

Prediction of Grades for Reviewing with Automated Peer-review and Reputation Metrics

Da Young Lee, Ferry Pramudianto, Edward F. Gehringer
North Carolina State University
Raleigh, NC 27695
[dlee10, fferry, efg]@ncsu.edu

ABSTRACT

Peer review is an effective and useful method for improving students' learning through review by student peers. Peer review has been used in classes for several decades. To ensure the success of peer review, research challenges such as the quality of peer review must be addressed. It is challenging to identify how good the reviewer is. We develop a prediction model to assess students' reviewing capability. We investigate several important factors that influence students' reviewing capability, which corresponds to instructor-assigned grades for reviewing. We use machine learning techniques algorithms to build models for grade prediction for reviewing. Our models are based on the several metrics such as the reviewer assigning different scores to different rubric items and automated metrics to assess the textual feedback given by the reviewers. To improve the models, we also use reputation score of students' as reviewers. We present results of experiments to show the effectiveness of the models.

Keywords

Peer reviews, rubrics, prediction model

1. INTRODUCTION

Peer review [1, 4] is an effective and useful method for improving students' learning by reviewing peer students' work. Peer review has been used in classes for several decades. In recent years, peer review has been used not only for traditional classes but also for online courses such as Massive Open Online Courses (MOOC) [4]. For example, in Coursera [2], several online courses are offered, in which thousands of students from around the world are enrolled. In such cases, instructors are not able to give feedback to such a large number of students in a timely manner. Therefore, development of peer review methods based on observing peer behaviors is important, and the technology should be improved to be more reliable and useful to users.

The classroom peer review process is as follows. Students submit their assignments. Reviewers (peer students) provide reviews of the assignments. The students have a chance to improve their submitted work by incorporating scores and comments in the reviews. Because reviewers in educations are peer students, they may lack sufficient peer reviewing experience. Therefore, they need to be guided through the peer review process to ensure the provision of high-quality reviews.

The assessment of reviews is a challenging problem in education. Meta-reviewing is a manual process [5] where instructor might assign grades and provide feedback as a measure of the students' reviewing capability. The problem is that the manual process of meta-reviewing is tedious and time-consuming.

To address this issue, this study aims at investigating methods to help identify good reviewers who write high-quality reviews. To attain this goal, we examine factors that may influence review scores and propose a model to predict how good the reviewers are based on the reviews written by them using machine learning algorithms.

We investigate several important factors that influence instructor-assigned grades, especially reviewers assigning scores behaviors for instructor-assigned grades. In this paper, we refer instructor-assigned grades (i.e., grades) as the students' reviewing capability score assigned by the instructor. Another factor is automated peer-review metrics, which are text metrics [5, 6] such as tone for assessing the textual feedback given by the reviewers. The other factor is a reputation metric [11] to determine who is good reviewer based on history review scores across artifacts. This reputation metric is calculated based on the measure of the reviewer's leniency ("bias").

In this paper, we first investigate strong/weak correlation between reviewers with high reviewing capability and spread between scores. Note that the spread between scores corresponds to deviation for reviewer assigning different scores described in Section 3.3. We then investigate whether development of a model based on reviewer assigning different scores would be effective for predicting how good the reviewer is. For this task, we apply machine learning techniques such as a decision tree [12] and k-Nearest Neighbors [13] algorithms to build a model for prediction. We then investigate whether this model incorporating textual feedback shows positive results for predicting how good the reviewer is. Lastly, we investigate whether our model combined with text metrics and reputation scores shows positive results for predicting how good the reviewer is. For these tasks, we investigate following research questions:

- RQ1: Is there correlation between reviewer assigning different scores (i.e., "spread" between scores) to different rubric items, and instructor-assigned grades?
- RQ2: How well does our model, based on reviewer assigning different scores, predicts instructor-assigned grades?
- RQ3: How well does our model combined with text metrics of reviews predict instructor-assigned grades?
- RQ4: How well does our model combined with text metrics and reputation scores of reviewers predict instructor-assigned grades?

The rest of the paper is organized as follows. In Section 2, we briefly introduce peer review process and peer review system, called Expertiza [3]. In Section 3, we describe our methodology for

the study. In Section 4, we present our experimental results. Finally, we give concluding remarks in Section 5.

2. BACKGROUND

This section discusses background for this study.

2.1 Peer Review System: Expertiza

There are many tools to help peer-review process [3, 7, 8]. Expertiza is a web-based education system where a feature for enabling peer reviews is integrated. This feature is a part of an active learning process from peer students.

Using Expertiza, in classes, students are able to select tasks from assignment list. After students complete their tasks, they submit their outputs to receive reviews from peers in the peer-review system. The submissions will be reviewed by anonymous peers who can provide helpful comments and give scores based on rubrics. Researchers have worked on peer review systems for decades. Researchers improved Expertiza for effective learning management systems and peer-review systems.

Students expect to receive author feedback. Typically, a double-blinded review process makes difficult for students to explain what they have done, especially when reviewers may misunderstand the contents of the submissions and give low grades. In Expertiza, peer review may have multiple rounds where the reviewers give feedback for improvements and check if the suggestions have been implemented in next round. Each round have its several deadlines which are useful for organizing reviewing and resubmission.

In Expertiza, the functionality for supporting wikis is integrated for collaboration among students. Also, for submissions, students may use a wiki, which is very helpful in supporting collaborative work in writing assignments. These wikis provide several features for easy editing and keeping track of the past edition.

2.2 Peer Review

Each student can select more than one submission to review within one assignment period. Each review consists of a review rubric to guide students in the completion of the review. Each rubric may include multiple questions, called criteria. Appendix A. is an example of rubric, which consists of 12 rubric criteria. For example, each question may ask for assessments of the organization, originality, grammar issues or clarity of a writing submission under review. The rubric also asks whether the submission contains the acceptable quality of the definitions, examples, and links found in the submission.

In the peer review process, reviewers often provide two kinds of feedback: quantitative (scores) and qualitative feedback. Reviewers measure numeric scores for certain rubric criteria. In other words, after the reviewers read the rubric, they submitted their textual feedback and numeric scale scores for each criterion.

For example, rubric criteria can be, “on a scale of 1 (worst) to 5 (best), how easy is it to understand the code?” Moreover, reviewers are often required to provide formative textual feedback where their comments incorporate issues identified, suggestions, and comments. As numeric scores may be helpful, but textual feedback also gives more concrete ideas on the submissions.

3. METHODOLOGY

This section discusses methodology for this study.

3.1 Data

3.1.1 Data Collection

We assemble peer-review data from Expertiza [3]. This tool is a web-based educational learning application that helps students review peers’ work. We analyze 703 records submitted by students where the students are assigned to grade assignments of peers.

The data set is collected from two graduate-level courses: CSC 517 (Object-Oriented Design and Development) and CSC 506 (Architecture of Parallel Computers). Both are offered at North Carolina State University. For example, in CSC 517, programming assignments and writing assignments are used for peer reviews. These assignments are team-based assignments where more than two students collaborate together. We use six review assignments where four of six are related to writing and results and two out of six are related to programming assignments.

In this study, instructors manually assess submitted reviews and assign scores within one review period where each student may review multiple submissions. A final grade is given based on the students’ submission and the quality of their reviews when assessing their peers’ submissions.

3.1.2 Data Preparation

Data cleaning process is required before we process data analytics, which includes combining multiple Database and Excel tables based on the user’s id using SAS. During this process, we remove entries where numeric scores are 0 or NULL, which indicate empty. Invalid numeric scores can be assigned when students dropped their courses and did not assign scores on submissions of peer students. In addition, a rubric may require only textual feedback, which is not included in this study.

3.2 Research Questions

We investigate several important factors that influence instructor-assigned grades, especially reviewers assigning scores behaviors for instructor-assigned grades. As we explained in Section 1, we refer instructor-assigned grades (i.e., grades) as the students’ reviewing capability score assigned by the instructor.

To study the usefulness of review quality assessment, we investigate the following research questions:

- RQ1: Is there correlation between reviewer assigning different scores (i.e., “spread” between scores) to different rubric items, and instructor-assigned grades?
- RQ2: How well does our model, based on reviewer assigning different scores, predicts instructor-assigned grades?
- RQ3: How well does our model combined with text metrics of reviews predict instructor-assigned grades?
- RQ4: How well does our model combined with text metrics and reputation scores of reviewers predict instructor-assigned grades?

We describe more details about research questions. For RQ1, reviewers may assign grades for multiple submissions within the same review. This research question investigates strong/weak correlation between reviewers with high reviewing capability and spread between scores. Note that the spread between scores is measured by weighted standard deviation described in Section 3.3.

RQ2 investigates whether development of a model based on reviewer assigning different scores would be effective for predicting how good the reviewer is. RQ3 investigates whether this

model incorporating textual feedback shows positive results for predicting how good the reviewer is. RQ4 investigates whether our model combined with text metrics and reputation scores shows positive results for predicting how good the reviewer is. Note that we use a text analysis tool to automatically extract text metrics [5, 6]. We measure text metrics for given textual feedback such as content type, tone, and volume.

3.3 Metrics

We utilize the following metrics to address research questions.

- **Pearson Correlation Coefficients:** Pearson Correlation Coefficients measures simple linear correlation between sets of data. This shows a degree of how well they are related. The correlation is measured as follows:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

We measure the correlation between the reviewer assigning different scores to different rubric items, and that reviewer being given a high grade by the instructor. The correlation coefficient ranges between -1 to 1 where 1 implies perfect linear relation between X and Y, and -1 implies that, when X values increases, Y values decreases linearly. 0 implies no linear relation.

- **Weighted Standard Deviation:** This weighted standard deviation metric is measured as follows.

$\hat{w} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - M)^2}$ where standard deviation is a degree to measure the spread of observed numbers (x_1, x_2, \dots, x_n) in a data set with the mean value M of the observation numbers and weight \hat{w} . We measure this value to the degree of spread of scores given by reviewers. \hat{w} is the number of reviews assigned to each reviewer within one assignment.

- **Average Number of Words (Avg. # Words):** Given more than one review comment, this metric is the average number of words.

We measure weighted standard deviation and average number of words, which are used as inputs for machine learning algorithms for predicting instructor-assigned grades.

- **Root Mean Square Error (RMSE):** The RMSE between predicted values and actual values is computed as square root of the mean of the squares of the deviations.
- **Score Difference (Score Diff):** This metric is the gap between predicted values and actual values.

RMSE and Score Diff are used to measure the effectiveness of models. Especially, if RMSE and Score Diff are larger, a prediction model is less effective. If RMSE and Score Diff are smaller, a prediction model is more effective for prediction.

For determining the quality of the textual feedback, we use various text metrics, which can be collected automatically via a text analysis tool [5, 6].

- **Content Type:** Identification of content types in reviews: reviews may include different types of contents. This metric classifies contents into one of three categories: summative, problem-detection, or advisory content.

- **Summative content:** This content type is positive feedback or a summary of the submission. For example, "The page is organized logically" is classified into summative content.

- **Problem-detection content:** This content type Identifies problems in the submission. For example, "The page lacks a qualitative approach and an overview" is classified into problem-detection content.

- **Advisory content:** this content type provides suggestions to the students for improving their work. For example, "The page could contain more ethics related links" is classified into advisory content.

- **Tone:** reviews may include different tones, which refer to the semantic orientation of a text given words and presentation written by reviewers. This metric classify contents into one of three tones: positive, negative or neutral. This metric is
 - **Positive:** A review is classified as having a positive tone when it contains positive feedback overall. For example, positive words or phrase such as "well-organized paper" and "complete" indicate positive semantic orientation.
 - **Negative:** A review is classified as having a negative tone when it contains negative feedback overall. For example, negative words or phrase such as "copied", "poor", and "not complete" indicate negative semantic orientation.
 - **Neutral:** A review is classified into a neutral tone when it is contains neutral feedback and a mix of positive and negative feedback. For example, a mix of positive and negative words or phrase such as "The organization looks good overall; however, we did not understand the terms." indicate neutral semantic orientation: "looks good" can be positive and "did not understand" can be negative semantic orientation.
- **Volume:** reviews may include different words. This metric refers to the quantity of unique tokens in the review excluding stop words such as pronouns.

We use Lauw's Reputation Score identified by Song et al. [11]. Lauw-peer algorithm is based on the measure of the reviewer's leniency ("bias"), which can be either positive or negative.

- **Lauw's Reputation Score:** this metric is the measure who is good reviewer based on history review scores across artifacts. The reputation range calculated by the Lauw algorithm is [0,1]. A reputation score close to 1 means the reviewer is credible.

We measure text metrics, which are used as additional inputs for machine learning algorithms for predicting instructor-assigned grades.

3.4 Approach

Machine learning approaches [12, 13] such as K-nearest neighbor, decision tree, and neural network are useful for prediction. For our experiments, we use K-nearest neighbor and decision tree, which are based on supervised learning [13]. We first choose K-nearest neighbor classifiers because these ones are based on learning by analogy of input values. With this model, we can observe the closeness patterns based on Euclidean distance between training and validation data sets. As this model is rooted on analogy, we use another approach, which is known high performance on classification for categories of values. Decision tree model is a good fit for our input data set with different values. In addition, this approach is useful for comprehensibility to show a tree structure. In this paper, we did not consider other machine

Table 1. Examples of writing assignment reviews for a submission where numeric score and textual feedback given by reviewers on assignments. Detailed questions related to rubric criteria is found in Appendix A.

Rubric Criteria	Score	Textual Feedback
A1. Organization	5	The organization is good and nicely gives intro, features and then examples of the framework
A2. Clarity	4	They are clear and the language is easily understood
A3. Revision	2	No, I don't see any changes from previous version

learning algorithms SVM and Naïve Bayes because these require complicated calculation of formula and it is not easy to track how values are classified.

We first propose a simple baseline model, which predicting instructor-assigned grades by the average grades for reviewing for a training data set. This baseline model is used to justify whether machine learning algorithms would be useful for grade prediction. We develop models based on machine learning algorithms, which can show better prediction than the simple baseline model.

As Expertiza has been used for several years, we have sufficient data concerning students' reviews and instructor grade assignment. We divide our data into a training set and a validation set. Our goal for using the machine learning approach is to predict instructor grade assignments. This problem is related to the reputation of peers. The peers who carefully review other students' submissions are likely to receive higher scores. We first use a training data set for training our decision tree model and then apply the model on a validation set for score prediction.

In this paper, we use a decision tree [12] to explore and model our data. Decision trees are typically used in operations research, especially for decision analysis. As one would like to predict specific decisions related to scores, decision models are applied. We used an SAS tool, called JMP [12], in which machine learning approaches are integrated. In this tool, the partition function recursively partitions data according to the relationship between data sets. Given the relationship between X and Y values, a tree is created determining how to generate a tree of partitions. In this process, the tool automatically searches groups and continues splitting off separate groups. This process is conducted recursively until the tool reaches a specific desired fit. The tool stops when the prediction result is no longer improved.

We also consider the k-Nearest Neighbors algorithm (or k-NN for short). The k-NN algorithm is a widely used algorithm among all machine learning algorithms. k-NN is a non-parametric method used for classification and is supported by SAS/JMP tool. The input consists of the k closest training set, which contains similar features. k-NN is a type of learning to find approximated locally similar classification.

We describe steps of our work using the example reviews in Table 1. We compare automated metrics and reputation score assessed by reputation algorithm, called Lauw's model for predicting the instructor-assigned grades for reviewing. We also

Table 2. Input data sets used for our decision tree and k-Nearest Neighbors algorithm for predicting instructor-assigned grades for reviewing.

Name	Input Data Set
Base	<ul style="list-style-type: none"> Weighted Standard Deviation Average Number of Words
Text	<ul style="list-style-type: none"> Content Type Tone Volume
Rep	<ul style="list-style-type: none"> Lauw's Reputation scores

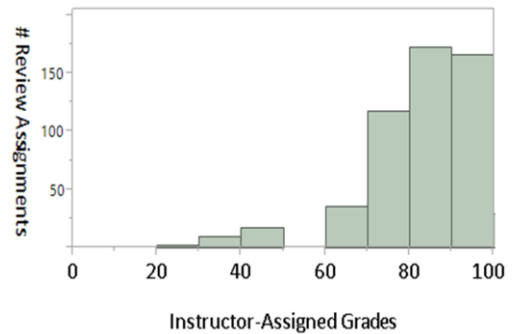


Figure 1. Instructor-assigned grade distribution where X-axis represents scores and Y-axis represents the number of review assignments.

combine two approaches whether the combined models show better prediction.

Consider that Alice gives scores and comments to Bob. Note that rubric criteria A1, A2, and A3 are found in the Appendix A.

Step 1. Collect a list of scores that students gave to assignments. Consider that Alice gave scores and comments to Bob's assignment. As shown in Table 1, Alice rated 5, 4, and 2 for the assignment of Bob. This score is used for calculating weighted standard deviation. Weighted standard deviation is 4.5 considering that its standard deviation is 1.5 and its weight is 3.

Step 2. Collect list of comments given to a reviewer. The number of word per each textual feedback is 15, 9, and 9. The total count of words is 33. The average number of words is 11.

Step 3. Consider that Alice's reviewing score is 95 assigned by the instructor. Using weighted standard deviation and the average number of words, we use a decision tree to predict this instructor-assigned grade. From the collected data set, we use 2/3 of data as a training set and 1/3 of data as a validation data set. This decision tree is trained to predict the instructor-assigned grade.

Step 4. Using weighted standard deviation and the average number of words, the k-NN algorithm is used to predict this instructor-assigned grade. We set k with 10.

Step 5. Calculate reputation of a reviewer using Lauw's Reputation Model. Then, this reputation score is converted to a predicted instructor-assigned grades.

Step 6. Compare performance of prediction. RMSE and Score Diff are used to measure the effectiveness of approaches.

Then, we also extend these models using different sets of metrics for decision tree and k-NN algorithm, especially, described in Table 2.

4. RESULTS

In this Section, we investigate factors to influence instructor-assigned grades. Figure 1 shows instructor-assigned grade distribution. As shown in Figure 1, the grades are well distributed above 75. There are some low scores.

4.1 Effect of Reviewer Assigning Different Scores

We describe *hypothesis 1* to answer RQ1.

Hypothesis 1: There is a strong correlation between a reviewer assigning different scores to different rubric items, and instructor-assigned grades. That is, a reviewer who is careful to consider what score a student should receive for each rubric item (and therefore gives different scores for different rubric items) is likely to be assigned a higher grade by the instructor, than is a student who tends to assign the same score (e.g., 4 out of 5) to all or almost all rubric items.

The purpose of this study is to investigate the effect of reviewer assigning different scores for grade prediction. This research question investigates whether reviewers with high quality reviews may show some correlation between the scores assigned to different rubric items and grades.

The first step is to find and collect review scores within the same assignments per student. The second step is to calculate weighted standard deviation from the list of scores. The third step is to calculate the relationship between this deviation and instructor-assigned grades. Assumption is that student who give scores differently would be more a careful reviewer. This student may receive higher grades for their review assignments.

For the Pearson correlation, good fit is useful to predict an anticipated future rate. We assess the statistical significance using statistical testing methods. In this context, we measure p-value with regard to those correlation models. The p-value represents the probability of satisfying a model. The p-value is considered to be an estimate of the 'goodness of fit' of the model. Typically, the test of satisfying the model is statistically significant if the p-value < 0.05 . We used SAS software for conducting this analysis.

A Pearson product-moment correlation coefficient was computed to assess the relationship the deviation metric and scores. There was a positive, but weak linear correlation between the two variables, $r = 0.1$, $p = 0.03$. Note that r is correlation coefficient and p is p-value. As r is small, we observe that there is a positive, but weak correlation between a reviewer assigning different scores to different rubric items, and instructor-assigned grades. Note that this shows only linear correlation.

In addition, as shown in Section 4.2, we use the metric, a reviewer assigning different scores to different rubric items, for building models because decision tree models with this metric are effective in terms of grade prediction.

With regards to Pearson product-moment correlation, we conclude that data does not support *hypothesis 1*.

4.2 Prediction of Instructor-Assigned Grades for Reviewing

We describe *hypothesis 2* to answer RQ2.

Hypothesis 2: Our decision tree and K-nearest neighbor models based on reviewer assigning different scores are effective for predicting instructor-assigned grades. That is, decision tree and k-nearest neighbor models have smaller RMSEs than that of Lauw's Reputation Model.

The purpose of this study is to investigate whether development of models based on a reviewer assigning different scores would be effective for predicting instructor-assigned grades.

The first step is to apply the decision tree model to partition data for the best performance. We divide the data into training set and validation set. We assign 2/3 of a set as a training data set for modelling. The remaining 1/3 of a data set is used as a validation data set for comparison. The second step is to calculate the average difference and RMSE between actual grades and instructor-assigned grades based on the decision tree model. The third step is to compare results with results gained from K-nearest neighbor, baseline and prediction model based on reputation system, called Lauw's algorithm [11].

When we use the decision tree and k-nearest neighbor models, we employ two inputs: weighted standard deviations, and the average number of words for reviews given by students within one assignment. The output is the predicted grades. To compare performance, we measure the average of the absolute value of score difference between an actual grade and corresponding predicted grade. RMSE is also used to measure how close the predicted values are to actual values. Note that, as grades vary from low to high, accurate grade prediction cannot be achieved with high accuracy. Instead, we measure the average difference between actual grades and instructor-assigned grades.

All available valid peer-review records are used in this experiment. We measure that score difference range, average absolute bias and root mean square error (RMSE) in Tables 3 and 4. Tables 3 and 4 present the results from decision tree (DT) model, k-nearest neighbor (k-NN) model, baseline and Lauw's Rep Model for different data sets inputs. For example, Base+Text means that base and text input data sets in Table 2 are used.

We observe a case of base data set input only since this research question is related to only base metric. For base data set inputs, we observe that the decision tree and k-nearest neighbor models have smaller RMSEs than that of baseline and Lauw's Reputation models for writing and programming assignments. Therefore, *the decision tree and K-nearest neighbor models are more effective for prediction in this case*. DT model and k-NN model are data-driven models, which assess input data and find the best fit to correlate these inputs with an output. We observe that Lauw's Rep Model is dependent on data sets. For example, the range of $[0,1]$ is useful for reputation score. However, if one receives 0 as a reputation score, then, she/he may be expected to receive the lowest grade (e.g., 0), but this cannot happen because instructor consider many aspects other than reputation.

We conclude that data supports *hypothesis 2*.

4.3 Prediction of Instructor-Assigned Grades for Reviewing using Text Metrics

We describe *hypothesis 3* to answer RQ3.

Hypothesis 3: Our decision tree and k-nearest neighbor models based on additional text metrics are more effective for predicting instructor-assigned grades than the preceding models (based on reviewer assigning different scores). That is, decision tree and k-

Table 3. Experimental results for writing assignments based on our decision tree (DT) model, k-nearest neighbor (k-NN) model, baseline and Lauw’s Rep Model. The decision tree has a lower RMSE than the baseline and the Lauw’s reputation model, and the RMSE decreases each time the decision tree is refined. The k-nearest neighbor has a lower RMSE than the Lauw’s reputation model and the RMSE is similar each time the k-nearest neighbor is refined.

	Decision Tree				K-Nearest Neighbors				Baseline	Lauw’s Rep Model
	Base	Base+ Text	Base+ Rep	Base+ Text+Rep	Base	Base+ Text	Base+ Rep	Base+ Text+Rep		
Avg. Abs. Score Diff	9.4	8.8	8.7	8.0	8.8	9.1	8.6	8.9	9.9	16.2
RMSE	13.0	12.6	11.4	10.1	12.6	13.6	12.2	12.9	13.2	20.8
Avg. RMSE	11.7				14.4				13.2	20.8

Table 4. Experimental results for programming assignments based on decision tree (DT) model, k-nearest neighbor (k-NN) model, baseline and Lauw’s Rep Model. The decision tree has a lower RMSE than the baseline and the Lauw’s reputation model, and the RMSE is similar each time the decision tree is refined except one based Base+Text+Rep. The k-nearest neighbor has a lower RMSE than the Lauw’s reputation model and the RMSE tends to be increased each time the k-nearest neighbor is refined.

	Decision Tree				k-Nearest Neighbors				Baseline	Lauw’s Rep Model
	Base	Base+ Text	Base+ Rep	Base+ Text+Rep	Base	Base+ Text	Base+ Rep	Base+ Text+Rep		
Avg. Abs. Score Diff	8.7	8.4	8.6	9.9	8.4	8.5	8.0	9.0	8.9	16.2
RMSE	11.9	11.0	11.7	13.0	9.0	10.8	12.3	11.2	13.1	20.8
Avg. RMSE	11.9				10.8				13.1	20.8

nearest neighbor models based on additional text metrics have smaller RMSEs.

We investigate whether our models with additional text metrics derived from textual feedback show more effective results for predicting instructor-assigned grades than the preceding models. In this study, we measure text metrics from textual feedback: content, tone, and volume.

The purpose of this study is to investigate whether additional text metrics can be useful as predictive metric for improving decision tree prediction results with regard to instructor-assigned grades. The first step is to measure text metrics from reviews. We create our models to partition data for best performance. For this model, we use metrics such as weighted standard deviations, the average number of words, content type, tone, and volume. We divide the data into training and validation sets. The second step is to calculate the average difference between actual grades and instructor-assigned grades for our models. The third step is that we compare results with the one resulting from our models in the Section 4.2.

Tables 3 and 4 show the results our models with text metrics for grade prediction. All available valid peer-review records are used in this experiment. We observe the average score difference and root mean square error (RMSE) in Tables 3 and 4. From this result, when we compare RMSE results between base and base+text cases, we see that for the decision tree model, *additional text metrics help improve the prediction power for grades*. We see that for k-nearest neighbor model, *additional text metrics do help improve the prediction power for grades*.

K-nearest neighbor model results are based on analogy, which is not be effective for prediction in this case. Volume is already accounted for in the number of words. Therefore, volume may not

show substantial improvement. Review contents and tone generated by our meta-review service are not highly analogous for the similar grades. We observe that some students may have higher review grades with negative tones and summary content. But other students might have higher review grades with positive tones and problem-detection content. However, K-nearest neighbor models cannot distinguish these cases. Additionally, K-nearest neighbor models incorporated with more, yet unrelated variables may be less effective than those limited to selected and related variables.

Our results are dependent on which models would be used. *We conclude that data analyzed by decision tree models supports hypothesis 3. We conclude that data analyzed by the k-nearest neighbor models assignments does not support hypothesis 3.*

4.4 Prediction of Instructor-Assigned Grades for Reviewing using Text Metrics and Reputation Models

We describe *hypothesis 4* to answer RQ4.

Hypothesis 4: Our decision tree and k-nearest neighbor models based on additional reputation scores improve prediction of instructor-assigned grades. That is, decision tree and k-nearest neighbor models based on additional reputation scores have smaller RMSEs.

We investigate whether our models with additional text metrics and reputation model scores shows positive results for predicting instructor-assigned grades. In this study, we measure text metrics from textual feedback: content, tone, and volume.

The purpose of this study is to investigate whether additional reputation scores can be useful as predictive variables for improving decision tree prediction results with regard to instructor-assigned grades. The first step is to calculate reputation scores [11] from reviews. We create our models to partition data for best

performance. The second step is to calculate the average difference between actual grades and instructor-assigned grades for our models. The third step is that we compare results with the one resulting from the model in the preceding Section.

Tables 3 and 4 shows the results of grade prediction. All available valid peer-review records are used in this experiment. We observe the average absolute score difference and root mean square error (RMSE) in Tables 3 and 4. From this results, *for writing assignments, the decision tree model with base+text+rep data inputs is the most effective in terms of RMSE.* We infer that reputation score helps improve the performance of grades prediction in this case. However, *for programming assignments, the decision tree model with base+text+rep data inputs is not the most effective in terms of RMSE.* The reasons would be for programming assignments, the focus of reviewing is to check the correctness of program behaviors and requirements with shorter textual feedback compared with ones from writing assignments.

Our results are dependent on which assignments would be used. *We conclude that our data does partially support hypothesis 4: the decision tree models with writing assignments supports hypothesis 4, but the decision tree models with programming assignments does not support hypothesis 4.*

5. CONCLUSIONS AND FUTURE WORK

Peer review is an effective and useful method for improving students' learning by reviewing peer students' work. The quality of peer reviews is important when guiding students. To improve the quality of peer reviews, instructors grade their reviews based on students' scores and feedback. However, this process is manual, and automated decisions would be helpful. Prediction of the instructor-assigned grades is a complex and challenging problem in peer review systems. We used machine learning techniques algorithms to build models for grade prediction for reviewing. Experimental results showed that the decision tree model and K-nearest neighbor (k-NN) model are more effective than Lauw's Reputation Model in terms of RMSE. We also compared the average RMSE values for the decision tree and k-NN models. Experimental results showed that the decision tree models (avg. RMSE: 11.7) are more effective than k-NN models (avg. RMSE: 14.4) for writing assignments in terms of the average value of RMSE. Experimental results showed that the k-NN models (avg. RMSE: 10.8) are slightly more effective than decision tree models (avg. RMSE: 11.9) for programming assignments in terms of the average of RMSE. Text metrics may be useful for classifying contents, but showed less effect on grade prediction. Future work includes the followings. First, we improve the prediction capabilities of the model. We investigate any other metric to capture a certain feature of data, which can improve the performance. Second, we explore semantics of text, which also help guide modelling with higher performance.

6. ACKNOWLEDGMENTS

This study is partially funded by the PeerLogic project under the National Science Foundation grants 1432347, 1431856, 1432580, 1432690, and 1431975.

7. REFERENCES

- [1] Topping, K.. "Peer assessment between students in colleges and universities." *Review of educational Research* 68.3 (1998): 249-276.
- [2] Cloudera: <http://www.cloudera.com/>, 2016

- [3] Gehringer, E., "Expertiza: information management for collaborative learning." *Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-Learning Support*, pp 143-159, 2009.
- [4] Kulkarni, Chinmay, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R. Klemmer. "Peer and self assessment in massive online classes." In *Design Thinking Research*, pp. 131-168. Springer International Publishing, 2015.
- [5] Ramachandran, L. and Gehringer, E., "Automated assessment of review quality using latent semantic analysis," 11th IEEE International Conference on Advanced Learning Technologies, 2011.
- [6] Ramachandran, L. and Gehringer, E., "An automated approach to assessing the quality of code reviews," American Society for Engineering Education, San Antonio, TX, 2012.
- [7] Margerum, L., Gulsrud, M., Manlapez, R., "Application of calibrated peer review (CPR) writing assignments to enhance experiments with an environmental chemistry focus." *J. Chemical Education* 84, no. 2 (2007): 292.
- [8] Luca de Alfaro and Michael Shavlovsky. *CrowdGrader: a tool for crowdsourcing the evaluation of homework assignments.* Proc. 45th ACM technical symposium on Computer science education (SIGCSE '14). ACM, pp 415-420, 2014.
- [9] Jonsson, A. and Svingby, G., *The Use of Scoring Rubrics: Reliability, Validity and Educational Consequences* Educational Research Review, v2 n2, pp130-144, 2007
- [10] Song, Y., Hu, Z. and Gehringer, E.F., *Closing the Circle: Use of Students' Responses for Peer-Assessment Rubric Improvement.* Proc. *Advances in Web-Based Learning--ICWL 2015*, pp 27-36, 2015
- [11] Song, Y., Hu, Z. and Gehringer, E.F., *Pluggable reputation systems for peer review: A web-service approach.* *FIE* pp 1-5, 2015
- [12] JMP Decision Tree Model https://www.jmp.com/support/downloads/pdf/jmp11/Specialized_Models.pdf
- [13] Phyu, Thair Nu. "Survey of classification techniques in data mining." In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, pp. 18-20. 2009.

8. APPENDIX

Appendix A. Examples of Rubric Criteria of Writing Assignments in CSC 517

No	Question	Score Range
A1	Organization: how logical and clear is the organization?	(Terrible organization) 0 to 5 (Very logical and clear)
A2	Clarity: Are the sentences clear, and non-duplicative? Does the language used in this artifact simple and basic to be understood?	(Terrible English usage) 0 to 5

		(Good English usage)
A3	Did the authors revise their work in accordance with your suggestions?	(Not agree) 0 to 5 (Strong agree)
A4	Originality: If you found any plagiarism in round 1, has it been removed? Then, randomly pick some sentences or paragraphs and search for them with a search engine. Describe any text that may infringe copyrights.	(Several places of plagiarism spotted) 0 to 5 (No plagiarism spotted)
A5	Coverage: does the artifact cover all the important aspects that readers need to know about this topic? Are all the aspects discussed at about the same level of detail?	(Not agree) 0 to 5 (Strong agree)
A6	Definitions: are the definitions of unfamiliar terms clear and concise? Are the definitions adequately supported by explanations or examples?	(Several definitions are missing or incomplete) 0 to 5 (Strong agree)
A7	References: do the major concepts have citations to more detailed treatments? Are there any unavailable links?	(Many more references should be added) 0 to 5 (Strong agree)
A8	List the unfamiliar terms used in this wiki. Are those unfamiliar terms well defined or linked to proper references?	(neither defined nor linked) 1 to 5 (well defined or links are added)
A9	Rate the overall readability of the article. Explain why you give this score.	(not readable and confusing) 1 to 5 (readable and not confusing)
A10	Rate the English usage. Give a list of spelling, grammar, punctuation mistakes or language usage mistakes you can find in this wiki (e.g. ruby on rails -> Ruby on Rails).	(terrible English usage) 1 to 5 (good English usage)
A11	List any related terms or concepts for which the writer failed to give adequate citations and links. Rate the helpfulness of the citations.	(more citations are needed) 1 to

		5 (adequate citations)
A12	Rate how logical and clear the organization is. Point out any places where you think that the organization of this article needs to be improved.	(terrible organization) 1 to 5 (very logical and clear)

Appendix B. Snapshot of Decision Tree Model for Writing Assignments for Base+Text+Rep Metrics

