A Natural Language Tutorial Dialogue System for Physics*

Pamela W. Jordan, Maxim Makatchev, Umarani Pappuswamy,
Kurt VanLehn and Patricia Albacete
Learning Research and Development Center
University of Pittsburgh
Pittsburgh PA, 15260
{pjordan,maxim,umarani,vanlehn,albacete}@pitt.edu

Abstract

We describe the WHY2-ATLAS intelligent tutoring system for qualitative physics that interacts with students via natural language dialogue. We focus on the issue of analyzing and responding to multi-sentential explanations. We explore approaches for achieving a deeper understanding of these explanations and dialogue management approaches and strategies for providing appropriate feedback on them.

Introduction

In a tutorial system that interacts with a student through natural language, the system needs to understand the user just well enough to respond appropriately. What it means to understand well enough and what it means to respond appropriately vary according to the application.

Most natural language tutorial applications have focused on coaching either problem solving or procedural knowledge (e.g Steve (Johnson & Rickel 1997), Circsim-tutor (Evens *et al.* 2001), BEETLE (Zinn, Moore, & Core 2002), SCoT (Pon-Barry *et al.* 2004), *inter alia*). When coaching problem solving, simple short answer analysis techniques are frequently sufficient because the primary goal is to lead a trainee step-by-step through problem solving. There is a narrow range of possible responses and the context of the previous dialogue and the question invite a short answer.

Any deeper analysis of short answers in these cases results in a small return on investment when the focus is eliciting a step during problem solving. It isn't until the instructional objectives shift and a tutorial system attempts to explore a student's chain of reasoning behind an answer or decision that deeper analysis can begin to pay off. And having the student construct more on his own is important for learning perhaps in part because he reveals what he does and does not understand (Chi *et al.* 2001). But the difficulty in understanding the explanation increases with the length of the chain of reasoning being elicited. If just one step in the reasoning is sought, then only deeper single sentence analysis is needed. This was the case with the GEOMETRY EXPLANATION TUTOR (Aleven *et al.* 2003). Since all the reasons sought were definitions, terminological classification was a good fit for understanding well enough to respond appropriately.

When the student is invited to provide a longer chain of reasoning, the explanations become multi-sentential. Compare the short explanations requested in Figure 1 to the longer ones in Figures 2 and 3. The explanation in Figure 2 is part of an initial student response and Figure 3 shows the explanation from the same student after several follow-up dialogues with the WHY2-ATLAS tutoring system. A longer explanation is unlikely to strictly follow the problem solving structure because the student may reorganize it (e.g. give an overview before going into details) and may leave out some of the reasoning, which are both common things to do in natural language.

GEOMETRY EXPLANATION TUTOR: Base angles in what type of geometric figure are congruent
Student: the bottom angles in an isoceles triangle are congruent <approximately 3 propositions expressed> (Aleven *et al.* 2003)

WHY2-AUTOTUTOR: Once again, how does Newton's third law of motion apply to this situation?
Student: Does Newton's law apply to opposite forces? <approximately 2 propositions expressed> (Graesser *et al.* 2005).

WHY2-ATLAS: Fine. Using this principle, what is the value of the horizontal component of the acceleration of the egg? Please explain your reasoning.
Student: zero because there is no horizontal force acting on the egg <approximately 3 propositions expressed>

Figure 1: Examples of 1 sentence explanations from the domains of geometry and qualitative physics.

The only previous tutoring system that has attempted to address longer explanations is AUTOTUTOR (Graesser *et al.* 2005). It uses a latent semantic analysis (LSA) approach where the structure of sentences is not considered. Thus the degree to which details of the explanation are understood is limited. But this approach is appropriate given AUTOTUTOR's pedagogical strategy of eliciting a single unit of the explanation (about one sentence or more), when LSA determines it is missing. It first hints with a short answer question

*This research was supported by ONR Grant No. N00014-00-1-0600 and by NSF Grant No. 9720359.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Natural Language Tutorial Dialogue System for Physics	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Pittsburgh,Learning Research and Development Center,3939 O'Hara Street,Pittsburgh,PA,15260**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the 19th International FLAIRS Conference. Menlo Park, CA: AAAI Press

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**6**	

Question: Suppose a man is in an elevator that is falling without anything touching it (ignore the air, too). He holds his keys motionless right in front of his face and then just releases his grip on them. What will happen to them? Explain.

<omitted approximately 15 correct propositions>... Yet the gravitational pull on the man and the elevator is greater because they are of a greater weight and therefore they will fall faster then the keys. I believe that the keys will float up to the cieling as the elevator continues falling.

Figure 2: Part of a verbatim student response to the stated problem before interacting with the tutoring system.

<omitted approximately 16 correct propositions>... Since <Net force = mass * acceleration> and <F= mass*g> therefore <mass*acceleration= mass*g> and acceleration and gravitational force end up being equal. So mass does not effect anything in this problem and the acceleration of both the keys and the man are the same. <omitted approximately 46 correct propositions>...we can say that the keys will remain right in front of the man's face.

Figure 3: Part of a verbatim response from the same student in Figure 2 after completing interaction with the system.

and if that fails, prompts with a fill-in-the-blank question and if that fails, bottoms-out with the missing unit. One way to possibly improve is to add pedagogical strategies that elicit increasingly greater precision as students' explanations become less vague. (e.g. "what can you say about the forces in this problem?", "you are right that the net force is zero but how did you determine this?"). But to do so, deeper understanding of multi-sentential explanations is likely necessary (Chi *et al.* 2001).

In this paper we will describe the WHY2-ATLAS qualitative physics tutoring system's approach for supporting a wider range of pedagogical strategies and for achieving a deeper understanding. We will end with a discussion of the system's most recent evaluation in which student learning gains were measured. Although the results are promising, much work remains to be done to assess interactions between the system's understanding performance and learning.

Dialogue Management in Why2-Atlas

Lower-level dialogue management. At the lowest-level dialogue management is a finite state network with a stack that is implemented using a reactive planner (APE (Freedman 2000)). Finite state approaches are appropriate for dialogues in which the task to be discussed is well-structured and the dialogue is to be system-led (McTear 2002), as was the case for WHY2-ATLAS.

A state in the network is either a push to a sub-network as with the right-most and left-most nodes in Figure 4 or a tutor turn plus an optional student response as with the top node and its three branches in Figure 4. There is a sub-network for each complex topic to discuss in dialogue so that a state is the equivalent of a step in a recipe for covering the topic.

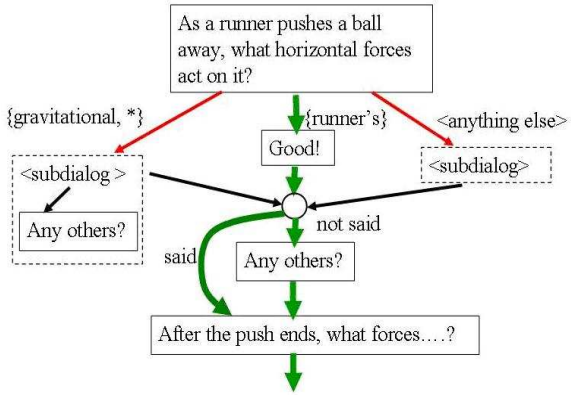

Figure 4: Finite State Model with answer classes and optional steps.

A tutor turn is a *ready-to-utter* string. When a tutor turn sets up a discourse obligation for the student (e.g. tutor asks a question as with the top node in Figure 4), there is a set of anticipated classes to recognize for each conceptually different satisfactory and unsatisfactory response. The classification of the student response decides the next state to which to move. Thus each response selects an arc between two states in the network. Classes that correspond to unsatisfactory responses lead to a state that is a push to a recipe that addresses the unsatisfactory response. These remediation recipes are written to anticipate an eventual return to a state that is the next step in the parent recipe. By default, if a tutor turn does not setup an obligation for the student to respond then the transition is to the next step in the recipe.

The anticipated student response classes for each state are further categorized as either correct answers, vague answers, expected wrong answers or unanticipated responses. This categorization of the answer classes helps determine feedback (e.g. "Correct!") which is prepended to the *ready-to-utter* strings in the network and helps in tracking the student's performance over time when analyzing the dialogue history.

Different classification techniques can be designated for each state. The default classification technique is short-answer classification since a majority of responses are still anticipated to be short-answers. But when the response for a state is expected to be an explanation then the explanation classifier is designated for that state. Both classification approaches will be described in more detail later in the paper.

In addition to answer classes, three other conditions can be used in deciding which state to go to next. One is a test to skip a state if the content of that state is already in the discourse history as with the "said" and "not said" arcs in Figure 4. The second transition condition is a test of which difficulty level is appropriate for a student. For example, there could be an alternate state relative to the last node in Figure 4 and the two alternate states could have different difficulty levels associated with them. The past performance of the student is evaluated to determine which is the appropriate one to select. The last transition condition is just before a

pop from a remediation sub-network and tests that the state before the push is still in the student's focus of attention according to the dialogue history. If it is not in the student's focus of attention then the tutor turn before the push is repeated and otherwise the pop is completed. In this case part of the original network is copied and inserted just before the pop; just the correct and the unanticipated response conditions and transitions are copied. But the path for the unanticipated response instead leads to a tutor turn that states the correct answer just before the pop is completed.

Higher-level dialogue management. This level of dialogue management oversees the finite state network and picks between three types of recipes that were authored for WHY2-ATLAS (1) a high-level walkthrough of the problem solution or parts of the problem solution, (2) short elicitations of particular pieces of knowledge and (3) remediations. Walkthrough recipes are selected when the student is unable to provide much in direct response to the qualitative physics problem or when the system is unable to classify much of what the student wrote. Short elicitations are selected if the student's response is partially complete with a few scattered gaps in order to encourage the student to fill in missing pieces of the explanation. Remediations are selected if errors or misconceptions are detected in the response. While executing a recipe, pushes to recipes for subdialogues that are of the same three types (i.e. walkthrough, elicitation or remediation) are possible but typically are limited to remediations.

In the case of single elicitation recipes, the dialogue manager will present a summary of what is correctly covered according to the response analysis. The content selected for the summary includes all nodes in a solution graph that are on the path between the node that is to be elicited and the first node that is in focus in the dialogue history (i.e. what was last talked about in dialogue). The summaries are generated using templates with clause slots, and clauses associated with the selected nodes of the graph fill those slots.

Authoring. High-level dialogue management is assumed or built into the dialogue manager but an instructor must author the lower-level finite state network. Instructors use a scripting language (Jordan, Rosé, & VanLehn 2001) to do so. The author must first define recipes and their steps, define the initial answer class labels, assign optional semantic labels to be used in implementing optional step and difficulty level transitions, and indicate the difficulty levels for each arc and which steps are optional. The reasking states, transition conditions and arcs are generated automatically from the authored network. Finally the author must define the answer classes associated with the labels in the script. How answer classes are defined is done differently for short-answers and explanations and is described in more detail in the next section.

Analyzing Student Contributions in Why2-Atlas

When a student contribution is to be analyzed, first an equation identifier tags any physics equations in the student's response and then classification is done to complete the assessment of the student's natural language contributions. In the case of explanations, the classification is with respect to steps in correct and buggy chains of reasoning. All answer classes for explanation states (including the initial response to the qualitative physics problem) are selected from precomputed chains of reasoning. In the case of short answers the classification is with respect to classes that the author defines specifically for each state. Some of these classes can be reused for other states but it is much less frequent than with explanations. First we will describe how explanations are classified and then short-answers. Finally we will briefly describe the equation identifier.

Explanation Classification

Explanation classification is broken into two stages, (1) single sentence analysis, which outputs a first-order predicate logic (FOPL) representation and then (2) an assessment of correctness and completeness of those representations with respect to nodes in correct and buggy chains of reasoning. The nodes matched in this final stage determine what classes are associated with the explanation. First we will discuss single sentence analysis and then the assessment of correctness and completeness.

Single Sentence Analysis. Single sentence analysis uses three competing single sentence analysis methods and a heuristic selection process to choose one of the output representations for each sentence (Jordan, Makatchev, & VanLehn 2004). The rationale for using multiple approaches is that the techniques available vary considerably in accuracy, processing time and whether they tend to be brittle and produce no analysis vs. a partial one. There is also a trade-off between these performance measures and the amount of domain specific setup required for each technique and there are no formal return on investment studies to give us insight into which technique is the best one to pick for an application.

The first method, CARMEL, provides combined syntactic and semantic analysis using the LCFlex syntactic parser along with semantic constructor functions (Rosé 2000). Given a specification of the desired representation language, it then maps the analysis to this language. Then discourse level processing attempts to resolve nominal and temporal anaphora and ellipsis to produce the final FOPL representation for each sentence (Jordan & VanLehn 2002). Since the knowledge engineering effort for creating semantic constructor functions is considerable there are gaps in the coverage of these functions. Also there are known gaps in the discourse level processing with respect to the WHY2-ATLAS domain.

The second method, RAINBOW, is a tool for developing *bag of words* (BOW) text classifiers (McCallum & Nigam 1998). The classes of interest must first be identified and then a text corpus annotated for example sentences for each class. From this training data a bag of words representation is derived for each class and a number of algorithms can be tried for measuring similarity of a new input segment's BOW representation to each class.

For WHY2-ATLAS, the classes we use are targeted nodes

in the correct and buggy chains of reasoning. But there were many misclassifications of sentences due to overlap in the classes; that is, words that discriminate between classes are shared by many other classes (Pappuswamy *et al.* 2005). By aggregating classes and building three tiers of BOW text classifiers that use a kNN measure, we obtained a 13% improvement in classification accuracy over a single classifier approach (Pappuswamy *et al.* 2005). The first tier classification identifies which second tier classifier to use and likewise the second tier classifier selects the third tier classifier. The third tier then identifies which if any node a sentence expresses. But even with these improvements, the current training data for WHY2-ATLAS is too sparse for some classes to achieve good accuracy.

With the BOW approach, an assessment of correctness and completeness can be skipped since a BOW class equates to a targeted node. However, a representation of the class is still needed by the single sentence selection process described below. This representation translation is obtained by looking up a stored translation of the node associated with the identified class.

Finally, the third method, RAPPEL, is a hybrid approach that uses symbolically-derived syntactic dependency features (obtained via MINIPAR (Lin & Pantel 2001)) to train for classes that are defined at the representation language level (Jordan, Makatchev, & VanLehn 2004). Each proposition in the representation language corresponds to a template in RAPPEL. Each template has its own set of classes that cover all possible ways in which the template's slots could be filled. A class indicates which slots in a particular proposition template are filled with which constants. There is a one-to-one correspondence between a filled template and an instance of a proposition in the representation language. An exception is body slots which are handled by separate binary classifiers; one for propositions involving one body and another for those involving two bodies.

A separate classifier is trained for each template. For example, there is a classifier that specializes in the velocity template and another that specializes in the acceleration template. For the WHY2-ATLAS domain, there are 27 templates and thus 27 classifiers. Each classifier returns either a nil which indicates that no form of that proposition is present or a class label that corresponds to one of the possible completions of the template. Classifiers and classes have been defined that cover the entire WHY2-ATLAS representation language but the training data is sparse relative to the number of classes.

Next one of the three possible outputs of the single sentence analyzers must be selected. The selection process is independent of the single sentence analysis techniques used; it depends only on the system's FOPL representation language. Heuristics estimate whether a resulting representation either over or under represents the sentence by matching the root forms of the words in the natural language sentence to the constants in the representation returned by each method.

If the selected representation is not a product of the multi-level BOW approach, then the representation is assessed for correctness and completeness, as described next. Recall that the multi-level BOW approach directly identifies which targeted node in the chain of reasoning a sentence represents.

Analyzing correctness and completeness As the final step in analyzing a student's explanation, an assessment of correctness and completeness is performed by matching the FOPL representations of the student's response to nodes of an augmented assumption-based truth maintenance system (ATMS) (Makatchev & VanLehn 2005). An ATMS for each physics problem is generated off-line. The ATMS compactly represents the deductive closure of a problem's givens with respect to a set of both good and buggy physics rules. That is, each node in the ATMS corresponds to a proposition that follows from a problem statement. Each anticipated student misconception is treated as an assumption (in the ATMS sense), and all conclusions that follow from it are tagged with a label that includes it as well as any other assumptions needed to derive that conclusion. This labelling allows the ATMS to represent many interwoven deductive closures, each depending on different misconceptions, without inconsistency. The labels allow recovery of how a conclusion was reached. Thus a match with a node containing a buggy assumption indicates the student has a common error or misconception and which error or misconception it is.

Completeness in WHY2-ATLAS is relative to an informal two-column proof generated by a domain expert. A human author should control which proof is used for checking completeness, and it is probably less work for an author to write an acceptable proof than to find one in the ATMS. The informal proof for the problem in Figure 2 is shown in Figure 5 where facts appear in the left column and justifications that are physics principles appear in the right column. Justifications are further categorized as vector equations (e.g. <Average velocity = displacement / elapsed time>, in step (12) of the proof), or qualitative rules (e.g. "so if average velocity and time are the same, so is displacement" in step (12)). A two-column proof is represented in the system as a directed graph in which nodes are facts, vector equations, or qualitative rules that have been translated to the FOPL representation language off-line. The single sentence analyzer can be used to assist in this translation but a developer must still review and refine the result. The edges of the graph represent the inference relations between the premise and conclusion of modus ponens.

Matches of input representations against the ATMS and the two-column proof (we collectively referred to these earlier as the correct and buggy chains of reasoning) do not have to be exact. Further flexibility in the matching process is provided by examining a neighborhood of radius N (in terms of graph distance) from matched nodes in the ATMS to determine whether it contains any of the nodes of the two-column proof. This provides an estimate of the proximity of a student's utterance to nodes of the two-column proof. Additional details on correctness and completeness analysis are provided in (Makatchev & VanLehn 2005).

Short-answer classification

Short-answer classification is accomplished using the LCFlex flexible left corner parser that is part of CARMEL

Step	Fact	Justification
1	The only force on the keys and the man is the force of gravity	Forces are either contact forces or the gravitational force
2	The magnitude of the force of gravity on the man and the keys is its mass times g	The force of gravity on an object has a magnitude of its mass times g, where g is the gravitational acceleration
...	...	...
10	At every time interval, the keys and the man have the same final velocity	<Acceleration = (final velocity - initial velocity)/elapsed time>, so for two objects, if the acceleration, initial velocity and time are the same, so is final velocity.
11	The man and the keys have the same average velocity while falling	If acceleration is constant, then <average velocity = (vf+vi)/2>, so if two objects have the same vf and vi, then their average velocity is the same.
12	The keys and the man have the same displacements at all times	<Average velocity = displacement / elapsed time>, so if average velocity and time are the same, so is displacement.
13	The keys and the man have the same initial vertical position	given
14	The keys and the man have the same vertical position at all times	<Displacement = difference in position>, so if the initial positions of two objects are the same and their displacements are the same, then so is their final position
15	The keys stay in front of the man's face at all times	

Figure 5: Part of the informal "proof" used in WHY2-ATLAS for the Elevator problem in Figure 2.

(Rosé 2000) and a separate semantic grammar for each state in which a short answer response is expected, although some rules may be shared by other states. The classes in each state grammar correspond to the expected responses. For instance, if the anticipated responses for a state are "down" and "up", then the semantic grammar would have two rules such as "state1_resp_class1 => down_class" and "state1_resp_class2 => up_class" where down_class and up_class are classes that may be shared by semantic grammars for other states. The classes are further defined by rules such as "down_class => 'down' or 'downward' or 'toward earth'. Because the LCFlex parser can skip words, it can find certain key words or phrases in the student's response even if they are surrounded by extra words, (e.g. "It is downward."). Thus when the author scripts the answer classes for a state, the author needs to list as many phrasings as possible that have similar semantics but can omit words that won't help distinguish it from a phrase with different semantics (e.g. "it" or "is").

Equation Identification

Equations can be expressed in natural language (e.g. net force is the mass times the acceleration), in algebraic form (e.g. f=ma), or in natural language mixed with algebraic symbols (e.g. net force is ma). The equation identifier tags each of these expressions in a student's input as a semantic unit. Since there is a small set of equations to consider (twelve correct and seven buggy ones) it is feasible to match directly against the representations of these equations. The equation identifier does this matching by applying a series of regular expressions before invocation of explanation or short-answer classification. Both types of classification are tolerant of formulas that have been replaced by tags since they can either skip unknown words (CARMEL), treat them as nouns (RAPPEL), or be trained with text that has been tagged for equations (RAPPEL and RAINBOW).

System Evaluation

The system was evaluated in the context of testing the hypothesis that even when content is equivalent, students who engage in more interactive forms of instruction learn more. To test this hypothesis we compared students who received human tutoring with students who read a short text. WHY2-ATLAS and WHY2-AUTOTUTOR provided a third type of condition that served as an interactive form of instruction where the content is better controlled than with human tutoring. With the computer tutors only the same content covered in the text condition can be presented. But if the system misinterprets any of a student's multi-sentential answers it may skip material covered in the text that the student needs. In all conditions the students solved four problems that require multi-sentential answers, one of which is shown in Figure 2.

After conducting a number of experiments with different subpopulations and adjustments in content and assessment materials, we found that overall students learn and learn equally well in all three types of conditions when the content is appropriate to the level of the student (VanLehn *et al.* 2005). That is, the learning gains for *human tutoring* and the content controlled text were the same. Thus, learning gains alone for this experimental setup can only reveal whether the computer tutors were the same or worse than the text. A system could perform worse if it too frequently misinterprets multi-sentential answers and skips material covered in the text that a student may need.

For the version of WHY2-ATLAS we described, the learning gains were the same on two of three different types of post-tests administered. On multiple-choice and essay post-tests, there was no reliable difference. However, on fill-in-the-blank post-tests, the WHY2-ATLAS students scored higher than the text students ($p=0.010$; $F(1,74)=6.33$), and this advantage persisted when the scores were adjusted by factoring out pre-test scores in an ANCOVA ($p=0.018$; $F(1,72)=5.83$). Although this difference was in the expected

direction, it was not accompanied by similar differences for the other two post-tests. These learning measures show that, relative to the text, the two systems' overall performance at selecting content is good. But since the dialogue strategies in the two systems are different and selected relative to the understanding techniques used, we next need to do a detailed corpus analysis of the language data collected to track successes and failures of understanding and dialogue strategy selection relative to knowledge components in the post-test.

During an informal review of the WHY2-ATLAS corpus we saw that the strategy of walking through a problem had a positive impact on students who could explain little initially. But the impact of eliciting missing pieces of an explanation was mixed and requires a detailed corpus analysis. While similar to WHY2-AUTOTUTOR's hints, these elicitations first summarize the correct components of a student's explanation that lead up to a missing or incorrect component. We expect these dialogues to be more cohesive, compared to ones using decontextualized hints, because they use problem-solving structure to present an integrated partial explanation.

Conclusion

We described a tutoring system that explores deeper understanding techniques for multi-sentential explanations and dialogue strategies that depend on deeper understanding. Compared to a system that uses shallower understanding techniques, there were no measurable differences in overall learning. However, overall learning measures do not adequately evaluate the utility of deeper understanding and its associated dialogue strategies since it assumes that understanding performance and strategy choices are correct. Thus our next step will be a detailed corpus analysis that examines correlations between student learning and system performance during tutoring.

References

Aleven, V.; Popescu, O.; Ogan, A.; and Koedinger, K. R. 2003. A formative classroom evaluation of a tutorial dialogue system that supports self-explanation. In *AIED Workshop on Tutorial Dialogue Systems: with a view toward the classroom*.

Chi, M. T. H.; Siler, S. A.; Jeong, H.; Yamauchi, T.; and Hausmann, R. G. 2001. Learning from human tutoring. *Cognitive Science* 25(4):471–533.

Evens, M.; Brandle, S.; Chang, R.; Freedman, R.; Glass, M.; Lee, Y.; Shim, L.; Woo, C.; Zhang, Y.; Zhou, Y.; Michael, J.; and Rovick, A. 2001. Circsim-tutor: An intelligent tutoring system using natural language dialogue. In *Proceedings of 12th Midwest AI and Cognitive Science Conference*, 16–23.

Freedman, R. 2000. Plan-based dialogue management in a physics tutor. In *Proceedings of the 6th Applied Natural Language Processing Conference*.

Graesser, A. C.; Olney, A.; Haynes, B. C.; and Chipman, P. 2005. AutoTutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue. In Forsythe, C.; Bernard, M.; and Goldsmith, T., eds., *Cognitive systems: Human cognitive models in systems design*. Mahwah: Erlbaum.

Johnson, W. L., and Rickel, J. 1997. Stev: An animated pedagogical agent for procedural training in virtual environments. *SIGART Bulletin* 16–21.

Jordan, P., and VanLehn, K. 2002. Discourse processing for explanatory essays in tutorial applications. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*.

Jordan, P. W.; Makatchev, M.; and VanLehn, K. 2004. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of the Intelligent Tutoring Systems Conference*.

Jordan, P.; Rosé, C.; and VanLehn, K. 2001. Tools for authoring tutorial dialogue knowledge. In *Proceedings of AI in Education 2001 Conference*.

Lin, D., and Pantel, P. 2001. Discovery of inference rules for question answering. *Journal of Natural Language Engineering* 7(4):343–360.

Makatchev, M., and VanLehn, K. 2005. Analyzing completeness and correctness of utterances using an ATMS. In *Proceedings of Int. Conference on Artificial Intelligence in Education, AIED2005*. IOS Press.

McCallum, A., and Nigam, K. 1998. A comparison of event models for naive bayes text classification. In *Proceeding of AAAI/ICML-98 Workshop on Learning for Text Categorization*. AAAI Press.

McTear, M. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys* 34(1):90–169.

Pappuswamy, U.; Bhembe, D.; Jordan, P. W.; and VanLehn, K. 2005. A multi-tier NL-knowledge clustering for classifying students' essays. In *Proceedings of 18th International FLAIRS Conference*.

Pon-Barry, H.; Clark, B.; Bratt, E. O.; Schultz, K.; and Peters, S. 2004. Evaluating the effectiveness of SCoT—a spoken conversational tutor. In Heffernan, N., and Wiemer-Hastings, P., eds., *Workshop on Dialog-based Intelligent Tutoring Systems*, 23–32.

Rosé, C. P. 2000. A framework for robust semantic interpretation. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, 311–318.

VanLehn, K.; Graesser, A.; Jackson, G. T.; Jordan, P.; Olney, A.; and Rosé, C. P. 2005. When is reading just as effective as one-on-one interactive human tutoring? In *Proceedings of CogSci2005*.

Zinn, C.; Moore, J. D.; and Core, M. G. 2002. A 3-tier planning architecture for managing tutorial dialogue. In *Proceedings of Intelligent Tutoring Systems Conference (ITS 2002)*, 574–584.