121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014
Paper ID #9508

Using Natural Language Processing Tools to Classify Student Responses to Open-Ended Engineering Problems in Large Classes

Dr. Matthew A Verleger, Embry-Riddle Aeronautical Univ., Daytona Beach

Matthew Verleger is Assistant Professor in Freshman Engineering at Embry-Riddle Aeronautical University. He has a BS in Computer Engineering, an MS in Agricultural & Biological Engineering, and a PhD in Engineering Education, all from Purdue University. Prior to joining the Embry-Riddle faculty, he spent two years as an Assistant Professor of Engineering Education at Utah State University. His research interests include Model-Eliciting Activities, online learning, and the development of software tools to facilitate student learning.

Using Natural Language Processing Tools to Classify Student Responses to Open-Ended Engineering Problems in Large Classes

Peer review can be a beneficial pedagogical tool for providing students both feedback and varied perspectives. Despite being a valuable tool, the best mechanism for assigning reviewers to reviewees is still often blind random assignment. This research represents the first step in a larger effort to find an improved method for matching reviewers to reviewees. By automating the classification of student work, reviewer quality and reviewee need can potentially be assessed. With that assessment, the best reviewers can be assigned to the neediest teams, while the most self-sufficient teams can be assigned reviewers who may need to see higher quality work.

The purpose of this paper is to present the preliminary findings from an effort to classify student team performance on Model-Eliciting Activities (MEAs) using natural language processing tools. MEAs are realistic, open-ended, client-driven engineering problems where teams of students produce a written document describing the steps of how to solve the problem.

Archival data containing expert evaluations to MEAs were used to test different natural language processing tools in an attempt to identify which tools could most accurately assign scores similar to an expert. The research did not re-implement the selected algorithms, but rather used off-the-shelf libraries to explore the value of their application to this context.

Using a split-sample training-testing set, the "Bagged Decision Tree" and "Random Forest" algorithms were used to classify sample solutions against 11 MEA rubric dimensions. Performance on each rubric item averaged between 60% and 85% accurate, depending on the item. The implementation of these algorithms also revealed words and phrases commonly used in higher quality samples.

This paper will focus on how the data was obtained and prepared, how the different algorithms were utilized, how the algorithms performed in the classification tests, what the results indicate about our implementation of MEAs and how the results will be informing the next stages of the research project.

Introduction

Peer review is a cornerstone of the modern scientific process. It is meant to act as a gateway, allowing good research through, while filtering out junk science; to separate the wheat from the proverbial chaff. Yet many scientists, academics, and even the US Supreme Court agree that peer review, while essential to the scientific process, is far from a perfect system[1]. The problem with peer review is that it is a theoretically sound process that can easily fall apart on implementation. It is a methodology whose success is heavily dependent on having the most appropriate reviewer for the situation providing the right review.

When used in the classroom, peer review can be a useful tool for providing students with additional feedback and perspectives while not significantly increasing the workload of graders or course administrators. Much like it's research counterpart, classroom peer

review suffers from issues related to proper reviewer selection. Ballantyne, Hughes, and Mylonas[2] noted multiple studies describing how students do not necessarily believe that they or their peers are capable reviewers. In their study, 40% of the participants agreed that their peers could not fairly assess their work. Fundamentally, this is a case of "one bad apple spoils the bunch." When a student receives even a single poorly formed peer review, their attitude towards all of their received reviews can be spoiled. While this issue can be reduced through significant training and careful rubric design, the need for understanding effective reviewer matching is essential for improving the long-term effectiveness and implementation of peer review in the classroom.

Verleger, et al.[3] highlighted some of the complexities of viewing the reviewer-reviewee relationship as a variable that can be adjusted and explored to different effects. The key outcome of that research was an understanding that making peer review assignments in an algorithmic repeatable fashion requires careful monitoring of the driving assumptions in order to be more successful. Building on that concept, this paper represents the first step in a larger effort to develop a proof-of-concept peer-review matching algorithm and demonstrate if it is a valuable and viable methodology for conducting peer review? To that end, data used in Verleger, et al.[3] was reexamined for the purpose of developing a better approach to predicting reviewee need.

Literature Review

Peer Review

Editorial peer review has been a cornerstone component of scientific achievement since the mid-1950's[4]. Despite its tremendous post-war boom to become the de facto standard for scientific and technical publications and the largely similar goal of providing feedback to improve quality, peer review is still only moderately used as a pedagogical tool within the higher education classroom. The single greatest hindrance toward utilizing peer review in the classroom is getting students to accept that it is a viable source for feedback and assessment. Ballantyne, et al.[2] undertook a study of 1,654 first- and second-year students spanning three semesters studying four different courses. Despite continual efforts based on feedback from students and faculty to improve the process, some of the attitudes of the participants towards the process remained relatively consistent throughout the entire study. In a follow-up survey given to all 1,654 students, 734 gave a response to a question regarding the worst aspect of the peer review process. 31% of those 734 responders (14% of the total) mentioned concerns about the competency of either themselves or their peers.

Despite a lack of confidence in the quality of the review, the majority of students report liking peer review. Of the 30 undergraduate computer science students in their study, Moreira and Silva[5] found that 77% of the students indicated that they liked peer review, and another 13% were neutral towards peer review. Liu et al.[6] reported that 64% of participants viewed peer review as beneficial and effective for learning. Despite students' concerns about peer review, multiple studies indicate that it improves the quality of the products being submitted subsequent to the review. Sitthiworachart and Joy[7] indicated that 69% of first-year undergraduate students in computer science reported that they discovered mistakes in their own code while reviewing code written by their peers. Eighty

percent of the students felt that seeing other students' work was helpful for their learning. Ballantyne et al.[2] reported that the majority of the 939 respondents "agreed that peer assessment was an awareness-raising exercise because it made them consider their own work more closely, highlighted what they needed to know in the subject, helped them make a realistic assessment of their own abilities, and provided them with skills that would be valuable in the future."

In addition to the immediate skills provided by participating in peer review, many researchers recognize the long-term benefits provided to reviewers. Boud[8] posited that the focus of assessment as a whole must be rethought to promote lifelong learning skills. Learning to perform and to respond to formative feedback given by both peer- and self-review are essential skills for succeeding in a continuous working world that doesn't assign an end-of-project grade. Teaching students how to perform peer review and how to utilize constructive criticism for improvement is essential for their future. Yet despite the long-term benefits recognized by academia, students are largely unfamiliar with peer review. Sitthiworachart and Joy[9] reported that of their 215 first-year students taking a computer programming course, 89% of them had not ever experienced peer review prior to the start of the course. Guilford[10] found that only 39% of undergraduate engineering students understood peer review as it related to scientific publishing. Ballantyne et al.[2] indicated that only 10% of all the students studied recognized the value of peer review towards their future employment, though 35% of the education students in their study recognized the long-term value.

Numerous software tools now exist to handle the collection, review, and redistribution process of electronic documents[5,6,11–16]. The most significant advantage to systems such as these is that they streamline the overall process, reducing the time-intensive overhead associated with using peer review in a classroom. Despite the numerous software tools available, one element which all are lacking is an informed mapping system for assigning reviewers to reviewees, most likely because informed mapping systems require data about the reviewer to effectively perform the mapping. Most systems rely on some form of random or instructor-based assignment. While these methodologies may work for small classes, their effectiveness quickly breaks down in larger classes.

Because random assignment requires no prior knowledge, assignments can be made on demand with no regard for the participants' skills and abilities. Prior to Verleger et al.[3], Crespo, Pardo, and Kloos[17] proposed the most ambitious attempt at producing higher quality reviewer-reviewee mappings. The authors developed an adaptive model that assigns students a "proficiency score". They then mapped reviewers to reviewees in such a way as to produce complementary proficiencies, i.e. high proficiency reviewers mapped to low proficiency reviewees and vice-versa. This strategy produced "promising experimental results", however no discussion of the educational impact has ever been published. One of the reasons their model is flawed is that it reduces a participant down to a single numerical value. It assumes that as a reviewer, an individual is equally capable at all aspects of reviewing as well as being an equally capable reviewee; that reviewer skill and reviewee need are the same for any individual. Likewise, while the Calibrated Peer Review[11] system

does assign a reviewer competency score, it only uses the score to weight final grades, not to make reviewer assignments.

In Verleger et al.[3], individuals were given a reviewer quality score based on their performance on a calibration exercise. Teams were given a reviewee need score based on a TA's evaluation of an earlier draft of their work. Different assignment methodologies were then employed to match reviewers to reviewees. Verleger then reevaluated the student work of 147 teams to measure the change seen across many aspects of a team's solution. The primary finding from this research was a better understanding about how sensitive the algorithmic assignment was to the driving assumptions. A description of those failed assumptions and how they are being removed or mitigated is presented as part of the project methodology for how the algorithms are being developed and validated.

MEAs

This research will initially be explored in the context of Model-Eliciting Activities. Model-Eliciting Activities (MEA) are realistic, client-driven, open-ended problems that are designed to be both model-eliciting and thought-revealing[18]. They require students to mathematize (e.g., quantify, organize, dimensionalize) information in context. An engineering-based MEA requires that students be provided with a realistic problem that a client needs solved. The solution of an MEA requires the development of one or more mathematical, scientific, or engineering concepts that are unspecified by the problem – students must grapple with their existing knowledge to develop a generalizable mathematical model to solve the problem. The point is for students to be involved in the creation of the initial ideas underlying the concept or system, thus establishing the need and motivation to go through cycles of expressing their initial ideas, testing, and refining them. An MEA creates an environment where skills such as communication, verbalization, and an ability to work cooperatively and collaboratively are valued. Carefully constructed MEAs can begin to prepare students to communicate and work effectively in teams; to create, adopt and adapt conceptual tools; to construct, describe, and explain complex systems; and cope with complex systems. The attributes of MEAs support the development of the abilities and skills required of graduates of accredited engineering programs as stated in ABET Criterion 3 a to k[19].

Instructors' MEA Assessment/Evaluation Packages (I-MAPs) are currently used to provide formative and summative evaluation of student work across four dimensions (mathematical model, re-usability, modifiability, and share-ability) that align to the MEA design principles[20]. These four dimensions were designed to specifically assess issues which practicing engineers valued[21]. Each I-MAP includes a quantitative rubric, the use of which is to be supplemented with qualitative feedback to the students. Quantitative rubric items are used to broadly assess the quality of the response as well as to establish metrics used in assigning a grade. The qualitative feedback is aimed at helping student teams improve the quality of their solutions.

The mathematical model dimension encompasses the assessment of (1) the quality of the solution in terms of how well it addresses the complexity of the problem and accounts for all data provided, and (2) the use of rationales to support the solution method. The root of

this dimension is assessing how good the procedure is at providing a solution to the specific problem being given. Does the procedure do what it is explicitly required to do?

The re-usability dimension focuses on the quality of the solution in terms of how easily it can be used by the client in new but similar situations. A re-usable procedure (1) identifies who the direct user is and what the direct user needs in terms of the product, criteria for success, and constraints, (2) provides an overarching description of the procedure, and (3) clarifies assumptions and limitations concerning the use of the procedure. The underlying idea is that engineers rarely develop a procedure specifically to solve a single problem, but often design solutions around a class of problems. Part of that development involves explicitly defining that class in such a way as to make it clear what problems can and can not be solved with the given procedure.

The modifiability dimension assesses how well the procedure can be modified by the direct user for use in different situations. A modifiable procedure (1) contains acceptable rationales for critical steps in the procedure and (2) clearly states assumptions associated with individual procedural steps. Unlike the re-usability dimension, which defines the larger context in which a procedure can and cannot be used, the modifiability dimension is concerned with how difficult it is to modify each step in order to adapt the procedure while maintaining the team's intentions. For example, if a specific value is selected as a threshold value (e.g., "remove the top 10% of the data"), modifiability seeks to measure if the reasoning behind "10%" is made clear.

The share-ability dimension is used to evaluate the quality of the solution in terms of (1) how well the client can understand the procedure, and (2) how accurately the client can replicate results given in the procedure for the provided data set. A portion of this includes responding to all of the client's requests for results. An underlying component of this dimension is not only clarity, but also brevity and avoiding extraneous and unnecessary information.

Methods

Project Goal

The goal of this phase of the research is to find an accurate, automated method for predicting how an expert would score the procedures produced by a team of students. The primary vehicle for making these predictions will be to leverage off-the-shelf methods from the field of Natural Language Processing.

Archive Data

As part of a prior study[3], the author evaluated the solutions to 147 team's responses to three drafts of the Paper Airplane MEA, resulting in 441 total evaluations. Specifics about the MEA can be found in [22], however the overarching focus of the MEA is on developing a procedure to rank competitors in a paper airplane competition for four (4) awards based on data from multiple throws and three (3) different measurements per throw. Based on the author's extensive history developing, evaluating, and researching MEAs, as well as

test-retest evaluations (documented in [3]), the author's scores are considered to be expert evaluations with regards to MEAs.

For the predictions made in [3], teaching assistant (TA) evaluations of first draft scores were used. TAs went through an extensive training sequence, but ultimately proved inconsistent and largely inaccurate. While their evaluations are not the focus of this research, they do represent a baseline metric upon which to improve.

MEA Rubrics

For that evaluation, each MEA was evaluated using a rubric that consisted of 8 numerical items which were then translated into 3 dimensional scores and an overall score. One of the eight items ("No progress has been made toward a model.") was evaluated, but was removed from analysis due to a lack of variability, with only 4 of the 441 solutions being rated as having no progress. A full discussion on the development, reliability, and validity of the MEA Rubric can be found in Diefes-Dux, Zawojewski et al.[21].

The rubric was divided along three dimensions; Mathematical Model, Re-Usability & Modifiability, and Audience (Share-ability). Each dimension contained numeric and free response feedback items, though only the numeric items were used for the expert evaluations. Each numeric MEA Rubric item was assigned point values corresponding to levels of achievement. Items are divided into two categories; true/false items and mutually exclusive items. True/False items are assigned one of two possible point values depending on the item. Mutually exclusive items are items where multiple statements are presented, each with its own associated point value, and only a single statement may be selected. All of the items are presented in Table 1. The score for each dimension is calculated as the minimum of the items in that dimension; the overall score is calculated as the minimum score of the three dimension scores. Minimums are taken for the dimensional and overall scores to encourage continuous broad-spectrum improvement. This is also a philosophical stance by the instructors – the student work is only as good as the weakest dimension.

As an example, assume an evaluator selects Level 2 for the Mathematical Model Complexity, "False" for Data Usage, and "True" for Rationales. As the Mathematical Model dimensional score is calculated as a minimum of the dimension's items, the Mathematical Model dimension score is a minimum of 2 (Mathematical Model Complexity), 3 (Data Usage), and 4 (Rationales), resulting in a score of 2 for the Mathematical Model Dimension. The overall score is then calculated as a minimum of the three dimensional scores, meaning that regardless of how this theoretical team performs on the remainder of the items, the best score this team could receive is an overall score of 2 because of the 2 given for the Mathematical Model Dimension.

Dim.	Item Label	Full Item Wording	Points	
Mathematical Model	Mathematical Model Complexity	The procedure fully addresses the complexity of the problem.	4	
		A procedure moderately addresses the complexity of the problem or contains embedded errors.	3	
		A procedure somewhat addresses the complexity of the problem or contains embedded errors.	2	
		Does not achieve the above level.	1	
	Data Usage	The procedure takes into account all types of data provided to generate results OR justifies not using some of the data types provided.	True	4
			False	3
	Rationales	The procedure is supported with rationales for critical steps in the procedure.	True	4
			False	3
Re-Usability/Modifiability	Re-Usability/ Modifiability	The procedure not only works for the data provided but is clearly re-usable and modifiable. Re-usability and modifiability are made clear by well articulated steps and clearly discussed assumptions about the situation and the types of data to which the procedure can be applied.	4	
		The procedure works for the data provided and might be re-usable and modifiable, but it is unclear whether the procedure is re-usable and modifiable because assumptions about the situation and/or the types of data that the procedure can be applied to are not clear or not provided.	3	
		Does not achieve the above level.	2	
Audience (Share-ability)	Results	Results from applying the procedure to the data provided are presented in the form requested.	True	4
			False	1
	Audience Readability	The procedure is easy for the client to understand and replicate. All steps in the procedure are clearly and completely articulated.	4	
		The procedure is relatively easy for the client to understand and replicate. One or more of the following are needed to improve the procedure: (1) two or more steps must be written more clearly and/or (2) additional description, example calculations using the data provided, or intermediate results from the data provided are needed to clarify the steps.	3	
		Does not achieve the above level.	2	
	Extraneous Information	There is no extraneous information in the response.	True	4
			False	3

Table 1. MEA Rubric – Numerical Items

Data Preparation & Cleaning
Using the archival data, each team's procedure was downloaded into individual text files with unique file names to delineate both the source team and the draft number. Numeric scores from the expert evaluation were downloaded as an Excel spreadsheet, mapping the team/draft numbers to the expert scores. The algorithms being tested are considered "Bag-of-Words" based algorithms, meaning that word ordering, placement, and meaning are not considered, only the frequency with which the words are used. Based on that, a MATLAB script was developed to process each procedure and count the frequency of each word within each procedure. To further expand the power of the algorithms, the occurrence of multi-word sequences were also determined, allowing for sequences of up to 10 words. Finally, to reduce the data pool to a manageable size, words that occurred in fewer than 5% of the procedures or in more than 95% of the procedures were removed from consideration. This reduced the dataset from 1,977,409 word or word sequences to just 4506 word or word sequences, with each word occurring a median of 54 times across all of the procedures.

Algorithm Implementation
Two algorithms were examined; Random Forest and Bagged Decision Tree with Forward Feature Selection (BDTFFS). The BDTFFS is included in MATLAB's statistics toolbox, while the Random Forest algorithm is an external package found at [23] based on the description found in [24].

At a high level, both algorithms function by building a decision tree based on word appearances. The idea being that, procedures that contain for example, the phrase "to the target", are placed in one half of the tree, while those that don't are segmented into a different branch. Each branch is further subdivided until eventually, the resulting tree has grouped similarly rated procedures together. Figure 1 shows a final tree determined by the BDTFFS algorithm for the Re-Usability/Modifiability Scale. The values on each line represent the number of times the parent word or phrase appears in the completed procedure. The colored bubbles correspond to the final score that is predicted by the algorithm. For example, a procedure that contains the word "judges" at least once, does not contain the phrase "and best overall" and contains one or more occurrences of "in case of a tie" will typically be rated as a level 3 procedure with regards to its Re-Usability/Modifiability scale.

For the Random Forest algorithm, the data is split into training and testing groups. For this analysis, the test group consisted of a randomly selected group of 20% of the samples. The remaining 80% training group is used to construct a classifier. The algorithm works by generating 500 (or some other user selected number) decision trees, each one based on a randomly selected subset of the training data. Each of the 500 trees are then "voted" on by the entire training data set and the best tree is the resulting classifier. This tree is then applied to the 20% test group to determine an actual error rate. Each item took between 60 and 90 minutes to identify the final tree and the corresponding error rate.

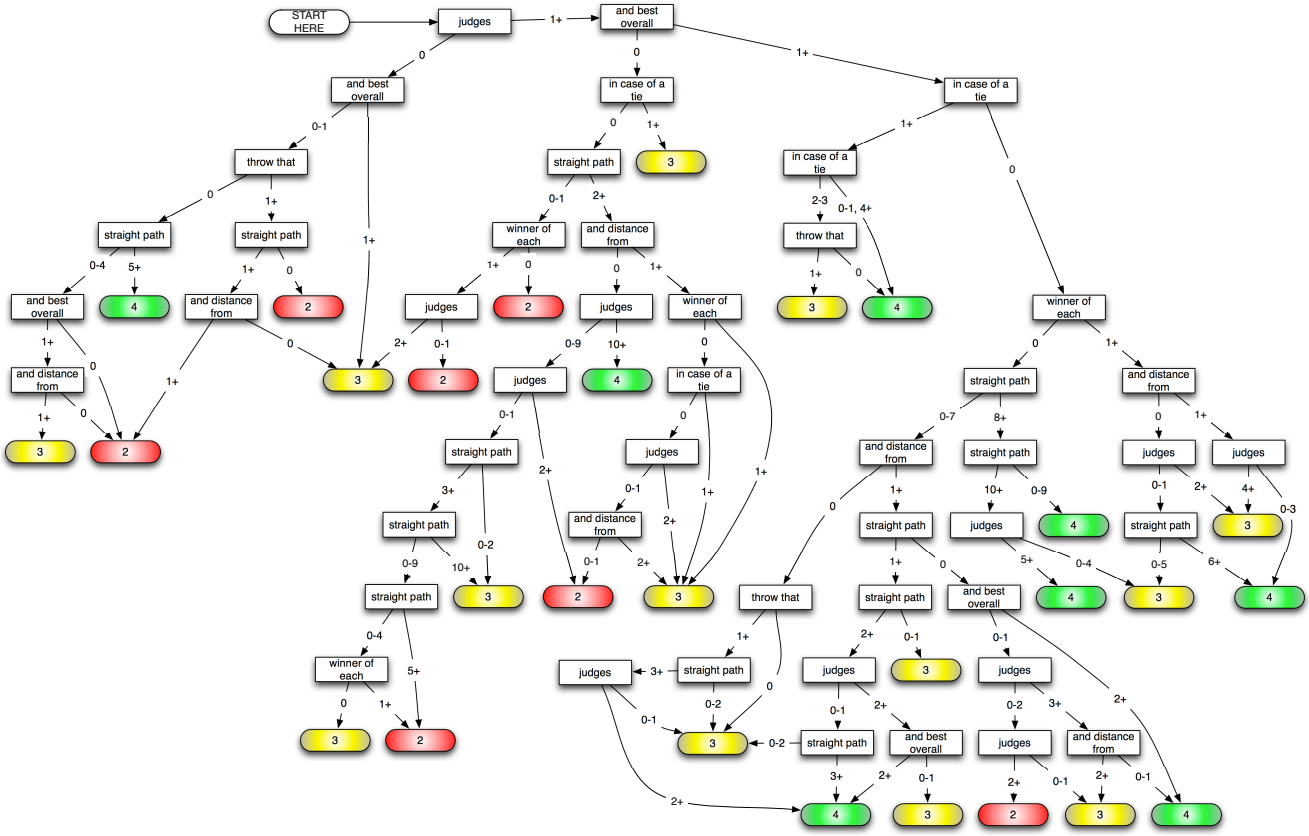

Figure 1. Re-Usability/Modifiability Scale Decision Tree

The BDTFFS algorithm is more methodical, but is also much slower. Similar to the Random Forest algorithm, but instead of using the entire tree returned by the training set, it only selects each branching point. The algorithm then makes multiple iterations, selecting the best word(s) to branch on at each point to reduce the overall error. The result tends to be a more accurate tree (as each branching word is explicitly chosen to reduce the classification error), but for a non-trivial increase in the amount of time needed to identify the appropriate words. Each item took between 8 and 10 hours for this algorithm to identify the final tree and the corresponding error rate.

Findings

Classification Results – Random Forest

Because the algorithm heavily relies on randomization, there is variability between consecutive runs. To gauge the overall accuracy of the algorithm, 10 runs were conducted to predict each of the 11 MEA Rubric items. The error rates for each of the 11 tests are presented in Table 2. For 7 of the 11 items, the worst Random Forest measurement was still better than the corresponding error in the teaching assistant (TA) evaluations. The TA evaluation was better than the best Random Forest in only 2 of the items (Data Usage and Results). For the other 2 items (Rationales and the Audience Scale), the TAs evaluations were not meaningfully different than the Random Forest evaluation was able to predict.

Item/Scale/Score	Min	Max	Mean	St. Dev.	TA Error
Mathematical Model Complexity	0.318	0.409	0.348	0.025	0.667
Data Usage	0.352	0.477	0.409	0.041	0.291
Rationales	0.239	0.409	0.332	0.047	0.310
Re-Usability/ Modifiability	0.170	0.307	0.236	0.044	0.549
Results	0.295	0.420	0.366	0.039	0.255
Audience Readability	0.307	0.455	0.366	0.045	0.569
Extraneous Information	0.227	0.409	0.312	0.051	0.425
Mathematical Model Scale	0.330	0.432	0.384	0.035	0.660
Re-Usability/ Modifiability Scale	0.170	0.250	0.208	0.032	0.549
Audience Scale	0.443	0.580	0.511	0.047	0.467
Final Score	0.352	0.455	0.399	0.036	0.520

Table 2. Random Forest Error Rates (n = 10 forests)

Classification Results – Bagged Decision Tree with Forward Feature Selection

For the BDTFFS, the extensive running time precluded multiple tests, however a single run was made for each of the items and is shown in Table 3. 10 of the 11 items had less error than the TA evaluations and only the Audience Scale item had an error greater than 0.30.

Item/Scale/Score	Error Rate	# of Selected Word(s)	TA Error
Mathematical Model Complexity	0.285	9	0.667
Data Usage	0.224	9	0.291
Rationales	0.211	8	0.310
Re-Usability/ Modifiability	0.138	13	0.549

Results	0.256	6	0.255
Audience Readability	0.247	7	0.569
Extraneous Information	0.188	6	0.425
Mathematical Model Scale	0.297	9	0.660
Re-Usability/ Modifiability Scale	0.150	7	0.549
Audience Scale	0.361	10	0.467
Final Score	0.290	6	0.520

Table 3. Bagged Decision Tree with Forward Feature Selection Error Rate

Conclusions

MEA-centric Implications

This approach represents a first step in a larger attempt to predict student performance on MEAs. For this specific MEA, the results are favorable. While the Random Forest algorithm provides a quick analysis, the increased accuracy of the BDTFFS algorithm is worth the increased time investment, particularly given that a single run may generate trees that can be used for this MEA for an extended period of time. One potential application of these results may be in guiding teaching assistants towards more accurate evaluations and feedback. Pre-identifying for TAs that a particular piece of work has a particular percent chance of rating at certain level may help them re-examine if the work truly fits the level they choose. This is a double-edged sword that would need to be carefully monitored to make sure that TAs do not automatically select the predicted level without that re-evaluation.

One MEA-specific next step will be to compare the predicted values with TA evaluations to identify rubric items or TAs that need specific improvement. If a particular TA is consistently rating off of the predicted value, this may highlight a TA that needs additional intervention. Likewise, if a particular rubric item is not being consistently evaluated by many TAs, training for that that item may have been inadequate.

Finally, examining the makeup of each tree and the implications of the selected words may provide insight into the attributes that make a particular procedure better. Usage of the phrases 'to the target' and 'around the chair' (both phrases near the top of the BDTFFS Mathematical Model Complexity tree) highlight the importance of including directional specificity in the procedure's mathematical description. Providing students with explicit training that encourages this action may have a net positive effect on solution quality.

Broader Implications

For a broader audience, this work hopefully highlights one way that Natural Language Processing tools such as these may be useful for exploring student work in a new way, but that they are also not yet accurate enough to be used for assigning student grades. One potential value of these techniques would be to help instructors identify their own potentially mis-graded work. Similar to the TA recommendations described above, instructors may find that these techniques are helpful at checking that their own evaluations are internally consistent.

Next Steps

These results generate a number of next steps that must now be explored. First will be to identify procedures that are routinely being mis-classified to verify that the expert evaluations are accurate. One potential cause for the misclassification may be that the scores provided by the expert are incorrect. Correcting these mis-rated items would only improve the prediction accuracy. Likewise, additional methods must be examined to identify if a combination of these and other algorithms can even more accurately predict student performance.

For the multi-level items (Mathematical Model Complexity, Re-Usability/Modifiability, and Audience Readability), an investigation into how extreme the mis-predictions are may improve their accuracy measurement. For example, if a Mathematical Model Complexity true score is a 4, a predicted value of 3 is more accurate than a predicted value of 1, but the above analysis only considers exact matches.

Finally, these algorithms have been applied to a very specific MEA and a very specific archival data set. To broaden their value, we must examine how well the predictions holds up to a more recent (and presumably better trained because of other improvements in the MEA process) group of students, as well as how well they work for other MEAs. Are there keywords that are universal across all MEAs or are there items that are highly predictable regardless of the problem context? This work represents a first step in a larger effort to automatically predict student performance on MEAs and building an understanding of how these prediction methods work across multiple MEAs and from a variety of contexts and universities is an essential component of that effort.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. XXXXXX. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Blackmun, H. Blackmun, H., Daubert v. Merrell Dow Pharmaceuticals. *United States Reports* **509 US,** 579 (1993).

2. Ballantyne, R., Hughes, K. & Mylonas, A. Developing Procedures for Implementing Peer Assessment in Large Classes Using an Action Research Process. *Assess. Eval. High. Educ.* **27,** 427–441 (2002).

3. Verleger, M., Diefes-Dux, H., Ohland, M., Besterfield-Sacre, M. & Brophy, S. Challenges to Informed Peer Review Matching Algorithms. *J. Eng. Educ.* **99,** 397–408 (2010).

4. Burnham, J. C. The Evolution of Editorial Peer Review. *JAMA J. Am. Med. Assoc.* **263,** 1323–1329 (1990).

5. Moreira, D. A. & Silva, E. Q. A Method to Increase Student Interaction Using Student Groups and Peer Review over the Internet. *Educ. Inf. Technol.* **8,** 47–54 (2003).

6. Liu, E. Z., Lin, S. S. J., Yuan, S. & Chiu, C. Web-Based Peer Review: The Learner as Both Adapter and Reviewer. *IEEE Trans. Educ.* **44,** 246–251 (2001).

7. Sitthiworachart, J. & Joy, M. Deepening Computer Programming Skills by Using Web-based Peer Assessment. in *4th Annu. Conf. LTSN Cent. Inf. Comput. Sci.* (2003).

8. Boud, D., Cohen, R. & Sampson, J. Peer Learning in Higher Education - Learning From & With Each Other. (2001).

9. Sitthiworachart, J. & Joy, M. Effective Peer Assessment for Learning Computer Programming. *9th Annu. SIGCSE Conf. Innov. Technol. Comput. Sci. Educ.* (2004).

10. Guilford, W. H. Teaching Peer Review and the Process of Scientific Writing. *Adv. Physiol. Educ.* **25,** 167–175 (2001).

11. Chapman, O. L. Calibrated Peer Review - The White Paper: A Description of CPR. **2007,** (2003).

12. Gehringer, E. F. Electronic Peer Review and Peer Grading in Computer-Science Courses. *Thirty-second SIGCSE Tech. Symp. Comput. Sci. Educ.* (2001).

13. Ngu, A. H. H., Shepherd, J. & Magin, D. Engineering the "Peers" System: The Development of a Computer-Assisted Approach to Peer Assessment. in *Res. Dev. High. Educ.* **18,** 582–587 (1995).

14. Sitthiworachart, J. & Joy, M. Web-based Peer Assessment in Learning Computer Programming. in *4th Annu. Conf. LTSN Cent. Inf. Comput. Sci.* (2003).

15. Trahasch, S. Towards a Flexible Peer Assessment System. *Fifth Int. Conf. Inf. Technol. Based High. Educ. Train. 2004* (2004).

16. Tsai, C., Liu, E. Z., Lin, S. S. J. & Yuan, S. A Networked Peer Assessment System Based on a Vee Heuristic. *Innov. Educ. Teach. Int.* **38,** 220–230 (2001).

17. Crespo, R. M., Pardo, A. & Kloos, C. D. An Adaptive Strategy for Peer Review. *34th ASEE/IEEE Front. Educ. Conf.* (2004).

18. Lesh, R. A., Hoover, M., Hole, B., Kelly, A. & Post, T. in *Handb. Res. Des. Math. Sci. Educ.* (Kelly, A. & Lesh, R. A.) 591–645 (Lawrence Erlbaum, 2000).

19. Accreditation Board for Engineering and Technology. *Criteria for Accrediting Programs in Engineering*. (ABET, Inc., 2013).

20. Diefes-Dux, H., Hjalmarson, M., Miller, T. K. & Lesh, R. A. in *Model. Model. Eng. Educ. Des. Exp. All Students* (Zawojewski, J. S., Diefes-Dux, H. & Bowman, K.) 17–36 (Sense Publishers, 2008).

21. Diefes-Dux, H., Zawojewski, J. S. & Hjalmarson, M. Using Educational Research in the Design of Evaluation Tools for Open-Ended Problems. *Int. J. Eng. Educ.* **26,** 807–819 (2010).

22. Wood, T., Hjalmarson, M. & Williams, G. in *Model. Model. Eng. Educ. Des. Exp. All Students* (Zawojewski, J. S., Diefes-Dux, H. & Bowman, K.) 187–212 (Sense Publishers, 2008).

23. Jaiantilal, A. Random Forest (Regression, Classification and Clustering) implementation for MATLAB (and Standalone). at <https://code.google.com/p/randomforest-matlab/>

24. Breiman, L. Random Forests. *Mach. Learn.* **45,** 5–32 (2001).