Towards Effective Tutorial Feedback for Explanation Questions: A Dataset and Baselines

Myroslava O. Dzikovska[*] and **Rodney D. Nielsen**[†] and **Chris Brew**[‡]
[*]School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, UK
[†]Computational Language & Education Research Center
University of Colorado at Boulder, Boulder, CO 80309-0594, USA
[‡]Educational Testing Service, Princeton, NJ 08451, USA
m.dzikovska@ed.ac.uk, rodney.nielsen@colorado.edu, cbrew@ets.org

Abstract

We propose a new shared task on grading student answers with the goal of enabling well-targeted and flexible feedback in a tutorial dialogue setting. We provide an annotated corpus designed for the purpose, a precise specification for a prediction task and an associated evaluation methodology. The task is feasible but non-trivial, which is demonstrated by creating and comparing three alternative baseline systems. We believe that this corpus will be of interest to the researchers working in textual entailment and will stimulate new developments both in natural language processing in tutorial dialogue systems and textual entailment, contradiction detection and other techniques of interest for a variety of computational linguistics tasks.

1 Introduction

In human-human tutoring, it is an effective strategy to ask students to explain instructional material in their own words. Student self-explanation and large amounts of contentful talk focused on the domain are correlated with better learning outcomes (Chi et al., 1994; Litman et al., 2009). There has therefore been much interest in developing automated tutorial dialogue systems that ask students open-ended explanation questions (Graesser et al., 1999; Aleven et al., 2001; Jordan et al., 2006; VanLehn et al., 2007; Nielsen et al., 2009; Dzikovska et al., 2010a). In order to do this well, it is not enough to simply ask the initiating question, because students need the experience of engaging in meaningful dialogue about the instructional content. Thus, systems must respond appropriately to student explanations, and must provide detailed, flexible and appropriate feedback (Aleven et al., 2002; Jordan et al., 2004).

In simple domains, we can adopt a knowledge engineering approach and build a domain model and a diagnoser, together with a natural language parser to produce detailed semantic representations of student input (Glass, 2000; Aleven et al., 2002; Pon-Barry et al., 2004; Callaway et al., 2006; Dzikovska et al., 2010a). The advantage of this approach is that it allows for flexible adaptation of feedback to a variety of factors such as student performance. For example, it is easy for the system to know if the student made the same error before, and adjust its feedback to reflect it. Moreover, this approach allows for easy addition of new exercises within the domain: as long as an exercise relies on the concepts covered by the domain model, the system can apply standard instructional strategies to each new question automatically. However, this approach is significantly limited by the requirement that the domain be small enough to allow comprehensive knowledge engineering, and it is very labor-intensive even for small domains.

Alternatively, we can adopt a data-driven approach, asking human tutors to anticipate in advance a range of possible correct and incorrect answers, and associating each answer with an appropriate remediation (Graesser et al., 1999; Jordan et al., 2004; VanLehn et al., 2007). The great advantage of this approach is that it allows more complex and interesting domains and provides a good framework for eliciting the necessary information from the human experts. A weakness of this approach, which also arises in content-scoring applications such as ETS's c-rater (Leacock and Chodorow, 2003), is that human experts find it extremely difficult to predict with any certainty what the full range of student responses will be. This leads to a lack of adaptivity and generality – if the system designers have failed to predict the full range of possibilities, students will often receive the default feedback. It is frustrating and confusing for students to repeatedly receive the same feedback, regardless of their past performance or dialogue context (Jordan, 2004).

Our goal is to address the weaknesses of the data-driven approach by creating a framework for supporting more flexible and systematic feedback. Our approach identifies general classes of error, such as omissions, incorrect statements and off-topic statements, then aims to develop general remediation strategies for each error type. This has the potential to free system designers from the need to pre-author separate remediations for each individual question. A precondition for the success of this approach is that the system be able to identify error types based on the student response and the model answers.

A contribution of this paper is to provide a new dataset that will enable researchers to develop classifiers specifically for this purpose. The hope is that with an appropriate dataset the data-driven approach will be flexible and responsive enough to maintain student engagement. We provide a corpus that is labeled for a set of five student response types, develop a precise definition of the corresponding supervised classification task, and report results for a variety of simple baseline classifiers. This will provide a basis for the development, comparison and evaluation of alternative approaches to the error classification task. We believe that the natural language capabilities needed for this task will be directly applicable to a far wider range of tasks in educational assessment, information extraction and computational semantics. This dataset is publicly available and will be used in a community-wide shared task.

2 Corpus

The data set we developed draws on two established sources – a data set collected and annotated during an evaluation of the BEETLE II tutorial dialogue system (Dzikovska et al., 2010a) (henceforth, BEETLE corpus) and a set of student answers to questions from 16 science modules in the Assessing Science Knowledge (ASK) assessment inventory (Lawrence Hall of Science, 2006) (henceforth, the Science Entailments Bank or SCIENTSBANK).

In both corpora, each question was associated with one or more reference answers provided by the experts. Student answers were evaluated against these reference answers and assigned correctness labels using corpus-specific annotation schemes. We derived a new set of annotations, shown in Figure 1, in order to reconcile the two different schemes and to cast the task in terms of standard supervised machine classification at the sentence level.

Our label set has some similarity to the RTE5 3-way task (Bentivogli et al., 2009), which used "entailment", "contradiction" and "unknown" labels. The additional distinctions our in labels reflect typical distinctions made by a variety of tutorial dialogue systems and to match the human tutor's intuitions about the general error types made by students and corresponding teaching tactics. For example, a likely response to "partially_correct_incomplete" would be to tell the student that what they said so far was correct but it had some gaps, and to encourage them to fill in those gaps. In contrast, the response to "contradictory" would emphasize that there is a mistake and the student needs to change their answer rather than just expand it. Finally, the response to the "irrelevant" answer may point out that the student is not thinking about relevant concepts. The "non_domain" content could be treated as an indicator that the student is frustrated or confused, and may require special attention.

The annotations in the source corpora make some more fine-grained distinctions based on the needs of the corresponding systems. In principle, it is possible to have answers that have both correct and contradictory parts, and acknowledge correct parts before pointing out mistakes. There are also distinct classes of "non_domain" utterances, e.g., social and metacognitive statements, to which an ITS may want to react differently (described in Section 2.1). However, these situations were rare in our corpora, and we decided to use a single class for all contradictory answers and a single non-domain class. These may be expanded in the future as more data becomes available for new versions of this challenge task.

Label	Definition
non_domain	does not contain domain content, e.g., a help request or “I don’t know”
correct	the student answer is correct
partially_correct_incomplete	the answer does not contradict the reference answer and includes some correct nuggets, but parts are missing
contradictory	an answer that contradicts some part of the reference answer
irrelevant	contains domain content, but does not answer the question

Figure 1: The set of answer labels used in our task

We further discuss the relationship with recognizing textual entailment in Section 5. In the rest of this section, we describe our corpora and discuss how we obtained these labels from the raw data available in our datasets.

2.1 Beetle II data

The Beetle corpus consists of the interactions between students and the Beetle II tutorial dialogue system (Dzikovska et al., 2010b). The Beetle II system is an intelligent tutoring system that teaches students with no knowledge of high-school physics concepts in basic electricity and electronics. In the first system evaluation, students spend 3-5 hours going through prepared reading material, building and observing circuits in the simulator and interacting with a dialogue-based tutor. The interaction was by keyboard, with the computer tutor asking questions, receiving replies and providing feedback via a text-based chat interface. The data from 73 undergraduate volunteer participants at southeastern US university were recorded and annotated to form the Beetle human-computer dialogue corpus.

The Beetle II lesson material contains two types of questions. Factual questions require them to name a set of objects or a simple property, e.g., “Which components in circuit 1 are in a closed path?” or “Are bulbs A and B wired in series or in parallel”. Explanation and definition questions require longer answers that consist of 1-2 sentences, e.g., “Why was bulb A on when switch Z was open?” (expected answer “Because it was still in a closed path with the battery”) or “What is voltage?” (expected answer “Voltage is the difference in states between two terminals”). From the full Beetle evaluation corpus, we automatically extracted only the students’ answers to explanation and definition questions, since reacting to them appropriately requires processing more complex input than factual questions.

The extracted answers were filtered to remove duplicates. In the Beetle II lesson material there are a number of similar questions and the tutor effectively had a template answer such as ”Terminal X is connected to the negative/positive battery terminal”. A number of students picked up on this and used the identical pattern in their responses (Steinhauser et al., 2011). This resulted in a number of answers to certain questions that came from different speakers but which were exact copies of each other. We removed such answers from the data set, since they were likely to be in both the training and test set, thus inflating our results. Note that only exact matches were removed: for example, answers that were nearly identical but contained spelling errors were retained, since they would need to be handled in a practical system.

Student utterances were manually labeled using a simplified version of the DEMAND coding scheme (Campbell et al., 2009) shown in Figure 2. The utterances were first classified as related to domain content, student’s metacognitive state, or social interaction. Utterances addressing domain content were further classified with respect to their correctness as described in the table. The Kappa value for this annotation effort was $\kappa = 0.69$.

This annotation maps straightforwardly into our set of labels. The social and metacognitive statements are mapped to the “non_domain” label; “pc_some_error”, “pc” and “incorrect” are mapped to the “contradictory” label; and the other classes have a one-to-one correspondence with our task labels.

2.2 SciEntsBank data

The SciEntsBank corpus (Nielsen et al., 2008) consists of student responses to science assessment

Category	Subcategory	Description
Metacognitive	positive negative	content-free expressions describing student knowledge, e.g., "I don't know"
Social	positive negative neutral	expressions describing student's attitudes towards themselves and the computer (mostly negative in this data, e.g., "You are stupid")
Content		the utterance addresses domain content.
	correct	the student answer is fully correct
	pc_some_missing	the student said something correct, but incomplete
	incorrect	the student's answer is completely incorrect
	pc_some_error	the student's answer contains correct parts, but some errors as well
	pc	the answer contains a mixture of correct, incorrect and missing parts
	irrelevant	the answer may be correct or incorrect, but it is not answering the question.

Figure 2: Annotation scheme used in the Beetle corpus

questions. Specifically, around 16k answers were collected spanning 16 distinct science subject areas within physical sciences, life sciences, earth sciences, space sciences, scientific reasoning and technology. The tests were part of the Berkeley Lawrence Hall of Science Assessing Science Knowledge (ASK) standardized assessments covering material from their Full Option Science System (FOSS) (Lawrence Hall of Science, 2011). The answers came from students in grades 3-6 in schools across North America.

The tests included a variety of questions including fill in the blank and multiple choice, but the SciEntsBank corpus only used a subset that required students to explain their beliefs about topics, typically in one to two sentences. We reviewed the questions and a sample of the responses and decided to filter the following types of questions from the corpus, because they did not mesh with our goals. First, we removed questions whose expected answer was more than two full sentences (typically multi-step procedures), which were beyond the scope of our task. Second, we removed questions where the expected answer was ill-defined or very open-ended. Finally, the most frequent reason for removing questions was an extreme imbalance in the answer classifications (e.g., for many questions, almost all of the answers were labeled "partially_correct_incomplete"). Specifically, we removed questions where more than 80% of the answers had the same label and questions with fewer than three correct answers, since these questions were unlikely to be useful in differentiating between the quality of assessment systems.

The SciEntsBank corpus was developed for the purpose of assessing student responses at a very fine-grained level. The reference answers were broken down into several facets, which consisted roughly of two key terms and the relation connecting them. Nielsen et al. annotated student responses to indicate for each reference answer facet whether the response 1) implied the student understood the facet, 2) implied they held a contradictory belief, 3) included a related, non-contradicting facet, or 4) left the facet unaddressed. Reported agreement was 86.2% with a Kappa statistic (Cohen, 1960) of 0.728, which is in the range of substantial agreement.[1]

Because our task focuses on answer classification rather than facet classification, we developed a set of rules indicating which combinations of facets constituted a correct answer. We were then able to compute an answer label from the gold-standard facet annotations, as follows. First, if any facet was annotated as contradictory, the answer was also labeled "contradictory". Second, if all of the expected facets for any valid answer were annotated as being understood, the answer was labeled "cor-

[1] These statistics were actually based on five labels, but we chose to combine the fifth, a self-contradiction, with other contradictions for the purposes of our task.

rect". Third, the remaining answers that included some but not all of the expected facets were labeled "partially_correct_incomplete". Fourth, if an answer matched none of the expected facets, and had not been previously labeled as "contradictory" it was given the label "irrelevant". Finally, all "irrelevant" answers were reviewed manually to determine whether they should be relabeled as "non_domain". However, since Nielsen et al. had already removed most of the responses that originally fell into this category, we only found 24 "non_domain" answers.

3 Baselines

We established three baselines for our data set – a straightforward majority class baseline, an existing system baseline (BEETLE II system performance, which we report only for the BEETLE portion of the dataset), and the performance of a simple classifier based on lexical similarity, which we report in order to offer a substantial example of applying the same classifier to both portions of the dataset.

3.1 BEETLE II system baseline

The interpretation component of the BEETLE II system uses a syntactic parser and a set of hand-authored rules to extract the domain-specific semantic representations of student utterances from the text. These representations were then matched against the semantic representations of expected correct answers supplied by tutors. The resulting system output was automatically mapped into our target labels as discussed in (Dzikovska et al., 2012).

3.2 Lexical similarity baseline

To provide a higher baseline that is comparable across both subsets of the data, we built a simple decision tree classifier using the Weka 3.6.2 implementation of C4.5 pruned decision trees (weka.classifiers.trees.J48 class), with default parameters. As features, we used lexical similarity scores computed by the `Text::Similarity` package with default parameters[2]. The code computes four similarity metrics – the raw number of overlapping words, F1 score, Lesk score and cosine score. We compared the learner response to the expected answer(s) and the question, resulting in eight total features (the four values indicated above for the comparison with the question and the highest of each value from the comparisons with each possible expected answer).

This baseline is based on the lexical overlap baseline used in RTE tasks (Bentivogli et al., 2009). However, we measured overlap with the question text in addition to the overlap with the expected answers. Students often repeat parts of the question in their answer and this needs to be taken into account to differentiate, for example, "partially_correct_incomplete" and "correct" answers.

4 Results

4.1 Experimental Setup

We held back part of the data set for use as standard test data in the future challenge tasks. For BEETLE, this consisted of all student answers to 9 out of 56 explanation questions asked by the system, plus approximately 15% of the student answers to the remaining 47 questions, sampling so that the distribution of labels in test data was similar to the training data. For SCIENTSBANK, we used a previous train-test split (Nielsen et al., 2009). For both data sets, the data was split so that in the future we can test how well the different systems generalize: i.e., how well they perform on answers to questions for which they have some sample student answers vs. how well they perform on answers to questions that were not in the training data (e.g., newly created questions in a deployed system). We discuss this in more detail in Section 5.

In this paper, we report baseline performance on the training set to demonstrate that the task is sufficiently challenging to be interesting and that systems can be compared using our evaluation metrics. We preserve the true test data for use in the planned large-scale system comparisons in a community shared task.

For the lexical similarity baseline, we use 10-fold cross-validation.[3] For the BEETLE II system baseline, the language understanding module was de-

[2] http://search.cpan.org/dist/Text-Similarity/

[3] We did not take the student id into account explicitly during cross-validation. While there is some risk that the classifiers will learn features specific to the student, we concluded (based on our understanding of data collection specifics for both data sets) that there is little enough overlap in cross-validation on the training data that this should not have a big effect on the results.

veloped based on eight transcripts, each taken from the interaction of a different student with an earlier version of the system. These sessions were completed prior to the beginning of the experiment during which the Beetle corpus was collected, and are not included in the corpus presented here. Thus, the dataset used in the paper constitutes unseen data for the Beetle II system.

We process the two corpora separately because the additional system baseline is available for beetle, and because the corpora may be different enough that it will be helpful for shared task participants to devise processing strategies that are sensitive to the provenance of the data.

4.2 Evaluation Metrics

Table 1 shows the distribution of codes in the annotated data. The distribution is unbalanced, and therefore in our evaluation results we report per-class precision, recall and F_1 scores, plus the averaged scores using two different ways to average over per-class evaluation scores, micro- and macro- averaging.

For a set of classes C, each represented with N_c instances in the test set, the macro-averaged recall is defined as

$$R_{macro} = \frac{1}{|C|} \sum_{c \epsilon C} R(c)$$

and the micro-averaged recall as

$$R_{micro} = \sum_{c \epsilon C} \frac{1}{N_c} R(c)$$

Micro- and macro-averaged precision and F_1 are defined similarly.

Micro-averaging takes class sizes into account, so a system that performs well on the most common classes will have a high micro-average score. This is the most commonly used classifier evaluation metric. Note that, in particular, overall classification accuracy (defined as the number of correctly classified instances out of all instances) is mathematically equivalent to micro-averaged recall (Abudawood and Flach, 2011). However, macro-averaging better reflects performance on small classes, and is commonly used for unbalanced classification problems (see, e.g., (Lewis, 1991)). We report both values in our results.

	Beetle		SciEntsBank	
Label	Count	Freq.	Count	Freq.
correct	1157	0.42	2095	0.40
partially_correct_incomplete	626	0.23	1431	0.27
contradictory	656	0.24	526	0.10
irrelevant	86	0.03	1175	0.22
non_domain	204	0.07	24	0.005
total	2729		5251	

Table 1: Distribution of annotated labels in the data

In addition, we report the system scores on the binary decision of whether or not the corrective feedback should be issued (denoted “corrective feedback” in the results table). It assumes that a tutoring system using a classifier will give corrective feedback if the classifiers returns any label other than “correct”. Thus, every instance classified as “partially_correct_incomplete”, “contradictory”, “irrelevant” or “non_domain” is counted as true positive if the hand-annotated label also belongs to this set (even if the classifier disagrees with the annotation); and as false positive if the hand-annotated label is “correct”. This reflects the idea that students are likely to be frustrated if the system gives corrective feedback when their answer is in fact a fully accurate paraphrase of a correct answer.

4.3 Beetle baseline performance

The detailed evaluation results for all baselines are presented in Table 2.

The majority class baseline is to assign “correct” to every test instance. It achieves 42% overall accuracy. However, this is obviously at the expense of serious errors; for example, such a system would tell the students that they are correct if they are saying something contradictory. This is reflected in a much lower macro-averaged F_1 score.

The Beetle II system performs only slightly better than the baseline on the overall accuracy (0.44 vs. 0.42 micro-averaged recall). However, the macro-averaged F_1 score of the Beetle II system is substantially higher (0.46 vs. 0.12). The micro-averaged results show a similar pattern, although the majority-class baseline performs slightly better than in the macro-averaged case, as expected.

Comparing the Beetle II parser to our lexical similarity baseline, Beetle II has lower overall accuracy, but performs similarly on micro- and macro-averaged scores. Beetle II precision is higher than that of the classifier in all cases except for the binary decision as to whether corrective feedback should be issued. This is not unexpected given how the system was designed – since misunderstandings caused dialogue breakdown in pilot tests, the parser was built to prefer rejecting utterances as uninterpretable rather than assigning them an incorrect class, leading to a considerably lower recall. Around 31% of utterances could not be interpreted.

Our recent analysis shows that both incorrect interpretations (in particular, confusions between "partially_correct_incomplete" and "contradictory") and rejections have significant negative effects on learning gain (Dzikovska et al., 2012). Classifiers can be tuned to reject examples where classification confidence falls below a given threshold, resulting in precision-recall trade-offs. Our baseline classifier classified all answer instances; exploring the possibilities for rejecting some low-confidence answers is planned for future work.

4.4 SciEntsBank baseline performance

The accuracy of the majority class baseline (which assumes all answers are "correct") is 40% for SciEntsBank, about the same as it was for Beetle. The evaluation results, based on 10-fold cross-validation, for our simple lexical similarity classifier are presented in Table 3. The lexical similarity based classifier outperforms the majority class baseline by 0.18 and 3% on the macro-averaged F_1-measure and accuracy, respectively. The F_1-measure for the two-way classification detecting answers which need corrective feedback is 0.66.

The scores on SciEntsBank are noticeably lower than those for Beetle. The SciEntsBank includes questions from 12 distinct science subject areas, rather than a single area as in Beetle. This decision tree classifier learns a function from the eight text similarity features to the desired answer label. Because the features do not mention particular words, the model can be applied to items other than the ones on which it was trained, and even to items from different subject areas. However, the correct weighting of the textual similarity features depends on the extent and nature of the expected textual overlap, which does vary from subject-area to subject-area. We suspect that the differences between subject areas made it hard for the decision-tree classifier to find a single, globally appropriate strategy. Nielsen (2009) reported the best results for classifying facets when training separate question-specific or even facet-specific classifiers. Although separate training for each item reduces the amount of relevant training data for each classifier, it allows each classifier to learn the specifics of how that item works. A comparison using this style of training would be a reasonable next step,

Predictn	correct	pc_inc	contra	irrlvnt	nondom
correct	1213	553	209	392	2
pc_inc	432	497	128	241	2
contra	115	109	58	74	3
irrlvnt	335	272	131	468	17
nondom	0	0	0	0	0

Figure 4: Confusion matrix for lexical classifier on SciEntsBank. Predictions in rows, gold labels in columns

5 Discussion and Future Work

The results presented satisfy two critical requirements for a challenge task. First, we have shown that it is feasible to develop a system that performs significantly better than the majority class baseline. On the macro-averaged F_1-measure, our lexical classifier outperformed the majority-class baseline by 0.33 (on Beetle) and 0.18 (on SciEntsBank) and by 13% and 3% on accuracy. Second, we have also shown, as is desired for a challenge task, that the task is not trivial. With a system specifically designed to parse the Beetle corpus answers, the macro-averaged F_1-measure was just 0.46 and on the binary decision regarding whether the response needed corrective feedback, it achieved just 0.63.

One contribution of this work was to define a general classification scheme for student responses that allows more specific learner feedback. Another key contribution was to unify two, previously incompatible, large student response corpora under this common annotation scheme. The resultant corpus will enable researchers to train learning algorithms to classify student responses. These classifications can then be used by a dialogue manager to generate targeted learner feedback. The corpus is available

Classifier:	majority			lexical similarity			BEETLE II		
Predicted label	prec.	recall	F1	prec.	recall	F1	prec.	recall	F1
correct	0.42	1.00	0.60	0.68	0.75	0.72	0.93	0.53	0.68
partially_correct_incomplete	0.00	0.00	0.00	0.41	0.38	0.39	0.43	0.53	0.47
contradictory	0.00	0.00	0.00	0.39	0.34	0.36	0.58	0.23	0.33
irrelevant	0.00	0.00	0.00	0.05	0.02	0.03	0.23	0.17	0.20
non_domain	0.00	0.00	0.00	0.66	0.82	0.73	0.92	0.46	0.61
macroaverage	0.09	0.20	0.12	0.44	0.46	0.45	0.62	0.39	0.46
microaverage	0.18	0.42	0.25	0.53	0.55	0.54	0.71	0.44	0.53
corrective feedback	0.00	0.00	0.00	0.80	0.74	0.77	0.73	0.56	0.63

Table 2: Evaluation results for BEETLE corpus

Classifier:	lexical similarity					BEETLE II				
Predicted label	corrct	pc_inc	contra	irrlvnt	nondom	corrct	pc_inc	contra	irrlvnt	nondom
correct	870	187	199	20	2	617	20	23	0	3
part_corr_incmp	138	239	178	24	11	249	332	146	29	20
contradictory	139	153	221	33	22	68	38	149	3	0
irrelevant	3	20	12	2	1	4	22	23	15	1
non_domain	7	27	46	7	168	3	3	1	1	94
uninterpretable	n/a	n/a	n/a	n/a	n/a	216	211	314	38	86

Figure 3: Confusion matrix for BEETLE corpus. Predictions in rows, gold labels in columns

Classifier:	baseline			lexical similarity		
Predicted label	prec.	recall	F1	prec.	recall	F1
correct	0.40	1.00	0.57	0.51	0.58	0.54
partially_correct_incomplete	0.00	0.00	0.00	0.38	0.35	0.36
contradictory	0.00	0.00	0.00	0.16	0.11	0.13
irrelevant	0.00	0.00	0.00	0.38	0.40	0.39
non_domain	0.00	0.00	0.00	0.00	0.00	0.00
macroaverage	0.08	0.20	0.11	0.29	0.29	0.29
microaverage	0.16	0.40	0.23	0.41	0.43	0.42
corrective feedback	0.00	0.00	0.00	0.69	0.63	0.66

Table 3: Evaluation results for SCIENTSBANK baselines

for general research purposes and forms the basis of SEMEVAL-2013 shared task "Textual entailment and paraphrasing for student input assessment".[4]

A third contribution of this work was to provide basic evaluation benchmark metrics and the corresponding evaluation scripts (downloadable from the site above) for other researchers, including shared task participants. This will facilitate the comparison and, hence, the progress, of research.

The work reported here is based on approximately 8000 student responses to questions covering 12 distinct science subjects and coming from a wide range of student ages. These responses comprise the training data for our task. The vast majority of prior work, including BEETLE II, which was included as a benchmark here, has been designed to provide ITS feedback for relatively small, well-defined domains. The corpus presented in this paper is intended to encourage research into more generalizable, domain-independent techniques. Following Nielsen (2009), from whom the SCIENTSBANK corpus was adapted, our shared task evaluation corpus will be composed of three types of data: additional student responses for all of the questions in the training data (Unseen Answers), student responses to questions that were not seen in the training data, but that are from the same subject areas (Unseen Questions), and responses to questions from three entirely different subject areas (Unseen Domains), though in this case the questions are still from the same general domain – science. Unseen Answers is the typical scenario for the vast majority of prior work – training and testing on responses to the same questions. Unseen Questions and Unseen Domains allow researchers to evaluate how well their systems generalize to near and far domains, respectively.

The primary target application for this work is intelligent tutoring systems, where the classification of responses is intended to facilitate specific pedagogic feedback. Beneath the surface, the baseline systems reported here are more similar to grading systems that use the approach of (Leacock and Chodorow, 2003), which uses classifier technology to detect expressions of facet-like concepts, then converts the result to a numerical score, than to grading systems like (Mohler et al., 2011), which directly produces a numerical score, using support vector regression and similar techniques. Either approach is reasonable, but we think that feedback is the more challenging test of a system's ultimate abilities, and therefore a better candidate for the shared task. The corpora from those systems, alongside with new corpora currently being collected in BEETLE and SCIENTSBANK domains, can serve as sources of data for future tasks extensions.

Future systems developed for this task can benefit from the large amount of existing work on recognizing textual entailment (Giampiccolo et al., 2007; Giampiccolo et al., 2008; Bentivogli et al., 2009) and on detecting contradiction (Ritter et al., 2008; De Marneffe et al., 2008). However, there are substantial challenges in applying the RTE tools directly to this data set. Our set of labels is more fine-grained than RTE labels to reflect the needs of intelligent tutoring systems (see Section 2). In addition, the top-performing systems in RTE5 3-way task, as well as contradiction detection methods, rely on NLP tools such as dependency parsers and semantic role labelers; these do not perform well on specialized terminology and language constructs coming from (typed) dialogue context. We chose to use lexical similarity as a baseline specifically because a similar measure was used as a standard baseline in RTE tasks, and we expect that adapting the more complex RTE approaches for purposes of this task will result in both improved results on our data set and new developments in computational linguistics algorithms used for RTE and related tasks.

Acknowledgments

We thank Natalie Steinhauser, Gwendolyn Campbell, Charlie Scott, Simon Caine, Leanne Taylor, Katherine Harrison and Jonathan Kilgour for help with data collection and preparation. The research reported here was supported by the US ONR award N000141010085 and by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A110811 to Boulder Language Technologies Inc. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

[4] See http://www.cs.york.ac.uk/semeval-2013/task4/

References

Tarek Abudawood and Peter Flach. 2011. Learning multi-class theories in ilp. In *The 20th International Conference on Inductive Logic Programming (ILP'10)*. Springer, June.

V. Aleven, O. Popescu, and K. R. Koedinger. 2001. Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In *Proceedings of the 10th International Conference on Artificial Intelligence in Education (AIED '01)"*.

Vincent Aleven, Octav Popescu, and Koedinger Koedinger. 2002. Pilot-testing a tutorial dialogue system that supports self-explanation. *Lecture Notes in Computer Science*, 2363:344–354.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernando Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Notebook papers and results, Text Analysis Conference (TAC)*.

Charles Callaway, Myroslava Dzikovska, Colin Matheson, Johanna Moore, and Claus Zinn. 2006. Using dialogue to learn math in the LeActiveMath project. In *Proceedings of the ECAI Workshop on Language-Enhanced Educational Technology*, pages 1–8, August.

Gwendolyn C. Campbell, Natalie B. Steinhauser, Myroslava O. Dzikovska, Johanna D. Moore, Charles B. Callaway, and Elaine Farrow. 2009. The DeMAND coding scheme: A "common language" for representing and analyzing student discourse. In *Proceedings of 14th International Conference on Artificial Intelligence in Education (AIED), poster session*, Brighton, UK, July.

Michelene T. H. Chi, Nicholas de Leeuw, Mei-Hung Chiu, and Christian LaVancher. 1994. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439–477.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):3746.

M.C. De Marneffe, A.N. Rafferty, and C.D. Manning. 2008. Finding contradictions in text. *Proceedings of ACL-08: HLT*, pages 1039–1047.

Myroslava Dzikovska, Diana Bental, Johanna D. Moore, Natalie B. Steinhauser, Gwendolyn E. Campbell, Elaine Farrow, and Charles B. Callaway. 2010a. Intelligent tutoring with natural language support in the Beetle II system. In *Sustaining TEL: From Innovation to Learning and Practice - 5th European Conference on Technology Enhanced Learning, (EC-TEL 2010)*, Barcelona, Spain, October.

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 *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010) demo session*, Uppsala, Sweden, July.

Myroslava O. Dzikovska, Peter Bell, Amy Isard, and Johanna D. Moore. 2012. Evaluating language understanding accuracy with respect to objective outcomes in a dialogue system. In *Proceedings of the 13th Conference of the European Chapter of the Association for computational Linguistics*, Avignon, France, April.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June.

Danilo Giampiccolo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, Elena Cabrio, and Bill Dolan. 2008. The fourth PASCAL recognizing textual entailment challenge. In *Proceedings of Text Analysis Conference (TAC) 2008*, Gaithersburg, MD, November.

Michael Glass. 2000. Processing language input in the CIRCSIM-Tutor intelligent tutoring system. In *Proceedings of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*.

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

Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. 2004. Combining competing language understanding approaches in an intelligent tutoring system. In James C. Lester, Rosa Maria Vicari, and Fábio Paraguaçu, editors, *Intelligent Tutoring Systems*, volume 3220 of *Lecture Notes in Computer Science*, pages 346–357. Springer.

Pamela Jordan, Maxim Makatchev, Umarani Pappuswamy, Kurt VanLehn, and Patricia Albacete. 2006. A natural language tutorial dialogue system for physics. In *Proceedings of the 19th International FLAIRS conference*.

Pamela W. Jordan. 2004. Using student explanations as models for adapting tutorial dialogue. In Valerie Barr and Zdravko Markov, editors, *FLAIRS Conference*. AAAI Press.

Lawrence Hall of Science. 2006. Assessing Science Knowledge (ask). University of California at Berkeley, NSF-0242510.

Lawrence Hall of Science. 2011. Full option science system.

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

David D. Lewis. 1991. Evaluating text categorization. In *Proceedings of the workshop on Speech and Natural Language*, HLT '91, pages 312–318, Stroudsburg, PA, USA.

Diane Litman, Johanna Moore, Myroslava Dzikovska, and Elaine Farrow. 2009. Using natural language processing to analyze tutorial dialogue corpora across domains and modalities. In *Proceedings of 14th International Conference on Artificial Intelligence in Education (AIED)*, Brighton, UK, 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, James H. Martin, and Martha Palmer. 2008. Annotating students' understanding of science concepts. In *Proceedings of the Sixth International Language Resources and Evaluation Conference, (LREC08)*, Marrakech, Morocco.

Rodney D. Nielsen, Wayne Ward, and James H. Martin. 2009. Recognizing entailment in intelligent tutoring systems. *The Journal of Natural Language Engineering*, 15:479–501.

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 *Proceedings of ITS-2004*, pages 390–400.

A. Ritter, D. Downey, S. Soderland, and O. Etzioni. 2008. It's a contradiction—no, it's not: a case study using functional relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 11–20.

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 *Proceedings of the 15th International Conference on Artificial Intelligence in Education (AIED-2011)*.

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