Analyzing Completeness and Correctness of Utterances Using an ATMS

Maxim Makatchev [1] and Kurt VanLehn
Learning Research and Development Center, University of Pittsburgh

Abstract. Analyzing coverage of a student's utterance or essay (completeness) and diagnosing errors (correctness) can be treated as a diagnosis problem and solved using a well-known technique for model-based diagnosis: an assumption-based truth maintenance system (ATMS). The function-free first-order predicate logic (FOPL) representation of the essay is matched with nodes of the ATMS that are then analyzed for being within the sound part of the closure or relying on a particular misconception. If the matched nodes are sound they are analyzed for representing a particular required physics statement. If they do not represent the required statement, a neighborhood (antecedent and consequent nodes within N inference steps) of these nodes can be analyzed for matching the statement, to give a measure of how far the student utterance is, in terms of a number of inferences, from the desired one.

Keywords. Dialogue-based intelligent tutoring systems, formal methods in natural language understanding, ATMS

1. Introduction

Analyzing student input to an intelligent tutoring system for coverage (completeness) and errors (correctness) is essential for generating adequate feedback. When the student input is spoken or typed natural language (NL), analysis of the input becomes a significant problem. While statistical methods of analysis in many cases are sufficient [2], our tutoring system, Why2-Atlas [11], must analyze coverage and errors at a fine grain-size so that it can pinpoint students' mistakes and help students learn from them. This finely detailed analysis requires a large number of classes whose representatives have nearly the same bags of words and syntactic structures. This makes it very difficult for statistical classifiers to determine which classes best fit the student's input. Thus, Why2-Atlas is relying increasingly on non-statistical NLU in order to produce an adequately detailed analysis of student input.

In previous work [6], we demonstrated the feasibility of using an abductive reasoning back-end for analyzing students' NL input. A major part of this work involved defining and refining the knowledge representation language. As the development progressed, it became clear that adequate tutoring depended on being able to make fine distinctions, so the language became increasingly fine-grained. As the granularity decreased, the number

[1] Correspondence to: Maxim Makatchev, LRDC, 3939 O'Hara Street, Pittsburgh, PA 15260, USA. Tel.: +1 412 624 7498 ; Fax: +1 412 624 7904; E-mail: maxim@pitt.edu.

of inferences required to connect utterances increased. The abductive reasoning backend would make these inferences at run-time using the Tacitus-lite+ theorem prover. As the number of inferences to be made at run-time increased, it became more difficult to provide a guaranteed bound on the response time of the tutoring system.

In order to improve the response time of Why2-Atlas and to increase the maintainability of the knowledge base, we have switched to precomputing as much of the reasoning as possible. In particular, Why2-Atlas now precomputes all the reasoning that depends only on the problem and not on the student's solution to the problem. Reasoning that depends on the student input is of course still done at runtime. Because so much reasoning is done in advance, we can check each problem's precomputed reasoning thoroughly in order to guarantee that no flaws have crept into the knowledge base.

For this purpose, we adopted an augmented assumption-based truth maintenance system (ATMS) to precompute the desired reasoning [1]. Essentially, the precomputation requires computing the deductive closures of a set of rules of physics (e.g., "zero net force implies zero acceleration") and a set of propositions representing a particular problem (e.g., "the truck has a larger mass than the car"). However, our knowledge representation includes rules for student misconceptions, such as "zero force implies velocity decreases." Including both buggy rules and correct ones in the same deductive closure introduces inconsistencies. Thus, each 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/misconceptions needed to derive that conclusion. This labeling essentially allows the ATMS to represent many interwoven deductive closures, each depending on different misconceptions, without inconsistency.

This also makes is much easier to check the precomputed reasoning for flaws. By examining the labels, one can easily figure out how a conclusion was reached, which facilitates debugging the knowledge base. Moreover, it allows us to automate regression testing. Whenever a significant change is made to the knowledge base, one compares the newly computed conclusions to those saved just before making a change. Similar advantages have driven other ITS projects to use precomputed reasoning as well [12,9].

This paper begins by reviewing the NLU task of Why2-Atlas and its knowledge representation in Sections 2 and 3. We then discuss the design choices for the ATMS (Section 4) and the structure of the completeness and correctness analyzer (Section 5). We end with the preliminary evaluation results in Section 6 and the conclusions in Section 7.

2. Role of NLU in Why2-Atlas tutoring system

The Why2-Atlas tutoring system is designed to encourage students to write their answers to qualitative mechanics problems along with detailed explanations supporting their arguments [11]. A typical problem and a student explanation is shown in Figure 1.

Each problem has an ideal "proof" designed by expert physics tutors that contains steps of reasoning, i.e. facts and their justifications, and ends with the correct answer. The proof for the Clay Balls problem from Figure 1 is given in Figure 2. Not all of the proof facts and justifications are required to be present in an acceptable student essay. The task of the NLU module is to identify whether the required points have been mentioned and whether any of the essay propositions are related to a set of known common misconceptions.

Problem: A heavy clay ball and a light clay ball are released in a vacuum from the same height at the same time. Which reaches the ground first? Explain.
Explanation: Both balls will hit at the same time. The only force acting on them is gravity because nothing touches them. The net force, then, is equal to the gravitational force. They have the same acceleration, g, because gravitational force=mass*g and f=ma, despite having different masses and net forces. If they have the same acceleration and same initial velocity of 0, they have the same final velocity because acceleration=(final-initial velocity) elapsed time. If they have the same acceleration, final, and initial velocities, they have the same average velocity. They have the same displacement because average velocity=displacementtime. The balls will travel together until the reach the ground.

Figure 1. The statement of the problem and a verbatim student explanation.

Step	Proposition	Justification
1	Both balls are near earth	Unless the problem says otherwise, assume objects are near earth
2	Both balls have a gravitational force on them due to the earth	If an object is near earth, it has a gravitational force on it due to the earth
3	There is no force due to air friction on the balls	When an object is in a vacuum, no air touches it
4	The only force on the balls is the force of gravity	Forces are either contact forces or the gravitational force
5	The net force on each ball equals the force of gravity on it	[net force = sum of forces], so if each object has only one force on it, then the object's net force equals the force on it
6	**Gravitational force is w = m*g for each ball**	**The force of gravity on an object has a magnitude of its mass times g, where g is the gravitational acceleration**
⋮	⋮	⋮
18	**The balls have the same initial vertical position**	given
19	The balls 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
20	**The balls reach the ground at the same time**	

Figure 2. A fragment of an ideal "proof" for the Clay Balls problem from Figure 1. The required points are in bold.

After the essay analysis is complete the tutoring feedback may be a dialogue that addresses missing required points or erroneous propositions. During a dialogue an analysis similar to that performed during the essay stage may be required for some student turns: does the student's dialogue turn include a required point or is it related to a known misconception.

3. Knowledge representation

The difficulty of converting unconstrained natural language into a formal representation is one of the main obstacles to using formal reasoning techniques for NLU. We designed FOPL representation that is expressive enough to cover the physics domain propositions we are interested in, and is able to preserve formal and informal descriptions of the domain concepts (for example, "the force is downward" versus "the horizontal component of the force is zero and the vertical component is negative", "the balls' positions are the same" versus "the balls move together") [5], and can incorporate algebraic expressions (for example, "F=ma"). This relatively fine granularity of representation for degrees of formality in NL is useful for providing more precise tutoring feedback, and can be generated by language understanding approaches that include statistical classifiers [3].

To demonstrate the flexibility of the KR with an example, we include a few slightly abridged representations below:

"the balls' positions are the same"

```
(position p1 big-ball ?comp1 ?d-mag1 ?d-mag-num1
      ?mag-zero1 ?mag-num1 ?dir1 ?dir-num1 ?d-dir1 ?time1 ?time2)
(position p2 small-ball ?comp1 ?d-mag1 ?d-mag-num1
      ?mag-zero1 ?mag-num1 ?dir1 ?dir-num1 ?d-dir1 ?time1 ?time2)
```

"the balls move together"

```
(motion m1 big-ball ?comp2 ?traj-shape2 ?traj-speed2 ?d-mag2
    ?d-mag-num2 ?mag-zero2 ?dir2 ?dir-num2 ?d-dir2 ?time3 ?time4)
(motion m1 small-ball ?comp ?traj-shape ?traj-speed ?d-mag2
    ?d-mag-num2 ?mag-zero2 ?dir2 ?dir-num2 ?d-dir2 ?time3 ?time4)
```

In these examples the equality of arguments of two predicates is represented via the use of shared variables.

4. ATMS design

ATMS's have been used for tasks that are closer to the front end of the NLU processing pipeline such as for parsers that perform reference resolution (e.g. [7]), but there are few systems that utilize an ATMS at deeper levels of NLU [4,13]. In our view, given that a formal representation of student input is obtained, the task of analyzing its completeness and correctness can be treated as a diagnosis problem and solved by methods of model-based diagnosis. In this section we describe in detail the ATMS we designed for the task of diagnosing formal representations of NL utterances.

For the description of ATMS features below we adopt the terminology from [1]:

- *Premises* are givens of the physics problem ("initial positions of balls are the same," etc.)
- *Assumptions* are statements about student beliefs in a particular misconception ("Student believes that heavier objects fall faster").
- *Deduction rules* are the rules of inferences in the domain of mechanics ("zero force implies zero acceleration").

- *Nodes* are the atoms of the FOPL representation that are derived from the givens and assumptions via forward chaining with the deduction rules.
- *Labels* are assumptions that were made on the way to derive the particular node.
- *Environment* is a consistent set of assumptions that are sufficient to infer a node.

Our implementation of the ATMS relaxes the usual requirement of consistency of the deductive closure, because in our context students may hold inconsistent beliefs. While this certainly increases the size of the deductive closure, it may potentially provide a better explanation of the student's actual reasoning. The degree of ATMS consistency needed to best match with the observed student's reasoning is a topic we will explore during a future evaluation.

5. Completeness and correctness analyzer Cocoro

All domain statements that are potentially required to be recognized in the student's explanation or utterances are divided into principles and facts. The principles are versions of general physics (and "buggy physics") principles that are either of a vector form (for example, "F=ma") or of a qualitative form (for example,"if total force is zero then acceleration is zero"), while facts correspond to concrete instantiations of the principles (for example, "since there is no horizontal force on the ball its horizontal acceleration is zero") or to derived conclusions (for example, "the horizontal acceleration of the ball is zero"). As a natural consequence of the fact that the ATMS deductive inferences are derived from the problem givens, which are instantiated facts, the ATMS includes only facts. Therefore the recognition of both general principles and facts must be restricted to the actual input representations, while the ATMS is used only for recognizing and evaluating the correctness of facts closely related to the student's utterances, as shown in Figure 3 and elaborated below.

The nodes of the ATMS that match the representation of the input utterance are analyzed for correctness by checking whether their labels contain only environments with buggy assumptions. If there are no environments that are free of buggy assumptions in the label of the node, the node can only be derived using one of the buggy assumptions and therefore represents a buggy fact. These buggy assumptions are then reported to the tutoring-system strategist for possible remediation. If the nodes are correct (labels contain assumption-free environments) they are matched with required statements and the list of matched statements is then reported to the tutoring-system strategist for possible elicitation of any missing points. Additionally, a neighborhood of radius N (in terms of a graph distance) of the matched nodes can be analyzed for whether it contains any of the required principles to get an estimate of the proximity of a student's utterance to a required point.

For example, given the formal representation for the student utterance "the balls have the same vertical displacement," Cocoro attempts to both directly match it with stored statement representation (the right branch in the diagram in Figure 3) and find a set of matching nodes in the ATMS (the left branch in the diagram in Figure 3). If the direct match succeeds this already provides information about whether the student statement is correct or not. If the direct match fails, namely we do not have a stored representation for this fact, then we arrive at a conclusion about the correctness of the student's statement by examining the labels of the ATMS nodes that matched the input statement, if there are

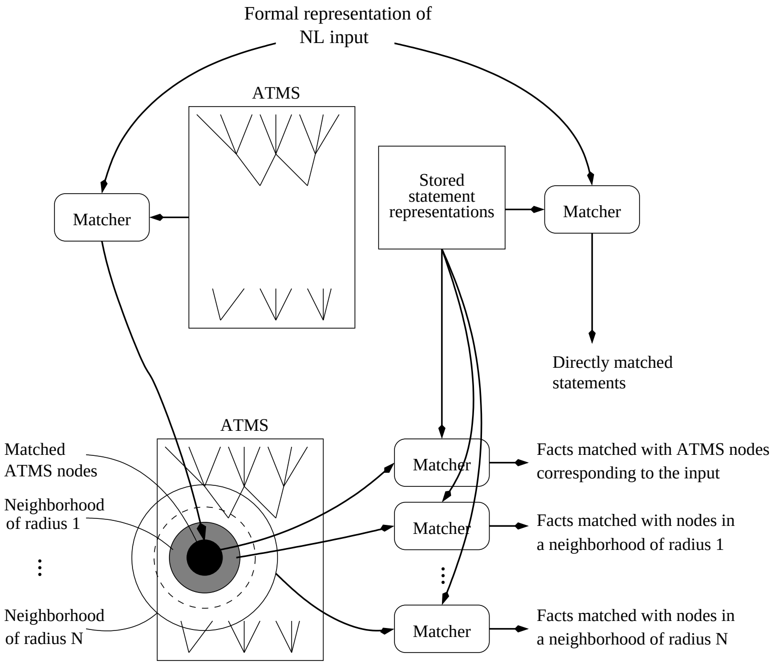
Formal representation of NL input
ATMS
Matcher
Stored statement representations
Matcher
Directly matched statements
ATMS
Matched ATMS nodes
Neighborhood of radius 1
Neighborhood of radius N
Matcher
Matcher
Matcher
Facts matched with ATMS nodes corresponding to the input
Facts matched with nodes in a neighborhood of radius 1
Facts matched with nodes in a neighborhood of radius N

Figure 3. Completeness and correctness analyzer Cocoro. A description of the diagram is in the text.

any (represented by the black circle in the ATMS block in Figure 3). The neighborhoods of the matched ATMS nodes can also be examined for matching with stored statements. For example, the nodes for the stored required fact “The balls have the same vertical position” would be within distance 1 from the set of nodes that matched the student utterance “The balls have the same vertical displacement.” This information can lead to an encouraging feedback to let the student know that she is one inference away from the desired answer.

Formal representations are matched 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 [10], that is particularly fast when one of the graphs to match is small and known in advance, as is the case with all but one of the Matcher blocks shown in Figure 3. In case of the Matcher for the formal representation of the NL input, which is not known in advance, the set of ATMS nodes is known but large. For this case we settle for an approximated evaluation of the match via a suboptimal largest common subgraph.

6. Preliminary evaluation

The Cocoro analyzer is being deployed in an ongoing evaluation of the full Why2-Atlas tutoring system. Figure 4 shows results of classifying 135 student utterances for two physics problems using only direct matching (66 utterances with respect to 46 stored representations and 69 utterances with respect to 44 stored representations). To generate

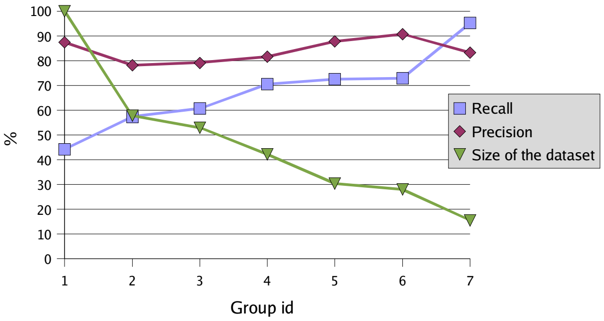

Figure 4. Average recall and precision of utterance classification by Cocoro. The size of a group of entries is shown relative to the size of the overall data set. Average processing time is 0.46 seconds per entry on a 1.8 GHz Pentium 4 machine with 2Gb of RAM.

these results, the data is divided into 7 groups based on the quality of conversion of NL to FOPL, 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 matching algorithm allows classification even of utterances that have mediocre representations, resulting in 70.6% average recall and 81.6% average precision for 42.2% of all entries (group 4). However, large numbers of inadequately represented utterances (at least 47%) result in 44.3% average recall and 87.4% average precision for the whole data set (group 1). Note that Cocoro analyzes only utterances for which *some* representation in FOPL has been generated. Figure 4 does not include data on utterances for which no formal representation has been generated; such utterances are classified relying on a statistical classifier only [8].

At the same time we are investigating the computational feasibility of utilizing the full Cocoro analyzer with ATMS. One of the concerns is that as the depth of the inferencing increases, ATMS size can make real-time matching infeasible. Our results show that an ATMS of depth 3, generated using just 11 physics inference rules, and containing 128 nodes, covers 55% of the relevant problem facts. It takes about 8 seconds to analyze an input representation consisting of 6 atoms using an ATMS of this size, which is a considerable improvement over the time required for the on-the-fly analysis performed by the Tacitus-lite+ abductive reasoner [6]. The knowledge engineering effort needed to increase the coverage is currently under way and involves enriching the rule base.

7. Conclusions

In this paper we described how we alleviate some of the performance and knowledge engineering drawbacks associated with using an on-the-fly abductive reasoner by deploying a precomputed ATMS as a back-end for an analyzer of completeness and correctness of student utterances. Besides the improvement in time response, the ATMS-based analysis provides the additional possibility of evaluating an “inferential neighborhood” of the student’s utterance which we expect to be useful for providing more precise tutoring feedback. The preliminary evaluation provided encouraging results suggesting that we can successfully deploy the ATMS-based reasoner as an NLU back-end of the Why2-Atlas tutoring system.

Acknowledgements

This research has been supported under NSF grant 0325054 and ONR grant N00014-00-1-0600. The authors would like to thank the Natural Language Tutoring group for their work on the Why2-Atlas system, in particular Pamela Jordan, Brian ‘Moses’ Hall, Umarani Pappuswamy, and Michael Ringenberg.

References

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

[2] Arthur C. Graesser, Peter Wiemer-Hastings, Katja Wiemer-Hastings, Derek Harter, Natalie Person, and the TRG. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments*, 8:129–148, 2000.

[3] Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 3220 of *LNCS*, pages 346–357, Maceió, Alagoas, Brazil, 2004. Springer.

[4] Yasuyuki Kono, Takehide Yano, Tetsuro Chino, Kaoru Suzuki, and Hiroshi Kanazawa. Animated interface agent applying ATMS-based multimodal input interpretation. *Applied Artificial Intelligence*, 13(4-5):487–518, 1999.

[5] Maxim Makatchev, Pamela W. Jordan, Umarani Pappuswamy, and Kurt VanLehn. Abductive proofs as models of students’ reasoning about qualitative physics. In *Proceedings of the 18th International Workshop on Qualitative Reasoning*, pages 11–18, Evanston, Illinois, USA, 2004.

[6] Maxim Makatchev, Pamela W. Jordan, and Kurt VanLehn. 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, 2004.

[7] Toyoaki Nishida, Xuemin Liu, Shuji Doshita, and Atsushi Yamada. Maintaining consistency and plausibility in integrated natural language understanding. In *Proceedings of COLING-88*, volume 2, pages 482–487, Budapest, Hungary, 1988.

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

[9] S. Ritter, S. Blessing, and L. Wheeler. User modeling and problem-space representation in the tutor runtime engine. In *Proceedings of the 9th International Conference on User Modelling*, volume 2702 of *LNAI*, pages 333–336. Springer, 2003.

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

[11] Kurt VanLehn, Pamela Jordan, Carolyn Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, Stephanie Siler, and Ramesh Srivastava. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 2363 of *LNCS*, pages 158–167. Springer, 2002.

[12] Kurt VanLehn, Collin Lynch, K. Schultz, Joel Shapiro, R. H. Shelby, Linwood Taylor, D. J. Treacy, Anders Weinstein, and M. C. Wintersgill. The Andes physics tutoring system: Lessons learned (under review). *Unpublished manuscript*.

[13] Uri Zernik and Allen Brown. Default reasoning in natural language processing. In *Proceedings of COLING-88*, volume 2, pages 801–805, Budapest, Hungary, 1988.