ETS: Domain Adaptation and Stacking for Short Answer Scoring*

Michael Heilman and **Nitin Madnani**
Educational Testing Service
660 Rosedale Road
Princeton, NJ 08541, USA
{mheilman,nmadnani}@ets.org

Abstract

Automatic scoring of short text responses to educational assessment items is a challenging task, particularly because large amounts of labeled data (i.e., human-scored responses) may or may not be available due to the variety of possible questions and topics. As such, it seems desirable to integrate various approaches, making use of model answers from experts (e.g., to give higher scores to responses that are similar), prescored student responses (e.g., to learn direct associations between particular phrases and scores), etc. Here, we describe a system that uses stacking (Wolpert, 1992) and domain adaptation (Daume III, 2007) to achieve this aim, allowing us to integrate item-specific n-gram features and more general text similarity measures (Heilman and Madnani, 2012). We report encouraging results from the Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge.

1 Introduction

In this paper, we address the problem of automatically scoring short text responses to educational assessment items for measuring content knowledge.

Many approaches can be and have been taken to this problem—e.g., Leacock and Chodorow (2003), Nielsen et al. (2008), *inter alia*. The effectiveness of any particular approach likely depends on the the availability of data (among other factors). For example, if thousands of prescored responses are available, then a simple classifier using n-gram features may suffice. However, if only model answers (i.e., reference answers) or rubrics are available, more general semantic similarity measures (or even rule-based approaches) would be more effective.

It seems likely that, in many cases, there will be model answers as well as a modest number of prescored responses available, as was the case for the Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge (§2). Therefore, we desire to incorporate both task-specific features, such as n-grams, as well as more general features such as the semantic similarity of the response to model answers.

We also observe that some features may themselves require machine learning or tuning on data from the domain, in addition to any machine learning required for the overall system.

In this paper, we describe a machine learning approach to short answer scoring that allows us to incorporate both item-specific and general features by using the domain adaptation technique of Daume III (2007). In addition, the approach employs stacking (Wolpert, 1992) to support the integration of components that require tuning or machine learning.

2 Task Overview

In this section, we describe the task to which we applied our system: the Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge (Dzikovska et al., 2013), which was task 7 at SemEval 2013.

The aim of the task is to classify student responses to assessment items from two datasets represent-

*System description papers for SemEval 2013 are required to have a team ID (e.g., "ETS") as a prefix.

ing different science domains: the Beetle dataset, which pertains to basic electricity and electronics (Dzikovska et al., 2010), and the Science Entailments corpus (SciEntsBank) (Nielsen et al., 2008), which covers a wider range of scientific topics.

Responses were organized into five categories: correct, partially correct, contradictory, irrelevant, and non-domain. The SciEntsBank responses were converted to this format as described by Dzikovska et al. (2012).

The Beetle training data had about 4,000 student answers to 47 questions. The SciEntsBank training data had about 5,000 prescored student answers to 135 questions from 12 domains (different learning modules). For each item, one or more model responses were provided by the task organizers.

There were three different evaluation scenarios: "unseen answers", for scoring new answers to items represented in the training data; "unseen questions", for scoring answers to new items from domains represented in the training data; and "unseen domains", for scoring answers to items from new domains (only for SciEntsBank since Beetle focused on a single domain).

Performance was evaluated using accuracy, macro-average F_1 scores, and weighted average F_1 scores.

For additional details, see the task description paper (Dzikovska et al., 2013).

3 System Details

In this section, we describe the short answer scoring system we developed, and the variations of it that comprise our submissions to task 7. We begin by describing our statistical modeling approach. Thereafter, we describe the features used by the model (§3.1), including the PERP feature that relies on stacking (Wolpert, 1992), and then the domain adaptation technique we used (§3.2).

Our system is a logistic regression model with ℓ_2 regularization. It uses the implementation of logistic regression from the scikit-learn toolkit (Pedregosa et al., 2011).[1] To tune the C hyperparameter, it uses a 5-fold cross-validation grid search (with $C \in 10^{\{-3,-2,\ldots,3\}}$).

During development, we evaluated performance using 10-fold cross-validation, with the 5-fold cross-validation grid search still used for tuning within each training partition (i.e., each set of 9 folds used for training during cross-validation).

3.1 Features

Our full system includes the following features.

3.1.1 Baseline Features

It includes all of the baseline features generated with the code provided by the task organizers.[2] There are four types of lexically-driven text similarity measures, and each is computed by comparing the learner response to both the expected answer(s) and the question, resulting in eight features in total. They are described more fully by Dzikovska et al. (2012).

3.1.2 Intercept Feature

The system includes an intercept feature that is always equal to one, which, in combination with the domain adaptation technique described in §3.2, allows the system to model the *a priori* distribution over classes for each domain and item. Having these explicit intercept features effectively saves the learning algorithm from having to use other features to encode the distribution over classes.

3.1.3 Word and Character n-gram Features

The system includes binary indicator features for the following types of n-grams:

- lowercased word n-grams in the response text for $n \in \{1, 2, 3\}$.
- lowercased word n-grams in the response text for $n \in \{4, 5, \ldots, 11\}$, grouped into 10,000 bins by hashing and using a modulo operation (i.e., the "hashing trick") (Weinberger et al., 2009).
- lowercased character n-grams in the response text for $n \in \{5, 6, 7, 8\}$

[1] The scikit-learn toolkit uses a one-versus-all scheme, using multiple binary logistic regression classifiers, rather than a single multiclass logistic regression classifier.

[2] At the time of writing, the baseline code could be downloaded at `http://www.cs.york.ac.uk/semeval-2013/task7/`.

3.1.4 Text Similarity Features

The system includes the following text similarity features that compare the student response either to a) the reference answers for the appropriate item, or b) the student answers in the training set that are labeled "correct".

- the maximum of the smoothed, uncased BLEU (Papineni et al., 2002) scores obtained by comparing the student response to each correct reference answer. We also include the word n-gram precision and recall values for $n \in \{1, 2, 3, 4\}$ for the maximally similar reference answer.
- the maximum of the smoothed, uncased BLEU scores obtained by comparing the student response to each correct training set student answer. We also include the word n-gram precision and recall values for $n \in \{1, 2, 3, 4\}$ for the maximally similar student answer.
- the maximum PERP (Heilman and Madnani, 2012) score obtained by comparing the student response to the correct reference answers.
- the maximum PERP score obtained by comparing the student response to the correct student answers.

PERP is an edit-based approach to text similarity. It computes the similarity of sentence pairs by finding sequences of edit operations (e.g., insertions, deletions, substitutions, and shifts) that convert one sentence in a pair to the other. Then, using various features of the edits and weights for those features learned from labeled sentence pairs, it assigns a similarity score. Heilman and Madnani (2012) provide a detailed description of the original PERP system. In addition, Heilman and Madnani (To Appear) describe some minor modifications to PERP used in this work.

To estimate weights for PERP's edit features, we need labeled sentence pairs. First, we describe how these labeled sentence pairs are generated from the task data, and then we describe the stacking approach used to avoid training PERP on the same data it will compute features for.

For the reference answer PERP feature, we use the Cartesian product of the set of correct reference answers ("good" or "best" for Beetle) and the set of student answers, using 1 as the similarity score (i.e., the label for training PERP) for pairs where the student answer is labeled "correct" and 0 for all others. For the student answer PERP feature, we use the Cartesian product of the set of correct student answers and the set of all student answers, using 1 as the similarity score for pairs where both student answers are labeled "correct" and 0 for all others.[3] We use 10 iterations for training PERP.

In order to avoid training PERP on the same responses it will compute features for, we use 10-fold stacking (Wolpert, 1992). In this process, the training data are split up into ten folds. To compute the PERP features for the instances in each fold, PERP is trained on the other nine folds. After all 10 iterations, there are PERP features for every example in the training set. This process is similar to 10-fold cross-validation.

3.2 Domain Adaptation

The system uses the domain adaptation technique from Daume III (2007) to support generalization across items and domains.

Instead of having a single weight for each feature, following Daume III (2007), the system has multiple copies with potentially different weights: a generic copy, a domain-specific (i.e., module-specific) copy , and an item-specific copy. For an answer to an unseen item (i.e., question) from a new domain in the test set, only the generic feature will be active. In contrast, for an answer to an item represented in the training data, the generic, domain-specific, and item-specific copies of the feature would be active and contribute to the score.

For our submissions, this feature copying approach was not used for the baseline features (§3.1.1) or the BLEU and PERP text similarity features (§3.1.4), which are less item-specific. Those features had only general copies. We did not test whether doing so would affect performance.

[3] The Cartesian product of the sets of correct student answers and of all student answers will contain some pairs of identical correct answers. We decided to simply include these when training PERP, since we felt it would be desirable for PERP to learn that identical sentences should be considered similar.

Submission	Beetle		SciEntsBank		
	A	Q	A	Q	D
Run 1	.552	.547	.535	.487	.447
Run 2	.705	.614	.625	.356	.434
Run 3	.700	.586	.640	.411	.414
maximum	.705	.614	.640	.492	.471
mean	.514	.398	.457	.377	.374
Run 1 (corrected)	.555	.542	.546	.492	.424
Run 2 (corrected)	.703	.608	.641	.372	.428
Run 3 (corrected)	.695	.587	.655	.393	.396

Table 1: Weighted average F_1 scores for 5-way classification for our SemEval 2013 task 7 submissions, along with the maximum and mean performance of the original submissions, for comparison. "A" = unseen answers, "Q" = unseen questions, "D" = unseen domains (see §2 for details). "(corrected)" denotes a corrected version of a submission, as discussed in §4.1.

3.3 Submissions

We submitted three variations of the system. For each variation, a separate model was trained for Beetle and for SciEntsBank.

- **Run 1**: This run included the baseline (§3.1.1), intercept (§3.1.2), and the text-similarity features (§3.1.4) that compare student responses to reference answers (but not those that compare to scored student responses in the training set).
- **Run 2**: This run included the baseline (§3.1.1), intercept (§3.1.2), and n-gram features (§3.1.3).
- **Run 3**: This run included all features.

4 Results

Table 1 presents the weighted averages of F_1 scores across the five categories for the 5-way subtask, for each dataset and scenario. The maximum and mean scores of all the submissions are included for comparison. These results were provided to us by the task organizers.

For conciseness, we do not include accuracy or macro-average F_1 scores here. We observed that, in general, the results from different evaluation metrics were very similar to each other. We refer the reader to the task description paper (Dzikovska et al., 2013) for a full report of the task results.

4.1 Corrections

There were two errors in the original submissions. First, there was an error in how the domain adaptation features (§3.2) were computed for the SciEntsBank dataset: answer IDs were incorrectly used in place of the domain (i.e., module) IDs (e.g., "EM.45b.299.1" instead of just "EM" for the "EM" module). Second, there was an error in how the PERP features (§3.1.4) were computed for both datasets.

We corrected the errors, recomputed system outputs, ran the evaluation script provided by the task organizers, and then computed weighted average F_1 scores. The corrected results for the 5-way subtask are shown in Table 1 along with the results for the original submissions. We observed similar trends for the 2-way and 3-way subtasks.

4.2 Discussion

Observe that Runs 1 and 2 use subsets of the features from the full system (Run 3). While Runs 1 and 2 are not directly comparable to each other, Runs 1 and 3 can be compared to measure the effect of the features based on other previously scored student responses (i.e., n-grams, and the PERP and BLEU features based on student responses). Similarly, Runs 2 and 3 can be compared to measure the combined effect of all BLEU and PERP features.

It appears that features of the other student responses improve performance for the unseen answers task. For example, the full system (Run 3) performed better than Run 1, which did not include features of other student responses, on the unseen answers task for both Beetle and SciEntsBank.

However, it is less clear whether the PERP and

BLEU features improve performance. The full system (Run 3) did not always outperform Run 2, which did not include these features.

We leave to future work various additional questions, such as whether student response features or reference answer similarity features are more useful in general, and whether there are any systematic differences between human-machine and human-human disagreements.

5 Conclusion

We have presented an approach for short answer scoring that uses stacking (Wolpert, 1992) and domain adaptation (Daume III, 2007) to support the integration of various types of task-specific and general features. Evaluation results from task 7 at SemEval 2013 indicate that the system achieves relatively high levels of agreement with human scores, as compared to other systems submitted to the shared task.

Acknowledgments

We would like to thank the task organizers for facilitating this research and Dan Blanchard for helping with scikit-learn.

References

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June. Association for Computational Linguistics.

Myroslava O. Dzikovska, Diana Bental, Johanna D. Moore, Natalie Steinhauser, Gwendolyn Campbell, Elaine Farrow, and Charles B. Callaway. 2010. Intelligent tutoring with natural language support in the BEETLE II system. In *Proceedings of Fifth European Conference on Technology Enhanced Learning (EC-TEL 2010)*.

Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. 2012. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210, Montréal, Canada, June. Association for Computational Linguistics.

Myroslava O. Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In **SEM 2013: The First Joint Conference on Lexical and Computational Semantics*, Atlanta, Georgia, USA, 13-14 June. Association for Computational Linguistics.

Michael Heilman and Nitin Madnani. 2012. ETS: Discriminative edit models for paraphrase scoring. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 529–535, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Michael Heilman and Nitin Madnani. To Appear. Henry: Domain adapation and stacking for text similarity. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

C. Leacock and M. Chodorow. 2003. c-rater: Scoring of short-answer questions. *Computers and the Humanities*, 37.

Rodney D. Nielsen, Wayne Ward, and James H. Martin. 2008. Classification errors in a domain-independent assessment system. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 10–18, Columbus, Ohio, June. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1113–1120, New York, NY, USA. ACM.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.