Diagnosing Natural Language Answers to Support Adaptive Tutoring

Myroslava O. Dzikovska[*], **Gwendolyn E. Campbell**[†], **Charles B. Callaway**[*],
Natalie B. Steinhauser[†], **Elaine Farrow**[*], **Johanna D. Moore**[*],
Leslie A. Butler[‡] and **Colin Matheson**[*]
[*]HCRC, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, United Kingdom
e-mail:m.dzikovska@ed.ac.uk
[†]Naval Air Warfare Center Training Systems Division, Orlando, Florida
[‡]Florida Atlantic University, Boca Raton, FL

Abstract

Understanding answers to open-ended explanation questions is important in intelligent tutoring systems. Existing systems use natural language techniques in essay analysis, but revert to scripted interaction with short-answer questions during remediation, making adapting dialogue to individual students difficult. We describe a corpus study that shows that there is a relationship between the types of faulty answers and the remediation strategies that tutors use; that human tutors respond differently to different kinds of correct answers; and that re-stating correct answers is associated with improved learning. We describe a design for a diagnoser based on this study that supports remediation in open-ended questions and provides an analysis of natural language answers that enables adaptive generation of tutorial feedback for both correct and faulty answers.

1 Introduction

Research in intelligent tutoring systems suggests that getting students to explain their reasoning is important for learning and should be encouraged (Chi *et al.* 1994), and that contentful student talk is correlated with learning gain (Litman & Forbes-Riley 2006). Thus, some tutorial dialogue systems encourage students to produce language by asking "why" questions (Aleven, Popescu, & Koedinger 2001; Jordan *et al.* 2006; Graesser *et al.* 1999). They use natural language understanding techniques to analyze student answers and choose feedback based on the faults (errors and missing parts) discovered. However, for remediation, current systems revert to scripted dialogue techniques, where instructors pre-author system responses to the range of student inputs they anticipate.

While systems using scripted dialogue can be built and deployed quickly, they are limited in their ability to adapt to the needs of individual learners. Scripted remediations usually ask only short-answer questions and restrict student input. If unrestricted input is allowed, typically only generic remediation can be offered without referencing the content of the student input, which is not how human tutors normally behave (Jordan 2004). Moreover, while scripted dialogue allows different types of feedback for different types of mistakes anticipated by the authors, there is no guarantee that the same tutorial strategies will be applied consistently to the same types of mistakes in different authored scripts. This causes additional difficulty in transferring the results of research on tutorial dialogue into a practical system, because a large number of scripts may have to be altered if a tutoring strategy needs to be changed.

We believe that there can be significant training benefits from a system designed to overcome these limitations. We report a corpus study showing that human tutors use different types of tutoring strategies depending on the combinations of correct, incorrect and missing parts in student answers. Tutors also pay attention to correct parts of student input, restating and summarizing correct student answers, and explicitly acknowledging correct parts of faulty answers. Moreover, our data show that re-stating correct answers is associated with higher learning gains. This is a significant new finding, as current research in tutoring (Litman & Forbes-Riley 2006; Kim *et al.* 2006; Rose *et al.* 2003) focuses only on fixing faults in student answers, and existing tutorial dialogue systems provide only contentless positive feedback if a student answers correctly.

Based on the results of this study, we describe an implemented diagnoser component built as part of developing the BEETLE2 tutorial dialogue system (Callaway *et al.* 2007). The diagnoser produces diagnosis structures containing detailed analyses of student answers to explanation questions, and identifies explicitly correct, incorrect, missing and irrelevant parts of the answer. We show how these diagnosis structures are used to automatically generate adaptive tutorial feedback, including restates of student answers.

We then propose a model for handling correct student answers in a way that is more similar to human tutor behavior. Based on our corpus analysis and discussions with experienced tutors, we propose to introduce differing categories of good answers that correspond to different tutoring strategies. This allows for greater flexibility of analysis in answers expressed in natural language, and encodes intuitions of experienced tutors about the range of acceptable answers to explanation questions, and the appropriate system responses.

Our system improves on existing language-based tutorial dialogue systems by providing a framework that allows for consistently applying tutorial strategies throughout the dialogue. In the future, this will allow us to systematically compare the effectiveness of different tutorial dialogue strategies by comparing learning gain and user satisfaction between versions of the system using different strategies.

The rest of the paper is organized as follows. Section 2 describes our corpus of human-human tutorial dialogue, the corpus analysis used to identify levels of accuracy in student answers and the patterns of response strategies used by human tutors across those levels. Section 3 presents the design for a diagnoser that takes natural language answers as input and produces analyses suitable for a tutorial component that generates adaptive feedback. Finally, Section 4 discusses how our approach relates to previous work on diagnosing natural language explanations, and our ongoing and planned efforts to evaluate the usefulness of adaptivity in dialogue.

2 Corpus

2.1 Data Collection

Two research team members, an experienced electrician and a psychologist, jointly developed a curriculum covering basic topics in D.C. circuits, including: the components of a circuit; how to build a functioning circuit; and how to find faults in a non-functioning circuit using a voltmeter. The instructional design philosophy was to interleave short presentations of information with interactive exercises, activities, and discussion. The student learning environment consisted of a window displaying lesson slides, a simulation-based workbench for building circuits, and dialogue history and chat windows. The student and tutor were physically separated and communicated over a network through the chat interface. The tutor had an extra monitor that mirrored the student's workstation. The corpus collection is reported in more detail in (Steinhauser, Butler, & Campbell 2007).

We present the results from thirty participants distributed across three experienced tutors. Exactly half of the participants were male. The mean age was 22.4 years ($\sigma = 5.0$). These participants scored an average of 41% on their pretests ($\sigma = 11\%$), and an average of 83% on their posttests ($\sigma = 10\%$), demonstrating a statistically significant improvement in performance, $t(29) = 20.04$, $p < .01$. Our corpus consists of 8,085 turns and 56,133 tokens.

2.2 Corpus Analysis

In this section we describe the analyses that we have conducted on our corpus to search for support for our hypotheses about human tutor behavior. More specifically, we investigate the following questions:

1. Do human tutors explicitly address content that has already been correctly provided by their students?
2. If so, can we identify any reliable patterns in when they do this, to provide guidance for system development?
3. Does the data suggest that this type of instructional strategy is actually effective?

Additionally, when student answers are incorrect, we look for reliable patterns between the nature of the student error and the instructional strategy selected by the tutor, again to provide guidance for system development.

Strategy	Subtypes
REINFORCE	Accept and Move On ("acc_moveon")
	Accept and Restate ("acc_restate")
REMEDIATE	Let it go("let_go")
	Try to get them to fix it ("student_fix")
	Fix it for them ("tutor_fix")
feedback	Positive, Negative, Hedged;
	Acknowledge good bits ("ack_good")
	Acknowledge bad bits ("ack_bad")

Figure 1: Overview of tutorial strategy annotation

First, we briefly describe our coding scheme. Each student utterance was assigned an accuracy value ("correct", "incorrect", "partially correct with some errors" or "partially correct but incomplete") by the tutor in real time during the dialogue. We applied a hierarchically organized coding scheme to the tutor utterances (summarized in Figure 1). At the highest level, we distinguished between strategies that typically followed correct answers and that appeared to signal a basic acceptance on the tutor's part of the student answer (labeled "reinforce") from strategies that typically followed the flawed answers (labeled "remediate"). The "reinforce" responses were divided into two sub-categories. Sometimes the tutor simply accepted the answer (usually with positive feedback) and allowed the student to move on, "accept & move on", while other times the tutor's response included substantive content, often some type of restatement of the content in the student's answer, "accept & restate".

"Remediate" responses were further broken down into three sub-categories, which can be conceptualized as "let it go", "fix it for them" and "try to get them to fix it." An additional annotation code describing explicit tutorial feedback was added where relevant. Three of the code values reflected feedback that conveyed an accuracy judgment, but did not contain any lesson content: "positive", "negative" and "hedged". The other two values were used when the utterance contained content, in addition to the accuracy judgment. These values were labeled "acknowledge good bits" and "acknowledge bad bits". A hypothetical example of a "try to get them to fix it" remediation that would also receive a code of "acknowledge good bits" is "Well, you're right that the battery and the light bulb must be connected, but would ANY connection at all be good enough?".

To assess the reliability of the annotation, we evaluated the pairwise percent agreement for segmentation, and Cohen's kappa for the segments with agreed boundaries, following (Carletta *et al.* 1997). For the segmentation of tutor utterances ("remediate" vs. "reinforce") two independent coders achieved a pair-wise percent agreement score of 91%. There was substantial agreement on whether a segment was a "reinforce" or "remediate" ($K = 0.88$), and near perfect agreement for differentiating between the three subcategories of remediate ($K = 0.98$) and two sub-categories of reinforce ($K = 0.92$).

Now we turn to our hypothesis testing. First, consider reinforcement strategies. The "acc_restate" strategy was used by our tutors 26% of the time in response to student an-

Did tutor use "ack_good"?	Incorrect	PartCorrect some errors	Incomplete
Yes	5% (6)	16% (10)	15% (16)
No	95% (142)	84% (53)	85% (94)

Table 1: Tutor feedback strategy by student error type.

Tutor Strategy	Incorrect	PartCorrect some errors	Incomplete
let_go	4% (7)	1% (1)	28% (69)
tutor_fix	27% (43)	47% (33)	18% (46)
student_fix	69% (106)	52% (37)	54% (134)

Table 2: Tutor remediation strategy by student error type.

swers that were considered correct. We did not ask our tutors to provide any finer level of classification of correct answers in real time, so we are not able to identify a pattern in our tutors' selection between the "acc_moveon" and "acc_restate" based on the nature of the students' answers. However, we can ask whether or not restating is an effective strategy. We conducted a hierarchical multiple linear regression, using post-test scores as our dependent variable and pre-test scores, amount of reinforcement received overall, and the frequency with which the "acc_restate" was used as our independent variables. The first model included only the amount of reinforcement received as the predictor which accounted for 44% of the variance and was significant, $F(1, 29) = 22.14$, $p < .01$. The second model incorporated the pre-test score and accounted for 54% of the variance, $F(2, 29) = 15.84$, $p < .01$. Finally, the third model included all three predictors and accounted for 61% of the variance, $F(3, 29) = 13.64, p < .01$. More specifically, reinforcement ($\beta = .84$, $p < .01$), pre-test ($\beta = .36$, $p = .007$), and restatements ($\beta = .35$, $p = .038$) all demonstrated significant effects on post-test score. In other words, after controlling for incoming knowledge of the topic and the extent to which a student was reinforced for being correct during the lesson itself, the tutor's increased use of the the "acc_restate" strategy was associated with higher student post-test scores.

We next looked at remediation strategies, which generally focused on the content that students misrepresented. Overall 10% of the tutor strategies coded as "remediate" also received a code of "ack_good". Table 1 breaks down the presence of this code across tutor responses to student utterances that vary in their level of inaccuracy. As Table 1 shows, tutors were approximately equally likely to use this code if the student answer contained any accurate information, and unlikely to use it if the student answer was completely incorrect. A Chi-Square Test of Independence was significant, $\chi^2(2) = 10.78$, $p < 0.01$, allowing us to conclude that tutors' use of the "acknowledge good bits" strategy does depend systematically on the nature of the student error.

In order to determine whether or not "ack_good" is an effective strategy, we again conducted a hierarchical multiple linear regression, using post-test scores as our dependent variable and pre-test scores, amount of remediation received overall, and amount of remediation with acknowledge good bits received as our independent variables. The first model included only the amount of remediation received as the predictor which accounted for 44% of the variance and was significant, $F(1, 29) = 21.81$, $p < .01$. The second model incorporated the pre-test score and accounted for 54%, $F(2, 29) = 15.71$, $p < .01$. Remediation ($\beta = -.63$, $p < .01$) and pre-test ($\beta = .32$, $p = .023$) demonstrated significant effects on post-test score. However, the frequency with which students received remediation that included an "ack_good" component did not account for a statistically significant amount of unique variance in post-test scores.

Finally, we examined our data to see if the tutor remediation strategy selection was related to the type of error in the student answer (Table 2). Here, a Chi-Square Test of Independence was significant, $\chi^2(4) = 64.29$, $p < 0.01$, allowing us to conclude that tutors' selection of a strategy is related to the nature of the student error. More specifically, it appears as if our tutors reacted more strongly as the nature of the student error became more severe. For example, our tutors were willing to ignore answers that were correct but incomplete over 25% of the time. But they almost never ignored a student answer that contained an actual error.

To summarize, our tutors did indeed allocate a certain amount of their dialogue to explicitly addressing lesson content that students had already presented accurately. When student answers were correct, tutors restated that information over 25% of the time. Even when student answers contained errors, if the answers also contained any correct information at all, tutors still addressed that content about 15% of the time. And while the "ack_good" strategy did not appear to have a significant effect on learning outcomes in our data, the "acc_restate" reinforcement strategy was a significant positive predictor of learning outcomes. Taken in combination, this suggests that the effectiveness of future tutorial dialogue systems may be improved by adding the capability to explicitly address and discuss lesson content that has already been correctly presented by the student.

In addition, our study demonstrates that tutor's choice of strategy for faulty answers depends on the different combinations of correct, missing and errorful parts of the student answer. This suggests a diagnoser design where such parts are identified explicitly, and then used in choosing the appropriate strategy. We discuss our design in the next section.

3 Diagnosing Student Answers

Based on the results of the data analysis, we built a diagnoser with the twin goals of being able to implement a variety of tutorial strategies (in particular strategies that restate the correct answers, as well as acknowledge correct parts of faulty answers), and to be able to apply those strategies systematically depending on the student answer category. An example remediation dialogue from the corpus is shown in Figure 2. Here the tutor acknowledges the good part of the answer and elicits the missing piece of information by hinting at the precise nature of the error.

We represent correct answers as lists of objects and rela-

[Question: What are the conditions that are required to make a light bulb light up?]

S: A closed circuit with both ends of the battery are necessary.
T: If both battery terminals are included, is that good enough? Because both battery terminals are involved in #5 but that didn't light.
S: Both terminals of the lightbulb also have to be included.
T: Yup! We usually say something like this: There must be a closed path (complete circuit) containing both a lightbulb and a battery

Figure 2: Example human-human dialogue from our corpus where an incomplete answer is improved

T1: What are the conditions that are required to make a light bulb light up?
S1: A battery must be in a closed path.
T2: Yes, a battery must be contained in a closed path. Anything else? Consider circuit 5.
S2: A bulb must be in a closed path as well.
T3: Perfect. A bulb and a battery must be contained in a closed path.

Figure 3: Example human-computer dialogue with the implemented system.

tions between them that a student is required to mention. We then use a deep parser to extract a list of objects and relations from a student answer, and attempt to match it with a known good answer. If a student answer matches a correct answer perfectly, then a corresponding reinforcement strategy can be applied (as discussed in Section 3.2). Otherwise, the diagnoser tries to classify each object and relationship that the student mentioned as one of the following:

- Correct parts – objects and relationships present in the standard answer;
- Direct errors - objects and relationships directly contradicting the standard answer (e.g., the student says the terminals are separated when they are connected);
- Irrelevant parts - objects and relationships not present in the standard answer, perhaps because an irrelevant (and therefore incorrect) reason was given in the explanation, or because the student was too verbose;
- Missing parts - parts in the standard answer but not present in the student's answer.

For example, consider a dialogue generated by our system, shown in Figure 3 (this is a slightly simplified version of the corpus dialogue from Figure 2). The deep parser and interpreter extract a set of relationships shown in Figure 4(a). The diagnoser implements a soft unification algorithm that attempts to match (including variable bindings) the student input with the expected answer (shown in Figure 4(b)), and, based on the results, produces the diagnosis structure shown in Figure 4(c). Each representation contains an answer code (discussed in Section 3.2), a match code that corresponds to the error category from the student answer analysis, and the detailed analysis of the student answer described above.

(a) (Battery _Batt1) (Path _Path1)
(is-closed _Path1 T) (contains _Path1 _Batt1)

(b) ((**Answer-type** Best)
((LightBulb ?bulb) (Battery ?batt) (Path ?p) (is-closed ?p T)
(contains ?p ?bulb) (contains ?p ?batt)

(c) ((**Answer-type** Best) (**Code** Partial)
(**Matched** ((Battery _Batt1) (Path _Path1)
(is-closed _Path1 T) (contains _Path1 _Batt1)))
(**Missing** ((LightBulb ?bulb) (contains _Path1 ?bulb)))
(**Contradictory** ()) (**Extra** ())

Figure 4: Representations used in interpreting student input: (a) the set of objects and relationships extracted by the interpreter from **S1**; (b) the encoding of the expected best answer, "A bulb must be in a closed path with a battery"; (c) the diagnosis produced by the system, later used for feedback.

In our example, the analysis contains both the matching parts (the mention of a battery contained in a closed path), and the missing part (the mention of the bulb in the same path). No parts were marked as contradicting the answer directly, or being irrelevant. This structure can then be used directly by the tutorial planner to generate the remediation in **T2**, confirming the correct part, adding a prompt for the missing information, and optionally generating a hint by pointing out a relevant circuit for which the student answer holds but the bulb is not lit. The process of generating tutorial feedback is discussed in more detail in the next section.

3.1 Generating Tutorial Feedback

The detailed answer diagnosis in Figure 4(c) allows the tutoring strategy to be adjusted depending on the results of the diagnosis, as we have observed human tutors to do. The tutorial planner first attempts to directly address any errors present in the student's answer (in all other cases, the answer will contain a mixture of correct, missing and extra information). Second, if the answer contains some but not all of the correct elements (*i.e.*, "matched" and "missing" are both non-empty) the system acknowledges the correct content and prompts the student to supply the missing content. Third, if the analysis indicates the student's answer contains neither correct parts nor direct errors, the tutorial planner will assume the student gave an irrelevant response, mention this to the student, and prompt for the missing information. Fourth, if the answer is fully correct and the irrelevant relationships are consistent with the state of the world (*i.e.*, too verbose), the tutor may apply a "reiterate key point" reinforcement strategy. Finally, if the answer was fully correct, the tutorial planner acknowledges that fact, potentially including encouragement depending on our student model.

We use a tutorial planner together with a deep generation system to implement this strategy. The tutorial planner defines classes of dialogue acts like "accept" and "generic prompt" which can each have many potential lexicalizations, as well as more contentful dialogue acts like "specific prompt" based on the underlying propositions in the diagnosis. The deep generator is then used to map propositions to syntactic structures and lexical items. Thus, different responses can be instantiated based on different combinations of dialogue acts:

Wrong → reject() + deny(**wrong**) + [try-again()]
"No, the battery should also be in a closed path."

Incomplete → accept() + restate(**correct**) + [prompt()]
"Exactly, the bulb must be in a closed path. What else?"

Incomplete → ack() + restate(**correct**) + [prompt(**missing**)]
"Yes, the bulb must be in a closed path. What about the battery?"

Correct → ack() + reiterate(**correct**)
"Right, they must both be in the same closed path."

Thus a tutor's reply may be a combination of "Yes" from an acknowledgment dialogue act, "You need a battery in a closed path" from the content in the **matched** part of the diagnosis, "What about a bulb?" from the tutorial planner's determination of a specific hint via the propositions in **missing**, as well as phrases like "but...also" derived from the fact that the answer has both correct and missing propositions.

3.2 Reinforcing Correct Answers

Since our data analysis showed that re-stating answers judged as correct by human tutors is correlated with improved learning gains, we took a deeper look at those in our corpus, in order to decide on an appropriate strategy for when and how to restate the good answers. First, we noted that the restatements can be classified into 3 broad categories: accepting a student answer as correct but modeling a better answer (for example, using a better terminology), re-iterating the key point (emphasizing the most important point in the explanation), and summarizing the answer (for example if, as in the previous section, the answer was elicited over multiple turns). We also found that there was significant variation in the answers that were rated as good by experienced tutors. To investigate this issue we asked one of the expert tutors involved in the experiment to supply a list of possible good answers to each question in the curriculum.

Based on this list and a discussion with the tutor we arrived at a more detailed classification of student answers into minimal, good, and best. Generally speaking, an answer is classified as best if it makes correct use of technical terminology, or is based on "deep" features of the context, or if it references unobservable, causal relationships between domain constructs. The answer is classified as good when it uses technical terminology correctly and describes actions and observable outcomes of manipulations, but is based on surface features of the context. The answer is classified as minimal when it alludes to the key point but does not use technical terminology or does not use it correctly, or requires some inferencing to be considered correct.

Figure 5 contains examples of different levels of correct answers listed by our tutor. The minimal answer is correct as it gives a functional description of how voltage readings behaved in previous exercises in the curriculum. The good answer provides a generalization on how the voltage can be used in fault-finding, the topic that the student is learning. The best answer (rarely seen with real students) gives the complete definition of voltage from the lesson. Based on discussions with the tutor, we decided that different reinforcement techniques are appropriate for each case. Minimal answers should be accepted as correct, but rephrased to model a better answer; for example, "That's right - voltage is a measurement of the difference in electrical states between two terminals". Good answers should be accepted, but with re-iterating the key point, for example "That's right - the point is that voltage gives you a comparison or difference". Finally, the system should just accept the best answers and move on (not using a restatement technique).

What is the relationship between the voltage reading between two terminals and their electrical states?

Minimal A voltage reading of 0 means the terminals are in the same state, a voltage reading of 1.5 tells you they are in different states

Good Voltage indicates if terminals are in the same state or in different states

Best Voltage is a difference in states between 2 terminals

Figure 5: Examples of different good answers to the same question encoded by an expert tutor

In an initial evaluation of our answer categories we had an expert tutor encode standard answers to a set of 49 questions from the curriculum, corresponding to 2 out of 3 lessons. Twenty two of the questions were factual questions where there could not be more than one type of correct answer, such as "which of the diagrams on the slide show a short circuit". Of the remaining 27, 19 (70%) had more than one type of correct answer encoded (i.e. either "minimal and best", "good and best", or "minimal, good and best"). Thirteen of those had all three types of correct answers encoded.

We implemented multiple levels of good answers in our system by allowing multiple expected answers for each question. Each answer is associated with a code (as shown in Figure 4), which can be one of **best**, **good** or **minimal**. The diagnoser compares the student answer to all possible answers, and selects the diagnosis that has the fewest errors, or, given an equal number of errors, the largest number of matching parts. The code for this answer is then passed to the tutorial planner in the diagnosis structure, and is used to select the appropriate reinforcement strategy.

4 Discussion and Future Work

In separating out the good, missing and incorrect parts of the student answer, our diagnoser design is similar to those in Why2-Atlas (Jordan *et al.* 2006) tutor, which uses uses similar categories of explicit and implicit correct and incorrect statements, along with missing statements. However, these categories are applied to essays only, and not to short answer questions in remediations, which are entirely pre-scripted. This often results in redundancy, because the system is not aware of the semantic content of tutor and student statements during remediation. A semantic tagging mechanism was added to Why2-Atlas (Jordan, Albacete, & VanLehn 2005) to control the redundancy that results from scripted remediations, but our system aims to avoid redundancy and allow the dialogue to be adaptive without need for additional labeling.

The Geometry Explanation tutor (Aleven, Popescu, & Koedinger 2001) adds adaptive feedback to the PACT cognitive tutor (Aleven *et al.* 1998) by encoding hand-authored hints for every possible combination of missing parts or errors in a definition (each combination is defined as a concept in a tutoring strategy ontology). Our diagnoser offers a potentially more flexible solution: if information about missing parts and errors was available from a cognitive model, then dialogue management and generation could be combined to generate adaptive feedback without pre-authoring (though significant effort to build the deep generation component would still be necessary).

The AUTOTUTOR system (Graesser *et al.* 1999) requires the human developers to pre-author a set of possible good answers to each question, which are matched with student input using LSA. This approach could be extended to include different types of good answers, but at the expense of pre-authoring additional tutorial feedback.

We have not yet evaluated the effectiveness of our diagnoser and tutorial strategies. The corpus is currently annotated with different reinforcement and remediation strategies, but not with different levels of good answers, and thus cannot be used directly to find associations between different types of good answers and reinforcements. The choice of strategy is therefore based on discussions with human tutors. However, we plan to evaluate its effectiveness as part of the overall system evaluation.

Our ultimate goal is to develop a system flexible enough to conduct a systematic program of research into the nature of effective adaptation in tutorial systems. The initial studies that we plan to conduct will focus on the effectiveness of the specific tutor response strategies described in this paper. We then plan to investigate hypotheses about additional variables (beyond student answer accuracy) that may also contribute to tutor response strategy selection, such as the student's confidence and history of success (or failure) on similar problems (Porayska-Pomsta 2004). The diagnoser will provide a foundation for such experimentation, as once a set of different tutoring moves is implemented, the choice of strategy can be easily altered within the tutorial planner, thus allowing us to directly compare versions of the system that use different strategies in the same contexts.

5 Conclusion

In this paper we presented evidence that tutors explicitly restate good parts of student answers, and that if these restatements occur after correct answers, they are correlated with better learning. We presented a diagnoser that provides detailed analyses of student input, and supports automatically generating restatements and other tutoring strategies. We also introduced the idea that in natural language explanations different levels of good answers are acceptable, and proposed an initial model for how a tutoring system would choose an appropriate reinforcement strategy to respond to such answers. The diagnoser and tutoring component will serve as a basis for further experimentation by allowing us to consistently apply tutorial strategies across an interaction, and compare versions of the system using different strategies. This should allow us to make more causal conclusions about the effectiveness of various tutoring strategies, instead of relying only on correlational data from corpus studies.

Acknowledgments This work was supported under grants from The Office of Naval Research numbers N000149910165 and N0001408WX20977.

References

Aleven, V.; Koedinger, K. R.; Sinclair, H. C.; and Snyder, J. 1998. Combatting shallow learning in a tutor for geometry problem solving. In *Proceedings of ITS-98*, 364–373.

Aleven, V.; Popescu, O.; and Koedinger, K. R. 2001. Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In *Proceedings of AIED-2001*.

Callaway, C. B.; Dzikovska, M.; Farrow, E.; Marques-Pita, M.; Matheson, C.; and Moore, J. D. 2007. The Beetle and BeeDiff tutoring systems. In *Proceedings of the SLaTE-2007 Workshop*.

Carletta, J.; Isard, A.; Isard, S.; Kowtko, J. C.; Doherty-Sneddon, G.; and Anderson, A. H. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics* 23(1):13–31.

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

Graesser, A. C.; Wiemer-Hastings, P.; Wiemer-Hastings, P.; and Kreuz, R. 1999. Autotutor: A simulation of a human tutor. *Cognitive Systems Research* 1:35–51.

Jordan, P.; Albacete, P.; and VanLehn, K. 2005. Taking control of redundancy in scripted tutorial dialogue. In *Proceedings of AIED-05*, 314–321.

Jordan, P.; Makatchev, M.; Pappuswamy, U.; VanLehn, K.; and Albacete, P. 2006. A natural language tutorial dialogue system for physics. In *Proceedings of FLAIRS'06*.

Jordan, P. W. 2004. Using student explanations as models for adapting tutorial dialogue. In Barr, V., and Markov, Z., eds., *FLAIRS Conference*. AAAI Press.

Kim, J. H.; Freedman, R.; Glass, M.; and Evens, M. W. 2006. Annotation of tutorial dialogue goals for natural language generation. *Discourse Processes* 42(1):37–74.

Litman, D., and Forbes-Riley, K. 2006. Correlations between dialogue acts and learning in spoken tutoring dialogues. *Natural Language Engineering* 12(2):161–176.

Porayska-Pomsta, K. 2004. *Influence of Situational Context on Language Production: Modelling Teachers' Corrective Responses*. Ph.D. Dissertation, The University of Edinburgh.

Rose, C. P.; Bhembe, D.; Siler, S.; Srivastava, R.; and VanLehn, K. 2003. The role of why questions in effective human tutoring. In *Proceedings of AIED-2003*.

Steinhauser, N. B.; Butler, L. A.; and Campbell, G. E. 2007. Simulated tutors in immersive learning environments: Empirically-derived design principles. In *Proceedings of I/ITSEC-07*.