

Automated Metareviewing: A Classifier Approach to Assess the Quality of Reviews

Ravi K. Yadav
Computer Science Department
North Carolina State University
Raleigh, NC, USA
rkyadav@ncsu.edu

Edward F. Gehringer
Computer Science Department
North Carolina State University
Raleigh, NC, USA
efg@ncsu.edu

ABSTRACT

A review's quality can be evaluated through metric-based automated metareview. But not all the metrics should be weighted the same when it comes to evaluating the overall quality of reviews. For instance, if a review identifies specific problems about the reviewed artifact, then even with a low score for other metrics it should be evaluated as a helpful review. To evaluate the usefulness of a review, we propose a use of decision-tree based classifier models computed from the raw score of metareview metrics, instead of using all the metrics, we can use a subset of them.

Keywords

Automated metareview, Decision tree classifier Peer-review systems; Education artifacts.

1. Introduction

MOOC-based education platforms as well as face-to-face classrooms are increasingly adopting peer assessment. Peer reviewing increases students' participation and fosters collaborative learning. Students are encouraged to review their peers' work and provide formative feedback. High-quality feedback can help the reviewee improvise his/her work. Reviewing (or evaluating) a review is known as metareviewing. For best results, a review should be metareviewed before being presented to the reviewee. Usually this is a manual task [1, 2] for the teaching staff, which becomes more demanding when the metareview is needed quickly. Automated metareviewing [3] is a technique of using a smart tool to evaluate the quality of a review using certain textual properties of the submitted feedback. These properties include tone, volume, content type, relevance, coverage, and plagiarism. Content type is further divided into problem identification, advisory, or summative evaluation of reviewed work. These properties are the metrics used by automated metareviewer to evaluate the usefulness of a review. Though a good review may contain all of these properties, we found that a good review need not contain all.

2. Metrics to assess a review

As mentioned above, a metareview evaluates a review based on certain textual properties, otherwise known as metrics. Below are the metrics used by our metareview evaluator.

Review relevance: A relevant review should discuss the work reviewed and try to identify problems/issues in author's work.

Review content: This metric is further divided into three metrics, such as: Summative, Problem Detection and Advisory.

Summative: A summative review provides either positive feedback or a summary of the author's work.

Problem detection: A review can detect one or more specific problems in the reviewed artifact.

Advisory: A reviewer can provide specific advice to the author, which can be used by the author to improve the artifact.

Coverage: Coverage is a measure of review's ability to cover the main points of the artifact.

Tone: Tone refers to the semantic orientation of a text. Tone is divided into three categories: positive, negative and neutral. A single review can contain various measures of positive, negative and neutral tone.

Volume: Volume measures the quantity of textual feedback provided by the reviewer.

Plagiarism: This metric is based on the originality of a review. If a review is copied, then it is marked as plagiarized. A review is compared against artifact, rubrics used and the internet search results based on the review text.

3. Experiments

Our automated metareview system is a Ruby on Rails-based web service [10]. All the statistical calculations are performed using packages available in R. The metareview web service generates quantitative scores, but to determine the overall quality of a review based on this score, we need a statistical model. This model, once trained, can be used to classify a review as a good or a bad one. To train this model, we performed an experiment in the form of a survey. We selected a collection of student artifacts from Expertiza [4]. We used the reviews they received from the other students in the class. These reviews were rated manually by survey participants, explained in next section. The questionnaire used to evaluate the reviews by survey participants was based on metareview metrics. Table 1 lists all the questions used in the questionnaire. Survey participants were asked to answer the questions by selecting a response on the scale of 1–5, where 1 is the lowest score and 5 as the highest. In this experiment, we ignored the Plagiarism metric, hence no question was asked related to this metric. The question on "Overall quality" was used to generate the class identifier for each review.

Experiment participants

Participants were former and current TAs from different departments of Engineering, Science and Business. We trained them by explaining the essence of each metareview metric used in automated metareviewing. Multiple participants were asked to rate the same reviews to generate a holistic model. We created

an anonymous system to prevent the reviewers from knowing the identity of the authors of the artifact and the reviews.

The artifacts selected for this experiment were taken from the articles created by Spring 2016 students in CSC 517 course at NC State University. As a part of this course, students wrote Wikipedia articles which were then given to other students in class for reviewing. Each student was required to review two articles. They were given an option to review two more articles to receive extra points.

Table 1: Questionnaire for the survey

Question text	Associated metareview metric (scale of 1–5)
How well does the review adequately reflect (summarize) the artifact?	Summative
How well is the problem identified by the reviewer about the artifact?	Problem detection
How specific is the advice provided by the reviewer to the author to improve the artifact?	Advisory
How relevant is the review to the artifact?	Relevance
Does the review cover all the parts of the artifact?	Coverage
What do you think about the tone used by reviewer? (1: strongly negative, 2: negative, 3: neutral, 4: positive, 5: strongly positive)	Tone
How satisfied are you with the quantity of comments provided by reviewer?	Volume
How would you rate the overall quality of the review?	Overall quality

4. Data model & Results

Preprocessing data

A total of 119 reviews were surveyed in this experiment. Since more than one survey participant reviewed the same artifact, each review was assigned the average of the scores it received from all the participants.

All the questions were answered on a scale of 1–5, with 5 being the “best” score. For the tone metric, we found that only two surveys assigned a score of 1 (highly negative) to a review, whereas about 60% reviews received a score of 4 (positive). About 10% received a score of 5 (highly positive). We normalized the survey score for tone and grouped them into three categories. A score less than 3 (<3) was translated to -1 (Negative), whereas 3 was translated to 0 (Neutral) and a score greater than 3 (>3) was converted to 1 (Positive). The survey question associated with the overall quality of the review was normalized as well. A score higher than 3 was translated to good review (1), otherwise it was marked as bad review (0). We used this metric as class identifier for our data modeling. This was done to create a holistic model.

Figure 1 shows the distribution of surveys scores for each metric individually. We can see from this figure that not all the metrics are dispersed equally, which correlates with the idea that each

metric is not equally important for evaluating the overall quality of the review.

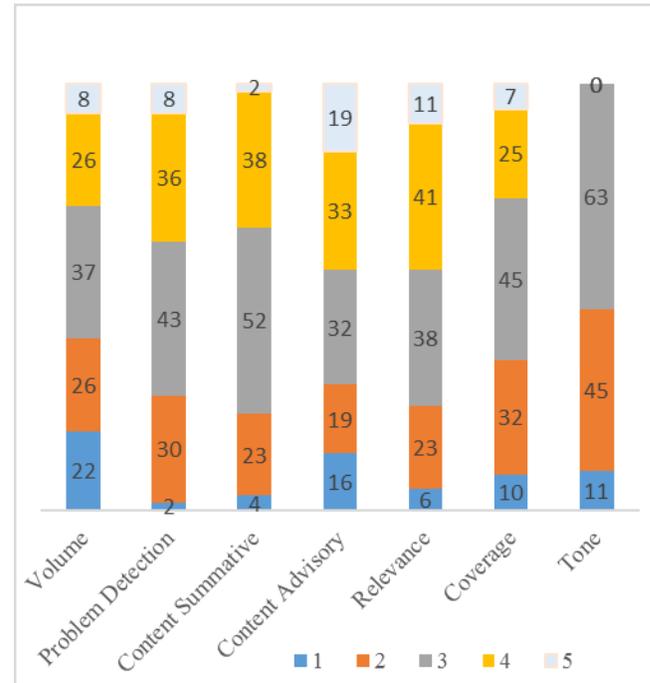


Figure 1: Distribution of count of expert surveys for the score they received on the scale of 1-5 for each metric. Total number of surveys were 119. As explained earlier Tone was measured on a scale (-1, 0, 1). In figure for tone, -1 is represented by 1, 0 by 2 and 1 by 3 respectively.

Each review used in the survey was evaluated using the automated metareviewer, which generated metareview score for each review. The metareview web service evaluates each sentence of a review and tries to identify positive or negative words used in it from a collection of word list. If the count is same, then it is marked as neutral. An aggregated score of all the sentences is calculated for the review. So if a review contains positive and negative sentences, then the overall score can have a score for positive metric as well as negative metric. But for our experiment, we scaled the overall tone score. If the overall positive score for a review was higher than the negative score, then it was translated to 1 (overall positive review). If overall negative score was higher than positive score, then it was translated to -1 (overall negative review), else it was converted to 0 (overall neutral review).

The survey participants had an absolute agreement (zero tolerance) of 38.8% with inter-rater reliability, calculated using weighted kappa [5], of 0.13. Inter-rater agreement increased to 80% when the tolerance was relaxed by one point (± 1). For reviews surveyed by more than one person, an average score was used to represent the final score. For some of the metrics in Figure 1, such as coverage, summative, and problem identification, the distribution is concentrated toward the center axis of graph. This explains the sudden increase of inter-rater agreement when the tolerance is relaxed by 1 point. Other metrics such as volume, relevance, and advisory shows a fair distribution cross the rating scale.

Sixty-five percent of reviews were rated as good whereas others were marked as bad by the survey experts. Table 2 lists the Pearson Correlation matrix between the score of the questions based on metareview metrics to the overall quality of the review as rated by survey participants. It can be easily inferred from Table 2, that each metric is highly correlated with the overall quality of the review, except tone. As per figure 1, volume and advisory are two most dispersed metrics and they also show greater correlation with the overall grade of a review, which makes them two most important metrics for data modelling.

Table 2: Pearson Correlation matrix for survey response for each metric and overall quality of a review (degree of freedom for each metric is 117, confidence interval: 95%)

Survey Metric	Pearson Correlation	p	t	95 % confidence interval
<i>Summative</i>	0.56	0	7.36	0.43 - 0.67
<i>Problem Identification</i>	0.57	0	7.56	0.44 - 0.68
<i>Advisory</i>	0.67	0	9.79	0.56 - 0.76
<i>Coverage</i>	0.68	0	10.0	0.57 - 0.77
<i>Relevance</i>	0.67	0	9.66	0.55 - 0.76
<i>Tone</i>	0.20	0.032	2.17	0.02 - 0.36
<i>Volume</i>	0.75	0	12.2	0.66 - 0.82

Table 3 shows the one-to-one correlation between the scores received for the survey question based on metareview metrics and metareview metrics from web service respectively. As per this table, web service and expert scores have the most agreement on the volume metric. Also other metrics such as summative, advisory and tone have appreciable agreements as well. The correlation between the relevance metric is very weak, which suggests that a changed strategy should be employed to improve performance of the relevance metric generator.

Table 3: Pearson Correlation between a metric score from survey and metareview system (degree of freedom for each metric is 117, confidence interval: 95%)

Metric	Pearson Correlation	p	t	95 % confidence interval
<i>Summative</i>	0.17	0.06	1.9	-0.01 - 0.34
<i>Problem Identification</i>	-0.03	0.74	-0.34	-0.21 - 0.15
<i>Advisory</i>	0.22	0.02	2.42	0.04 - 0.38
<i>Coverage</i>	0.02	0.87	0.16	-0.17 - 0.19
<i>Relevance</i>	0.01	0.94	0.08	-0.17 - 0.19
<i>Tone</i>	0.25	0.01	2.80	0.07 - 0.41
<i>Volume</i>	0.58	0	7.67	0.44 - 0.69

Table 4 shows the Pearson correlation between the scores from automated metareview metrics and the overall quality of the

review as per surveys experts. This translates to similar results, which we derived from Table 2. Based on the experiment and the data collected from automated metareviewing, volume, summative, and advisory are better suited metrics on which to create a model to categorize the quality of a review. Other metrics like tone, and problem identification should be used in modeling as well. But metrics such as relevance, and coverage are not performing well, so these metrics cannot be used for data modeling.

Table 4: Pearson Correlation between metareview metric score and overall quality of a review (degree of freedom for each metric is 117, confidence interval: 95%)

Metareview Metric	Pearson Correlation	p	t	95 % confidence interval
<i>Summative</i>	0.22	0.02	2.46	0.04 - 0.39
<i>Problem Identification</i>	0.13	0.16	1.42	-0.05 - 0.30
<i>Advisory</i>	0.25	0.01	2.77	0.07 - 0.41
<i>Coverage</i>	-0.02	0.81	-0.24	-0.20 - 0.16
<i>Relevance</i>	-0.05	0.61	-0.52	-0.23 - 0.13
<i>Tone</i>	0.15	0.11	1.60	-0.03 - 0.32
<i>Volume</i>	0.55	0	7.07	0.41 - 0.66

Decision-tree modeling and results

While selecting the model that can be used to differentiate between a good and a bad review, we investigated various modeling methodologies. We wanted a model that is inexpensive to construct, which can be retrained, and is extremely fast in classifying unknown reviews. Also since, we are ignoring two metrics in this modeling, we wanted a model that is flexible to incorporate these variables at a later stage. One modeling technique that looks ideal for these cases is a decision tree.

To create a decision tree, we started with Classification and Regression Trees (CART) modeling using the `rpart` [6] library in R. This library provides various ways to generate trees, such as classification and regression. The classification method is used in this experiment to generate the tree.

To find an optimal tree, a first attempt was made with volume, summative, advisory, problem identification and tone metrics. The summary function in `rpart` library shows that volume is a very important metric when generating the classification tree. Table 5 shows the result of the summary function, which states that the tone and problem identification were the least preferred metrics for classification

Table 5: Comparative variable importance for tree generation based on `rpart` library

Volume	Advisory	Summative	Tone	Problem Identification
64 %	15 %	13 %	4 %	4 %

From table 2, 3 and 4, Volume shows a stronger correlation with the class identifier (overall quality). Figure 2 shows that the volume metric alone can construct a classification tree to identify

review quality. This decision tree can be used to identify whether the review is good or bad on the basis of the volume score received from the automated metareview metric. For instance, if the volume metric score is greater than 68, then it is a good review, or if score is less than 26, that is a bad review. This is not pruned at the moment. Another algorithm discussed later generates a more pruned tree.

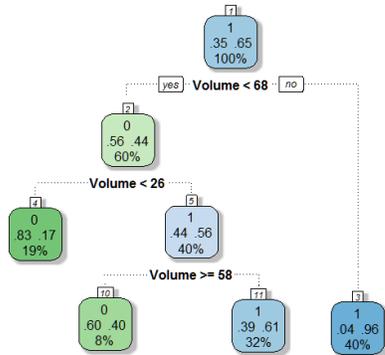


Figure 2: Unpruned Classification tree based on metareview score (using rpart).

Node 1 divided the sample space into two sets containing 42 and 77 observations respectively. A review with a score of 68.5 for metareview metric volume is used as the first split criterion. Each node number is marked in Figure 2, with split criteria and class probabilities.

Though volume can be a good classifier, volume alone should not be used to identify the quality of a review. We found in another study [7] that review volume may be related to the rubrics used in review phase. Some rubrics can ask for more feedback from reviewers than others. The volume metric can often be misleading and can result in higher number of false positives. A reviewer can provide gibberish comments which can result in a good metareview score for volume. We should consider other metrics as well to evaluate the overall quality of such a review.

This calls for another decision tree based on other metrics. Then, we can use both of these decision trees to classify a review. If any one tree classifies a review as a bad review, then that information can be shown to the reviewer as a guidance. This information can help the reviewer to correct issues with the review.

Figure 3 shows the decision tree created without the volume metric. We saw earlier that advisory and summative were next two stronger metrics after volume. As per the decision-tree construction algorithm, these two metrics can create the decision tree as well. Since these metrics suppress tone and problem-identification metrics, we could have created another decision tree based on tone and problem identification to further classify the review. But we chose to ignore them, since as per the `rpart` library’s metric importance their importance is very low compared to other three metrics used to generate trees in Figure 2, and Figure 3.

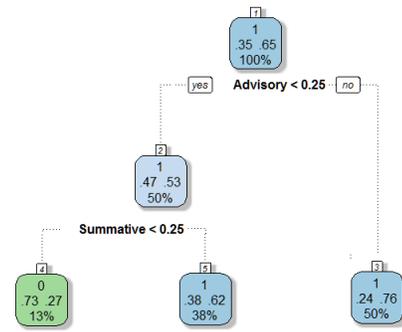


Figure 3: Classification Tree based on metareview scores, excluding volume (using rpart)

According to the tree in Figure 3, if a review receives a score in excess of 0.25 for advisory, then it is a good review, else we can check the score it receives for summative metric. If a review receives a score less than 0.25 for advisory and a review score in excess of 0.25 on summative, it is classified as a good review, else it is a bad review. As we can see that once the decision tree is created, process of classification of a review becomes easy.

In order to validate the results received from the `rpart` library, another method of tree construction was explored. One such method is C5.0 [8], which is an extension to C4.5 [9]. C5.0 is the package implemented in R, which is used to generate the tree based on the automated metareview score. 10-fold validation was used in decision tree construction. Figure 4 shows the final tree which includes all the metrics. As was noticed earlier in the tree constructed using the `rpart` classification method, the volume metric dominates the tree, and root node partition is based on volume “> 68”. This tree is shorter than tree in figure 2, because C5.0 uses tree pruning to create a shorter tree. Sometimes this pruning in result in increased classification error rate. The classification error rate for this tree is 22.7%. The majority class probability for this classifier tree is 80.7%, which is higher when compared to the baseline and the classification tree generated in Figure 3. One more tree was constructed without the volume metric, as is shown in Figure 5. The classification error rate for the tree is 29.4% which is higher than the similar CART based tree. The majority class probability using this tree comes to 87.4%, which is again higher compared to the baseline score and the other classification tree generated in Figure 3. This shows that the tree generated using `rpart` fits the data better than the similar tree generated using C5.0. C5.0 seems to generate a more pruned tree, which is smaller in size, but with an increased classification error rate.

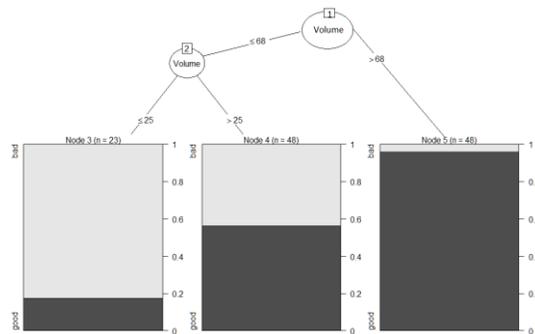


Figure 4: Classification tree based on metareview scores (using C5.0)

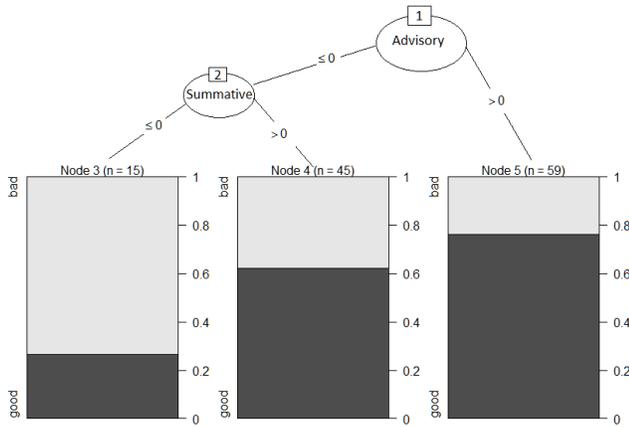


Figure 5: Decision Tree based on metareview scores, without volume metric (using C5.0)

Table 6 compares the performance for majority class prediction using different classification methods. Higher majority class probability compared to baseline probability means more false positives. C5.0 generates shorter trees compared to CART, at the cost of reduced accuracy at times. We found that CART based classification tree is better at classification than C5.0.

Table 6: Comparison of majority class probability using different classification methods.

Classification method	Majority class probability
Base line (based on experiments)	64.7%
CART (Metareview)	72%
CART (Metareview without volume)	88%
C5.0 (Metareview)	80.7%
C5.0 (Metareview without volume)	87.4%

5. Discussion and conclusions

Metareviewing is an essential tool, which can improve the quality of reviewing. A reviewer can write a good review if timely feedback can be provided on the review before he/she submits it to the author.

As part of this work, we created a decision-tree data classifier based on the score a review receives from the automated metareviewer. Decision trees are fast and efficient classifiers for peer review metrics. We found that certain metrics, such as volume, dominate the decision trees. But reliance on the volume metric alone can generate false positives. We also created a decision tree excluding the volume metric. That decision tree uses content advisory, content summative, tone and problem detection metrics. We suggest the use of a hybrid model that includes use of both the trees. Each review is rated on both trees from Figure 2 and Figure 3. A good review should score well on both.

5.1 Future work

We used Wikipedia artifacts and reviews written for them in this experiment. To make the model more robust, more similar experiments can be done to include artifacts from other educational domains. We used supervised learning to create this model. Natural language processing (NLP) is becoming more and more efficient in determining the semantics of a text. The relevance metric generator should be updated to make it more robust, so that it can also be used in the classification decision tree.

6. Acknowledgement

This work has been supported by the U.S. National Science Foundation under grants 1432347, 1431856, 1432580, 1432690, and 1431975.

7. References

- [1] K. Cho, "Machine classification of peer comments in physics," in *Educational Data Mining*, 2008, pp. 192-196.
- [2] W. Xiong and D. Litman, "Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews," *Proceedings of 6th International Conference on Educational Data Mining (EDM)*, 2013.
- [3] L. Ramachandran, "Automated Assessment of Reviews," in *PhD Dissertation at North Carolina State University*, Raleigh, 2013.
- [4] E. F. Gehringer, "Expertiza: Managing feedback in collaborative learning," in *Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-Learning Support*, IGI Global Press, 2010, pp. 75-96.
- [5] J. Cohen, "Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit," in *Psychological Bulletin*, 1968.
- [6] T. Therneau, B. Atkinson and B. Ripley, "rpart: Recursive Partitioning and Regression trees," 2015. [Online]. Available: <https://cran.r-project.org/web/packages/rpart/index.html>.
- [7] R. K. Yadav and E. F. Gehringer, "Metrics for Automated Review Classification: What Review Data Show," in *State-of-the-Art and Future Directions of Smart Learning*, Springer Singapore, 2016, pp. 333-340.
- [8] M. Kuhn, S. Weston, N. Coulter and M. Culp, "C5.0 Decision Trees and Rule-Based Models," CRAN, 08 03 2015. [Online]. Available: <https://cran.r-project.org/web/packages/C50/C50.pdf>.
- [9] Q. R. C4.5: Programs for Machine Learning., Morgan Kaufmann Publishers, 1993.
- [10] R. K. Yadav "Web Services for Automated Assessment of Reviews", in *MS Thesis at North Carolina State University*, Raleigh, 2016