Assessing the Effectiveness of an Intelligent Tool that Supports Targeted Teacher Responses to Student Ideas

James P. Bywater
College of Education
James Madison University
MSC 6913, 395 S. High St.
Harrisonburg, VA 22807
bywatejx@jmu.edu

Jennifer L. Chiu
Curry School of Education and Human Development
University of Virginia
P.O. Box 400273
Charlottesville, VA 22903
jlchiu@virginia.edu

Ginger S. Watson, Ph.D.
Curry School of Education and Human Development
University of Virginia
P.O. Box 400273
Charlottesville, VA 22903
ginger.watson@virginia.edu

Keywords: Teacher feedback, intelligent tools.

Abstract

This paper reports on the design and development of an intelligent, natural language processing tool, the Teacher Responding Tool (TRT), that provides response recommendations to teachers to foster consistent, content-specific feedback based on student cognition. Placing student ideas at the center of instructional decisions promotes equitable teaching. Results indicate that the TRT selected accurate recommendations and that the interface promoted the teachers' thoughtful consideration of these recommendations. Future design recommendations are provided.

Introduction

Research in teacher education shows that placing student ideas at the center of instructional decisions is critical for promoting equitable student participation, achievement, and agency (NCTM, 2014). However, responding to students in the moment is complex. First, teachers must infer the current understanding of the student (Coffey, Hammer, Levin, & Grant, 2011). Second, teachers must prioritize which understandings to focus on for the sake of the student, the class, and the intended learning goals (Ball, 1993). Third, the delivery of the response must be student-specific, be given in manageable chunks, do more than highlight errors, and avoid comparisons with other students (Shute, 2008). Finally, teachers should ask questions that support further student discourse (Chapin, O'Conner, & Anderson, 2009).

Given this complexity, providing teachers with opportunities to develop these skills is important, and there have been calls to develop a variety of "approximations of practice" (Grossman, Compton, et al., 2009). This paper describes the design and development of an intelligent tool to scaffold teachers' skills at giving high quality, student specific feedback. The Teacher Responding Tool (TRT) is a natural language processing (NLP) tool grounded in design principles for worked examples and developing thinking skills (Clark & Mayer, 2016) that provides recommendations to support teachers while they respond to students. The TRT builds upon research with technologies that automatically respond directly to students (Aleven & Koedinger, 2002) and that provide teachers with insight into student thinking (McDonald, Bird, Zouaq, & Moskal, 2017).

Theoretical Framework

This study is informed by instructional design principles aligned with the cognitive theory of multimedia learning (Mayer, 2014). These research-based principles describe how to design for learning in contexts that involve text and images. While our design is text-based, the assumptions of this theory and many of the design principles that follow from it guided our design. For example, this study is aligned with the assumptions that learners have a limited capacity to process information, and that learners engage in active processing via selecting, organizing, and integrating text. We build upon those design principles that suggest limiting extraneous material, adding cues and highlights, using worked examples, and focusing on authentic job-relevant thinking skills (Clark & Mayer, 2016).

Design of the Teacher Responding Tool (TRT)

The TRT system consists of three interacting sub-systems: the training dataset creation sub-system, the natural language processing (NLP) recommendation engine sub-system, and the user interface sub-system. The TRT system is designed to be implemented within authentic learning contexts. To set-up the system for a given learning context, two steps need to be taken. First, the training dataset needs to be created and then used to train the NLP recommendation engine. The training dataset is created by consulting with the teacher users and collecting their prior or suggested responses to prior student explanations for the given context. Second, the TRT needs to be connected to the learning management system that the students will be using so that it can pull new student explanations after they have been written and push teacher responses back to students. When the TRT is in use, new student explanations are pulled from the learning management system and used to select feedback recommendations. The recommendations are presented to teachers via the user interface, and the teacher response is then made available to the students by pushing these responses to the learning management system. The TRT system is designed to include teachers into the feedback process and requires teacher user interaction. As such it is intentional that this system does not provide instantaneous feedback to students. Figure 1 illustrates the subsystems and overall system flow.

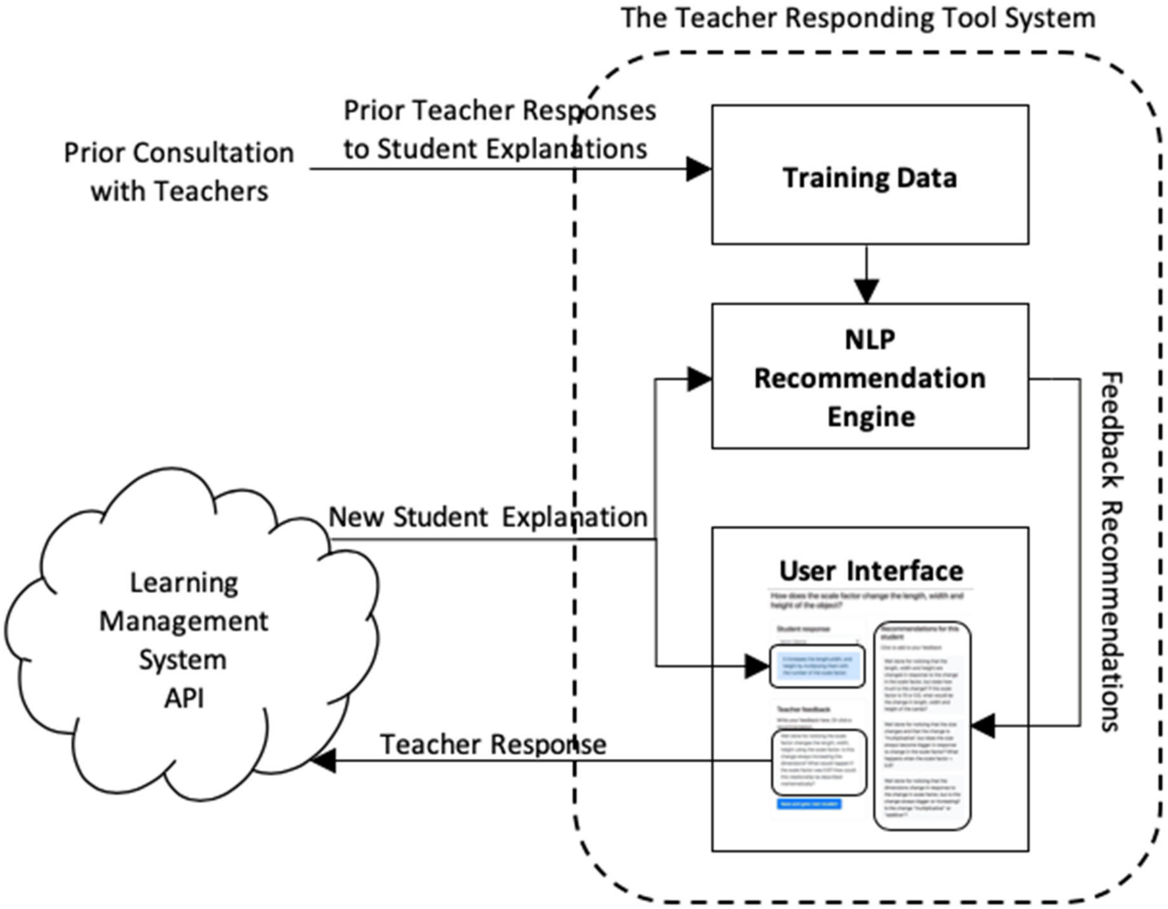

The Teacher Responding Tool System
Prior Consultation with Teachers
Prior Teacher Responses to Student Explanations
Training Data
NLP Recommendation Engine
Feedback Recommendations
Learning Management System API
New Student Explanation
User Interface
How does the scale factor change the length, width and height of the object?
Student response
Teacher feedback
Recommendations for this student
Teacher Response

Figure 1. The Teacher Responding Tool (TRT) system.

Training data subsystem

Training data creation should aim to use authentic student explanations. In our study we began with student explanations collected during a prior implantation of the project within the same context (i.e. same school, same course, similar student population). The teacher responses were created by: first, consulting with the teachers about how they would respond to each of the student explanations; second, drawing upon research finding for how to best provide formative feedback; and third, drawing upon researcher expertise to standardize the responses. This led to a training dataset consisting of pairs of student explanation and teacher response written in natural language text. Table 1 shows the first few records of training data used for one of the questions in this study. A different training dataset needs to be created for each question prompt.

Table 1. A sample of the 116 student explanation and teacher response pairs in the training dataset for the question prompt: "How does the scale factor change the length, width and height of the object?"

Student Explanation	Teacher Response
It gets bigger	Well done for noticing that there is a change, but what is "it"? Is the change always larger? What happens when the scale factor is 10 or 0.5?
It makes the size bigger. the scale factor times by the scale	Well done for noticing that the dimensions change in response to the change in scale factor and that change is "multiplicative", but is this change always bigger or increasing?

It multiplies the original length by the number of scale factor.	Well done for noticing that the length changes in response to the change in the scale factor and that the change is “multiplicative”, but what about the other dimensions – width and height?
Length and width and height multiplied by the 3	Well done for noticing that the dimensions changes and that the change is "multiplicative", but what change in the scale factor causes these changes? What happens to the dimensions when the scale factor is 0.5?
The bigger the scale factor the bigger the dimensions get.	Well done for noticing that the dimensions become bigger as the scale factor becomes bigger, but what happens when the scale factor is made smaller rather than bigger?
The new measurements are way higher than the old measurements, its shape has increased.	Well done for noticing that the size changes, but does the size always increase in response to change in the scale factor? What happens when the scale factor = 0.5? What is "it"? Expand on your answer.

We developed training datasets for each question that were on average about 100 records (pairs of student explanation and teacher response) long. Generally, larger datasets are known to improve the recommendation engine performance. However, factors such as the generality the question being asked and the number of different recommendations to choose from also impact how large the training dataset should be (Burrows, Gurevych, & Stein, 2015; Zehner, Sälzer, & Goldhammer, 2016). At the same time, the advantages of larger datasets are offset by the time and cost involved in creating them. Based on these considerations, we concluded that about 100 records represented a reasonable dataset size.

NLP recommendations engine subsystem

The NLP recommendation engine consists of two components: the preprocessing of student explanations, and a tf-idf (term frequency-inverse document frequency) model (see Figure 2). The recommendation engine is initialized using the training dataset. Each of the student explanations in the training dataset are preprocessed and then used to build the tf-idf model. The recommendation engine is used by querying the model with new student explanations that have been preprocessed in the same way, and the recommendations selected are outputted.

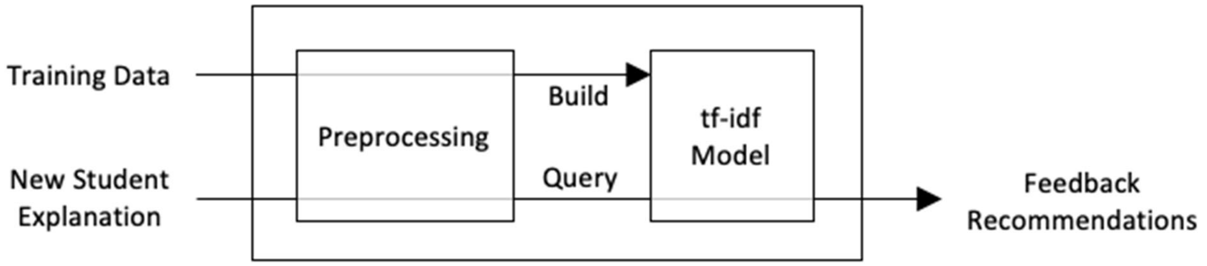


Figure 2. The natural language processing (NLP) recommendations engine subsystem.

Preprocessing involves separating a student explanation into individual words (tokenizing) and then applying automatic spelling correction to each word. Each word is then converted to its stem, so that words like “increases,” “increasing,” and “increased” all become “increase.” Finally, common high frequency words, or stop words, such as “a,” “the,” and “at” are removed. Tokenizing and stemming are performed using the Python *nltk* package, the spelling correction is performed using the Python *autocorrect* package, and the stop word list included in the Python *sklearn* package is used.

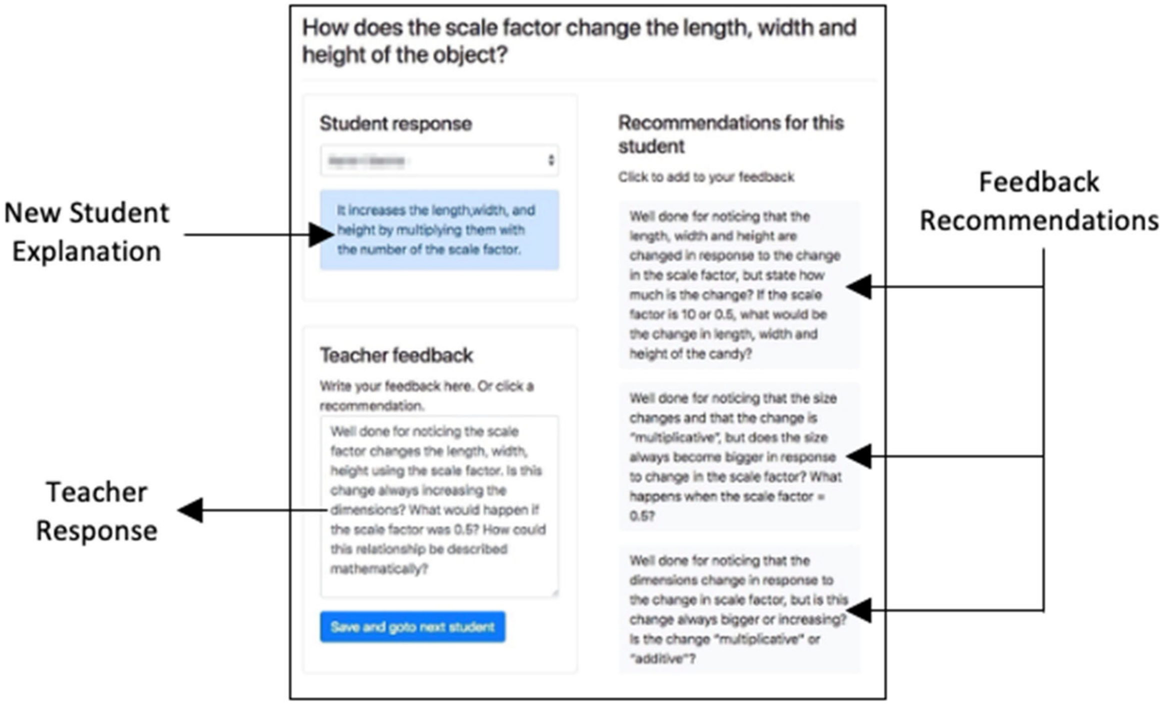
The model is built by creating vector representations of each preprocessed student explanation. Weights for each word are determined using tf-idf (term frequency-inverse document frequency) which assigns higher weights to words that occur less often in the training dataset (Zehner, Sälzer, & Goldhammer, 2016). The model is queried by finding how similar a new student explanation is when compared to each of the training dataset student explanations. Similarity is determined using cosine similarity, i.e. the cosine of the angle between vector representations of student explanations. The teacher feedback in the training dataset that corresponds to the student explanations that are most similar to the new student explanation are then examined and the top three unique teacher feedback responses are selected as the recommendations. The model building and querying was performed using *TfidfVectorizer* within the Python *sklearn* package.

User interface subsystem

The TRT user interface is an interactive webpage that presents teachers with the question prompt, the student explanation, and a teacher-response field (see Figure 3). The three TRT-recommended responses are shown in a column on the right side of the screen. When mouse-clicked, the text of the recommendation is copied to the teacher-response field, and any text in the teacher-response field can be edited. This allows teachers to use the recommendations without making changes, customize a recommendation, take parts of different recommendations, or ignore the recommendations and write their own response.

The layout of the user interface was designed to reduce the extraneous cognitive load that result from navigating the page, allowing the teachers to focus their working memory on considering of the recommendations. Recommendations were presented near to the student explanations, no scrolling was needed to navigate the page, clicking recommendations copy-and-pasted the text, and important text was subtly highlighted.

The decision to present three recommendations was based on a trade-off between providing teachers with enough recommendations to promote the thoughtful consideration of different perspectives, but not too many so as to make the cognitive load of the task high and overwhelming. For each recommendation, teachers were expected to read the text, consider it in relation to the student explanation, and compare it with other recommendations. From this perspective, and considering the length of the text of the recommendations, four recommendations were considered by the designer to be the upper limit, two a lower limit, so three were chosen.



How does the scale factor change the length, width and height of the object?

Student response

It increases the length,width, and height by multiplying them with the number of the scale factor.

Teacher feedback

Write your feedback here. Or click a recommendation.

Well done for noticing the scale factor changes the length, width, height using the scale factor. Is this change always increasing the dimensions? What would happen if the scale factor was 0.5? How could this relationship be described mathematically?

Recommendations for this student

Click to add to your feedback

Well done for noticing that the length, width and height are changed in response to the change in the scale factor, but state how much is the change? If the scale factor is 10 or 0.5, what would be the change in length, width and height of the candy?

Well done for noticing that the size changes and that the change is "multiplicative", but does the size always become bigger in response to change in the scale factor? What happens when the scale factor = 0.5?

Well done for noticing that the dimensions change in response to the change in scale factor, but is this change always bigger or increasing? Is the change "multiplicative" or "additive"?

Figure 3. The user interface subsystem. The teacher user interface displays the question prompt for which the training data was collected (top), the new student explanation (top left, in blue), the three recommendations from the NLP recommendations engine, and space for the teacher response (bottom left).

Method

Research Questions

To assess the effectiveness of the tool we asked:

1. How accurately does the TRT select recommendations?
2. How effectively do teachers interact with the TRT?

Data Collection

Context

The data for this study was collected in from high school geometry students and their teachers. The demographics of the school, located in a rural mid-Atlantic region of the United States, were 12% Black, 44% Hispanic, and 38% White students, with 68% of the students receiving free or reduced lunch and 39% of the students classified as having Limited English Proficiency. The students participated in a mathematical modeling project that focused on how scale factor impacts the dimensions, volume, and surface area of a rectangular prism. During the project, the students were asked to write explanations for three different question prompts. The student explanations and teacher responses from a prior implementation of the project were used to create the training datasets used to answer research question 1. The teacher interactions with the TRT during a subsequent implementation of the same project but with different students were used to answer research question 2.

Research question 1

In order to assess the effectiveness of the TRT in terms of how accurately it selects recommendations we created three training datasets as describe above, with a different dataset for each of three question prompts (see Table 2). High school geometry students

Table 2. *A summary of the three training datasets used to assess the accuracy of the TRT at selecting recommendations.*

Training dataset	Question prompt	Number of student explanation and teacher response pairs	Number of different teacher responses
1	How does the scale factor change the length, width and height of the object?	116	29
2	How does the scale factor change the volume of the object?	99	6
3	How does the scale factor change the surface area of the object?	85	6

For each training dataset we performed a leave-one-out cross-validation (Borra & Di Ciaccio, 2010). To do this for a given training dataset we first removed one of the student explanations and its corresponding teacher feedback. Second, we used the remaining training data to build a tf-idf model as described above. Third, we used the removed student explanation to query the model, and finally, recorded whether the recommendations obtained from the query included the teacher feedback corresponding to the removed student explanation. If so, we counted this as a success; if not, a failure. This process was repeated, leaving out a different student explanation from the training dataset each time, until all the student explanations in the dataset had had their turn to be left out. The proportion of successes for each training dataset was found by dividing the total number of successes by the size of the dataset, and to account for successes that are expected by random chance we calculated kappa for each dataset. A kappa of 0 indicates that all the success is due to randomness and a kappa of 1 indicates success every time (Cohen, 1968).

Research question 2

To assess the effectiveness of the TRT in terms of how teachers interact with the user interface, four classroom teachers were observed using the user interface while following a think-aloud protocol which encouraged the teachers to verbalized their thinking as they interacted with the tool. Video screen capture, audio, and researcher fieldnotes were recorded throughout. In addition, the four teachers also participated in individual interviews several days later with included questions about their use of the TRT. The transcripts of think-aloud and post-project interviews were analyzed to identify reoccurring themes and evidence that confirmed or diverged from the themes. The screen capture video was analyzed for the frequency that recommendations were selected or edited by the teachers and the time that they spent interacting with the TRT.

Results

System Effectiveness

Research question 1

After performing leave-one-out cross-validation for each of our training datasets we found kappa values of 0.51, 0.84 and 0.76 (see Table 3). These results are comparable to those found in other studies that used natural language processing technology with open-response items. Liu and colleagues (2014) report average kappa values for such studies to be between 0.62 and 0.81. However, the questions considered in these studies only distinguish between two and five categories of response, making them more comparable to the question 2 and question 3 dataset results rather than the result for the question 1 dataset which is lower, we expect, due to the high number of unique recommendations included in this dataset. Therefore, we conclude that the accuracy of the TRT recommendation selection is comparable to those used by other studies.

Table 3. Number of explanations, unique recommendations, proportion of successful recommendation selections, and kappa values by question

Training dataset	Number of student explanations	Number of unique recommendations	Proportion of successful selections	Kappa
Question 1	116	29	0.560	0.509
Question 2	99	6	0.919	0.838
Question 3	85	6	0.882	0.764

Research question 2

Teachers interacted intuitively with the TRT interface. None of the teachers were observed asking about how to use the interface or expressing frustrations with the interface while responding. Instead, teachers were positive about their interactions, for example, Henry commented that “it was a very clever interface” and that “it was nice to be able to see what [the students] did, try to give a tailored response to give them a hint towards where they supposed to be going, and it was also nice to be able to personalize it for them.”

Teachers interacted thoughtfully with the TRT interface. The teachers interacted with the recommendations provided by thoughtfully considering the merits of the different recommendations with respect to the student explanation. Mike described that he would “look at the recommendations and think ‘Well, that one clearly isn't what I see happening here. This one is the closest to [the student explanation], but I think I need to just qualify it a little bit, modify it to fit this situation.’” And Sam said that the recommendations were “something to start off of and decide if I agreed with what was there, or if I needed to make up my own.” Nina commented, it was beneficial to her that the recommendations were not “everything I want to say as verbatim exactly what I want … because if it was exactly like what I wanted to say, then I feel like [responding to students] would just be a little more mindless for me.”

Thoughtful teacher interactions were supported by the functionality of the TRT interface. The teachers thought that the TRT selected the recommendations well. For example, Henry commented that the TRT “generally, did a good job pulling recommendations that fit the situation. Many of them I was able to use.” However, because the recommendations were often not exactly how a teacher wished to respond to a student, the teachers made use of the user interface functionality for selecting and editing the recommendations. As Nina explained, “I could kind of pick apart different pieces. It was more of editing, manipulating, or rephrasing what was already given.” The results from the analysis of the user interaction data collected from the screen capture video confirm that teachers interacted with the recommendations often while responding to students. On average across all teachers, one fourth of the teacher responses were unedited recommendations and half of the responses were edited recommendations. At the same time, there was some divergence in how the teachers used the recommendations, with Sam mostly writing responses without using the recommendations, and Mike mostly using unedited recommendations (see Table 4).

Table 4. Average responding time (in seconds) and the use of recommendations in responding, by teacher.

		Number (percentage) of submitted teacher responses that used:		
Teacher	Average responding time (seconds)	*no* recommendation	an *edited* recommendation	an *unedited* recommendation
Sam	76.7	5 (56%)	3 (33%)	1 (11%)
Nina	69.6	4 (14%)	21 (75%)	3 (11%)
Mike	48.2	4 (14%)	7 (25%)	17 (61%)
Henry	84.7	5 (29%)	10 (59%)	2 (12%)
Total	66.2	18 (22%)	41 (50%)	23 (28%)

Discussion

The Teacher Responding Tool (TRT) was designed and developed to scaffold teachers' skills at giving high quality, student specific feedback. This required that the text of the recommendations aligned with research based-practices, that appropriate recommendations were selected for a given student explanation, and that teachers were able to interact thoughtfully with the selected recommendations via the user interface. The results from this study demonstrate that, for the context in which this pilot version of the TRT was tested, i.e. in the context of high school students writing explanations of their understanding during a mathematical modeling project, these requirements were satisfied. The TRT was able to select recommendations as accurately as other natural language processing tools and, importantly, the teacher users considered the recommendations selected to be appropriate. The user interface design supported thoughtful teacher interactions by providing three recommendations, functionality for selecting and editing recommendations, and a low extraneous cognitive load layout. A prior study (Bywater, Chiu, Hong, & Sankaranarayanan, 2019) demonstrated that thoughtful teacher interactions with the recommendations contributed to improved teacher responding practice. This study provides evidence that these interactions were facilitated by the TRT design.

Recommendations

Several recommendations for designing natural language tools for learning follow from the results of this study.

1) Natural language processing (NLP) tools that are designed for learning should understand their impact on teacher professional skills. In learning contexts, NLP tools are typically used to automatically respond to students so that teachers are able to focus their time on those students who are most in need (e.g. Gerard, Matuk, McElhaney, & Linn, 2015). In this study, the TRT design included teachers into the responding process so that teachers had opportunities to notice how their students were thinking and to develop their responding skills. We recommend that designers of NLP tools for learning consider how to they can support research-based teacher practices.
2) Training the system requires specific and purposeful data. The requirement of 'big data' is often associated with natural language applications and might be thought to limit the applicability of NLP techniques to specific learning contexts. This study suggests otherwise. Training a system with a smaller dataset that is specific to the applied context can be effective and can support the use in specialized, non-normative, or underrepresented learning contexts.
3) Teacher input into the training dataset content is critical to generate authentic, rich recommendations. The process of creating the training dataset might also be considered a novel professional development activity that builds upon established practices within the field for teachers collaboratively examining student work.
4) For tasks that are cognitively demanding, extraneous cognitive load can be reduced by using a minimalist user interface design that retains all necessary functionality. In this study, reducing extraneous cognitive load involved both how to best present information and how to reduce the load associated with interacting with the information. The TRT user interface combined the select-copy-select-paste steps that a user frequently repeats into a single click. This simplified the interaction steps for teachers when selecting and editing.
5) Connecting the TRT with different learning management systems requires permissions to share identifiable user data and technical expertise. These challenges continue to present adoption hurdles but are being addressed within the educational technology community (e.g. Learning Tools Interoperability, 2019) and we recommend common standards and protocols to mitigate these challenges.

References

Aleven, V. A., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science, 26*, 147–179.

Ball, D. L. (1993). With an eye on the mathematical horizon: Dilemmas of school mathematics. *The Elementary School Journal*, *93*(4), 373–397.

Borra, S., & Di Ciaccio, A. (2010). Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis, 54*, 2976-2989.

Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, *25*(1), 60-117.

Bywater, J. P., Chiu, J. L., Hong, J., & Sankaranarayanan, V. (2019). The Teacher Responding Tool: Scaffolding the teacher practice of responding to student ideas in mathematics classrooms. *Computers & Education.*

Chapin, S. H., O'Connor, C., & Anderson, N. C. (2009). *Classroom discussions: Using math talk to help students learn, Grades K-6.* Sausalito, CA: Math Solutions.

Clark, R. C., & Mayer, R. E. (2016). *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. San Francisco: John Wiley & Sons.

Coffey, J. E., Hammer, D., Levin, D. M., & Grant, T. (2011). The missing disciplinary substance of formative assessment. *Journal of Research in Science Teaching*, *48*(10), 1109–1136.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213.

Gerard, L., Matuk, C., McElhaney, K., & Linn, M. C. (2015). Automated, adaptive guidance for K-12 education. *Educational Research Review*, *15*, 41–58.

Grossman, P., Compton, C., Igra, D., Ronfeldt, M., Shahan, E., & Williamson, P. (2009). Teaching practice: A cross-professional perspective. *Teachers College Record*, *111*(9), 2055-2100.

Learning Tools Interoperability. (2019). Retrieved from http://www.imsglobal.org/activity/learning-tools-interoperability

Liu, O. L., Brew, C., Blackmore, J., Gerard, L., Madhok, J., & Linn, M. C. (2014). Automated scoring of constructed-response science items: Prospects and obstacles. *Educational Measurement: Issues and Practice*, *33*(2), 19–28.

Mayer, R. (2014). Cognitive Theory of Multimedia Learning. In R. Mayer (Ed.), *The Cambridge Handbook of Multimedia Learning* (Cambridge Handbooks in Psychology, pp. 43-71). Cambridge: Cambridge University Press. doi:10.1017/CBO9781139547369.005

McDonald, J., Bird, R. J., Zouaq, A., & Moskal, A. C. M. (2017). Short answers to deep questions: supporting teachers in large-class settings. *Journal of Computer Assisted Learning*, *33*(4), 306-319.

National Council of Teachers of Mathematics (2014). *Principles to actions: Ensuring mathematical success for all*. Reston, VA: Author.

Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, *78*(1), 153-189.

Zehner, F., Sälzer, C., & Goldhammer, F. (2016). Automatic coding of short text responses via clustering in educational assessment. *Educational and psychological measurement, 76*(2), 280-303.