Understanding Complex Natural Language Explanations in Tutorial Applications*

Pamela W. Jordan, Maxim Makatchev and Umarani Pappuswamy
Learning Research and Development Center
University of Pittsburgh
Pittsburgh PA, 15260
{pjordan,maxim,umarani}@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 an approach that combines a statistical classifier, multiple semantic parsers and a formal reasoner for achieving a deeper understanding of these explanations in order to provide appropriate feedback on them.

1 Introduction

Most natural language tutorial applications have focused on coaching either problem solving or procedural knowledge (e.g. Steve (Johnson and Rickel, 1997), Circsim-tutor (Evens and Michael, 2006), Atlas (Rosé et al., 2001), BEETLE (Zinn et al., 2002), SCoT (Peters 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 questions invite short answers. But when the instructional objectives shift and a tutorial system attempts to explore a student's chain of reasoning behind an answer or decision, deeper analysis techniques can begin to pay off. Having the student construct more on his own is important for learning perhaps in part because it reveals what the student does and does not understand (Chi et al., 2001).

When the student is invited to provide a longer chain of reasoning, the explanations become multi-sentential. Compare the short explanation in Figure 1 to the longer ones in Figures 2 and 3. The explanation in Figure 2 is part of an actual initial student response and Figure 3 shows the explanation from the same student after a follow-up dialogue with the WHY2-ATLAS tutoring system.

> 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 *[3 propositions expressed]*

Figure 1: Eliciting a one sentence explanation from a student.

> WHY2-ATLAS: 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.
>
> Student: *[omitted 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: An initial elicitation of a multi-sentence explanation from a student.

The only previous tutoring system that has attempted to address longer explanations is AUTOTUTOR (Graesser et al., 2004). It uses a latent semantic

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

Proceedings of the 3rd Workshop on Scalable Natural Language Understanding, pages 17–24,
New York City, June 2006.

[omitted 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 46 correct propositions]*...we can say that the keys will remain right in front of the man's face.

Figure 3: A subsequent response from the same student in Figure 2 after some interaction with WHY2-ATLAS.

analysis (LSA) approach where the structure of sentences is not considered. Thus the degree to which details of the explanation are understood is limited.

As can be seen from the examples, a student's explanation about a formal domain such as qualitative physics may involve a number of phenomena: algebraic formulas, NL renderings of formulas, various degrees of formality, and conveying the logical structure of an argument (Makatchev et al., 2005).

Tutoring goals involve eliciting correct statements of the appropriate degree of formality and their justifications to address possible gaps and errors in the explanation. To achieve these goals the NL understanding is required to answer the following questions:

- Does the student explanation contain errors? If yes, what are the likely buggy assumptions that have led the student to these errors?
- What required statements have not been covered by the student? Does the explanation contain statements that are logically close to the required statements?

These requirements imply that a logical structure needs to be imposed on the space of possible domain statements. Considering such a structure to be a model of the student's reasoning about the domain, the two requirements correspond to a solution of a model-based diagnosis problem (Forbus and de Kleer, 1993).

How does one build such a model? A desire to make the process scalable and feasible necessitates an automated procedure. The difficulty is that this automated reasoner has to deal with the NL phenomena that are relevant for our application. In turn, this means that the knowledge representation (KR) would have to be able to express these phenomena (e.g. NL renderings of formulas, various degrees of formality). The reasoner has to account for common reasoning fallacies, have flexible consistency constraints and perform within the tight requirements of a real-time dialogue application.

In this paper, we present a hybrid of symbolic and statistical approaches that attempts to robustly provide a model-based diagnosis of a student's explanation. In the next section, we provide a brief sketch of the KR used in WHY2-ATLAS. Section 3 describes our hybrid approach for analyzing student explanations while section 4 covers our most recent evaluations of the system and its explanation analysis components. Section 5 presents our conclusions along with future directions.

2 Knowledge representation

We selected an order-sorted first-order predicate logic (FOPL) as a base KR for our domain since it is expressive enough to reflect the hierarchy of concepts from the qualitative mechanics ontology (Ploetzner and VanLehn, 1997) and has a straightforward proof theory (Walther, 1987). Following the representation used in the abductive reasoner Tacitus-lite (Thomason et al., 1996), our KR is function-free, does not have quantifiers, Skolem constants or explicit negation. Instead all variables in facts or goals are assumed to be existentially quantified, and all variables in rules are either universally quantified (if they appear in premises) or existentially quantified (if they appear in conclusions only).

Although our KR has no explicit negation, some types of negative statements are represented by using (a) complimentary sorts, for example `constant` and `nonconstant`; (b) the value `nonequal` as a filler of the respective argument of comparison predicates.

Instead of parsing arbitrary algebraic expressions, an equation identifier module attempts shallow parsing of equation candidates and maps them into a finite set of anticipated equation labels (Makatchev et al., 2005).

NL understanding needs to distinguish formal versus informal physics expressions so that the tutoring system can coach on proper use of terminology. Many qualitative mechanics phenomena may

be described informally, for example "speed up" instead of "accelerate" and "push" instead of "apply a force." The relevant informal expressions fall into the following categories:

- relative position: "keys are behind (in front of, above, under, close, far from, etc.) man"
- motion: "move slower," "slow down," "moves along a straight line"
- dependency: "horizontal speed will not depend on the force"
- direction: "the force is downward"
- interaction: "the man pushes the keys," "the gravity pulls the keys"

Each of these categories (except for the last one) has a dedicated representation. While representing push and pull expressions via a dedicated predicate seems straightforward, we are still assessing the utility of distinguishing "man pushes the keys" and "man applies a force on the keys" for our tutoring application and currently represent both expressions as a nonzero force applied by the man to the keys.

One of the tutoring objectives of WHY2-ATLAS is to encourage students to provide argumentative support for their conclusions. This requires recognizing and representing the justification-conclusion clauses in student explanations. Recognizing such clauses is a challenging NLP problem due to the issue of quantifier and causality scoping. It is also difficult to achieve a compromise between two competing requirements for a suitable representation. First, the KR should be flexible enough to account for a variable number of justifications. Second, reasoning with the KR should be computationally feasible. We leave representing the logical structure of explanations for future work.

3 Analyzing Student Explanations

When analyzing a student explanation, first an equation identifier tags any physics equations in the student's response and then the explanation is classified to complete the assessment. Explanation classification is done by using either (a) a statistical classifier that maps the explanation directly into a set of known facts, principles and misconceptions, or (b) two competing semantic parsers that each generate an FOPL representation that is then matched against known facts, principles or misconceptions, as well as against pre-computed correct and buggy chains of reasoning. We present the approaches at a high-level in order to focus on how the approaches work when combined and our evaluation results.

3.1 Statistical classifier

RAINBOW is a tool for developing *bag of words* (BOW) text classifiers (McCallum and 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 are a subset of nodes in the correct and buggy chains of reasoning. Limiting the number of classes allows us to alleviate the problem of sparseness of training data, but the side-effect is that there are 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). We alleviate this problem some by aggregating classes and building three tiers of BOW text classifiers that use a kNN measure. By doing so, we obtain a 13% improvement in classification accuracy over a single classifier approach (Pappuswamy et al., 2005). The upper two tiers of classification describe the topic of discussion and the lower tier describes the specific principle or misconception related to the topic and subtopic. The first tier classifier identifies which second tier classifier to use and so on. The third tier then identifies which node (if any) in the chain of reasoning a sentence expresses.

But because the number of classes is limited, BOW has problems dealing with many of the NL phenomena we described earlier. For example, although it can deal with some informal language use (i.e. 'push the container' maps to 'apply force on the container'), it cannot provide accurate syntactic-semantic mappings between informal and formal language on the fly. This is because the informal language use is so varied that it is difficult to capture representative training data in sufficient quantities. Hence, a large portion of student statements either cannot be classified with high confidence or are

erroneously classified. We use a post-classification heuristic to try to filter out the latter cases. The filtering heuristic depends on the system's representation language and not on the classification technique. Given a classification of which node in the chain of reasoning the sentence represents, the heuristic estimates whether the node's FOPL 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 system's representation language.

For those statements BOW cannot classify or that the heuristic filters out, we attempt classification using an FOPL representation derived from semantic parsing, as described in the next two subsections.

3.2 Converting NL to FOPL

Two competing methods of sentence analysis each generate a FOPL candidate. The two candidates are then passed to a heuristic selection process that chooses the best one (Jordan et al., 2004). The rationale for using competing 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.

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 candidate FOPL representation for a sentence (Jordan and VanLehn, 2002).

The second method, RAPPEL, uses MINIPAR (Lin and Pantel, 2001) to parse the sentence. It then extracts syntactic dependency features from the parse to use in mapping the sentence to its FOPL representation (Jordan et al., 2004). Each predicate in the KR language is assigned a predicate template and a separate classifier is trained for each predicate template. For example, there is a classifier that specializes in predicate instantiations (atoms) involving the velocity predicate and another for instantiations of the acceleration predicate. Classes for each template represent combinations of constants that can fill a predicate template's slots to cover all possible instantiations of that predicate. Each predicate template classifier returns either a nil which indicates that there is no instantiation involving that predicate or a class label that corresponds to an instantiation of that predicate. The candidate FOPL representation for a statement is the union of the output of all the predicate template classifiers.

Finally, either the CARMEL or RAPPEL candidate FOPL output is selected using the same heuristic as for the BOW filtering. The surviving FOPL representation is then assessed for correctness and completeness, as described next.

3.3 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 and 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 labeling 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.

The completeness of an explanation is relative to a two-column proof generated by a domain expert. A human creates the proof that is used for checking completeness since it is probably less work for a person to write an acceptable proof than to find one in the ATMS. Part of the proof for the problem in Figure 2 is shown in Figure 4 where facts

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

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 edges of the graph represent the inference relations between the premise and conclusion of modus ponens.

Matches of an FOPL input 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. In addition, 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 the facts that are of interest.

Although matching against the ATMS deductive closure has been implemented, the current version of the system does not yet fully utilize this capability. Instead, the correctness and completeness of explanations is evaluated by flexibly matching the FOPL input against targeted relevant facts, principles and misconceptions in the chains of reasoning, using a radius of 0. This kind of matching is referred to as direct matching in Section 4.2.

4 Evaluations

WHY2-ATLAS, as we've just described it, has been fully implemented and 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 in that only some subset of the content covered in the text condition can be presented. In all conditions the students had to solve four problems that require multi-sentential explanations, one of which is shown in Figure 2.

In earlier evaluations, 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), i.e. the learning gains for *human tutoring* and the content controlled text were the same. For the latest evaluation of WHY2-ATLAS, which excluded a human tutoring condition, the learning gains on multiple-choice and essay post-tests were the same as for the other conditions. 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. A system could perform worse than the text condition if it too frequently misinterprets multi-sentential answers and skips material covered in the text that a student may need. 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. Next we will describe some component-level evaluations that focus on the parts of the system we just described.

4.1 Evaluating the Benefit of Combining Single Sentence Approaches

This first component-level evaluation focuses on the benefits of heuristically choosing between the results of BOW, CARMEL and RAPPEL. This particular evaluation used a prior version of the system which used BOW without tiers and hand-crafted pattern-matching rules instead of the ATMS approach to assessment. But this evaluation still reflects the potential benefits of combining single sentence approaches.

We used a test suite of 35 held-out multi-sentence student explanations (235 sentences total) that are annotated for the elicitation topics that are to be discussed with the student. We computed recall (R), precision (P) and false alarm rate (FAR) against the full corpus instead of averaging these measures for each explanation. Since F-measure does not allow error skewing as can be done with ROC areas (Flach, 2003) we instead look for cases of high recall with a low false alarm rate.

The top part of Table 1 compares the baseline of tutoring all possible topics and the individual performances of the three approaches when each is used in isolation from the others. We see that only the statistical approach lowers the false alarm rate but does so by sacrificing recall. The rest are not significantly different from tutoring all topics. The poor performances of CARMEL and RAPPEL is not totally unexpected because there are three potential failure points for these classification approaches; the syntactic analysis, the semantic mapping and the hand-crafted pattern matching rules for assessing correctness and completeness. While the syntactic analysis results for both approaches are good, the semantic mapping and assessment of correctness and completeness are still big challenges. The results of BOW, while better than that of the other two approaches, are clearly not good enough.

Table 1: Performance of NL to FOPL for actions taken in WHY2-ATLAS system.

Approach	R	P	FAR
tutor all topics	1.0	.61	1.0
CARMEL	1.0	.61	1.0
BOW without tiers	.60	.93	.07
RAPPEL	.94	.59	1.0
satisficing heuristic	.67	.80	.26
highest ranked heuristic	.73	.76	.36

The bottom part of Table 1, shows the results of combining the approaches and choosing one output heuristically. The satisficing[1] version of the heuristic checks each output in the order 1) CARMEL 2) BOW 3) RAPPEL, and stops with the first representation that is acceptable according to the filtering heuristic. This heuristic selection process modestly improves recall but at the sacrifice of a higher false alarm rate. The highest ranking heuristic scores each output and selects the best one. It provides the most balanced results of the combined or individual approaches. It provides the largest increase in recall and the false alarm rate is still modest compared to the baseline of tutoring all possible topics. It is clear, that a combined approach has a positive impact.

4.2 Completeness and Correctness Evaluation

The component-level evaluation for completeness and correctness was completed after the student learning evaluation. It focuses on the performance of just the direct matching procedure. Figure 5 shows the results of classifying 62 student utterances for one physics problem with respect to 46 stored statement representations using only direct matching. To generate these results, the data is manually divided into 7 groups based on the quality of the NL

[1] According to Newell & Simon (1972), satisficing is the process by which an individual sets an acceptable level as the final criterion and simply takes the first acceptable move instead of seeking an optimal one.

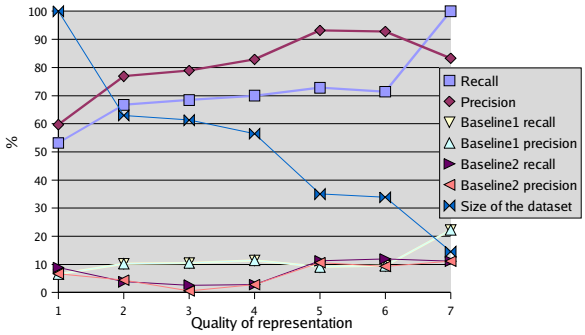

Figure 5: Average recall and precision of utterance classification. The size of a group of entries is shown relative to the size of the overall data set. Average processing time is 0.011 seconds per entry on a 1.8 GHz Pentium 4 machine with 2Gb of RAM.

to FOPL conversion, such that group 7 consists only of perfectly formalized entries, and for $1 \leq n \leq 6$ group n includes entries of group $n+1$ and additionally entries of somewhat lesser representation quality, so that group 1 includes all the entries of the data set. The flexibility of the direct matching algorithm even allows classification of utterances that have mediocre representations, resulting in 70% average recall and 82.9% average precision for 56.5% of all entries (group 4). However, large numbers of inadequately represented utterances (38.7% of all entries did not make it into group 3 of the data set) result in 53.2% average recall and 59.7% average precision for the whole data set (group 1). These results are still significantly better compared to the two baseline classifiers the best of which peaks at 22.2% average recall and precision. The first baseline classifier always assigns the single label that is dominant in the training set (average number of labels per entry of the training set is 1.36). The second baseline classifier independently and randomly picks labels according to their distributions in the training set. The most frequent label in the training set corresponds to the answer to the problem. Since in the test set the answer always appears as a separate utterance (sentence), recall and precision rates for the first baseline classifier are the same.

Although the current evaluation did not involve matching against the ATMS, we did evaluate the time required for such a match in order to make a rough comparison with our earlier approach. Matching a 12 atom input representation against a 128 node ATMS that covers 55% of relevant problem facts takes around 30 seconds, which is a considerable improvement over the 170 seconds required for the on-the-fly analysis performed by the Tacitus-lite+ abductive reasoner (Makatchev et al., 2004)—the technique used in the previous version of WHY2-ATLAS. The matching is done by a version of a largest common subgraph-based graph-matching algorithm (due to the need to account for cross-referencing atoms via shared variables) proposed in (Shearer et al., 2001), that has a time complexity $O(2^n n^3)$, where n is the size of an input graph. The efficiency can be further improved by using an approximation of the largest common subgraph in order to evaluate the match.

5 Conclusion

In this paper, we discussed an application that integrates a hybrid of semantic parsers and a symbolic reasoner with a statistical classifier to analyze student explanations. We attempted to address the problem that the leap made by statistical classifiers from NL to a feasible classification is too big since too many details of what was actually said by the student are lost. On the other hand, we showed that the hybrid semantic parsers allow for a slightly smaller leap by mapping to a symbolic representation that is sufficient for domain reasoning. Using deductive closure of problem givens and buggy assumptions, the correctness and completeness analyzer allows us to reason about the correctness of student statements that cannot be confidently classified statistically. Although formal and informal language expressions have unique underlying semantics, we attempt to paraphrase informal NL into formal NL by using the forward-chaining rules involved in creating the deductive closure for a problem from its givens. Our current symbolic representation is still too coarse to distinguish some fine nuances allowed by the domain of mechanics. We conjecture that extending our knowledge representation with more language-specific predicates would allow us to represent more fine-grained differences in student statements while still allowing feasible reasoning with the ATMS.

References

Michelene T. H. Chi, Stephanie A. Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G. Hausmann. 2001. Learning from human tutoring. *Cognitive Science*, 25(4):471–533.

M. Evens and J. Michael. 2006. *One-on-One Tutoring by Humans and Computers*. Lawrence Erlbaum Associates, Inc.

P. Flach. 2003. The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In *Proceedings of 20th International Conference on Machine Learning*.

Kenneth D. Forbus and Johan de Kleer. 1993. *Building Problem Solvers*. MIT Press, Cambridge, Massachusetts; London, England.

A.C. Graesser, S. Lu, G.T. Jackson, H. Mitchell, M. Ventura, A. Olney, and M.M. Louwerse. 2004. Autotutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, 36:180–193.

W. Lewis Johnson and Jeff Rickel. 1997. Steve: An animated pedagogical agent for procedural training in virtual environments. *SIGART Bulletin*, pages 16–21, Fall.

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

Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. 2004. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of the Intelligent Tutoring Systems Conference*.

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

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

Maxim Makatchev, Pamela W. Jordan, and Kurt VanLehn. 2004. Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems. *Journal of Automated Reasoning, Special issue on Automated Reasoning and Theorem Proving in Education*, 32:187–226.

Maxim Makatchev, Brian S. Hall, Pamela W. Jordan, Umarani Pappuswamy, and Kurt VanLehn. 2005. Mixed language processing in the Why2-Atlas tutoring system. In *Proceedings of the Workshop on Mixed Language Explanations in Learning Environments, AIED2005*, pages 35–42, July.

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

A. Newell and H.A. Simon. 1972. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ.

Umarani Pappuswamy, Dumisizwe Bhembe, Pamela W. Jordan, and Kurt VanLehn. 2005. A multi-tier NL-knowledge clustering for classifying students' essays. In *Proceedings of 18th International FLAIRS Conference*.

S. Peters, E. Bratt, B. Clark, H. Pon-Barry, and K. Schultz. 2004. Intelligent systems for training damage control assistants. In *In the Proceedings of I/ITSEC 2004*, Orlando, Florida.

Rolf Ploetzner and Kurt VanLehn. 1997. The acquisition of qualitative physics knowledge during textbook-based physics training. *Cognition and Instruction*, 15(2):169–205.

Carolyn Rosé, Pamela Jordan, Michael Ringenberg, Stephanie Siler, Kurt VanLehn, and Anders Weinstein. 2001. Interactive conceptual tutoring in atlas-andes. In *Proceedings of AI in Education 2001 Conference*.

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

Kim Shearer, Horst Bunke, and Svetha Venkatesh. 2001. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34(5):1075–1091.

Richmond H. Thomason, Jerry Hobbs, and Johanna D. Moore. 1996. Communicative goals. In K. Jokinen, M. Maybury, M. Zock, and I. Zukerman, editors, *Proceedings of the ECAI 96 Workshop Gaps and Bridges: New Directions in Planning and Natural Language Generation*.

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

Christof Walther. 1987. *A many-sorted calculus based on resolution and paramodulation*. Morgan Kaufmann, Los Altos, California.

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