Rutstein, D. W., Niekrasz, J., & Snow, E. (2016, April). *Automated scoring of constructed response items measuring computational thinking.* Paper presented at the annual meeting of the National Council on Measurement in Education, Washington, D.C.

Contract grant sponsor: National Science Foundation; Contract grant number: 1240625.

Abstract

More and more assessments are using constructed response items for hard-to measure concepts. These items are challenging to score reliably in a short time. This paper discusses the process of adapting an automated scoring engine designed for scoring essay responses to scoring responses on constructed response items. The research suggests that assessment design and rubric design have an effect on the reliability of an automated text scoring engine.

Background & Research Questions

There has been a growing focus on the concept of computational thinking (CT), both within and outside of computer science (CS) (e.g., Grover & Pea, 2013; NRC, 2012; NRC, 2010). Much of this work has been about defining computational thinking and creating learning experiences to help students develop it as a practice. However, without valid measurements, teachers cannot act on what students do and don't know in order to improve student learning.

Exploring Computer Science (ECS) is an academically rigorous yet engaging course that teaches problem-solving skills and computational thinking practices (CTP) (Margolis et al., 2012). ECS has begun to scale rapidly in response to demand from states, the CS industry, and CS education leaders. As part of this scale up, one cumulative and four unit assessments have been developed (under NSF funding) that measure the CT practices taught in ECS. The assessments are designed to measure how students engage in the practices and are not designed to focus on more procedural and factual knowledge. In order to accomplish this, the assessments rely on a series of open-ended constructed response tasks.

The scoring of open-ended tasks can be time-consuming, which makes it difficult to provide information to teachers about their students' performance soon after the students' have taken the assessment, which is when that information would be most useful to the teacher. With the increase in the development of on-line assessments, there is more opportunity for automated scoring of responses. While multiple choice items, and items in which the responses are constrained, such as numeric responses where students must enter a number, are straightforward to score, it is not straightforward to automatically score constructed response items. This paper discusses the use of an automated scoring engine to address the issues of scoring constructed response items.

Unit assessments for Exploring Computer Science

This analysis focuses on the first two of the unit assessments that were developed as part of the NSF funded Principled Assessment of Computational Thinking (PACT) suite of projects. Unit 1, Human Computer Interaction, provides a general coverage of computer science topics and Unit 2, Problem Solving, covers basic problem solving techniques and an introduction to algorithms. The assessments were developed using an Evidence-Centered Design process (Rutstein, Snow & Bienkowski, 2014) with a focus on the practices involved in computer science. The assessment for the first unit contains 6 multi-part tasks focusing on general computer science concepts such as: what is a computer, what are legal, ethical and privacy concerns of computational innovations, and what are ways to represent data. The Unit 2 assessment contained 4 tasks focused on the evaluation, comparison and development of algorithms or problem solutions. Each task contained multiple items, which were mostly open ended items, or multiple choice with an explain their choice questions. The assessments are currently paper/pencil assessments, and the answers were transcribed in order to have a digital copy of the student responses to the items.

Automated Text Scoring Engine

SRI's Automated Text Scoring Engine (ATSE) is a trainable, domain-independent software system that learns to assign numeric scores to texts. The engine uses advanced text analysis algorithms to identify features of interest in texts, such as word or phrase meanings and discourse relationships. Then, using a machine learning architecture and a training set of hand-scored example texts, the engine learns by example to assign scores based upon the identified features. The ATSE engine was initially developed at SRI as part of an evaluation study, the Literacy Courseware Challenge (LCC, funded by the Gates foundation), investigating the efficacy of literacy courseware products. The engine uses a technical approach informed by an analysis of the published approaches of winners of the Hewlett Foundation's Automated Essay Scoring competition, hosted on Kaggle[1]. SRI developed ATSE by selecting and combining the approaches used by the competition's winners, and enhancing it with some of SRI's key text analytic and machine learning innovations.

Our data in the LCC scoring evaluation consisted of approximately 4000 essays, normally 1-3 paragraphs in length, generated in response to 8 different prompts (giving about 500 essays per prompt). About 2500 of these were scored by hand (about 300 per prompt) according to 6 separate traits (introduction, conclusion, coherence and sequencing, relevance, language use, and written conventions) designed to align to the Common Core State Standards. All essays were scored using the same 6-trait rubric with scores from 0 to 4 for each trait. A scoring model was trained separately for each prompt and used only to score responses to that prompt.

Applying ATSE to these data, we found that by using 150 hand-scored responses as training data, the average agreement (across traits and rubrics) between the automatically predicted scores and human consensus scores was 93% of the level of inter-rater agreement between human scorers. For some of the traits this level was as high as 100%, meaning that the system’s accuracy was statistically no worse than agreement between independent human scorers.

One key innovation used within ATSE's approach is its lack of dependence upon pre-trained models of language use. Traditional natural language processing approaches rely upon large collections of annotated data, which are used to train models for such things as parsing and part of speech tagging.

[1] https://www.kaggle.com/c/asap-aes; https://www.kaggle.com/c/asap-sas

SRI's algorithms, which are the result of more than a decade of advanced research on unsupervised machine learning and computational linguistics, do not require pre-trained background models. Instead, our algorithms study the corpus of texts to be analyzed and automatically infer syntactic and semantic patterns directly from them. This allows ATSE to be applied across any domain and text type, without risk of misalignment between background models and the texts to be scored.

Application of ATSE to the ECS assessments

As part of the PACT project, we wanted to explore options to provide support for scoring the assessment. While scoring rubrics have been released along with the assessments to provide support for teachers to score, these rubrics do not fully address the issue that scoring the assessments can be very time-consuming. In addition, even with rubrics it is difficult to ensure the reliability across teachers or other scorers of the scores on the assessment. While inter-rater reliability may not be an issue for an individual teacher who is generating scores for their own students, this does become an issue for any type of comparison across classrooms. In order to address these issues, we decided to explore whether or not ATSE could be used to score the short constructed response items on these assessments. Our goal was to see if we could obtain comparable reliability with ATSE scoring as with human scorers.

Method and Analysis

In the first phase of our study, the goal was to understand how accurately the ATSE scoring engine would score the PACT assessments without modification. Though ATSE was designed to score longer essay-like responses using rubrics geared toward writing proficiency, we hypothesized that it would perform moderately well because many of the text analysis algorithms it uses are generically applicable across a range of response types.

Our first test of the scoring engine involved testing on 98 Unit 1 assessments and 120 Unit 2 assessments that were administered over the 2014-2015 school year, each of which have been scored by at least two scorers with a final score being determined by a third scorer any time there was a disagreement. We applied the ATSE system to the PACT assessments using a five-fold cross-validation scoring regime. Using this method, the responses were randomly partitioned into five sets. Then, in each of five separate rounds of scoring, one of the five sets (20% of the responses) are scored using a scoring model that has been trained using the other 80% of the data. The system’s accuracy is then determined by comparing the automated scores to the manual scores. Because of the cross-validation regime, the measurements indicate how well the system performs given a random 80% of the total number of responses. This method was applied for each of the individual items on the assessments. This makes it so that each scoring model is item-specific. The engine is trained using only the scores and responses for a specific item, and then used only to score responses to that same item. For the scoring of each item, a unique random five-fold partitioning of responses is used.

The version of ATSE used in the first round of PACT scoring was unchanged from the version applied to the LCC assessments. The PACT data exposed some technical problems that we hadn't encountered with the LCC data. First, there were some items for which the manual scores had very little or no variance (most or every student received the same manual score). This was not a condition the system had previously encountered, and some handling of such special cases needed to be implemented and added the system. Also, some pre-processing steps were added to allow for the aggregation of multiple response fields into a single response. This was necessary to accommodate the multi-part items.

After initial tests, some items were dropped from the remainder of the study. Four items were not included because of lack of variance in manual scoring. For these four items, fewer than 10 responses did not receive the most frequently assigned score. For item 2.1 from Unit 1 (item 1.2.1), students were asked to pick a set of instructions from a set of instructions and explain why a computer would have difficulty following this instruction. For this item, most students were able to get full credit. For task 3 from Unit 1 (items 1.3.1, 1.3.2 and 1.3.3) students discussed the benefits and issues around using social media sites for communication. These again were questions that most students got correct. This lack of variance meant that with five-fold validation, there were often only one or two responses in the training set that were scored differently from the rest. This represents too little information on which to train a machine learning based scoring system.

We also eliminated four other responses because they were multiple choice items rather than constructed response items (1.1.1, 1.5.1, 1.6.2, 2.3.4). These are items that can be scored automatically without using ATSE. After eliminating these eight items, 24 items remained and became the subject of our automated scoring analysis. All information presented henceforth concerns only these remaining 24 items.

The accuracy of automated scoring was measured using two statistics: (1) quadratic weighted kappa (see Vaughn and Justice, 2015 for a discussion on the use of this measure in recent evaluations of machine scoring) and (2) scaled root mean squared error (*sRMS*). Quadratic weighted kappa is also used to measure inter-rater reliability between independent scorers ($K_{q\ rater}$). For system performance, we measure agreement between the automated scores and the human rater consensus scores ($K_{q\ auto}$).

Using the same statistic for both inter-rater reliability and system performance also allows us to directly compare system performance against a human agreement ceiling ($K_{q\text{ auto}}$ / $K_{q\ rater}$). A value of 1.0 for this ratio would indicate optimal engine performance, since we would not expect the scoring engine to do any better than human agreement. For the sRMS statistic, scores are scaled as a proportion of the maximum possible score, i.e. (*score* / *maximum possible score*). Table 2 shows summary results of our scoring engine evaluation, along with summary statistics for the analyzed responses.

Table 1: Summary statistics, mean ATSE performance ($K_{q\ auto}$, $K_{q\ auto}$ / $K_{q\ rater}$, and *sRMS*), and mean inter-rater reliability ($K_{q\ auto}$) for items in each unit and for all items in both units.

Statistic	Unit 1	Unit 2	Both Units
Mean quad-weighted kappa: auto vs. consensus ($K_{q\ auto}$)	.446	.675	.570
Mean quad-weighted kappa: inter-rater reliability ($K_{q\ rater}$)	.754	.805	.782
Mean relative agreement score ($K_{q\ auto}$ / $K_{q\ rater}$)	.586	.825	**.715**
Mean scaled root mean squared error (*sRMS*)	.338	.332	.334
Total number of non-empty responses	97	120	*N/A*
Mean number of words per response	31	23	27

These results suggest that the system is performing well, but with room for improvement. The engine-predicted scores agree on average with the consensus scores at about **71.5%** of the rate of agreement between independent human scorers. While not ideal, these results are certainly not insignificant. This is especially true considering that for Unit 1, the scoring engine used only 76 training examples (80% of the total 97 responses were used in each round of cross-validation). For Unit 2, the amount of training data in each round was slightly higher, with 96 training examples in each round (80% of 120 total responses). These results are moderately lower than the performance we observed when using ATSE to score the LCC assessments. In that evaluation, we found that ATSE achieved 87% relative to human agreement with 50 training examples. Increasing the training size to 150 examples, the system performance was statistically no worse that human inter-rater agreement. We hypothesize a similar learning curve would be achieved with the PACT assessment data.

For the second phase of the project, we wanted to explore how we could improve the performance of the ATSE system for the PACT assessments. We conducted a diagnostic study of the scoring results, looking specifically at where the system did better or worse than expected. We began by looking at correlations between automated scoring accuracy and general characteristics of each item and the human scoring of the item. As expected, we found strong correlations with inter-rater reliability, the distribution of human scores for the item, and characteristics of responses elicited by the item.

To identify these correlations, we calculated the following statistics for each item: inter-rater reliability ($K_{q\ rater}$), scaled variance of the human scores, entropy of the human scores, mean number of words in the response, and number of examples used to train each scoring model. Table 2 shows these statistics for each of the 24 considered items. Then, using ordinary least squares, we estimated two linear models, one to predict $\mathrm{K}_{\mathrm{q\ auto}}$ from the four descriptive statistics, and another to predict sRMS.

Table 2: Reliability ($K_{q\ rater}$), score distribution (*variance*, *entropy*) and other summary statistics (*words*, *examples*) by item, along with per-item engine scoring accuracy ($K_{q\ auto}$, *sRMS*).

Item	$\mathrm{K}_{\mathrm{q\ rater}}$	variance	entropy	words	examples	$\mathrm{K}_{\mathrm{q\ auto}}$	sRMS
1.1.2	0.747	0.065	1.327	38	76	0.426	0.227
1.2.2	0.670	0.249	0.691	11	76	0.376	0.556
1.2.3	0.548	0.250	0.693	22	76	0.299	0.592
1.2.4	0.845	0.238	0.670	38	76	0.733	0.352
1.3.4	0.727	0.170	0.522	23	76	0.288	0.443
1.4.1	0.717	0.098	0.926	34	76	0.333	0.313
1.4.2	0.799	0.092	1.027	52	76	0.476	0.273
1.5.2	0.884	0.049	0.203	21	76	0.480	0.203
1.5.3	0.645	0.058	0.232	22	76	0.421	0.227
1.5.4	0.808	0.073	1.449	55	76	0.497	0.225
1.6.1	0.910	0.145	1.287	23	76	0.579	0.306
2.1.1	0.683	0.247	0.687	34	96	0.593	0.449
2.1.2	0.795	0.247	0.687	18	96	0.582	0.458
2.2.1	0.849	0.130	1.046	32	96	0.732	0.243
2.2.2	0.944	0.219	0.795	32	96	0.916	0.189
2.2.3	0.883	0.037	0.552	30	96	0.645	0.140
2.3.1	0.693	0.208	0.606	29	96	0.610	0.389
2.3.2	0.915	0.179	0.992	10	96	0.897	0.189
2.3.3	0.944	0.226	0.809	9	96	0.880	0.225
2.3.5	0.956	0.215	0.991	21	96	0.917	0.189
2.4.1	0.692	0.249	0.690	20	96	0.509	0.494
2.4.2	0.755	0.154	1.091	26	96	0.653	0.307
2.4.3	0.727	0.250	0.692	21	96	0.630	0.430
2.4.4	0.626	0.244	0.680	20	96	0.211	0.608

Interestingly, we found highly significant and very strong correlations between the statistics and the predicted scoring accuracy. For example, the Pearson correlation between human score variance and *sRMS* was 0.71. Using the linear model estimated from all four statistics, correlation between the model-predicted and actual *sRMS* values was **0.90**. Similarly strong predictions were obtained for $K_{q\ auto}$, with correlation of 0.79 for $K_{q\ rater}$ and 0.20 for entropy, and **0.91** for full linear model predictions. This result in itself is notable. It suggests that one can predict very accurately how well the scoring engine will do, using only characteristics of the human scoring, data quantity, and rudimentary characteristics of the responses (e.g., mean word count) as predictors.

Using this analysis, we were able to identify items on which the scoring engine did better or worse than expected (i.e., better or worse than the accuracies predicted by the linear models). We selected items at either end of the scale to study more closely — some where the system did much better than predicted (items 1.2.4, 2.3.5, 2.2.3), and some where the system did much worse than predicted (1.3.4, 1.5.2, 2.4.3).

For the three items on which the system did much better than expected, a consistent pattern emerged. Item 1.2.4 is a two-part question where students first answer yes or no as to whether or not a computer was “smart” enough to make up its own instruction and then explain their answer. Either yes or no were valid answers, but students were less likely to explain the “yes” answer appropriately. Out of the 29 students that responded “yes” only 3 of them received credit for their answer, compared to 56 students received credit out of the 68 students that responded “no”. The scoring engine was able to leverage the relationship between yes/no and the score without having to put much weight on the explanation. This suggests that the explanation might not be providing additional information about the student that is not already shown in the response, which in turn suggests possibly revising the item to only be multiple choice, or to change the item so that the explanation provides additional information about the student.

For item 2.3.5, an item where the system did better than expected, we found another instance of a multi-part item with an initial multiple-choice part. For this item students are asked to pick which of two algorithms meets the most conditions and to explain their answer. In this case, the multiple choice part was scored as either correct or not, as there was a clear benefit to one of the algorithms. Similarly, item 2.2.3 is a multi-part item for which the first part of the response is a single number, which was scored as correct or incorrect. The second part is the student’s explanation of how they found the number. For both 2.2.3 and 2.3.5, the score for the explanation was dependent on the score for the multiple choice item, and the scoring engine learned to identify a correct answer to the multiple choice part and was able to boost its scoring accuracy as a result. In both of these cases, the scoring engine likely took advantage of the fact that the multiple choice portions were strong predictors of the score. While this is not a detriment to the assessment, it does mean that we have less insight into how well the system would perform if they were only given the constructed response portion of the task.

Item 2.4.3 is a case where the scoring engine does much worse than expected. One notable characteristic of this item is that the students are asked in the prior item to invent an organizing procedure and then in this item to explain the procedure. The nature of this task (having students develop their own algorithm) led to a very wide variety of responses. This variety was likely too great for the scoring engine to ascertain strong patterns, and therefor had difficulty scoring responses outside of the training set. Item 1.5.2, on which the system also performed worse than expected, had similar characteristics. In this case, students were asked in an earlier item (within the same task) to list two communication methods they would use to tell friends about a funny thing they saw. Then, in this item, they were asked to provide an explanation of their choice. The resulting diversity of responses to both items was a likely cause of poor scoring engine performance on this item. One possible way to increase the reliability of the scoring of these items may be to join the response for the current item to the response on the related previous item. This would allow the scoring engine to observe both responses together, and measure important characteristics like relevance between the two responses.

In the final part of our study, we looked at specific examples where the engine failed to score a response correctly, focusing our study on item 1.3.4, on which the engine performed much worse than expected. On this item, the question was "Describe one legal, ethical, or privacy concern from using social media (such as Facebook or a blog)." The responses to this item are clearly diverse, but the responses tend to be short. Here are some examples of correct responses that were erroneously scored as incorrect by scoring engine:

- "People could copy your Identity and pretchd to be like you"
- "They can secretly access your messages and front camera."
- "Someone could put someones credit card stuff or something on there"

Notably, we observed that the human scorers marked 21 out of 97 responses as incorrect, whereas the scoring engine, on the other hand, marked only 11 as incorrect. We deduce from this pattern, and the nature of the responses, that there was likely an insufficient number of responses presented to the system where the students tried to reason an answer but failed. Instead, the scoring system likely attributed only trivial indicators of incorrectness to a false score, such as an empty or extremely cursory responses, such as with the correctly scored incorrect response “The things or pages more can be biased.”

Our hope in doing this more detailed analysis of specific errors was to diagnose the specific behaviors of the scoring engine that would suggest opportunities for scoring engine improvement. However, our analyses typically indicated that it was the diversity of scores, and of responses, in general that were the most powerful predictors of scoring accuracy. It is difficult at this stage to isolate the specific linguistic features that might provide the necessary additional evidence to boost system performance on these challenging items. Instead, assessment design, hand scoring, and endeavoring to implement system features that make use of very small numbers of training examples are the most likely avenue to improving performance overall.

Discussion

The analysis above highlights that point that the degree to which an automated scoring engine performs in not only dependent on the scoring engine itself, but also dependent on the items themselves and the scoring rubrics. Since ATSE learns from the training cases, it can only be as good as the human scorers that created the initial training set. For items in which there is a large degree of disagreement between the human scorers, the automated scoring is not going to do as well. One way to aid the reliability between humans is to develop strong rubrics that clearly define the borders between score points and includes discussion of the border cases.

Another aspect of the assessment that affects the reliability of the automated scoring is in the structure of the items that are developed. These results suggest that when designing a multi-part item with both a non-constructed and constructed part, each part ought to be separately scored. This way, the relationship between the constructed portion of the response and the training signal (human score) relating to that constructed part can be more easily identified by the engine, due to being isolated from the multiple choice part.

This deconstruction of the multiple choice section from the constructed response section could help as long as there is a range of scores for the constructed response section. Items which do not have a lot of variability in how students will respond may be difficult for ATSE to score, as there are not enough examples of responses at all of the different scores points included in the training set. This implies that an assessment designer needs to think about the purpose of the assessment, and the information that they hope to gain from the items. If the purpose of the assessment is to differentiate the students, then it would be appropriate to have constructed response items in which a range of scores is expected. However, for assessments or items in which most students are expected to perform well it may not be as appropriate to use the constructed response items if the responses are to be scored automatically. Since the scoring engine may have more difficulty recognizing the rare incorrect responses, the reliability of the automated scoring may be lower than with other constructed response. It may be more appropriate to determine if there are other formats that are easier to score that could be used instead.

One issue with the current work is the lack of data. Even though we started with about 120 responses for the Unit 1 data, there were a number of blank responses for each item and so we only had complete data for a small portion of students. Needless to say the algorithm should work better (have a higher reliability) with more data to train on, particularly for the cases in which the item difficulty was too high or too low. When using automated scoring it is important to think about the training set that you have, to make sure there is a range of responses that can be used to train the scoring engine.

Overall, there are two ceilings to the performance of the ATSE -- one ceiling on how much you are learning about the student skills/knowledge based on the item, and another ceiling on how much the engine can learn about scoring the responses to assess that knowledge. Increasing ground truth score entropy through assessment design choice and strong rubric design, will increase both ceilings and increase the reliability of the ATSE.

Future Work and Conclusion

While currently ATSE is not providing the reliability that we were able to obtain in the LCC project, we are seeing that there is promise in this work. In our next phase of this work we are collecting data from a larger sample in order to explore whether or not a larger sample size will improve the reliability, and if so how much more is needed. We will also continue to explore items that are not performing well to determine if there are improvements that can be made to either the assessment items or the scoring engine in order to improve reliability of ATSE.

The benefit of automated scoring is that teachers and researchers can use it obtain feedback on how their students performed on each of the items relatively quickly. Instead of waiting for days or months for human scorers to score, the engine can be run directly after the assessment is administered and results can be provided to stakeholders. The convenience of scoring could encourage the development and use of more constructed response items which will aid assessments in measuring deeper conceptual knowledge and practices that have typically been harder to measure.

References

Bienkowski, M., Snow, E. B., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computational thinking practices: A first look*. SRI International. Retrieved from http://pact.sri.com/resources.html

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Margolis, J., Ryoo, J. J., Sandoval, C. D. M., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads*, *3*(4), 72–78.

National Research Council. (2010). *Report of a workshop on the scope and nature of computational Thinking*. Washington, DC: The National Academies Press.

National Research Council. (2012). *Report of a workshop on the pedagogical aspects of computational thinking*. Washington, DC: National Academies Press.

Rutstein, D. W., Snow, E., & Bienkowski, M. (2014). Computational thinking practices: Analyzing and modeling a critical domain in computer science education. Paper presented at the annual meeting of the American Educational Research Association (AERA), Philadelphia, PA.

Vaughn, D., & Justice, D. (2015). On the Direct Maximization of Quadratic Weighted Kappa. *CoRR*, abs/1509.07107