# Context-based Automated Scoring of Complex Mathematical Responses

**Aoife Cahill, James H. Fife, Brian Riordan, Avijit Vajpayee, and Dmytro Galochkin**
acahill@ets.org, ejreed215@verizon.net,
{briordan, avajpayee, dgalochkin} @ets.org
ETS, Princeton, NJ 08541, USA

## Abstract

The tasks of automatically scoring either textual or algebraic responses to mathematical questions have both been well-studied, albeit separately. In this paper we propose a method for automatically scoring responses that contain both text and algebraic expressions. Our method not only achieves high agreement with human raters, but also links explicitly to the scoring rubric – essentially providing explainable models and a way to potentially provide feedback to students in the future.

## 1 Introduction

In this paper we present work on automatically scoring student responses to constructed-response mathematics items where the response should contain both text and mathematical equations or expressions. Existing work on automated scoring of mathematics items has largely focused on items where either only text is required (c.f. related work on automated short-answer-scoring (Galhardi and Brancher, 2018; Burrows et al., 2015)) or only an expression or equation is required (Drijvers, 2018; Fife, 2017; Sangwin, 2004). This is the first work, to our knowledge, that attempts to automatically score responses that contain both.

Items that elicit such responses could be algebra, trigonometry, or calculus items that ask the student to solve a problem and/or provide an argument. Items at levels much below algebra most likely would not require the student to include an equation – at least one that requires an equation editor for proper entry – in the text, and items at a higher level might require the student to include abstract mathematical expressions that would themselves present automated scoring difficulties. These kinds of items are quite common on paper-and-pencil algebra exams. However, they are less common on computer-delivered exams, primarily because the technology of calling up an equation editor to insert equations in text is new and not generally used.

The challenge with automatically scoring these kinds of responses, in a construct-valid way, is that the system needs to be able to interpret the correctness of the equations and expressions *in the context of* the surrounding text.

Our goal is not just to achieve accurate scoring but to also have explainable models. Explainable models have a number of advantages including (i) giving users evidence that the models are fair and unbiased; (ii) the ability to leverage the models for feedback; and (iii) compliance with new laws, e.g. the General Data Protection Regulation (EU) 2016/679 (GDPR) which requires transparency and accountability of any form of automated processing of personal data. In this paper we present an approach that not only achieves high agreement with human raters, but also links explicitly to the scoring rubric – essentially providing explainable models and a way to potentially provide feedback to students in the future.

## 2 Data

In this paper we use data from 3 pilot-study items that elicited responses containing both textual explanations as well as equations and expressions. An example item is given in Figure 1, and a sample response (awarded 2 points on a 0-3 point scale) is given in Figure 2.[1] The pilot was administered as part of a larger project in four high schools located in various regions of the United States. The items assumed one year of algebra and involved writing solutions to algebra problems, similar to what a student would be expected to write on a paper-based classroom test. Responses were collected digitally;

[1] This item corresponds to Item 2 in our dataset. The scoring rubric is given in Appendix A.1.



*Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 186–192
July 10, 2020. 

Explain, using words and equations, how you would use the quadratic formula to find two values of x for which

$$195 = -2x^2 + 40x.$$

You may also use the on-screen calculator.

Figure 1: Sample item that elicits textual explanations as well as equations and mathematics.

$x = \frac{-40+\sqrt{40^2-4(-2)(-195)}}{2(-2)}$

To solve this you must first put your equation in standard form, which gives you y=-2x+40x-195. You then plug your a, b, and c values into the quadratic formula. To start finding your x value, you must first multiply all your values in parentheses. You must then simplify the square root you get from multiplying. With your new equation, you make two more equations, one adding your simplified square root and one subtracting it. The two answers you get from those equations are your two values of x.

Figure 2: Sample response to the item in Figure 1 (2-point response). The student has put the equation into standard form with a slight error. $-2x^2$ has become $-2x$; the student was not using the equation editor and could not type the exponent. The student does not explicitly give the values of a, b, and c, but correctly substitutes these values into the formula, so we may assume that the student has determined these values correctly. We may also assume that the student has corrected the missing exponent in the standard form. The student talks about "two answers" but only gives one root, however, so this response is worth 2 points.

students used an interface that included an optional equation editor. The responses were captured as text, with the equations captured as MathML enclosed in <math> tags. Two of the items involved quadratic functions, requiring the student to use the equation editor to properly format equations in their responses. Nonetheless, many students did not use the equation editor consistently. In fact only 60% of all students used the equation editor. Of all equations entered by the students, only 34% were entered via the equation editor since most of the students preferred to write simple equations as regular text.[2]

There were over 1,000 responses collected for each item, however some responses were blank and therefore not included in this study. Table 1 gives some descriptive statistics for the final data used in this study. Items 2 and 3 were somewhat difficult for this pilot student population, with 71% and 78% of students receiving a score of 0 for those items. All responses were scored by two trained raters; the quadratic-weighted kappa values for the human-human agreement on the three items ranged from 0.91 to 0.95, indicating that humans were able to agree very well on the assignment of scores.

| Item | Total | % 0 | % 1 | % 2 | % 3 |
|---|---|---|---|---|---|
| 1 | 924 | 49.35 | 19.37 | 6.93 | 24.35 |
| 2 | 889 | 70.97 | 12.49 | 11.59 | 4.95 |
| 3 | 859 | 77.65 | 3.49 | 3.26 | 15.6 |

Table 1: Descriptive Statistics for the 3 items, including the total number of responses per item, as well as the percentage of responses at each score point.

## 3 Methods

### 3.1 Automatically scoring equations and expressions

We use m-rater, an automated scoring engine developed by Educational Testing Service (Fife, 2013, 2017) to automatically score the equations and mathematical expressions in our data. M-rater uses SymPy[3], an open-source computer algebra system, to determine if the student's response is mathematically equivalent to the intended response. M-rater can process standard mathematical format, with exponents, radical signs, fractions, and so forth. M-rater is a deterministic system, and as such has 100% accuracy, given well-formed input.

If, as in this study, the responses consist of a mixture of text and equations or mathematical expressions, m-rater can evaluate the correctness (or partial correctness) of the equations and expressions, but it cannot evaluate text.

### 3.2 Automatically identifying equations and expressions in text

While the students had access to an equation editor as part of the delivery platform, many did not use it consistently. This means that we cannot rely on the MathML encoding to identify all of the equations and mathematical expressions in the text. For example, a student may have wanted to enter the equation: $2x^2 - 40x + 195 = 0$. They may use the equation editor to enter the entire equation, or

[2]This presents obvious challenges for automatically scoring the mathematical components of the responses, since the first step is to even identify them (see Section 3.2 for how we address this).

[3]https://www.sympy.org/en/index.html

some of it (e.g. the piece after the = sign, or after the exponent expression), or none of it. This leads to construct-irrelevant variations in representations.

Therefore, we develop a regular-expression based system for automatically identifying equations and expressions in responses where all data from the equation editor has been rendered as plain text. Our processing includes the following assumptions which are appropriate for our dataset:

- Variables can only consist of single letters;
- We only detect simple functions (square root, absolute and very basic trigonometric functions);
- Equations containing line breaks are treated as two different equations.

We processed all responses to the three pilot items with this script and all identified equations and expressions were manually checked by a content expert. In almost all cases, the system correctly identified the equations or expressions. There were 9 incorrectly identified equations in total (out of 2,672). Mis-identifications were usually due to incorrect spacing in the equation – either too much space between characters in the equation or no space between the equation and subsequent text. A few students used the letter x to denote multiplication, which was read by the system as the variable x.

It is possible to convert the m-rater evaluations of the individual equations and expressions contained in a response into features. This is done by automatically extracting the equations and expressions and using m-rater to match each one to an element in the scoring rubric (also called concepts). These features encode a binary indicator of whether a particular concept is present or not in a response. Note that some concepts represent known or expected misconceptions in student responses. For example, the set of six binary features instantiated for each response to Item 2 are as follows: (i) has the equation been correctly transformed into standard form (rubric element 1); (ii) did the student answer a=2 (rubric element 2); (iii) did the student answer b=40 (rubric element 2); (iv) did the student answer c=195 (rubric element 2); (v) did the student find solution 1 (rubric element 3); (vi) did the student find solution 2 (rubric element 3).

### 3.3 Automatically scoring short texts for correctness

We use 4 approaches for automatically scoring short texts with mathematical expressions. The baseline system (LinReg$_m$) is an ordinary Linear Regression on the math features automatically extracted from m-rater evaluations and does not include any textual context. System 2 (SVR$_{csw}$) is a feature-based Support Vector Regressor (SVR) that encodes (1) key words and phrases (in the form of word ngrams); (2) character-ngrams as well as (3) key syntactic relationships in the text as binary features. Note that system 2 does not take any explicit math features into account, and the mathematical expressions are assumed to be captured through character level features. System 3 (SVR$_{msw}$) is a feature-based SVR taking into account both textual context (through word-ngrams and syntactic dependencies) as well as explicit math features, but no character-level ngrams. Our final system is a recurrant neural network (RNN) system. The RNN model uses pre-trained word embeddings encoded by a bidirectional gated recurrent unit (GRU). The hidden states of the GRU are aggregated by a max pooling mechanism (Shen et al., 2018). The output of the encoder is aggregated in a fully-connected feedforward layer with sigmoid activation to predict the score of the response. This architecture has achieved state-of-the-art performance on the ASAP-SAS benchmark dataset (Riordan et al., 2019). Additional information about steps to replicate the system can be found in the Appendix.

## 4 Experiments

We conduct a set of experiments to answer the following research questions:

1. How important is textual context for responses involving mathematical expressions with respect to automated scoring? (Comparing **Exp 0** and **Exp 1**)
2. Do character level features capture mathematical expressions? (**Exp 0**)
3. Can explainability be included in scoring models without severely compromising accuracy? (Comparing **Exp 0** to **Exp 1–3**)

For our baseline experiment (**Exp 0**), student responses are taken with all equations and expressions converted to plain text. For this experiment,

| System | Item 1 | Item 2 | Item 3 |
|---|---|---|---|
| LinReg$_m$ | 0.506 | 0.457 | 0.587 |
| SVR$_{csw}$ | 0.870 | 0.789 | 0.933 |
| SVR$_{msw}$ | **0.897** | 0.797 | **0.935** |
| Word RNN | 0.887 | **0.835** | 0.923 |

Table 2: Quadratically-weighted kappa results for **Exp 0** (plain text, no expression replacement)

| System | Exp 1 | | | Exp 2 | | | Exp 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Item 1 | Item 2 | Item 3 | Item 1 | Item 2 | Item 3 | Item 1 | Item 2 | Item 3 |
| SVR$_{msw}$ | 0.888 | 0.783 | 0.897 | 0.891 | 0.776 | 0.889 | 0.894 | 0.781 | 0.894 |
| SVR$_{csw}$ | 0.788 | 0.593 | 0.664 | 0.827 | 0.689 | 0.867 | 0.882 | 0.776 | 0.891 |
| Word RNN | 0.767 | 0.649 | 0.725 | 0.842 | 0.75 | 0.887 | 0.901 | 0.829 | 0.888 |

Table 3: Quadratically-weighted kappa results for explainability experiments

we use all 4 systems as described in Section 3.3. Subsequently, we perform 3 experiments where all expressions and equations (as identified by m-rater) are converted to pre-defined tokens with increasing degree of explainability:

**Exp 1** All equations and expressions automatically identified and converted to a single token (@expression@)

**Exp 2** All equations and expressions automatically identified and converted to one of @correct@ or @incorrect@. The correctness of an equation is determined automatically by matching against the scoring rubric using m-rater (see Section 3.1).

**Exp 3** All equations and expressions automatically identified and converted to one of @correct_N@ or @incorrect@, where N indicates the set of concept numbers from the scoring rubric and is automatically identified using m-rater.

For each pair of system and response variant, we conduct a 10-fold nested cross validation experiment. We split our data into 80% train, 10% dev and 10% test. For each fold, we train on the train+dev portions and make predictions on the held-out test portion, having tuned the hyperparameters on the dev set. There are no overlapping test folds. For evaluation, we pool predictions on test sets from all folds and compute agreement statistics between the rater 1 score and the machine predictions.

## 5 Results

Table 2 gives the results of all models used for the baseline experiment where all responses are converted to plain text. Even without pre-processing the mathematical expressions, textual context is very important, as we see by the poor performance of the Linear Regression model on purely mathematical features ($LinReg_m$). It can also be seen that character level features, while partially capturing mathematical expressions, do not perform as well as the SVR model with explicit math features (comparing $SVR_{csw}$ to $SVR_{msw}$). The difference, however, is not statistically significant for any item (details given in Appendix A.3). Another interesting result is that the RNN model without character level OR explicit math information performs well, being a close second to the SVR$_{msw}$ model and the differences between them are not statistically significant.

Table 3 gives the results for the explainability experiments i.e. **Exp 1 to 3** where mathematical expressions and equations were pre-identified and replaced in the response text. Comparing these with the results for the experiment on the original text responses (Table 2), it can be seen that the replacement that includes the mappings to rubric concepts (**Exp 3**) not only increases explainability but is also competitive in performance to models with explicit math features but no expression replacement (outperforming them on Item 1). Models SVR$_{csw}$ and WordRNN are not significantly different on any item for any of the 3 explainability experiments (**Exp 1 to 3**).

Coming back to our original research questions:

1. *How important is textual context for responses*

*involving mathematical expressions with respect to automated scoring?*
Context is important for automatically scoring responses that integrate text and algebraic information. Evaluating the mathematical expressions alone does not perform well (**Exp 0**). Additionally, **Exp 1** has no context for the mathematical expressions, and we see lower results for the system that still includes mathematical information as independent features, but out of context ($SVR_{msw}$), compared to systems that encode the mathematical information in some way *in context*.

2. *Do character level features capture mathematical expressions?*
Character level features certainly do capture a large portion of mathematical expressions. We see that in the **Exp 0** results, where there is no interpretation of the mathematical expressions, that systems perform almost as well as the systems that do explicit interpretation.

3. *Can explainability be included in scoring models without severely compromising accuracy?*

   Yes, we can include model interpretability without compromising scoring accuracy. The differences between the best models from **Exp 0** and **Exp3** ranged from -0.004 to +0.041). By explicitly linking aspects of the rubric to each response, we yield interpretable models that perform comparably to systems without this interpretative layer. Although the overall results are lower, they are not statistically significantly lower.

## 6 Conclusion

To summarize, this work presented a hybrid scoring model using a deterministic system for evaluating the correctness (or partial correctness) of mathematical equations, in combination with text-based automated scoring systems for evaluating the appropriateness of the textual explanation of a response.

We contribute the following:

1. Systems that produce extremely high agreement between an automated system and human raters for the task of automatically scoring items that elicit both textual and algebraic components
2. A method for linking rubric information to the automated scoring system, resulting in an more interpretable model than one based purely on the raw response


## Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. We would also like to thank Michael Flor, Swapna Somasundaran and Beata Beigman-Klebanov for their helpful comments.


## References


Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Rethinking Complex Neural Network Architectures for Document Classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4046–4051, Minneapolis, Minnesota. Association for Computational Linguistics.

Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25(1):60–117.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Paul Drijvers. 2018. Digital assessment of mathematics: Opportunities, issues and criteria. *Mesure et évaluation en éducation*, 41(1):41–66.

James H Fife. 2013. Automated scoring of mathematics tasks in the Common Core era: Enhancements to m-rater in support of CBAL™, mathematics and the Common Core assessments . *ETS Research Report Series*, 2013(2):i–35.

James H Fife. 2017. The m-rater Engine: Introduction to the Automated Scoring of Mathematics Items. *Research Memorandum, ETS RM-17-02*, pages 10–24.

Lucas Busatta Galhardi and Jacques Duílio Brancher. 2018. Machine learning approach for automatic short answer grading: A systematic review. In *Ibero-American Conference on Artificial Intelligence*, pages 380–391. Springer.

Peter Nemenyi. 1963. Distribution-free multiple comparisons,(mimeographed).

Brian Riordan, Michael Flor, and Robert Pugh. 2019. How to account for mispellings: Quantifying the benefit of character representations in neural content scoring models. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Chris Sangwin. 2004. Assessing mathematics automatically using computer algebra and the internet. *Teaching Mathematics and Its Applications: An International Journal of the IMA*, 23(1):1–14.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Melbourne, Australia. Association for Computational Linguistics.

Sudhir Varma and Richard Simon. 2006. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91.

# A Appendices

## A.1 Scoring Rubric for Item 2

- 1 pt. for writing the equation as $2x^2 - 40x + 195 = 0$ or $-2x^2 + 40x - 195 = 0$. It's acceptable to just write the expression $2x^2 - 40x + 195 = 0$ or $-2x^2 + 40x - 195 = 0$. It's also acceptable to say something like "Move 195 to the other side of the equation" if they find the correct values for $a$, $b$, and $c$ (with correct signs).

- 1 pt. for determining the values of $a$, $b$, and $c$. $a = 2$, $b = 40$, $c = 195$ OR $a = 2$, $b = 40$, $c = 195$ 0 pts. if they mix the values up (e.g., $a = 2$, $b = 40$, $c = 195$). 1 pt. if they implicitly complete this step by correctly substituting the correct values for $a$, $b$, and $c$ into the quadratic formula in the next step.

- 1 pt. for substituting the values of $a$, $b$, and $c$ into the quadratic formula and obtaining two solutions. Students do not need to simplify the answers. Students can write any equivalent expressions for the two values of $x$, including $x = \frac{40+\sqrt{40^2-4*2*195}}{2*2}$ and $x = \frac{40-\sqrt{40^2-4*2*195}}{2*2}$ OR $x = \frac{-40+\sqrt{40^2-4*-2*-195}}{2*-2}$ and $x = \frac{-40-\sqrt{40^2-4*-2*-195}}{2*-2}$. It's also acceptable for students to write $x = \frac{40\pm\sqrt{40^2-4*2*195}}{2*2}$ to mean both solutions. Or students may write that the two values of x are x = 11.5811... and x = 8.4188..., correct to at least one decimal place, provided they arrive at these numbers through the quadratic formula and not by solving the equation numerically.

- Max 2/3 for finding one correct solution.

- Max 2/3 for writing the two correct solutions with no explanation of where the values of $a$, $b$, and $c$ come from.

- 1/3 if the student provides an outline of the solution without actually carrying out any of the steps.

## A.2 Additional information for training the RNN model

The text is preprocessed with the spaCy tokenizer with some minor postprocessing to correct tokenization mistakes on noisy data. On conversion to tensors, responses are padded to the same length in a batch; these padding tokens are masked out during model training. Prior to training, responses are scaled to [0, 1] to form the input to the networks. The scaled scores are converted back to their original range for evaluation. Word tokens are embedded with GloVe 100 dimension vectors and fine-tuned during training. Word tokens not in the embeddings vocabulary are each assigned a unique randomly initialized vector. The GRUs were 1 layer with a hidden state of size 250. The network was trained with mean squared error loss. We optimized the network with RMSProp with hyperparameters set as follows: learning rate of 0.001, batch size of 32, and gradient clipping set to 10.0. An exponential moving average of the model's weights is used during training (Adhikari et al., 2019).

## A.3 Additional details on significance testing of results

Although nested cross-validation gives a fairly unbiased estimate of true error as shown by Varma and Simon (2006), we performed statistical significance testing to pair-wise compare 4 models for **Exp 0: no expression replacement** and 2 models for **Exp 3: expressions replaced with incorrect/correct along with concept numbers**.

Friedman's test as suggested by Demšar (2006) is run to compare 6 models (corresponding to treatments) across multiple repeated measures (10 folds) for each item individually. Note that such a setup of comparing multiple models across 10 folds on a dataset has to be regarded as non-independent data as even though the test folds will be distinct, the training data for each fold may partially overlap. Hence Friedman's test is appropriate here to

| | 0_SVR$_{csw}$ | 0_SVR$_{msw}$ | 0_WordRNN | 3_SVR$_{csw}$ | 3_WordRNN |
|---|---|---|---|---|---|
| 0_LinReg$_{m}$ | 1 / 3 | 2 / 3 | 3 / 3 | 1 / 3 | 2 / 3 |
| 0_SVR$_{csw}$ | - | 0 | 0 | 0 | 0 |
| 0_SVR$_{msw}$ | | - | 0 | 0 | 0 |
| 0_WordRNN | | | - | 1 / 3 | 0 |
| 3_SVR$_{csw}$ | | | | - | 0 |

Table 4: Pair-wise Comparisons of Models with fraction of datasets with significant difference between models

test whether any pair of models are statistically different.

Following Friedman's test, we do pair-wise post-hoc testing through Nemenyi's test (Nemenyi, 1963). Note that this testing is per-item and we report the fraction of times the differences were significant in table 4.