Predicting Changes in Level of Abstraction in Tutor Responses to Students

Michael Lipschultz[1], **Diane Litman**[1,2], **Pamela Jordan**[2], and **Sandra Katz**[2]
[1]Department of Computer Science and [2]Learning Research Development Center
University of Pittsburgh
Pittsburgh, PA 15260

Abstract

We examine a corpus of reflective tutorial dialogues between human tutor and student after the student completed introductory physics problems, to predict when the tutor abstracted from the student's preceding turn or when the tutor specialized from the student's preceding turn. Tutor abstraction occurs when the tutor repeats a segment of the student's turn using more general terms. Tutor specialization occurs when the tutor repeats a segment of the student's turn using more concrete terms. We find that features extracted from the reflective dialogue context produce the most predictive models. Also, the tutor abstracts more often when the student shows signs of working at a very detailed level for awhile, and prompts for specification when the student's responses are imprecise.

1 Introduction

Socio-cognitive theories of learning explain the effectiveness of one-on-one human tutoring (Bloom 1984) in terms of social interaction, or learning from dialogue (Chi et al. 2001; Boyer et al. 2010). Although there is abundant empirical evidence that interaction between a student and tutor (or student and peer) supports learning, much less is known about the specific features of effective instructional dialogue. This level of specificity is needed to plan tutorial dialogues in Intelligent Tutoring Systems (ITS).

Over the past decade, researchers in cognitive science and ITS have made significant progress in identifying specific features of human tutorial dialogue that predict learning (Chi et al. 2001; Forbes-Riley and Litman 2007; Chi, Roy, and Hausmann 2008; Ward et al. 2009). One aspect of these dialogues is cohesive ties, which have been shown to be beneficial for learning (Ward et al. 2009). Cohesion is considered to be the connectedness of a text (Halliday and Hasan 1976) and cohesive ties are the various forms of connectedness, such as synonymy and paraphrase. In this paper, we focus on two lexical cohesive ties that occur between student-tutor turns and have been shown to be correlated with learning: tutor abstraction and tutor specialization (Ward et al. 2009). Tutor abstraction occurs when the tutor repeats part of a student's utterance, but at a higher level of generality. Tutor specialization occurs when the tutor repeats part of a student's utterance at a lower level of generality. Figure 1 has an example of both.

Student: vf=vo+at, and plug in the values for vf, vo and t
Tutor: so you are telling me you can use a kinematics eq *[class=abstract]*?

Student: no so he would need a greater accel.
Tutor: exactly so! He'd have an ave. accel. *[class=specialize]* of (16 m/s /62 s) instead of (15 m/s /62 s)

Figure 1: Two dialogue snippets with matching segments underlined. The top shows when the tutor abstracts; the bottom shows when the tutor specializes.

The ultimate goal of our project, the Rimac Project[1], is to build a fully-automatic interactive post-problem reflective dialogue system for physics which abstracts and specializes from the student's preceding turn when appropriate. The system will be used to engage in interactive reflective dialogues with high school students after the students have solved introductory physics problems. To achieve the interactivity desired, we will identify *when* a computer tutor should abstract or specialize based on when human tutors did so during reflective dialogues. This paper presents our initial step. For this step, we are interested in identifying useful features and feature relationships that predict tutor abstraction and tutor specialization. Our next step will be to build models that can be used in our dialogue system to direct when the tutor should abstract or specialize.

In this paper, we use an existing human-human reflective dialogue corpus that has been tagged for when the tutor abstracted and specialized relative to the student's prior turn. From it, we extract features that could be used in a fully-automatic dialogue system. Others developing interactive tutorial systems have found certain types of features beneficial for identifying when to change the level of interactivity; we used these features as guidance when we selected our features. Research into emotion detection in tutorial dialogue systems found that dialogue context information, such as the number of main questions answered or the num-

[1]Rimac is the name of a river whose source is in the Andes. Its name is a Quechua word meaning "talking;" hence the nickname for Rimac: "talking river." We thus considered Rimac to be well-suited to a dialogue system embedded in the Andes tutoring system.

ber of characters in the student's turn, correlate with various emotional states (D'Mello and Graesser 2006). Others have found demographic information to be important for determining hint interactivity (Arroyo et al. 2000). Finally, student performance and student dialogue information have been used in research determining when a tutor should elicit information from the student versus telling them (Chi 2009).

We explore how useful each feature is in predicting abstraction and specialization by training two decision trees (one for each) per feature. In addition to looking at the prediction results, we examine the trees to gain intuitions about why the tutor may have abstracted or specialized what the student had said. To further investigate feature relationships, we group related features and train decision trees on these feature sets. From these trees, we are able to identify possible rules for why the tutor abstracted or specialized and the emerging rules suggest plausible explanations for tutor abstraction and specialization. Our results suggest that these features may be useful for predicting abstraction and specialization.

2 Corpus

Our corpus is from a previous study (Katz, Allbritton, and Connelly 2003) on the effectiveness of reflection questions after a physics problem-solving session with the Andes physics tutoring system (VanLehn et al. 2005). Students taking introductory physics courses at the University of Pittsburgh were recruited. They took a physics pretest, with nine quantitative and 27 qualitative physics problems. All 36 problems were tagged by physics experts for knowledge components (KCs) that students must have in order to correctly answer the problem. For example, one KC necessary for solving the problem shown in Figure 2 is "Tension in a cord or rope produces a force pulling in toward the center of the cord or rope." Following the pretest, students reviewed a workbook chapter developed for the experiment and received training on using Andes.

Although there were three conditions in the experiment, his paper only focuses on the Human Feedback (HF) condition since we are interested in building more interactive dialogues, which only this condition provides; see (Katz, Allbritton, and Connelly 2003) for complete details. Students in each condition began by solving a problem in Andes. After completing the problem, students in the HF condition were presented with a deep-reasoning reflection question which they needed to answer. After typing their answer, they would begin a teletyped dialogue with a human tutor on the student's answer. This dialogue continued until the tutor was satisfied that the student understood the correct answer. Three to eight reflection questions were asked per problem solved in Andes. There were 12 problems in all. An example problem and a reflection question associated with it can be found in Figure 2.

After the last problem's reflection dialogues, students took a posttest that was isomorphic to the pretest and counterbalanced. The study found that students who answered reflection questions learned more than students who did not answer reflection questions. However, there was no significant difference between the HF condition and the condition with canned feedback to students' answers to the reflection questions.

Andes Problem:
A rock climber of mass 55 kg slips while scaling a vertical face. Fortunately, her caribiner holds and she is left hanging at the bottom of her safety line. Find the tension in the safety line.
Reflection Question:
What minimum acceleration must the climber have in order for the rope not to break while she is rappelling down the cliff? (You do not have to come up with a numerical answer. Just solve for "a" without any substitution of numbers.)

Figure 2: Sample problem and the 3rd of 4 reflection questions.

There were 16 students in the HF condition (4 male, 12 female). Fifteen students participated in all 60 reflection question dialogues, one only participated in 53, giving a total of 953 dialogues. There are a total of 2,218 student turns and 2,135 tutor turns in these dialogues. Each dialogue has an average of 2.32 student turns and 2.24 tutor turns.

This HF condition data was used in a later study examining which cohesive ties during reflective dialogues correlate with learning (Ward et al. 2009). The reflection dialogue corpus was tagged by human annotators for cohesive ties. For each turn, the annotators identified segments containing a cohesive tie to a segment in the previous speaker's turn, then tagged that segment with the cohesive tie identified. The kappa for this annotation was 0.57[2]. The study found that two of these ties, tutor specialization and tutor abstraction, positively correlated with student learning. An example of each cohesive tie can be found in Figure 1. We use this tagged corpus to build two models predicting the tutor's next turn based on features described in Section 3: one model to predict whether the tutor abstracted and one to predict whether the tutor specialized.

3 Features

We partition our feature set into three groups based on the source of the information to allow us to explore not only which features are useful, but which sources are useful: *student*, *problem*, and *context*. Similar features have been used in previous work on emotion detection in tutorial dialogue systems (D'Mello and Graesser 2006), determining hint interactivity (Arroyo et al. 2000), and research on determining when a tutor should elicit information from the student or give them information (Chi 2009). From this literature, we selected the features we could extract from the data collected during the study. Since we are interested in developing a fully-automatic system that will be tested in high schools, features that could not be easily detected automatically (such as whether the student was paraphrasing the tutor) or are not available from high school students (such as college major) were not considered in this study.

[2]Although 0.57 is considered a moderate agreement by one standard (Landis and Koch 1977), interpreting such kappas is controversial (Eugenio and Glass 2004). However, even poor kappas can still be suitable for machine learning tasks (Reidsma and Carletta 2008).

student : features about the student and their background knowledge

PreQualScore – score on qualitative part of pretest (high, low)[3]

PreQuantScore – score on quantitative part of pretest (high, low)

Sex – **M**ale or **F**emale

problem : features from the problem-solving session in Andes; the median splits in this feature set are problem-specific

NextStepHelp – how often student requested help from Andes on what step to do next (high, low)

WhatsWrongHelp – how often student asked Andes what was wrong with their work (high, low)

UnsolicitedHelp – how often Andes offered an unsolicited hint (high, low)

NumErr – number of incorrect student entries during problem-solving process (high, low)

NumCorr – number of correct student entries during problem-solving process (high, low)

NumEntries – total number of student entries in the interface (sum of NumErr and NumCorr) (high, low)

CorrAns – total number of correct answers entered (not intermediate entries) by student (high, low)

Time2SolveNorm – time (in seconds) student took to solve problem, divided by the average time to solve this problem by the students in the other conditions of the study (slow, fast)

AvgKCScore – for KCs required by student to solve the particular Andes problem (as determined by physics experts), what was student's average score on pretest problems also requiring those KCs (high, low)

context : features from the reflection dialogue

RQPosition – reflection question for problem (1, 2, . . .)

PrevRQLength – number of turns in the previous reflection question's dialogue

Time2AnsNorm – how long (in seconds) it took for the student to respond to the tutor's previous message, normalized by the number of characters in the student's response (high, low)

TurnPosition – position in the reflection question dialogue

StuWordCount – count of words are in the student's preceding turn (high, low)

DomainWord% – of all words in student's preceding turn, what percentage are physics domain words[4] (high, low)

4 Machine Learning

As mentioned above, we are interested in using this corpus to predict when a computer-based tutor should abstract in a post-problem reflective dialogue and when such a tutor should specialize. Thus, we will be building from this corpus two models, one to predict when the human tutors abstracted and the other to predict when the human tutors specialized. The tags from (Ward et al. 2009) were done on segments of a turn. In this work, we predict at the turn level, so if any segment of a turn was labeled as abstraction or specialization, then the turn was labeled as abstraction or specialization. Since predictions are at the turn level, the segment-level tags were propagated to the turn level. Both of these prediction tasks are binary classifications, with *yes* meaning that the tutor provided an abstraction or specialization from the student's turn preceding the turn we are attempting to predict and *no* meaning that the tutor did not.

[3]Median splits were performed on most numerical features for ease of tree interpretation. Table 1 shows some value ranges and is described in Section 4.1.

[4]From: http://scienceworld.wolfram.com/physics/letters/

Feature	Low	Median	High
NumErr	0	14	72
NumEntries	2	22	86
Time2SolveNorm	0.08	0.95	3.06
RQPosition	1	3	8
PrevRQLength	1	3	34
TurnPosition	1	3	34
StuWordCount	0	6	81
DomainWord%	0	2.83	13

Table 1: Low, median, and high values for all features appearing in any trees presented in this paper.

The original data has a large bias towards not abstracting (93% of all tutor turns) and towards not specializing (94%), so we balanced the dataset for these tasks as others have done (Ang et al. 2002). To balance the dataset, we downsampled using WEKA's Resample filter[5]. The balanced dataset for abstraction contained 156 turns where the tutor did not abstract (56%) and 123 where the tutor did (44%). The balanced dataset for specialization contained 141 turns where the tutor did not specialize (60%) and 94 where the tutor did (40%). On each of these balanced datasets, we performed 10x10-fold cross-validation using J48 decision trees (so we could examine the relationships between the features) from WEKA. Although at this stage we are not attempting to optimize the prediction models, we do want to determine whether these features show any promise. Therefore, we use a majority class baseline, which is *no* for both tasks.

4.1 One-Feature Trees

First, we examined each feature individually. We trained one decision tree for each of the features listed in Section 3. Table 2 presents the performance of the decision trees. Trees that never performed differently from baseline are not shown in the table. Rows from NumErr to DomainWord% present the results for the one-feature trees. These rows are divided into two of the groups presented in Section 3. The top row, Baseline, shows the majority class results.

From these results, we see that the *student* features do no better than baseline. This is perhaps because between these features and the turn we are attempting to predict, there has been some learning, either during the Andes problem-solving session or the reflective dialogues. Therefore, the pretest scores may not be representative of the students' knowledge throughout most of the study.

Most of the *problem* features also do no better than baseline for predicting abstraction and none of the features do better than baseline for predicting specialization. This is perhaps also because *student* and *problem* features describe a past student state that is no longer relevant. However, three

[5]http://www.cs.waikato.ac.nz/ml/weka/

		Abstraction				Specialization			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Baseline		55.91	0.31	0.56	0.40	60.02	0.36	0.60	0.45
Single Features	*problem*								
	NumErr	59.35	**0.59**	0.59	**0.59**	60.02	0.36	0.60	0.45
	NumEntries	58.80	**0.59**	0.59	**0.58**	60.02	0.36	0.60	0.45
	Time2SolveNorm	60.57	**0.62**	0.61	**0.60**	60.02	0.36	0.60	0.45
	context								
	RQPosition	55.76	**0.51**	0.56	**0.51**	60.02	0.36	0.60	0.45
	StuWordCount	54.52	**0.48**	0.55	**0.50**	**68.53**	**0.72**	**0.69**	**0.69**
	DomainWord%	60.59	**0.64**	0.61	**0.60**	60.02	0.36	0.60	0.45
Feature Groups									
	problem	61.27	**0.61**	0.61	**0.61**	55.85	**0.44**	0.56	0.47
	context	**64.62**	**0.65**	**0.65**	**0.64**	**67.90**	**0.69**	**0.68**	**0.68**
Aggregated Feature Groups									
	StudentProblem	56.91	**0.57**	0.57	**0.57**	57.10	**0.46**	0.57	0.47
	StudentContext	61.53	**0.62**	0.62	**0.61**	**68.41**	**0.69**	**0.68**	**0.68**
	ProblemContext	61.51	**0.62**	0.62	**0.61**	64.19	**0.64**	0.64	**0.63**
	all	61.94	**0.62**	0.62	**0.61**	**67.93**	**0.68**	**0.68**	**0.67**

Table 2: Comparing feature sets across the weighted average metrics[6] for both Abstract and Specialize classification tasks. Bold values indicate results significantly better than baseline ($\alpha < 0.05$). All other values are not significantly different from the baseline[7]. The underlined values are the greatest in that column.

features do better than baseline for predicting abstraction on both precision and F1 (see Table 1 for the range and median for all features in the trees). The tutor tended to abstract when responding to the student if NumErr was high, NumEntries was high, or if Time2SolveNorm was slow. One possible interpretation is that the student had been focusing on the details long enough and needed to be encouraged to think more abstractly. Alternatively, abstraction may indicate that the tutor is focusing on basic concepts and principles – such as the formal statement of Newton's Second Law – when the student does not understand.

Of the six *context* features, three outperform the baseline for predicting abstraction and one outperforms the baseline for predicting specialization. The decision tree using RQPosition for predicting abstraction can be seen in Figure 3. From this tree, we see that for reflection questions 4, 5, and 8, the tutor abstracted. Although there were at most eight questions, a majority of the problems had four or five reflection questions. Later reflection questions build off of the earlier questions, often asking the student to work with equations and variables instead of giving values (see Figure 2). Examination of the dialogues suggests that students tended to instantiate variables too soon, so tutors would need to encourage the use of equations and variables, which was often done through abstracting the values into equations and variables. The decision tree using DomainWord% shows that the tutor abstracted when the DomainWord% was high. This may be because the student's turn was at a more specific level than the tutor preferred, so the tutor generalized the turn to show that the concept applies to more than just the specific instance the student was talking about.

Finally, StuWordCount is an important feature for both tasks. When the word count of the student's preceding turn is high, the tutor may either abstract or specialize. With high word counts, there may be more for the tutor to abstract or specialize over. Since the tutor rarely both abstracts and specializes in the same turn in this corpus, further analysis is needed to determine what influences the tutor to abstract or specialize when StuWordCount is high (see below).

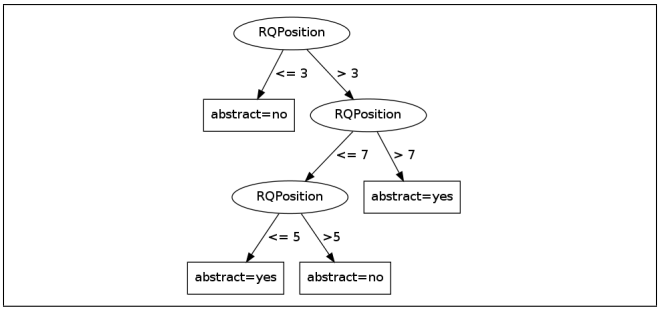

Figure 3: Decision tree to predict Abstract using only the RQPosition feature. Accuracy = 55.76%

4.2 Feature Group Trees

We next looked at how each feature group predicts tutor abstraction and specialization. The numeric results are in Table 2, in the rows for *problem* and *context*. The *student* feature group does no better than baseline. As with the one-feature trees, this is probably because these features are far-removed from the tutor's turn we are trying to predict.

The *problem* feature group does better than baseline on precision for both prediction tasks, but does not do signif-

[6]Weighted precision is calculated as $\frac{P_{no}*|no|+P_{yes}*|yes|}{|yes+no|}$; weighted recall and F1 are calculated similarly.

[7]Note that since the baseline model does not predict any turns as *yes*: (1) the baseline accuracy can only improve for *yes*; (2) the baseline recall is 0 for *yes* and 1 for *no* so there is only room for improvement for *yes*; (3) the baseline precision for *yes* and *no* both have room for improvement (0 and .559 respectively), which will lead to an increase in the F1 metric.

icantly better than Time2SolveNorm. For specialization, it does significantly better in precision, with no significant drop in the other three metrics.

The *context* feature group does significantly better than baseline on all four metrics for both prediction tasks. Figure 5 shows the decision trees for predicting abstraction (top) and specialization (bottom). Comparing the two trees, we see that if PrevRQLength was high, then the tutor abstracted more. Longer reflection dialogues may suggest the student did not, at first, understand the concepts, so the tutor may not want to overwhelm the student with details in the new dialogue. Alternatively, tutor abstraction may reflect the tutor's attempt to focus on basic concepts and principles whereas the student prefers to focus on the details (such as through variable instantiations). As the dialogue progresses (TurnPosition > 1), the tutor will specialize more than abstract, perhaps because the student's answers are less precise than the tutor would like (e.g. units missing, working with a vector rather than components). So, the tutor must guide the student to be more precise. The tutor does this through specializing the student's turns. Figure 4 shows an example, with the tutor guiding the student in specializing, which the student does in Student$_{11}$.

Student$_1$: 500 N (rope tension) / 55 (cliber mass) = acceleration
Tutor$_2$: hmm not quite. you have the right basic idea (use F = ma *[class=abstract]*) but the F in that equation is the net force. The force up would be 500 N, but would that be the net force?
. . .
Student$_9$: 39 N / 55 kg = a
Tutor$_{10}$: excellent. (and you should specify the direction, then we'll go on)
Student$_{11}$: accelerating downwards *[class=specialize]*

Figure 4: Student answering the reflection question in Figure 2. Tutor starts off abstracting, then later encourages student to specialize.

Returning to the question of when the tutor abstracts or specializes, if the student's word count is high (Section 4.1), these two trees provide some insights. The tutor will specialize starting after the student's second turn or on the first turn if the previous reflective dialogue was short. A short reflective dialogue can indicate that the student understands the concepts and is ready for a detailed dialogue. Beginning to specialize after the student's second turn overlaps with the tutor abstracting after the student's third turn (for few domain words only). Clearly, further investigation into when the tutor abstracts or specializes is needed. Finally, we see that when the StuWordCount is low, the tutor will neither abstract nor specialize, perhaps because there is little to abstract or specialize from.

4.3 Aggregating Feature Groups

We then looked at the interaction of features from different feature groups and their influence on the prediction tasks. Table 2's bottom four rows show the numeric results. The first three represent the results from the pairwise merging of the feature groups. The fourth row presents the results from merging all three feature groups together. We see that

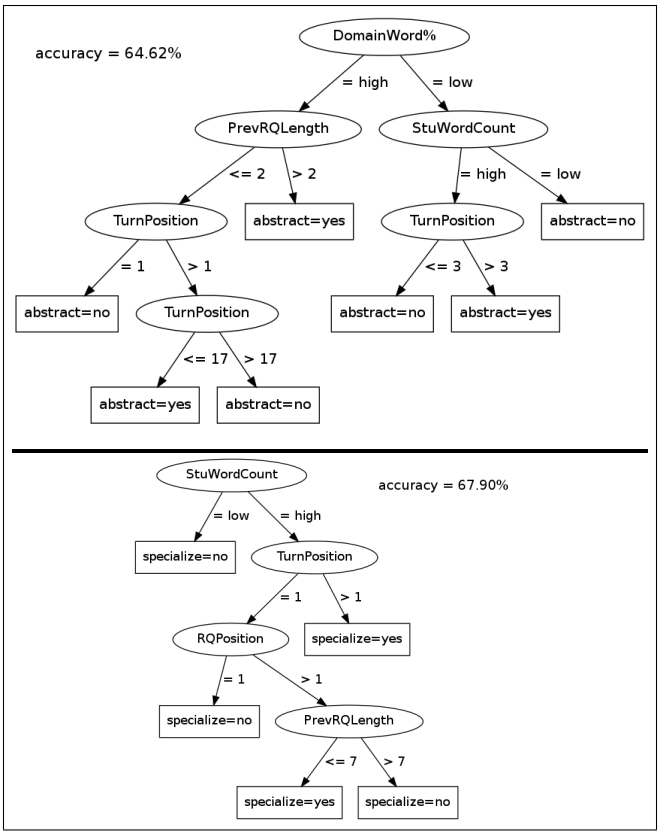

Figure 5: Decision trees using *context* features to predict abstraction (top) and specialization (bottom).

there are many aggregated feature groups that outperform the baseline, but none are significantly better than *context*. With a growing number of features (from 9 for StudentContext to 18 for all) and with datasets of size 279 (Abstraction) and 235 (Specialization), we believe we are starting to overfit the data.

5 Discussion & Future Work

This work builds off of previous work which showed that tutor abstraction and specialization positively predicted learning in post-problem reflective dialogues (Ward et al. 2009). The goal of the current analysis was to determine whether features and feature groups that can be automatically extracted from Andes tutoring logs and dialogue logs may be useful for predicting when a tutor will abstract or specialize. *Context* features appear to be the most useful. We were also interested in discovering relationships between these features and abstraction and specialization. We found evidence in the decision trees that tutors will abstract more often when students have been working at a very detailed level for awhile. Tutors appear to specialize more often when the student is more likely to understand the concepts (e.g. shorter dialogues). However, there is an ambiguity in the trees learned; for example, a tutor may either abstract or specialize when the student is verbose. We are currently investigating the contexts in which abstraction and specialization occur in order to resolve ambiguities such as this and

develop more robust decision rules.

Although we are able to outperform the baseline, there is still room for improvement. First, we are currently working on retagging the corpus for non-lexical forms of abstraction and specialization. Once this retagging is complete, we will have more instances of tutor abstraction and specialization, giving us larger balanced datasets on which to train our models. However, others have pointed out (Chi et al. 2001) that human tutors are not always consistent in their tutoring strategies and may miss opportunities to apply a particular strategy. Therefore, it may be difficult to build good models based on human-tutoring corpora alone.

We also plan on exploring other machine learning approaches. In addition to trying other machine learning algorithms (which may provide better results), we would also like to try creating a two-stage model. The first stage is intended to balance the dataset by identifying whether there is an opportunity for tutor abstraction or specialization. Turns classified as not having an opportunity will be labeled as *no* for both prediction tasks. To perform this first stage, since we have not tagged the corpus for opportunities, we plan to hand-code rules, drawing from some results of this paper (e.g., high student word count indicated both tutor abstraction and specialization) as well as relevant literature and intuition. Those turns classified as having an opportunity will be passed to the second stage, which will identify whether the tutor should abstract or specialize. The work presented in this paper focused on the second stage.

In this work, we only focused on automatic features. Certain features, such as student abstraction, that are predictive of student learning in this corpus (Ward et al. 2009), were not included in this study because they are not features that could be easily automated. Preliminary results suggest that cohesion-based context features may improve the prediction results by around five percentage points. If the addition of these features significantly improves the results, we could attempt to predict them using fully-automatic features. We will also attempt to identify additional automated features. We are currently tagging the student turns for correctness. We predict that the correctness of the student's turn also influences whether the tutor abstracted or specialized.

While the context features were the most useful features in this dataset, it is not clear whether that would still be true in other tutoring situations. We are interested in seeing whether these results transfer to other tutoring systems and domains, as well as similar prediction tasks. As a first step, we will be exploring whether we obtain similar results when predicting student abstraction and specialization. As mentioned in Section 4.2, the tutor will guide the student to be more specific or more abstract. In addition to exploring generalizability, we are interested in exploring what tutor moves predict student abstraction or specialization.

Acknowledgements

The authors thank the Rimac group, Joanna Drummond, and Wenting Xiong for their input. The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A100163 to the University of Pittsburgh. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

References

Ang, J.; Dhillon, R.; Krupski, A.; Shriberg, E.; and Stolcke, A. 2002. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In 7^{th} *Int'l Conf. on Spoken Language Processing*.

Arroyo, I.; Beck, J.; Woolf, B.; Beal, C.; and Schultz, K. 2000. Macroadapting animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism. In Gauthier, G.; Frasson, C.; and VanLehn, K., eds., *Intelligent Tutoring Systems*, volume 1839 of *LNCS*. 574–583.

Bloom, B. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13(6):4–16.

Boyer, K.; Phillips, R.; Ingram, A.; Ha, E.; Wallis, M.; Vouk, M.; and Lester, J. 2010. Characterizing the effectiveness of tutorial dialogue with hidden Markov models. In *ITS*, 55–64.

Chi, M.; Siler, S.; Jeong, H.; Yamauchi, T.; and Hausmann, R. 2001. Learning from human tutoring. *CogSci*.

Chi, M.; Roy, M.; and Hausmann, R. 2008. Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *CogSci*.

Chi, M. 2009. *Do Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning to Induce Pedagogical Tutorial Tactics*. PhD dissertation, University of Pittsburgh, Intelligent Systems Program.

D'Mello, S., and Graesser, A. 2006. Affect detection from human-computer dialogue with an intelligent tutoring system. In *Intelligent Virtual Agents*, 54–67. Springer.

Eugenio, B. D., and Glass, M. 2004. The kappa statistic: A second look. *Comput. Linguist.* 30(1):95–101.

Forbes-Riley, K., and Litman, D. 2007. Investigating human tutor responses to student uncertainty for adaptive system development. *Affective Computing and Intelligent Interaction* 678–689.

Halliday, M., and Hasan, R. 1976. *Cohesion in English*. Longman London.

Katz, S.; Allbritton, D.; and Connelly, J. 2003. Going beyond the problem given: How human tutors use post-solution discussions to support transfer. *IJAIED* 79–116.

Landis, J. R., and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics*.

Reidsma, D., and Carletta, J. 2008. Reliability measurement without limits. *Comput. Linguist.* 34:319–326.

VanLehn, K.; Lynch, C.; Schulze, K.; Shapiro, J.; Shelby, R.; Taylor, L.; Treacy, D.; Weinstein, A.; and Wintersgill, M. 2005. The Andes physics tutoring system: Lessons learned. *IJAIED* 147–204.

Ward, A.; Connelly, J.; Katz, S.; Litman, D.; and Wilson, C. 2009. Cohesion, Semantics and Learning in Reflective Dialog. In *Proc. AIED Workshop*.