Improving interpretation robustness in a tutorial dialogue system

Myroslava O. Dzikovska and **Elaine Farrow** and **Johanna D. Moore**
School of Informatics, University of Edinburgh
Edinburgh, EH8 9AB, United Kingdom
{m.dzikovska,elaine.farrow,j.moore}@ed.ac.uk

Abstract

We present an experiment aimed at improving interpretation robustness of a tutorial dialogue system that relies on detailed semantic interpretation and dynamic natural language feedback generation. We show that we can improve overall interpretation quality by combining the output of a semantic interpreter with that of a statistical classifier trained on the subset of student utterances where semantic interpretation fails. This improves on a previous result which used a similar approach but trained the classifier on a substantially larger data set containing all student utterances. Finally, we discuss how the labels from the statistical classifier can be integrated effectively with the dialogue system's existing error recovery policies.

1 Introduction

Giving students formative feedback as they interact with educational applications, such as simulated training environments, problem-solving tutors, serious games, and exploratory learning environments, is known to be important for effective learning (Shute, 2008). Suitable feedback can include context-appropriate confirmations, hints, and suggestions to help students refine their answers and increase their understanding of the subject. Providing this type of feedback automatically, in natural language, is the goal of tutorial dialogue systems (Aleven et al., 2002; Dzikovska et al., 2010b; Graesser et al., 1999; Jordan et al., 2006; Litman and Silliman, 2004; Khuwaja et al., 1994; Pon-Barry et al., 2004; VanLehn et al., 2007).

Much work in NLP for educational applications has focused on automated answer grading (Leacock and Chodorow, 2003; Pulman and Sukkarieh, 2005; Mohler et al., 2011). Automated answer assessment systems are commonly trained on large text corpora. They compare the text of a student answer with the text of one or more reference answers supplied by human instructors and calculate a score reflecting the quality of the match. Automated grading methods are integrated into intelligent tutoring systems (ITS) by having system developers anticipate both correct and incorrect responses to each question, with the system choosing the best match (Graesser et al., 1999; Jordan et al., 2006; Litman and Silliman, 2004; VanLehn et al., 2007). Such systems have wide domain coverage and are robust to ill-formed input. However, as matching relies on shallow features and does not provide semantic representations of student answers, this approach is less suitable for dynamically generating adaptive natural language feedback (Dzikovska et al., 2013).

Real-time simulations and serious games are commonly used in STEM learning environments to increase student engagement and support exploratory learning (Rutten et al., 2012; Mayo, 2007). Natural language dialogue can help improve learning in such systems by asking students to explain their reasoning, either directly during interaction, or during post-problem reflection (Aleven et al., 2002; Pon-Barry et al., 2004; Dzikovska et al., 2010b). Interpretation of student answers in such systems needs to be grounded in the current state of a dynamically changing environment, and feedback may also be generated dynamically to reflect the changing system state. This is typically achieved by employing hand-crafted parsers and semantic interpreters to produce structured semantic representations of student input, which are then used to instantiate ab-

Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 293–299,
Atlanta, Georgia, June 13 2013.

stract tutorial strategies with the help of a natural language generation system (Freedman, 2000; Clark et al., 2005; Dzikovska et al., 2010b).

Rule-based semantic interpreters are known to suffer from robustness and coverage problems, failing to interpret out-of-grammar student utterances. In the event of an interpretation failure, most systems have little information on which to base a feedback decision and typically respond by asking the student to rephrase, or simply give away the answer (though more sophisticated strategies are sometimes possible, see Section 4). While statistical scoring approaches are more robust, they may still suffer from coverage issues when system designers fail to anticipate the full range of expected student answers. In one study of a statistical system, a human judge labeled 33% of student utterances as not matching any of the anticipated responses, meaning that the system had no information to use as a basis for choosing the next action and fell back on a single strategy, giving away the answer (Jordan et al., 2009).

Recently, Dzikovska et al. (2012b) developed an annotated corpus of student responses (henceforth, the SRA corpus) with the goal of facilitating dynamic generation of tutorial feedback.[1] Student responses are assigned to one of 5 domain- and task-independent classes that correspond to typical flaws found in student answers. These classes can be used to help a system choose a feedback strategy based only on the student answer and a single reference answer. Dzikovska et al. (2013) showed that a statistical classifier trained on this data set can be used in combination with a semantic interpreter to significantly improve the overall quality of natural language interpretation in a dialogue-based ITS. The best results were obtained by using the classifier to label the utterances that the semantic interpreter failed to process.

In this paper we further extend this result by showing that we can obtain similar results by training the classifier directly on the subset of utterances that cannot be processed by the interpreter. The distribution of labels across the classes is different in this subset compared to the rest of the corpus. Therefore we can train a subset-specific classifier, reducing the amount of annotated training data needed without compromising performance of the combined system.

The rest of the paper is organized as follows. In Section 2 we describe an architecture for combining semantic interpretation and classification in a system with dynamic natural language feedback generation. In Section 3 we describe an experiment to improve combined system performance using a classifier trained only on non-interpretable utterances. We discuss future improvements in Section 4.

2 Background

The SRA corpus is made up of two subsets: (1) the SciEntsBank subset, consisting of written responses to assessment questions (Nielsen et al., 2008b), and (2) the Beetle subset consisting of utterances collected from student interactions with the BEETLE II tutorial dialogue system (Dzikovska et al., 2010b). The SRA corpus annotation scheme defines 5 classes of student answers ("correct", "partially-correct-incomplete", "contradictory", "irrelevant" and "non-domain"). Each utterance is assigned to one of the 5 classes based on pre-existing manual annotations (Dzikovska et al., 2012b).

We focus on the Beetle subset because the Beetle data comes from an implemented system, meaning that we also have access to the semantic interpretations of student utterances produced by the BEETLE II interpretation component. The system uses fine-grained semantic analysis to produce detailed diagnoses of student answers in terms of correct, incorrect, missing and irrelevant parts. We developed a set of rules to map these diagnoses onto the SRA corpus 5-class annotation scheme to support system evaluation (Dzikovska et al., 2012a).

In our previous work (Dzikovska et al., 2013), we used this mapping as the basis for combining the output of the BEETLE II semantic interpreter with the output of a statistical classifier, using a rule-based policy to determine which label to use for each instance. If the label from the semantic interpreter is chosen, then the full range of detailed feedback strategies can be used, based on the corresponding semantic representation. If the classifier's label is chosen, then the system can fall back to using content-free prompts, choosing an appropriate

[1] http://www.cs.york.ac.uk/semeval-2013/task7/index.php?id=data

prompt based on the SRA corpus label.

We evaluated 3 rule-based combination policies, chosen to reduce the effects of the errors that the semantic interpreter makes, and taking into account tutoring goals such as reducing student frustration. The best performing policy takes the classifier's output if and only if the semantic interpreter is unable to process the utterance.[2] This allows the system to choose from a wider set of content-free prompts instead of always telling the student that the utterance was not understood.

As discussed earlier, non-interpretable utterances present a problem for both rule-based and statistical approaches. Therefore, we carried out an additional set of experiments, focusing on the performance of system combinations that use policies designed to address non-interpretable utterances. We discuss our results and future directions in the rest of the paper.

3 Improving Interpretation Robustness

3.1 Experimental Setup

The Beetle portion of the SRA corpus contains 3941 unique student answers to 47 different explanation questions. Each question is associated with one or more reference answers provided by expert tutors, and each student answer is manually annotated with the label assigned by the BEETLE II interpreter and a gold-standard correctness label.

In our experiments, we follow the procedure described in (Dzikovska et al., 2013), using 10-fold cross-validation to evaluate the performance of the various stand-alone and combined systems. We report the per-class F_1 scores as evaluation metrics, using the macro-averaged F_1 score as the primary evaluation metric.

Dzikovska et al. (2013) used a statistical classifier based on lexical overlap, taken from (Dzikovska et al., 2012a), and evaluated 3 different rule-based policies for combining its output with that of the semantic interpreter. In two of those policies the interpreter's output is always used if it is available, and the classifier's label is used for a (subset of) non-interpretable utterances:

1. `NoReject`: the classifier's label is used in all cases where semantic interpretation fails, thus creating a system that never rejects student input as non-interpretable

2. `NoRejectCorrect`: the classifier's label is used for non-interpretable utterances which are labeled as "correct" by the classifier. This more conservative policy aims to ensure that correct student answers are always accepted, but incorrect answers may still be rejected with a request to rephrase.

We conducted a new experiment to evaluate these two policies together with an enhanced classifier, discussed in the next section.

3.2 Classifier

For this paper, we extended the classifier from the previous study (Dzikovska et al., 2013), which we will call `Sim8`, with additional features to improve handling of lexical variability and negation.

`Sim8` uses the Weka 3.6.2 implementation of C4.5 pruned decision trees, with default parameters. It uses 8 features based on lexical overlap similarity metrics provided by Perl's `Text::Similarity` package v.0.09: 4 metrics measuring overlap between the student answer and the expected answer, and the same 4 metrics applied to the student's answer and the question text.

In our enhanced classifier, `Sim20`, we extended the baseline feature set with 12 additional features. 8 of these are direct analogs of the baseline features, this time computed on the stemmed text to reduce the impact of syntactic variation, using the Porter stemmer from the `Lingua::Stem` package.[3] In addition, 4 features were added to improve negation handling and thus detection of contradictions. These are:

- `QuestionNeg, AnswerNeg`: features indicating the presence of a negation marker in the question and the student's answer respectively, detected using a regular expression. We distinguish three cases: a negation marker

[2] We will refer to such utterances as "non-interpretable" following (Bohus and Rudnicky, 2005).

[3] We also experimented with features that involve removing stop words before computing similarity scores, and with using SVMs for classification, but failed to obtain better performance. We continue to investigate different SVM kernels and alternative classification algorithms such as random forests for our future work.

	Standalone			Sem. Interp. + Sim20		Sem. Interp. + Sim20NI	
	Sem. Interp.	Sim8	Sim20	no_rej	no_rej_corr	no_rej	no_rej_corr
correct	0.66	0.71	0.71	0.70	0.70	0.70	0.70
pc_inc	0.48	0.38	0.40	0.51	0.48	0.50	0.48
contra	0.27	0.40	0.45	0.47	0.27	0.51	0.27
irrlvnt	0.21	0.05	0.08	0.22	0.21	0.22	0.21
nondom	0.65	0.73	0.78	0.83	0.65	0.83	0.65
macro avg	0.45	0.45	0.48	**0.55**	0.46	**0.55**	0.46

Table 1: F_1 scores for three stand-alone systems, and for combination systems using the Sim20 and Sim20NI classifiers together with the semantic interpreter. Stand-alone performance for Sim20NI is not shown since it was trained only on the non-interpretable data subset and is therefore not applicable for the complete data set.

likely to be associated with domain content (e.g., "not connected"); a negation marker more likely to be associated with general expressions of confusion (such as "don't know"); and no negation marker present.

- BestOverlapNeg: true if the reference answer that has the highest F_1 overlap with the student answer includes a negation marker.
- BestOverlapPolarityMatch: a flag computed from the values of AnswerNeg and BestOverlapNeg. Again, we distinguish three cases: they have the same polarity (both the student answer and the reference answer contain negation markers, or both have no negation markers); they have opposite polarity; or the student answer contains a negation marker associated with an expression of confusion, as described above.

3.3 Evaluation

Evaluation results are shown in Table 1. Unless otherwise specified, all performance differences discussed in the text are significant on an approximate randomization significance test with 10,000 iterations (Yeh, 2000).

Adding the new features to create the Sim20 classifier resulted in a performance improvement compared to the Sim8 classifier, raising macro-averaged F_1 from 0.45 to 0.48, with an improvement in contradiction detection as intended. But these improvements did not translate into improvements in the combined systems. Combinations using Sim20 performed exactly the same as the combinations using Sim8 (not shown due to space limitations, see (Dzikovska et al., 2013)). Clearly, more sophisticated features are needed to obtain further performance gains in the combined systems.

However, we noted that the subset of non-interpretable utterances in the corpus has a different distribution of labels compared to the full data set. In the complete data set, 1665 utterances (42%) are labeled as correct and 1049 (27%) as contradictory. Among the 1416 utterances considered non-interpretable by the semantic interpreter, 371 (26%) belong to the "correct" class, and 598 (42%) to "contradictory" (other classes have similar distributions in both subsets). We therefore hypothesized that a combination system that uses the classifier output only if an utterance is non-interpretable, may benefit from employing a classifier trained specifically on this subset rather than on the whole data set.

If our hypothesis is true, it offers an interesting possibility for combining rule-based and statistical classifiers in similar setups: if the classifier can be trained using only the examples that are problematic for the rule-based system, it can provide improved robustness at a significantly lower annotation cost.

We therefore trained another classifier, Sim20NI, using the same feature set as Sim20, but this time using only the instances rejected as non-interpretable by the semantic interpreter in each cross-validation fold (1416 utterances, 36% of all data instances). We again used the NoReject and NoRejectCorrect policies to combine the output of Sim20NI with that of the semantic interpreter. Evaluation results confirmed our hypothesis. The system combinations that use Sim20 and Sim20NI perform identically on

macro-averaged F_1, with `NoReject` being the best combination policy in both cases and significantly outperforming the semantic interpreter alone. However, the `Sim20NI` classifier has the advantage of needing significantly less annotated data to achieve this performance.

4 Discussion and Future Work

Our research focuses on combining deep and shallow processing by supplementing fine-grained semantic interpretations from a rule-based system with more coarse-grained classification labels. Alternatively, we could try to learn structured semantic representations from annotated text (Zettlemoyer and Collins, 2005; Wong and Mooney, 2007; Kwiatkowski et al., 2010), or to learn more fine-grained assessment labels (Nielsen et al., 2008a). However, such approaches require substantially larger annotation effort. Therefore, we believe it is worth exploring the use of the simpler 5-label annotation scheme from the SRA corpus. We previously showed that it is possible to improve system performance by combining the output of a symbolic interpreter with that of a statistical classifier (Dzikovska et al., 2013). The best combination policy used the statistical classifier to label utterances rejected as non-interpretable by the rule-based interpreter.

In this paper, we showed that similar results can be achieved by training the classifier only on non-interpretable utterances, rather than on the whole labeled corpus. The student answers that the interpreter has difficulty with have a distinct distribution, which is effectively utilized by training a classifier only on this subset. This reduces the amount of annotated training data needed, reducing the amount of manual labor required.

In future, we will further investigate the best combination of parsing and statistical classification in systems that offer sophisticated error recovery policies for non-understandings. Our top-performing policy, `NoReject`, uses deep parsing and semantic interpretation to produce a detailed semantic analysis for the majority of utterances, and falls back on a shallower statistical classifier for utterances that are difficult for the interpreter. This policy assumes that it is always better to use a content-free prompt than to reject a non-interpretable student utterance. However, interpretation problems can arise from incorrect uses of terminology, and learning to speak in the language of the domain has been positively correlated with learning outcomes (Steinhauser et al., 2011). Therefore, rejecting some non-interpretable answers as incorrect could be a valid tutoring strategy (Sagae et al., 2010; Dzikovska et al., 2010a).

The BEETLE II system offers several error recovery strategies intended to help students phrase their answers in more acceptable ways by giving a targeted help message, e.g., "I am sorry, I'm having trouble understanding. Paths cannot be broken, only components can be broken" (Dzikovska et al., 2010a). Therefore, it may be worthwhile to consider other combination policies. We evaluated the `NoRejectCorrect` policy, which uses the statistical classifier to identify correct answers rejected by the semantic interpreter and asks for rephrasings in other cases. Using this policy resulted in only a small improvement in system performance. A different classifier geared towards more accurate identification of correct answers may help, and we are planning to investigate this option in the future.

Alternatively, we could consider a combination policy which looks for rejected answers that the classifier identifies as contradictory and changes the wording of the targeted help message to indicate that the student may have made a mistake, instead of apologizing for the misunderstanding. This has the potential to help students learn correct terminology rather than presenting the issue as strictly an interpretation failure.

Ultimately, all combination policies must be tested with users to ensure that improved robustness translates into improved system effectiveness. We have previously studied the effectiveness of our targeted help strategies with respect to improving learning outcomes (Dzikovska et al., 2010a). A similar study is required to evaluate our combination strategies.

Acknowledgments

We thank Natalie Steinhauser, Gwendolyn Campbell, Charlie Scott, Simon Caine and Sarah Denhe for help with data collection and preparation. The research reported here was supported by the US ONR award N000141010085.

References

Vincent Aleven, Octav Popescu, and Kenneth R. Koedinger. 2002. Pilot-testing a tutorial dialogue system that supports self-explanation. In *Proc. of ITS-02 conference*, pages 344–354.

Dan Bohus and Alexander Rudnicky. 2005. Sorry, I didn't catch that! - An investigation of non-understanding errors and recovery strategies. In *Proceedings of SIGdial-2005*, Lisbon, Portugal.

Brady Clark, Oliver Lemon, Alexander Gruenstein, ElizabethOwen Bratt, John Fry, Stanley Peters, Heather Pon-Barry, Karl Schultz, Zack Thomsen-Gray, and Pucktada Treeratpituk. 2005. A general purpose architecture for intelligent tutoring systems. In JanC.J. Kuppevelt, Laila Dybkjr, and NielsOle Bernsen, editors, *Advances in Natural Multimodal Dialogue Systems*, volume 30 of *Text, Speech and Language Technology*, pages 287–305. Springer Netherlands.

Myroslava O. Dzikovska, Johanna D. Moore, Natalie Steinhauser, and Gwendolyn Campbell. 2010a. The impact of interpretation problems on tutorial dialogue. In *Proc. of ACL 2010 Conference Short Papers*, pages 43–48.

Myroslava O. Dzikovska, Johanna D. Moore, Natalie Steinhauser, Gwendolyn Campbell, Elaine Farrow, and Charles B. Callaway. 2010b. Beetle II: a system for tutoring and computational linguistics experimentation. In *Proc. of ACL 2010 System Demonstrations*, pages 13–18.

Myroslava O. Dzikovska, Peter Bell, Amy Isard, and Johanna D. Moore. 2012a. Evaluating language understanding accuracy with respect to objective outcomes in a dialogue system. In *Proc. of EACL-12 Conference*, pages 471–481.

Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. 2012b. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proc. of 2012 Conference of NAACL: Human Language Technologies*, pages 200–210.

Myroslava O. Dzikovska, Elaine Farrow, and Johanna D. Moore. 2013. Combining semantic interpretation and statistical classification for improved explanation processing in a tutorial dialogue system. In *In Proceedings of the The 16th International Conference on Artificial Intelligence in Education (AIED 2013)*, Memphis, TN, USA, July.

Reva Freedman. 2000. Using a reactive planner as the basis for a dialogue agent. In *Proceedings of the Thirteenth Florida Artificial Intelligence Research Symposium (FLAIRS 2000)*, pages 203–208.

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

Pamela Jordan, Maxim Makatchev, Umarani Pappuswamy, Kurt VanLehn, and Patricia Albacete. 2006. A natural language tutorial dialogue system for physics. In *Proc. of 19th Intl. FLAIRS conference*, pages 521–527.

Pamela Jordan, Diane Litman, Michael Lipschultz, and Joanna Drummond. 2009. Evidence of misunderstandings in tutorial dialogue and their impact on learning. In *Proc. of 14th International Conference on Artificial Intelligence in Education*, pages 125–132.

Ramzan A. Khuwaja, Martha W. Evens, Joel A. Michael, and Allen A. Rovick. 1994. Architecture of CIRCSIM-tutor (v.3): A smart cardiovascular physiology tutor. In *Proc. of 7th Annual IEEE Computer-Based Medical Systems Symposium*.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proc. of EMNLP-2010 Conference*, pages 1223–1233.

Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405.

Diane J. Litman and Scott Silliman. 2004. ITSPOKE: an intelligent tutoring spoken dialogue system. In *Demonstration Papers at HLT-NAACL 2004*, pages 5–8, Boston, Massachusetts.

Merrilea J. Mayo. 2007. Games for science and engineering education. *Commun. ACM*, 50(7):30–35, July.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA, June. Association for Computational Linguistics.

Rodney D. Nielsen, Wayne Ward, and James H. Martin. 2008a. Learning to assess low-level conceptual understanding. In *Proc. of 21st Intl. FLAIRS Conference*, pages 427–432.

Rodney D. Nielsen, Wayne Ward, James H. Martin, and Martha Palmer. 2008b. Annotating students' understanding of science concepts. In *Proceedings of the Sixth International Language Resources and Evaluation Conference, (LREC08)*, Marrakech, Morocco.

Heather Pon-Barry, Brady Clark, Karl Schultz, Elizabeth Owen Bratt, and Stanley Peters. 2004. Advantages of spoken language interaction in dialogue-based intelligent tutoring systems. In *Proc. of ITS-2004 Conference*, pages 390–400.

Stephen G Pulman and Jana Z Sukkarieh. 2005. Automatic short answer marking. In *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, pages 9–16, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Nico Rutten, Wouter R. van Joolingen, and Jan T. van der Veen. 2012. The learning effects of computer simulations in science education. *Computers and Education*, 58(1):136 – 153.

Alicia Sagae, W. Lewis Johnson, and Stephen Bodnar. 2010. Validation of a dialog system for language learners. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '10, pages 241–244, Stroudsburg, PA, USA. Association for Computational Linguistics.

Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research*, 78(1):153–189.

Natalie B. Steinhauser, Gwendolyn E. Campbell, Leanne S. Taylor, Simon Caine, Charlie Scott, Myroslava O. Dzikovska, and Johanna D. Moore. 2011. Talk like an electrician: Student dialogue mimicking behavior in an intelligent tutoring system. In *Proc. of 15th international conference on Artificial Intelligence in Education*, pages 361–368.

Kurt VanLehn, Pamela Jordan, and Diane Litman. 2007. Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In *Proc. of SLaTE Workshop on Speech and Language Technology in Education*, Farmington, PA, October.

Yuk Wah Wong and Raymond J. Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, Prague, Czech Republic, June.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th International Conference on Computational linguistics (COLING 2000)*, pages 947–953, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 658–666, Arlington, Virginia. AUAI Press.