Effective Feature Integration for Automated Short Answer Scoring[*]

Keisuke Sakaguchi
CLSP, Johns Hopkins University
Baltimore, MD
keisuke@cs.jhu.edu

Michael Heilman **Nitin Madnani**
Educational Testing Service
Princeton, NJ
{mheilman,nmadnani}@ets.org

Abstract

A major opportunity for NLP to have a real-world impact is in helping educators score student writing, particularly content-based writing (i.e., the task of automated short answer scoring). A major challenge in this enterprise is that scored responses to a particular question (i.e., labeled data) are valuable for modeling but limited in quantity. Additional information from the scoring guidelines for humans, such as exemplars for each score level and descriptions of key concepts, can also be used. Here, we explore methods for integrating scoring guidelines and labeled responses, and we find that stacked generalization (Wolpert, 1992) improves performance, especially for small training sets.

1 Introduction

Educational applications of NLP have considerable potential for real-world impact, particularly in helping to score responses to assessments, which could allow educators to focus more on instruction.

We focus on the task of analyzing short, content-focused responses from an assessment of reading comprehension, following previous work on short answer scoring (Leacock and Chodorow, 2003; Mohler et al., 2011; Dzikovska et al., 2013). This task is typically defined as a text regression or classification problem: we label student responses that consist of one or more sentences with scores on an ordinal scale (e.g. correct, partially correct, or incorrect; 1–5 score range, etc.). Importantly, in addition to the student response itself, we may also have available other information such as reference answers or descriptions of key concepts from the scoring guidelines for human scorers. Such information can be cheap to acquire since it is often generated as part of the assessment development process.

Generally speaking, most work on short answer scoring takes one of the following approaches:

- A **response-based** approach uses detailed features extracted from the student response itself (e.g., word n-grams, etc.) and learns a scoring function from human-scored responses.
- A **reference-based** approach compares the student response to reference texts, such as exemplars for each score level, or specifications of required content from the assessment's scoring guidelines. Various text similarity methods (Agirre et al., 2013) can be used.

These two approaches can, of course, be combined. However, to our knowledge, the issues of how to combine the approaches and when that is likely to be useful have not been thoroughly studied.

A challenge in combining the approaches is that the response-based approach produces a large set of sparse features (e.g., word n-gram indicators), while the reference-based approach produces a small set of continuous features (e.g., similarity scores between the response and exemplars for different score levels). A simple combination method is to train a model on the union of the feature sets (§3.3). However, the dense reference features may be lost among the many sparse response features.

[*] Work done when Keisuke Sakaguchi was an intern at ETS. Michael Heilman is now a data scientist at Civis Analytics.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1049–1054,
Denver, Colorado, May 31 – June 5, 2015.

Therefore, we apply stacked generalization (i.e. stacking) (Wolpert, 1992; Sakkis et al., 2001; Torres Martins et al., 2008) to build an ensemble of the response- and reference-based approaches. To our knowledge, there is little if any research investigating the value of stacking for NLP applications such as automated scoring.[1]

The contributions of this paper are as follows: (1) we investigate various reference-based features for short answer scoring, (2) we apply stacking (Wolpert, 1992) in order to combine the reference- and response-based methods, and (3) we demonstrate that the stacked combination outperforms other models, especially for small training sets.

2 Task and Dataset

We conduct our experiments on short-answer questions that are developed under the *Reading for Understanding* (RfU) assessment framework. This framework is designed to measure the reading comprehension skills of students from grades 6 through 9 by attempting to assess whether the reader has formed a coherent mental model consistent with the text discourse. A more detailed description is provided by Sabatini and O'Reilly (2013).

We use 4 short-answer questions based on two different reading passages. The first passage is a 1300-word short story. A single question ("Q1" hereafter) asks the reader to read the story and write a 5–7 sentence synopsis in her own words that includes all the main characters and action from the story but does *not* include any opinions or information from outside the story. The second passage is a 700-word article that describes the experiences of European immigrants in the late 19th and early 20th centuries. There are 3 questions associated with this passage: two that ask the reader to summarize one section each in the article ("Q2" and "Q4") and a third that asks to summarize the entire article ("Q3"). These 3 questions ask the reader to restrict his or her responses to 1–2 sentences each.

Each question includes the following:

- **scored responses**: short responses written by students, scored on a 0 to 4 scale for the first question, and 0 to 3 for the other 3.
- **exemplars**: one or two exemplar responses for each score level, and
- **key concepts**: several ($\leq$ 10) sentences briefly expressing key concepts in a correct answer.

The data for each question is split into a training and testing sets. For each question, we have about 2,000 scored student responses.

Following previous work on automatic scoring (Shermis and Burstein, 2013), we evaluate performance using the quadratically weighted κ (Cohen, 1968) between human and machine scores (rounded and trimmed to the range of the training scores).

3 Models for Short Answer Scoring

Next, we describe our implementations of the response- and reference-based scoring methods. All models use support vector regression (SVR) (Smola and Schölkopf, 2004), with the complexity parameter tuned by cross-validation on the training data.[2]

3.1 Response-based

Our implementation of the response-based scoring approach ("resp" in §4) uses SVR to estimate a model to predicts human scores for text responses. Various sparse binary indicators of linguistic features are used:

- binned response length (e.g. the `length-7` feature fires when the character contains 128-255 characters.)
- word n-grams from $n = 1$ to 2
- character n-grams from $n = 2$ to 5, which is more robust than word n-gram regarding spelling errors in student responses
- syntactic dependencies in the form of Parent-Label-Child (e.g. `boy-det-the` for "the boy")
- semantic roles in the form of PropBank[3] style (e.g. `say.01-A0-boy` for "(the) boy said")

[1] Some applications have used stacking but not analyzed its value. For example, many participants used stacking in the ASAP2 competition `http://http://www.kaggle.com/c/asap-sas`. Also, Heilman and Madnani (2013) used stacking for Task 7 of SemEval 2013.

[2] We used the implementation of SVR in scikit-learn (Pedregosa et al., 2011) via SKLL (`https://github.com/EducationalTestingService/skll`) version 0.27.0. Other than the complexity parameter, we used the defaults.

[3] `http://verbs.colorado.edu/~mpalmer/projects/ace.html`

The syntactic and semantic features were extracted using the ClearNLP parser.[4] We used the default models and options for the parser. We treat this model as a strong baseline to which we will add reference-based features.

3.2 Reference-based

Our implementation of the reference-based approach (“ref” in §4) uses SVR to estimate a model to predict human scores from various measures of the similarity between the response and information from the scoring guidelines provided to the human scorers. Specifically, we use the following information from §2: (a) sentences expressing key concepts that should be present in correct responses, and (b) small sets of exemplar responses for each score level. For each type of reference, we use the following similarity metrics:

- **BLEU**: the BLEU machine translation metric (Papineni et al., 2002), with the student response as the translation hypothesis. When using BLEU to compare the student response to the (much shorter) sentences containing key concepts, we ignore the brevity penalty.
- **word2vec cosine**: the cosine similarity between the averages of the word2vec vectors (Mikolov et al., 2013) of content words in the response and reference texts (e.g., exemplar), respectively.[5,6]
- **word2vec alignment**: the alignment method below with word2vec word similarities.
- **WordNet alignment**: the alignment method below with the Wu and Palmer (1994) WordNet (Miller, 1995) similarity score.

The WordNet and word2vec alignment metrics are computed as follows, where S is a student response, R is one of a set of reference texts, W_s and W_r are content words in S and R, respectively, and $Sim(W_s, W_r)$ is the word similarity function:

$$\frac{1}{len(S)} \sum_{W_s} \max_{W_r \in R} Sim(W_s, W_r) \qquad (1)$$

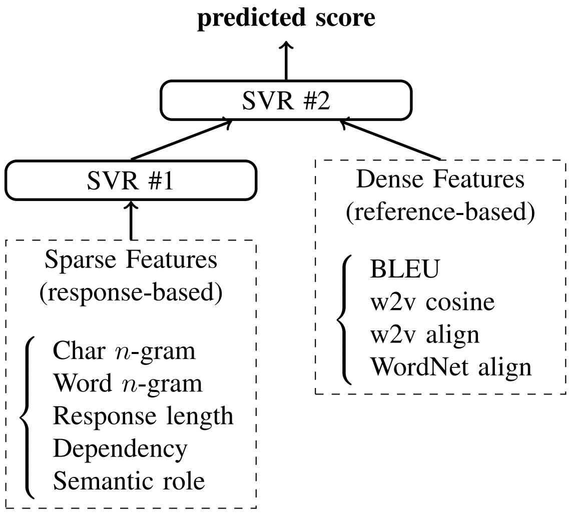

Figure 1: Stacking model for short answer scoring

When R is one of a set of reference texts (e.g., one of multiple exemplars available for a given score point), we use the maximum similarity over available values of R. In our data, there are multiple exemplars per score point, but only one text (usually, a single sentence) per key concept. In other words, we select the most similar exemplar response for each score level.

3.3 Simple Model Combination

One obvious way to combine the response- and reference-based models is to simply train a single model that uses both the sparse features of the student response and the dense, real-valued similarity features. Our experiments (§4) include such a model as a strong baseline, using SVR to estimate feature weights.

3.4 Model Combination with Stacking

In preliminary experiments with the training data, we observed no gains for the simple combination model over the component models. One potential challenge of combining the two scoring approaches is that the weights for the dense, reference-based features may be difficult to properly esti-

[4] http://www.clearnlp.com, v2.0.2

[5] The word2vec model was trained on the English Wikipedia as of June 2012, using gensim (http://radimrehurek.com/gensim/) with 100 dimensions, a context window of 5, a minimum count of 5 for vocabulary items, and the default skip-gram architecture.

[6] We define content words as ones whose POS tags begin with “N” (nouns), “V” (verbs), “J” (adjectives), or “R” (adverbs).

mate due to regularization[7] and the large number of sparse, mostly irrelevant linguistic features from the response-based approach. In fact, the reference-based sparse features constitute almost 90% of the entire feature set, while the response-based dense features constitute the remaining 10%.

This leads us to explore stacking (Wolpert, 1992), an ensemble technique where a top-layer model makes predictions based on predictions from lower-layer models. Here, we train a lower-layer model to aggregate the sparse response-based features into a single "response-based prediction" feature, and then train an upper-layer SVR model that includes that feature along with all of the reference-based features. Figure 1 shows the details.[8]

For training our stacking model, we first train the response-based regression model (SVR #1 in Figure 1), and then train the reference-based regression model (SVR #2) with an additional prediction feature value from the response-based model. Specifically, the lower-layer model concentrates sparse and binary features into a single continuous value, which accords with reference-based dense features in the upper-layer model. In training the lower-layer SVR on the training data, computing the response-based prediction feature (i.e., output of the lower-layer SVR) from the sparse features is similar to k-fold cross-validation ($k = 10$ here): the prediction feature values are computed for each fold by response-based SVR models trained on the remaining folds. In training the upper-layer SVR on the testing data, this prediction feature is computed by a single model trained on the entire training set.

4 Experiments

This section describes two experiments: an evaluation of reference-based similarity metrics, and an evaluation of methods for combining the reference- and response-based features by stacking. As mentioned in §2, we evaluate performance using quadratically weighted κ between the human and predicted scores.

	Q1	Q2	Q3	Q4
BLEU	.72	.45	.60	.52
word2vec cosine	.75	.45	.61	.52
word2vec alignment	.76	.47	.61	.51
WordNet alignment	.73	.49	.59	.51
All ("ref")	.78	.52	.66	.59
length	.68	.42	.59	.51
response-based ("resp")	.82	.72	.75	.74

Table 1: Training set cross-validation performance of reference-based models, in quadratically weighted κ, with baselines for comparison. The response-based ("resp") model is a stronger baseline as described in §3.3. Note that each reference-based model includes the length bin features for a fair comparison to "resp".

4.1 Similarity Metrics

We first evaluate the similarity metrics from §3.2 using 10-fold cross-validation on the training data. We evaluated SVR models for each metric individually as well as a model combining all features from all metrics. In all models, we included the response length bin features (§3.1) as a proxy for response-based features. We compare to the response-based model (§3.1) and to a model consisting of only the response length bin feature.

The results are shown in Table 1. Each similarity metric by itself does not always improve the performance remarkably from the baseline (i.e., the response length bin features). However, when we incorporate all the similarity features, we obtained substantial gain in all 4 questions. In the subsequent model combination experiment, therefore, we used all similarity features to represent the reference-based approach because it outperformed the other similarity models.

4.2 Model Combination

Next, we tested models that use both response- and reference-based features on a held-out test set, which contains 400 responses per question. We evaluated the response-based ("resp", §3.1) and reference-based ("ref", §3.2) individual models as well as the two combination methods ("ref+resp", §3.3 and "ref+resp stacking", §3.4). We also eval-

[7] Another possible combination approach would be to use the combination method from §3.3 but apply less regularization to the reference-based features, or, equivalently, scale them by a large constant. We only briefly explored this through training set cross-validation. The stacking approach seemed to perform at least as well in general.

[8] It would also be possible to also make a lower-layer model for the reference-based features, though doing this did not show benefits in preliminary experiments.

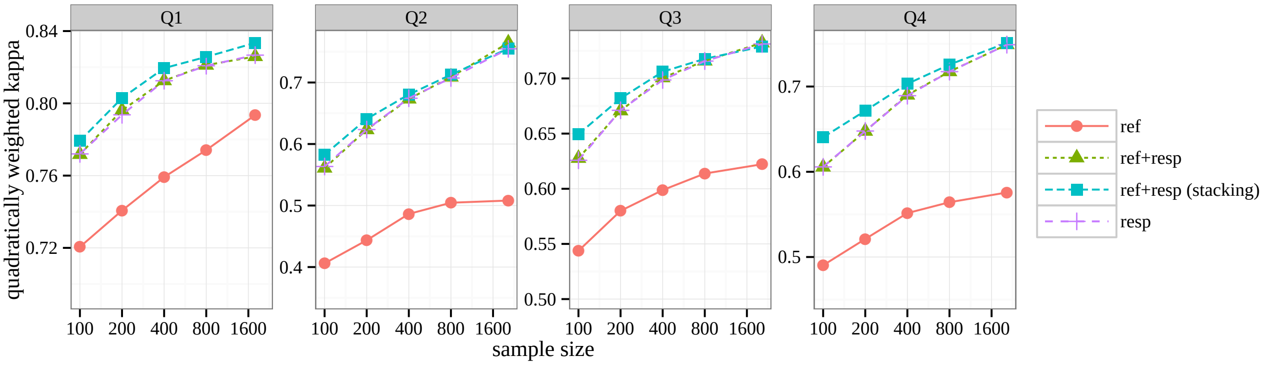

Figure 2: Test set results for various models trained on differently sized random samples of the training data. Each point represents an average over 20 runs, except for the rightmost points for each question, which correspond to training on the full training set. Note that the "resp" and "ref+resp" lines mostly overlap.

uated models trained on differently sized subsets of the training data. For each subset size, we averaged over 20 samples. The results are in Figure 2.

The performance of all models increased as training data grew, though there were diminishing returns (note the logarithmic scale). Also, the models with response-based features outperform those with just reference-based features, as observed previously by Heilman and Madnani (2013).

Most importantly, while all models with response-based features perform about the same with 1,000 training examples or higher, the stacked model tended to outperform the other models for cases where the number of training examples was very limited.[9] This indicates that stacking enables learning better feature weights than a simple combination when the feature set contains a mixture of sparse as well as dense features, particularly for smaller data sizes.

5 Conclusion

In this paper, we explored methods for using different sources of information for automatically scoring short, content-based assessment responses. We combined a response-based method that uses sparse features (e.g., word and character n-grams) with a reference-based method that uses a small number of features for the similarity between the response and information from the scoring guidelines (exemplars and key concepts).

On four reading comprehension assessment questions, we found that a combined model using stacking outperformed a non-stacked combination, particularly for the most practically relevant cases where training data was limited. We believe that such an approach may be useful for dealing with diverse feature sets in other automated scoring tasks as well as other NLP tasks.

As future work, it might be interesting to explore a more sophisticated model where the regression models in different layers are trained simultaneously by back-propagating the error of the upper-layer, as in neural networks.

Acknowledgments

We would like to thank John Sabatini and Kietha Biggers for providing us with the RfU datasets. We would also like to thank Dan Blanchard, Aoife Cahill, Swapna Somasundaran, Anastassia Loukina, Beata Beigman Klebanov, and the anonymous reviewers for their help.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared

[9] We are not aware of an appropriate significance test for experiments where subsets of the training data are used. However, the benefits of stacking seem unlikely to be due to chance. For all 4 items, stacking outperformed the non-stacking combination for 18 or more of the 20 200-response training subsets (note that under a binomial test, this would be significant with $p < 0.001$). Also, for the 100-response training subsets, stacking was better for 16 or more of the 20 subsets ($p < 0.01$).

task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June.

J. Cohen. 1968. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4).

Myroslava 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 *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 263–274, Atlanta, Georgia, USA, June.

Michael Heilman and Nitin Madnani. 2013. ETS: Domain adaptation and stacking for short answer scoring. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279, Atlanta, Georgia, USA, June.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

George A. Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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 ACL:HLT*, pages 752–762, Portland, Oregon, USA, June.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

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.

J. Sabatini and T. O'Reilly. 2013. Rationale for a new generation of reading comprehension assessments. In B. Miller, L. Cutting, and P. McCardle, editors, *Unraveling Reading Comprehension: Behavioral, Neurobiological and Genetic Components*. Paul H. Brooks Publishing Co.

Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. 2001. Stacking classifiers for anti-spam filtering of e-mail. In L. Lee and D. Harman, editors, *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)*, pages 44–50.

Mark D. Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.

André Filipe Torres Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 157–166, Honolulu, Hawaii, October.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241 – 259.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.