence of utterances rather than for labeling one single utterance because we found it very difficult to pick one single utterance as the one providing feedback, when the tutor may include e.g. an explanation that we consider to be part of feedback. Positive feedback occurs when the student says or does something correct, either spontaneously or after being prompted by the tutor. The tutor acknowledges the correctness of the student's utterance, and possibly elaborates on it with further explanation. Negative feedback occurs when the student says or does something incorrect, either spontaneously or after being prompted by the tutor. The tutor reacts to the mistake and possibly provides some form of explanation.

After developing a first version of the coding manual, we refined it iteratively. During each iteration, two human annotators independently annotated several dialogues for one DA at a time, compared outcomes, discussed disagreements, and fine-tuned the scheme accordingly. This process was repeated until a sufficiently high inter-coder agreement was reached. The Kappa values we obtained in the final iteration of this process are listed in Table 2 (Di Eugenio and Glass, 2004; Artstein and Poesio, 2008). In Table 2, the "Double Coded*" column refers to the sessions that we double coded to calculate the inter-coder agreement. This number does not include the sessions which were double coded when coders were developing the coding manual. The numbers of double-coded sessions differ by DA since it depends on the frequency on the particular DA (recall that we coded for one DA at a time). For example, since Student Initiatives (SI) are not as frequent, we needed to double code more sessions to find a number of SI's high enough to compute a meaningful Kappa (in our whole corpus, there are 1157 SIs but e.g. 4957 Prompts).

Category	Double Coded*	Kappa
DPI	10	.7133
Feedback	5	.6747
DDI	10	.8018
SI	14	.8686
Prompt	8	.9490

Table 2: Inter-Coder Agreement in Corpus

The remainder of the corpus was then independently annotated by the two annotators. For our final corpus, for the double coded sessions we did not come to a consensus label when disagreements arose; rather, we set up a priority order based on

topic and coder (e.g., during development of the coding scheme, when coders came to consensus coding, which coder's interpretation was chosen more often), and we chose the annotation by a certain coder based on that order.

As a final important note, given our coding scheme some utterances have more than one label (see Figures 2 and 4), whereas others are not labelled at all. Specifically, most student utterances, and some tutor utterances, are not labelled (see Figures 1 and 4).

3 Method

3.1 Linear Regression Models

In this work, we adopt a multiple regression model, because it can tell us how much variation in learning outcomes is explained by the variation of individual features in the data. The features we use include pre-test score, the length of the tutoring sessions, and DAs, both the single DAs we annotated for and DA *n-grams*, i.e. DA sequences of length n. Pre-test score is always included since the effect of previous knowledge on learning is well established, and confirmed in our data (see all Models 1 in Table 4); indeed multiple linear regression allows us to factor out the effect of previous knowledge on learning, by quantifying the predictive power of features that are added beyond pre-test score.

3.2 n-gram Dialogue Act Model

n-grams (sequences of n units, such as words, POS tags, dialogue acts) have been used to derive language models in computational linguistics for a long time, and have proven effective in tasks like part-of-speech tagging, spell checking.

Our innovation with regard to using DA n-grams is to allow gaps in the sequence. This allows us to extract the sequences that are really effective, and to eliminate noise. Note that from the point of view of an effective sequence, *noise* is anything that does not contribute to the sequence. For example, a tutor's turn may be interrupted by a student's acknowledgments, such as "OK" or "Uh-hah" (see Figure 1). Whereas these acknowledgments perform fundamental functions in conversation such as grounding (Clark, 1992), they may not directly correlate with learning (a hypothesis to test). If we counted them in the sequence, they would contribute two utterances, transforming a 3 DA sequence into a 5 DA sequence. As well known, the higher the n, the sparser the data becomes, i.e., the fewer sequences of length n we find, making the task of discovering significant correlations all the harder. Note that some of the bigrams in (Litman and Forbes-Riley, 2006) could be considered to have gaps, since they pair one student move (say SI) with each tutor move contained in the next tutor turn (eg, in our Figure 6 they would derive two bigrams [SI, FB], and [SI, Prompt]). However, this does not result in a systematic exploration of all possible sequences of a certain length n, with all possible gaps of length up to m, as we do here.

The tool that allows us to leave gaps in sequences is part of Apache Lucene,[1] an open source full text search library. It provides strong capabilities to match and count efficiently. Our counting method is based on two important features provided by Lucene, that we already used in other work (Chen and Di Eugenio, 2010) to detect uncertainty in different types of corpora.

- Synonym matching: We can specify several different tokens at the same position in a field of a document, so that each of them can be used to match the query.
- Precise gaps: With Lucene, we can precisely specify the gap between the matched query and the indexed documents (sequences of DAs in our case) using a special type of query called *SpanNearQuery*.

To take advantage of Lucene as described above, we use the following algorithm to index our corpus.

1. For each Tutor-Topic session, we generate n-gram utterance sequences – note that these are sequences of utterances at this point, not of DAs.
2. We prune utterance sequences where either 0 or only 1 utterance is annotated with a DA, because we are mining sequences with at least 2 DAs. Recall that given our annotation, some utterances are not annotated (see e.g. Figure 1).
3. After pruning, for each utterance sequence, we generate a Lucene document: each DA label on an utterance will be treated as a token, multiple

[1] http://lucene.apache.org/

labels on the same utterance will be treated as "synonyms".

By indexing annotations as just described, we avoid the problem of generating too many combinations of labels. After indexing, we can use SpanNearQuery to query the index. SpanNearQuery allows us to specify the position distance allowed between each term in the query.

Figure 3 is the field of the generated Lucene document corresponding to the utterance sequences in Figure 4. We can see that each utterance of the tutor is tagged with 2 DAs. Those 2 DAs produce 2 tokens, which are put into the same position. The tokens in the same position act as synonyms to each other during the query.

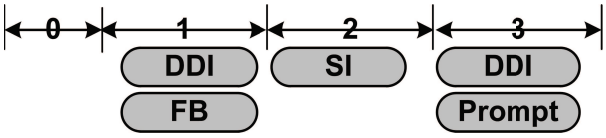

Figure 3: Lucene Document Example for DAs

258: okay.
JAC: its right child is eight. [*DDI*, *FB*]
258: uh no it has to be greater than ten. [*SI*]
JAC: right so it's not a binary search tree # it's not a b s t, right? [*DDI*,*Prompt*]

Figure 4: {FB, SI, DDI} is most effective in Trees

4 Experiments and Results

Here we build on our previous results reported in (Di Eugenio et al., 2009). There we had shown that, for lists and stacks, models that include positive and negative feedback are significant and explain more of the variance with respect to models that only include pre-test score, or include pre-test score and session length. Table 4 still follows the same approach, but adds to the regression models the additional DAs, DPI, DDI, Prompt and SI that had not been included in that earlier work. The column M refers to three types of models, Model 1 only includes Pre-test, Model 2 adds session length to Pre-test, and Model 3 adds to Pre-test all the DAs. As evidenced by the table, only DPI provides a marginally significant contribution, and only for lists. Note that length is not included in Model 3's. We did run all the equivalent models to Model 3's including length. The R^2's stay the same (literally, to the second decimal digit), or minimally decrease. However, in all these Model 3+'s that include length no DA is significant, hence we consider them as less explanatory than the Model 3's in Table 4: finding that a longer dialogue positively affects learning does not tell us what happens during that dialogue which is conducive to learning.

Note that the β weights on the pre-test are always negative in every model, 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*. Also noticeable is that the R^2 for the Trees models are much higher than for Lists and Stacks, and that for Trees no DA is significant (although there will be significant trigram models that involve DAs for Trees). We have observed that Lists are in general more difficult than Stacks and Trees (well, at least than binary search trees) for students.

Topic	Pre-Test	σ	Gain	σ
Lists	.40	.27	.14	.25
Stacks	.29	.30	.31	.24
Trees	.50	.26	.30	.24

Table 3: Learning gains and t-test statistics

Indeed Table 3 shows that in the CS-tutoring corpus the average learning gain is only .14 for Lists, but .31 for Stacks and .30 for Trees; whereas students have the lowest pre-test score on Stacks, and hence they have more opportunities for learning, they learn as much for Trees, but not for Lists.

We now examine whether DA sequences help us explain why student learn. We have run 24 sets of linear regression experiments, which are grouped as the following 6 types of models.

- With DA bigrams (DA sequences of length 2):
  - Gain $\sim$ DA Bigram
  - Gain $\sim$ DA Bigram + Pre-test Score
  - Gain $\sim$ DA Bigram + Pre-test Score + Session Length
- With DA trigrams (DA sequences of length 3):
  - Gain $\sim$ DA Trigram
  - Gain $\sim$ DA Trigram + Pre-test Score
  - Gain $\sim$ DA Trigram + Pre-test Score + Session Length

For each type of model:

Topic	M	Predictor	β	R^2	P
Lists	1	Pre-test	$-.47$	.20	$< .001$
	2	Pre-test	$-.43$	.29	$< .001$
		Length	.01		$< .001$
	3	Pre-test	$-.500$	.377	$< .001$
		+FB	.020		$< .01$
		-FB	.039		ns
		DPI	.004		$< .1$
		DDI	.001		ns
		SI	.005		ns
		Prompt	.001		ns
Stacks	1	Pre-test	$-.46$	.296	$< .001$
	2	Pre-test	$-.46$	.280	$< .001$
		Length	$-.002$		ns
	3	Pre-test	$-.465$	.275	$< .001$
		+FB	$-.017$		$< .01$
		-FB	$-.045$		ns
		DPI	.007		ns
		DDI	.001		ns
		SI	.008		ns
		Prompt	$-.006$		ns
Trees	1	Pre-test	$-.739$	.676	$< .001$
	2	Pre-test	$-.733$	.670	$< .001$
		Length	.001		ns
	3	Pre-test	$-.712$	.667	$< .001$
		+FB	$-.002$		ns
		-FB	$-.018$		ns
		DPI	$-.001$		ns
		DDI	$-.001$		ns
		SI	$-.001$		ns
		Prompt	$-.001$		ns
All	1	Pre-test	$-.505$	.305	$< .001$
	2	Pre-test	$-.528$	.338	$< .001$
		Length	.06		$< .001$
	3	Pre-test	$-.573$	.382	$< .001$
		+FB	.009		$< .001$
		-FB	$-.024$		ns
		DPI	.001		ns
		DDI	.001		ns
		SI	.001		ns
		Prompt	.001		ns

Table 4: Linear Regression – Human Tutoring

1. We index the corpus according to the length of the sequence (2 or 3) using the method we introduced in section 3.2.
2. We generate all the permutations of all the DAs we annotated for within the specified length; count the number of occurrences of each permutation using Lucene's SpanNearQuery allowing for gaps of specified length. Gaps can span from 0 to 3 utterances; for example, the excerpt in Figure 1 will be counted as a {DPI, DDI} bigram with a gap of length 1. Gaps can be discontinuous.
3. We run linear regressions[2] on the six types of models listed above, generating actual models by replacing a generic DA bi- or tri-gram with each possible DA sequence we generated in step 2.
4. We output those regression results, in which the whole model and every predictor are at least marginally significant ($p < 0.1$).

The number of generated significant models is shown in Figure 5. In the legend of the Figure, B stands for **B**igram DA sequence, T stands for **T**rigram DA sequence, L stands for session **L**ength, P stands for **P**re-test score. Not surprisingly, Figure 5 shows that, as the allowed gap increases in length, the number of significant models increases too, which give us more models to analyze.

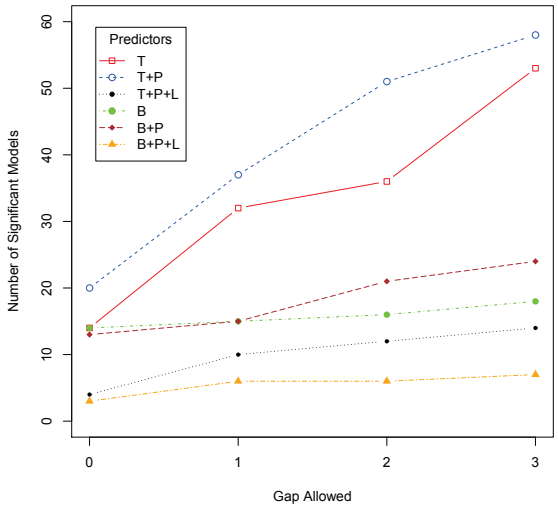

Figure 5: Gaps Allowed vs. Significant Models

Figure 5 shows that there are a high number of significant models. In what follows we will present first of all those that improve on the models that do not use sequences of DAs, as presented in Table 4. Improvement here means not only that the R^2 is higher, but that the model is more appropriate as an approximation of a tutor strategy, and hence, constitutes a better guideline for an ITS. For example, take model 3 for Lists in Table 4. It tells us that positive feedback (+FB) and direct procedural instruction (DPI) positively correlate with learning

[2]We used rJava, http://www.rforge.net/rJava/

gains. However, this obviously cannot mean that our ITS should only produce +FB and DPI. The ITS is interacting with the student, and it needs to tune its strategies according to what happens in the interaction; model 3 doesn't even tell us if +FB and DPI should be used together or independently. Models that include sequences of DAs will be more useful for the design of an ITS, since they point out what sequences of DAs the ITS may use, even if they still don't answer the question, when should the ITS engage in a particular sequence – we have addressed related issues in our work on iList (Fossati et al., 2009; Fossati et al., 2010).

4.1 Bigram Models

{**DPI, Feedback**} **Model** Indeed the first significant models that include a DA bigram include the {DPI, Feedback} DA sequence. Note that we distinguish between models that employ Feedback (FB) without distinguishing between positive and negative feedback; and models where the type of feedback is taken into account (+FB, -FB). Table 5 shows that for Lists, a sequence that includes DPI followed by any type of feedback (Feedback, +FB, -FB) produces significant models when the model includes pre-test. Table 5 and all tables that follow include the column *Gap* that indicates the length of the gap within the DA sequence with which that model was obtained. When, as in Table 5, multiple numbers appear in the *Gap* column, this indicates that the model is significant with all those gap settings. We only show the β, R^2 and P values for the gap length which generates the highest R^2 for a model, and the corresponding gap length is in bold font: for example, the first model for Lists in Table 5 is obtained with a gap length = 2. For Lists, these models are not as predictive as Model 3 in Table 4, however we believe they are more useful from an ITS design point of view: they tell us that when the tutor gives direct instruction on how to solve the problem, within a short span of dialogue the tutor produces feedback, since (presumably) the student will have tried to apply that DPI. For Stacks, a {DPI, -FB} model (without taking pre-test into account) significantly correlates ($p < 0.05$) with learning gain, and marginally significantly correlates with learning gain when the model also includes pre-test score. This latter model is actually more predictive than Model 3 for Stacks in Table 4 that includes +FB but not DPI. We can see the β weight is negative for the sequence {DPI, -FB} in the Stacks model. No models including the bigram {DPI, -FB} are significant for Trees.

Topic	Predictor	β	R^2	P	Gap
Lists	DPI, -FB	.039	.235	$<.001$	**2**, 3
	Pre-test	$-.513$		$< .001$	
	DPI, +FB	.019	.339	$<.001$	0, **1**, 2, 3
	Pre-test	$-.492$		$< .001$	
	Length	.011		< 0.05	
	DPI, FB	.016	.333	$<.05$	0, **1**, 2, 3
	Pre-test	$-.489$		$< .001$	
	Length	.011		< 0.05	
Stacks	DPI, -FB	$-.290$	.136	$<.05$	**0**, 1, 2, 3
	DPI, -FB	$-.187$	.342	$<.1$	0, **1**, 2, 3
	Pre-test	$-.401$		$< .001$	

Table 5: DPI, Feedback Model

{**FB, DDI**} **Model** A natural question arises: since Feedback following DPI results in significant models, are there any significant models which include sequences whose first component is a Feedback move? We found only two that are significant, when Feedback is followed by DDI (Direct Declarative Instruction). Note that here we are not distinguishing between negative and positive feedback. Those models are shown in Table 6. The Lists model is not more effective than the original Model 3 for Lists in Table 4, but the model for Trees is slightly more explanatory than the best model for Trees in that same table, and includes a bigram model, whereas in Table 4, only pre-test is significant for Trees.

Topic	Predictor	β	R^2	P	Gap
Lists	FB, DDI	.1478	.321	$<.1$	1
	Pre-test	$-.470$		$< .001$	
	Length	.011		$< .05$	
Trees	FB, DDI	.0709	.6953	$<.05$	0
	Pre-test	$-.7409$		$< .001$	

Table 6: {FB, DDI} Model

4.2 Trigram Models

{**DPI, FB, DDI**} **Model** Given our significant bigram models for DPI followed by FB, and FB followed by DDI, it is natural to ask whether the combined trigram model {DPI, FB, DDI} results in a significant model. It does for the topic List, as shown in table 7, however again the R^2 is lower than

that of Model 3 in Table 4. This suggests that an effective tutoring sequence is to provide instruction on how to solve the problem (DPI), then Feedback on what the student does, and finally some declarative instruction (DDI).

Topic	Predictor	β	R^2	P	Gap
Lists	DPI, FB, DDI	.156	.371	<.01	1
	Pre-test	−.528		< .001	
	Length	.012		< .05	

Table 7: {DPI, FB, DDI} Model

More effective trigram models include Prompt and SI. Up to now, only one model including sequences of DAs was superior to the simpler models in Table 4. Interestingly, different trigrams that still include some form of Feedback, DPI or DDI, and then either Prompt or SI (Student Initiative) result in models that exhibit slightly higher R^2; additionally in all these models the trigram predictor is highly significant. These models are listed in table 8 (note that the two Trees models differ because in one FB is generic Feedback, irregardless of orientation, in the other it's +FB, i.e., positive feedback). In detail, improvements in R^2 are 0.0382 in topic Lists, 0.12 in topic Stacks and 0.0563 in topic Trees. The highest improvement is in Stacks.

Topic	Predictor	β	R^2	P	Gap
Lists	PT,DPI,FB	.266	.415	<.01	0
	Pre-test	−.463		< .001	
	Length	.011		< .05	
Stacks	DDI,FB,PT	−.06	.416	<.01	1
	Pre-test	−.52		< .001	
Trees	+FB,SI,DDI	.049	.732	<.01	1
	Pre-test	−.746		< .001	
Trees	FB,SI,DDI	.049	.732	<.01	1
	Pre-test	−.746		< .001	

Table 8: Highest R^2 Models

It is interesting to note that the model for Lists add *Prompt* at the beginning to a bigram that had already been found to contribute to a significant model. For Trees, likewise, we add another DA to the bigram {FB,DDI} that had been found to be significant; this time, it is Student Initiative (SI) and it occurs in the middle. This indicates that, after the tutor provides feedback, the student takes the initiative, and the tutor responds with one piece of information the student didn't know (DDI). Of course, the role of Prompts and SI is not surprising, although interestingly they are significant only in association with certain tutor moves. It is well known that students learn more when they build knowledge by themselves, either by taking the initiative (SI), or after the tutor prompts them to do so (Chi et al., 1994; Chi et al., 2001).

LOW: it's backwards # it's got four elements, but they are backwards. [*DDI*]
234: so we have do it again. [*SI*]
LOW: so do it again. [*FB*]
LOW: do what again? [*Prompt*]

Figure 6: {DDI, FB, PT} is most effective in Stacks

4.3 Other models

We found other significant models, specifically, {DDI,DPI} for all three topics, and {-FB,SI} for Lists. However, their R^2 are very low, and much lower than any of the other models presented so far. Besides models that include *only one* DA *sequence* and pre-test score to predict learning gain, we also ran experiments to see if adding multiple DA sequences to pre-test score will lead to significant models – namely, we experimented with models which include two sequences as predictors, say, the two bigrams {-FB,SI} and {FB,DDI}. However, no significant models were found.

5 Conclusions

In this paper, we explored effective tutoring strategies expressed as sequence of DAs. We first presented the CS-Tutoring corpus. By relaxing the DA n-gram definition via the fuzzy matching provided by Apache Lucene, we managed to discover several DA sequences that significantly correlate with learning gain. Further, we discovered models with higher R^2 than models which include only one single DA, which are also more informative from the point of view of the design of interfaces to ITSs.

6 Acknowledgments

This work was mainly supported by ONR (N00014-00-1-0640), and by the UIC Graduate College (2008/2009 Dean's Scholar Award). Partial support is also provided by NSF (ALT-0536968, IIS-0905593).

References

John R. Anderson. 1986. Knowledge compilation: The general learning mechanism. *Machine learning: An artificial intelligence approach*, 2:289–310.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596. Survey Article.

Kristy Elizabeth Boyer, Robert Phillips, Amy Ingram, Eun Young Ha, Michael Wallis, Mladen Vouk, and James Lester. 2010. Characterizing the effectiveness of tutorial dialogue with Hidden Markov Models. In *Intelligent Tutoring Systems*, pages 55–64. Springer.

Whitney L. Cade, Jessica L. Copeland, Natalie K. Person, and Sidney K. D'Mello. 2008. Dialogue modes in expert tutoring. In *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 470–479. Springer Berlin / Heidelberg.

Lin Chen and Barbara Di Eugenio. 2010. A lucene and maximum entropy model based hedge detection system. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 114–119, Uppsala, Sweden, July. Association for Computational Linguistics.

Michelene T. H. Chi, Nicholas de Leeuw, Mei-Hung Chiu, and Christian LaVancher. 1994. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439–477.

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

Min Chi, Kurt VanLehn, and Diane Litman. 2010. The more the merrier? Examining three interaction hypotheses. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society (CogSci2010)*, Portland,OR.

Michelene T.H. Chi. 1997. Quantifying qualitative analyses of verbal data: A practical guide. *Journal of the Learning Sciences*, 6(3):271–315.

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

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

Barbara Di Eugenio, Trina C. Kershaw, Xin Lu, Andrew Corrigan-Halpern, and Stellan Ohlsson. 2006. Toward a computational model of expert tutoring: a first report. In *FLAIRS06, the 19th International Florida AI Research Symposium*, Melbourne Beach, FL.

Barbara Di Eugenio, Davide Fossati, Stellan Ohlsson, and David Cosejo. 2009. Towards explaining effective tutorial dialogues. In *Annual Meeting of the Cognitive Science Society*, pages 1430–1435, Amsterdam, July.

Martha W. Evens and Joel A. Michael. 2006. *One-on-one Tutoring by Humans and Machines*. Mahwah, NJ: Lawrence Erlbaum Associates.

Davide Fossati, Barbara Di Eugenio, Christopher Brown, Stellan Ohlsson, David Cosejo, and Lin Chen. 2009. Supporting Computer Science curriculum: Exploring and learning linked lists with iList. *IEEE Transactions on Learning Technologies, Special Issue on Real-World Applications of Intelligent Tutoring Systems*, 2(2):107–120, April-June.

Davide Fossati, Barbara Di Eugenio, Stellan Ohlsson, Christopher Brown, and Lin Chen. 2010. Generating proactive feedback to help students stay on track. In *ITS 2010, 10th International Conference on Intelligent Tutoring Systems*. Poster.

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

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

Arthur C. Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M. Louwerse. 2004. AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, 36:180–193.

Mark R. Lepper, Michael F. Drake, and Teresa O'Donnell-Johnson. 1997. Scaffolding techniques of expert human tutors. In K. Hogan and M. Pressley, editors, *Scaffolding student learning: Instructional approaches and issues*. Cambridge, MA: Brookline.

Diane Litman and Kate Forbes-Riley. 2006. Correlations between dialogue acts and learning in spoken tutoring dialogues. *Natural Language Engineering*, 12(02):161–176.

Brian MacWhinney. 2000. *The Childes Project: Tools for Analyzing Talk: Transcription format and programs*, volume 1. Psychology Press, 3 edition.

Johanna D. Moore, Kaska Porayska-Pomsta, Sebastian Varges, and Claus Zinn. 2004. Generating Tutorial Feedback with Affect. In *FLAIRS04, Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*.

Stellan Ohlsson, Barbara Di Eugenio, Bettina Chow, Davide Fossati, Xin Lu, and Trina C. Kershaw. 2007. Beyond the code-and-count analysis of tutoring dialogues. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, pages 349–356, Los Angeles, CA, July. IOS Press.

Stellan Ohlsson. 2008. Computational models of skill acquisition. *The Cambridge handbook of computational psychology*, pages 359–395.

Ron Sun, Paul Slusarz, and Chris Terry. 2005. The Interaction of the Explicit and the Implicit in Skill Learning: A Dual-Process Approach. *Psychological Review*, 112:159–192.

Kurt VanLehn, Arthur C. Graesser, G. Tanner Jackson, Pamela W. Jordan, Andrew Olney, and Carolyn P. Rosé. 2007. When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1):3–62.