Evaluation of the student's input in a Intelligent Tutoring System

Kepa Joseba Rodriguez
HS-Nr: 2028271
kepa@coli.uni-sb.de

Abstract

A common problem for Intelligent Tutoring Systems (ITS) is how to evaluate properly the student's input. In this paper I will try to give an overview about different approaches.

1 Introduction: ITS and free text input

In order to give tailored feedback how to evaluate the student's input is the first problem that an ITS has to address.

The answer of the student to a question of the ITS can have different degrees of correctness. It can be correct, wrong, partialy correct, etc. The input understander of an ITS must be able to distinguish and categorize these different possibilities. Also the ITS must be able to find the reason of the mistakes of the student.

An ITS must be able to handle unconstrained student input. The main reasons are:

- Students can be linguistically productive, even in a very small domain. When the student answers with the correct expected concept, she/he can express it in an unusual way. In the student's input there are often extra words and syntax that do not contribute to the system's understanding. It complicates the parsing.
- Sometimes the student does not know how to express a concept, and she/he tries to give some vague explanations of it. The system must be able to extract the correct partial information. A very common example is the substitution of sense for reference, instead of naming a concept (reference) the student describes it (sense).
- A number of unexpected simple answers stimulate useful tutoring. Free-text input gives the student the opportunity to produce such unexpected answers.
- Student's self explanations increase the didactical efficiency of the tutorial dialogue.

In the next chapters I will give an overview of different ITSs that use different approaches to interpret and evaluate free input.

The first system described is the CIRCSIM-Tutor which uses finite state techniques and looks up the conceptual content of the input in domain ontologies.

A quite different approach is to interpret the student's input without any syntax. The Latent Semantic Analysis (LSA) is a statistically based method to analyze texts. LSA has its origin in Information Retrieval, and it is used by AutoTutor.

There are systems that parse the input into a propositional representation. Here I give descriptions of two systems, which use different approaches. Why-Atlas uses logical abduction to infer the reasoning line of the student, and PACT-Geometry Explanation Tutor uses knowledge-based understanding.

At the end there is a comparison between the different systems and a discussion about the application of their approaches to different kinds of tutorial dialogue.

2 Processing student's Input in CIRCSIM-Tutor ITS

2.1 About CIRCSIM-Tutor

CIRCSIM-Tutor is an ITS for teaching the baroreceptor reflex mechanism of blood preasure control to medical students. The student is required by the ITS to predict the direct effect of seven physiological variables, like central venous pressure, heart rate, etc.

The dialogue is under the tutor's control, who asks questions in a Socratic tutorial style.

In CIRCSIM-Tutor the questions can admit short answers of a limited expectations list. The students usually does not produce longer utterances to answer the ITS's questions.

2.2 Categorization of student's answer

The categorization process categorizes the student's utterance and uses the categorized utterance to derive the correct preconditions in the dialogue planner.

The complete classification of answer recognized by the system is:

1. Correct.
2. Partially correct answer.
3. Near miss answer: The response is not what the ITS expected, but it is close enough to introduce subdialogues to drive the student to the desired answer.
4. "I don't know" answer.
5. "Grain of truth" answer. Incorrect but it indicates a partially correct understanding.
6. Misconception: The system has a catalogue of common misconceptions with corresponding hinting strategies.
7. Other incorrect answer.
8. Mixed answers.

2.3 Low level understander processing

Earlier versions parsed the student's input with a Lexical Functional Grammar (LFG)-based parser. This approach was not satifying because of:

- Sometimes extragrammaticality was not handled properly.
- With the inclusion of a variety of unexpected student answers the range of possible f-structures expanded dramatically.

The latest version uses only a little parser to extract syntactic information from the student's input.

The processing consists of the following steps:

1. **Lexicon lookup:** The lexicon contains many of the standard abbreviations used by students, phrasal entries with morphological and phrasal variantes. Only 5% of all lexical entries have meaning tokens.
2. **Spelling corrections:** Used when the lexicon lookup fails.
3. **Recognition by finite state transducers (FST):** the finite state approach is used to extract only the information that the system needs, ignoring other information.
   In CIRCSIM-Tutor a library of small FSTs is coded for specific kinds of answers.
   Some syntactic information is implemented in the transducer, e.g. to differentiate "is" as copula or as abbreviation IS, to recognize negation, etc.
4. **Lookup of meaning tokens in concept ontologies.**
5. **Matching the question:** Produces a representation of the answer.

3 AutoTutor and LSA

AutoTutor is a ITS that supports mixed-initiative dialogue. The application domains of the system are computer literacy and computer physics.

AutoTutor uses LSA (Latent Semantic Analysis) to evaluate the answer of the student. The main idea of AutoTutor's input understander is that human tutors do not get very deep in the understanding of the responses of the students. Most commonly a shallow understanding of the responses is enough to determine the vicinity of the expected answer.

3.1 The notion of curricular script

The knowledge about the topics to teach is organized in curriculum scripts. Each curriculum script covers one topic. For each topic there is an information delivery item to set up the common ground between the student and the ITS.

The remainder of the curriculum script consists of a set of subtopics. There are four kinds:

1. simple question/answer
2. question/answer with didactic content
3. problem solution
4. picture question/answer

3.2 The Latent Semantic Analysis (LSA)

LSA is a statistical, corpus based natural language understanding technique supporting similarity comparisons between texts.

LSA was developed for information retrieval, and it provides the critical backbone for representing word knowledge in AutoTutor. The main idea of the LSA is that particular words appear in particular texts.

The training of LSA starts with a corpus separated into units, called texts. For AutoTutor corpus curriculum scripts are used where each item is a separate text for training purposes. In the corpus there is also additional information from textbooks and articles.

Each paragraph of this additional information constitutes a text. The paragraph is considered to be at a good level of granularity because a paragraph tends to hold a well-developed coherent idea.

LSA computes a co-occurrence matrix of terms and texts that specifies the number of times the word W_i occurs in the document D_j. The cells of the matrix contain the number of times a particular term occurs in a particular text. A long entropy weighting is performed on this matrix to distinguish when a word occurs in a particular text and when it occurs across texts.

A statistical method called *singular value decomposition* reduces the large $W \times D$ co-occurrence matrix to K dimensions (usually between 100 and 300). Each word, sentence or text is represented as a weighted vector on K dimensions.

The *similarity* between two words, sentences or texts is computed as the cosine between the two vectors (from -1 to 1).

The *truth* of a student's response is evaluated by computing the maximum cosine match between the student's utterance and the entire corpus of related texts.

The *relevance* is evaluated by computing its match with expected answers/solutions to a question/problem.

A vector for the student input is calculated by summing the vectors of the terms included in the contribution. The result is compared with the vectors of the texts of the curriculum script items for the current topic. How good or bad a student contribution is, is calculated by comparing it with a set of good and bad responses in the curriculum script.

Two measures are calculated:

- **Completeness:** The percentage of the aspects of the ideal answer for the current topic which matches the student response.
- **Compatibility:** The percentage of the student response that matches some aspect of the ideal answer.

3.3 Adding syntactic information to LSA

In the initial LSA approach there was no syntax. Some experiments tried to add syntax to improve the performance of the understander.

The way to add the syntactic preprocessing of student's input and expected answer was the following:

1. Syntactic segmentation of the sentences. The sentences were broken in strings which corresponded to:
   - subject: noun phrase
   - verb: including adverbs and adverbial phrases
   - object noun phrase

   This additional information was achieved in the following way:
   - grouping of words which belong together into "components"
   - the "semantic" role of the components as derived from syntactic argument structure
2. Anaphora resolution in the sentences, replacing pronouns by their antecedents.
3. When there was a conjunction the arguments were distributed.

(1) Subject Verb Object1 and Object2 → Subject Verb Object1 and Subject Verb Object2

At the beginning there were three hypotheses of the possible effect of using this additional syntactic information:

1. Component grouping will increase discrimination because it adds information
2. Component grouping will hurt discrimination because LSA works better on longer strings
3. Component grouping will hurt grouping due to some complexity of combining individual component similarity scores.

After a number of experiments there are no results that support the hypothesis 1. The performance of the input understander does not increase, and sometimes it decreases.

4 Using logical abduction The Why-Atlas System

Why-Atlas is an ITS for physics that presents questions and encourages the students to answer them using natural language.

Why-Atlas creates and uses a proof-based representation of the student responses.

The responses that are handled by the system are short essays with an average of 6.9 sentences to explain the question of the instructor. The Why-Atlas system takes the propositions of this response and constructs abductive proofs of them. The abductive proofs use a backward inferencing from consequent to antecedent. It gives intuitions about the line of reasoning of the student across multiple sentences, because proofs of the propositions share subproofs. That means that one proposition's entire proof may be a subproof of the next proposition.

4.1 Overview of the Why-Atlas ITS

As the student introduces a response or explanation a sentence-level understanding module builds sets of propositions and sends it to the discourse-level understanding module. Each set of propositions represents one interpretation of a sentence.

Using language and domain reasoning axioms the discourse-level understanding module builds a set of proofs that offer an explanation for the student's essay and give some intuition into how the student applies his/her knowledge. In this module the propositions and search queue for proofs are updated in the history with the results from the theorem prover Tacitus-lite+.

The tutorial strategist modul identifies the relevant comunicative goals and classifies them in two groups of categories:

a. The first group is formed by categories that disambiguate terminology and carify the essay.
b. In the second group there are comunicative goals that repare misconceptions and elicite more complete explanations.

Misconceptions are detected when the proof includes an axiom that is incorrect or inapplicable.

Incompletness is detected under two conditions:

1. If there are multiple proofs that are equally good. That indicates that the essay of the student did not give enough information.
2. If the student fails to explicitly state a proposition that the system requires of any acceptable complete essay.

4.2 Weighted abduction in Tacitus-lite+

In weighted abduction there is a cost associated with making an assumption during the inference process. Tacitus-lite is a collection of axioms, and each axiom is stated as a Horn clause. Each term of the Horn clause p_i has a weight w_i associated. The weight is useful in order to calculate the cost of the assumption of p_i, where $cost(p_i) = cost(r) * w_i$, r is the conclusion and the Horn-Clause is:

$$p_1 w_1 \wedge p_2 w_2 \wedge ... \wedge p_n w_n \Rightarrow r$$

Given a goal to be proven, Tacitus-lite+ takes one of these actions:

1. assumes the goal at the cost associated with it
2. unifies with a fact for zero cost
3. unifies with a literal that has already been assumed or proven at no additional cost
4. attempts to prove it with an axiom

Tacitus-lite+ maintains a queue of proofs where the first reflects the assumption of all the observations and each of the above given actions adds a new proof into the queue.

Tacitus-lite+ uses a best-first search guided by heuristics to select which proof to expand, which action to apply, etc.

Why-Atlas uses Tacitus-lite+ for shallow qualitative physics reasoning and discourse-level language reasoning.

4.3 Phenomena handled by the system

4.3.1 Pronominal anaphora

The System is able to resolve simple intrasentential pronominal references with language axioms.

To resolve intersentential anaphora the system relies on the domain axioms, which will bind the body variables to their most likely referents.

If the referent is a physical quantity such as velocity or force the language axioms must be extended to handle these cases because it can involve a lot of unconstrained search for the domain axioms to resolve these.

4.3.2 Temporal anaphora

If the student indicates when an event or state occurs relative to another one, the domain-level reasoning will be able to unify the anchor event or state with an already known event or state in the proof it is constructing.

When there is no temporal anchor the world knowledge is used to infer the temporal relations.

Otherwise the language axioms use information about tense and aspect and default orderings relative to these to guide inferences about temporal relations.

4.3.3 Embedded relationships

In the physics domain there is a tendency to express multiple relations within a single sentence.

(2) When the magnitude of the punpkin's velocity equals the man's, the punpkin's velocity is in the opposite direction.

Here the "equal" and "opposite" relations are embedded in a temporal "when" relation. The sentence-level understandig module cannot indicate the specific constraints of each of these relations. This would be handled by the discourse-level understanding module that increases the load on the domain-level proof building.

4.3.4 Quantifiers and plurals

In the system there is a specification for a sentence-level representation of quantifiers and plurals. From this representation the language axioms will generate an appropiate number of new propositions to build the proofs, given the context of the problem and the expression recognized from sentence-level processing. This is still in project.

5 ITS with knowledge-based understanding

The PACT-Geometry Explanation Tutor takes a knowledge-based approach to natural language understanding and uses a statistical text classification as a backup. The main features are: robust parsing, logic-based representation of semantic content, representation of pedagogical content knowledge in the form of a hierarchy of partial and complete explanations and reactive dialogue management.

5.1 Fundaments

From the analysis of several corpora of written explanations of geometry theorems follows:

- There are a lot of ways to express correctly the same geometry theorem.
- Students fail in the explanation more often because they give rather incomplete than wrong explanations.
- Students state often the arithmetic operations by which they found the result.
- Students have the tendency to abbreviate in ways that violate the semantic constraints.
- Students use language that sometimes is ill-formed.

From this analysis follows that an ITS at the explanation-level must have a robust language understanding component that can deal with ill-formed explanations and can respond intelligently to the types of incomplete explanations that students produce.

Because of this reason the PACT-Geometry Explanation Tutor choses a knowledge-based approach to language understanding having a cognitive model that represents the skills of an ideal student in the given domain, represented in the form of production rules. It was expected that in this form more detailed and accurate analyses of the student input can be produced.

5.2 Analysis of student explanations

The student input is parsed using a unification-based chart-parser. The grammar uses feature structure unification.

A classifier is used to test the coherence of possible semantic representations with respect to the (system's hierarchy of explanation categories) semantic constraints of the system's domain ontology. This classifier uses Loom, which is a tool for building ontologies.

6 Comparison between the systems

The CIRCSIM-Tutor ITS takes syntactic information from the student's input with the help of a small parsing component and a keyword lookup. But its input understander is very hard-coded in a limited model, and it is no easy to scale it up.

Large-scale systems based on FST are built and trained automatically, but their training requires enormous quantities of the same kind of texts that are to be processed. There are not enough large corpora of student's answers to computer tutor's questions

The Latent Semantic Analysis used by AutoTutor is more robust and less brittle than CIRCSIM-Tutor's approach. The training of the system was easier than in the case of CIRCSIM-Tutor. The ITS was trained with written textbooks and articles, and learns the semantics latent in vocabulary co-occurrence. The fact that LSA works shows that students seem to use many of the words experts use.

LSA is able to process large texts, and its ability correlates positively with the length of the input. However it has difficulties to evaluate short input.

The main problem of the statistical approach is that it is not able to distinguish between good and bad explanations. It is insensible to negations, anaphoras (temporal and pronominal references), argument ordering variations. Its inferencing is weak.

The two last systems Why-Atlas and PACT-Geometry Explanation Tutor parse the student input into a propositional representation. PACT-Geometry tutor uses a knowledge base to classify the propositions into fine-grainded categories. Why-Atlas construct abductive proofs of the propositions.

The second approach can give more information about the student's reasoning. The first seems to be more efficient in eliciting self-explanations from students.

7 A little discussion

Tutorial dialogue is a very open concept that includes a vaste amount of different kinds of dialogue. There are tutorials where a short answer of the student is expected, while in other cases that the student has the tendency to give an entire essay as answer.

In restricted domains where the amount of possible answers is not very large and the students tend to give short answers a FST approach to parse the student's input and a lookup in ontologies seems to be enough.

In domains where the student is required to give long answers to the tutor's questions a statistically based approach like LSA could be a good alternative, because the statistically based approach are robust to handle whith large essays.

The systems that parse the student's contribution into a propositional representation (Why-Atlas and PACT-Geometry Tutor) are independent of the length of the input. They are more robust than the FST based approach.

Contrary to the statistical approaches like LSA Why-Atlas and PACT-Geometry Tutor are sensible to the phenomena that LSA cannot handle, like anaphora or negation.

A possible problem of Why-Atlas and PACT-Geometry Tutor is that the parser can give different possible propositional representations for an input. A kind of heuristics should be used that helps to decide which of these representation is the correct one, but in the documentation of the systems there are no references to such techniques.

It is worth to briefly look at a different domain of tutoring and see how the above methods of student answer evaluation serve that domain. The mathematics domain has the advantage that theorem provers can be used for dynamic planning.

The possible features of the student's input in mathematics domain are:

- Short answers are expected. Often one or two words.
- The range of possible answers is relatively small.
- The correct inpretation of negation or anaphora has a high relevance.

The LSA approach is not suitable to evaluate the student input in the mathematics domain because the potential answers would be too short. A second problem is the insensibility of statistical approaches to recognize negation, anaphora and other linguistic phenomena. This is not enough to handle the precision of the matematical language.

The approach used in Why-Atlas seems to be problematic as well. The reason is that it uses logical abduction, and mathematics and especially the set theory uses logical reasoning. In fact one of the main sources of students mistakes is that the students tend to use abduction.

A possible adaptation of this system is to use abduction only to detect misconceptions and to detect when the students themselves are using abduction, but not to evaluate the content of the input.

The approaches used in the other systems, CIRCSIM-Tutor and PACT-Geometry Tutor, seem to be suitable for the mathematics domain. But there is not a corpus to train FST.

8 Literature

Vicent Aleven, Octav Popescu and Kenneth R. Koedinger. 2001. *A tutorial dialogue system with knowledge-based understanding and classification of student explanations.* In Working Notes of 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems. 2001. Seattle. USA.

Michael Glass. 2001. *Processing Language Input in the* CIRCSIM*-tutor Intelligente Tutoring System.* Artificial Intelligence in Education. J.D.Moore et all eds. IOS Press, pp 210-211. Amsterdam.

Atrhur Graesser, Ashish Karnavat, Victoria Pomeroy and Katja Wiemer-Hastings. 2000 *Latent Semantic Analysis captures causal, goal-oriented and taxonomic structures.* COGSCI-2000. Philadelphia. USA.

Pamela Jordan and Kurt VanLehn.2002 .*Discourse Processing for Explanatory Essays in Tutorial Applications.* Procedings of the 3rd SIGdial Workshop on Discourse and Dialogue. Philadelphia. USA.

Peter Wiemer-Hastings. 2000.*Adding syntactic information to LSA*. Proceedings of the Twenty-second Annual Conference of the Cognitive Science Society. Philadelphia. USA.

Peter Wiemer-Hastings, Arthur C. Graesser, Derek Harter, and the Tutoring Research Group. 1998. *The fundations and architechture of AutoTutor.* Proceedingsof the 4th International Conference on Intelligent Tutoring Systems. pp 334-343. Springer Verlag. Berlin. BRD.

Peter Wiemer-Hastings, Katja Wiemer-Hastings, Arthur C. Graesser. 2000. *Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis.* Artificial Intelligence in Education. pp 535-542. IOS Press. Amsterdam.