
AC 2012-4776: AUTOMATIC QUALITY ASSESSMENT FOR PEER REVIEWS OF STUDENT WORK

Lakshmi Ramachandran, North Carolina State University
Dr. Edward F. Gehringer, North Carolina State University

Ed Gehringer is an Associate Professor in the departments of Computer Science and Electrical and Computer Engineering at North Carolina State University. He received his Ph.D. from Purdue University and has also taught at Carnegie Mellon University and Monash University in Australia. His research interests lie mainly in computer-supported cooperative learning.

Automated Quality Assessment for Peer Reviews of Student Work

Abstract

Reviews are text-based feedback provided by a reviewer to the author of a submission. Reviews are used not only in education to assess student work, but also in e-commerce applications, to assess the quality of products on sites like Amazon, ebay etc. Since reviews play a crucial role in providing feedback to people who make assessment decisions (deciding on a student's grade, purchase decision of a product etc.), it is important to ensure that reviews are of a good quality. In our work we propose the use of metrics such as content, tone and quantity of feedback to suitably represent a review. We use supervised classification techniques to determine content and tone of the feedback. Our approach predicts the metareview score for a review, which is indicative of its quality. The individual metrics together with the metareview score give reviewers immediate feedback that is likely to help them improve the quality of their reviews. We conducted experiments, to evaluate our automated metareviewing approach, on reviews collected using Expertiza, a web-based collaborative learning environment. Our approach produces accuracy values greater than 60% in predicting content and tone of reviews and an accuracy of 62.9% in predicting metareview scores.

Keywords: automated metareviewing, latent semantic analysis, review quality

1. Introduction

A review is considered to be of a good quality if it can help the author identify mistakes in his work and also learn possible ways of fixing them. Reviewers often tend to provide vague, unjustified comments, which are not relevant to the author's submission. The first two reviews in Table 1 are generic and do not refer to a specific object in the author's submission. The first two comments, which praise the author's work with adjectives such as "good" and "correct", when taken independently, do not contain any information that could help authors improve their work.

Reviews aid in the decision making process, whether it is a student's grade or the decision to accept or reject a scientific paper. It is therefore important to ensure that the reviews are of a good quality, i.e., they provide detailed information that can be used by the author. For example, they might point out problems in the author's work or provide suggestions to improve the work, similar to that in the last two comments in Table 1, below.

Reviewer feedback can be evaluated by a process referred to as metareviewing. Metareviewing is defined as the process of *reviewing* reviews, i.e., the process of identifying the quality of reviews. Metareviewing is a manual process and just as with any process that is manual; metareviewing is (a) slow, (b) prone to errors and is (c) likely to be inconsistent. An automated review process ensures consistent (bias-free) reviews to all reviewers. It also provides immediate feedback to reviewers, which is likely to motivate reviewers to improve their work and provide more useful feedback to authors.

Table 1: Some examples of review comments.

Reviews
“The language used is good.”
“Yes, the illustrations are technically correct.”
“The example code for delegation is taken from one of the references listed at the bottom of the page.”
“I would like to see a better definition/explanation of each technique before getting into the advantages and disadvantages.”

This paper discusses an automated metareviewing technique that uses machine learning to provide quick and reliable feedback to reviewers on their assessment of students’ submissions and help them produce better quality reviews.

In order to measure review quality, we first define metrics that suitably represent the features of the review. These features include *review content*, *tone* and the *amount of feedback* provided by the reviewer [6]. Content of a review identifies the type of feedback provided by the reviewer. Depending on the purpose for which a review is written, it could be classified as either being indicative or evaluative or a combination of both [5]. *Indicative* reviews contain brief summaries of the work under evaluation, while *evaluative* reviews provide criticisms on the author’s work with possible suggestions for improvement. Similarly, we classify the content of reviews based on if they were indicative of the content, were identifying problems or were suggesting possible improvements to the work.

Our approach to categorizing reviews is similar to that proposed by Kwangsu Cho in his paper, “Machine Classification of Peer Comments in Physics” [1]. Cho uses naïve Bayes, support vector machines (SVM) and decision trees to classify complete reviews (using all the words in the review as features) as *praise*, *criticism*, *problem detection* or *solution suggestion*, *summary* or *off-task* comment. His approach does not involve identifying review features such as content, tone and quantity to identify review quality.

Tone of feedback is important especially because, while providing negative criticism, reviewers might unknowingly use words or text that might offend the authors. In order to avoid doing so, we use tone to help guide us while writing reviews. A review can have three types of tones – *positive*, *negative* or *neutral*. Tone of feedback can be identified by studying the semantic orientation of a review, which is indicated by the presence or absence of positively or negatively oriented words [8]. Turney uses semantic orientation to determine whether a review can be classified as recommended or not recommended. Turney’s approach to differentiating positive from negative reviews involves identifying similarity between phrases containing adverbs and

adjectives to terms “excellent” and “poor” to determine the semantic orientation of the phrase. In our approach we use pre-annotated text to identify the tone of new reviews that come into the system.

In our approach, review content and tone are determined using a supervised text-classification technique – latent semantic analysis (LSA) [4]. Latent semantic analysis produces a concise representation of term-document relationships, which aids in the classification of reviews for content and tone. Review quantity is a count of the number of unique tokens in the review text.

Reviews are given scores, referred to as *metareview scores* that indicate their quality on a Likert scale of values 1 to 5, where 1 is the lowest and 5 is the highest score. Reviews containing similar content, tone and quantity are more likely to have similar metareview scores. Therefore new reviews represented in terms of their content, tone and quantity and are compared with previously metareviewed reviews to determine their metareview or review quality scores.

The rest of this paper is organized as follows. Section 2 describes our approach, which includes the text pre-processing technique applied to the reviews, followed by a description of the different review metrics and a brief description of the text classification technique - latent semantic analysis. Section 3 describes the experiments conducted and Section 4 concludes the paper.

2. Approach

The process of predicting metareview scores for text-based reviews involves the following steps.

2.1. Text Pre-processing:

Before metrics are calculated, reviews are subjected to pre-processing, which involves breaking the reviews down at *transition keywords*. Transition keywords are words such as “but”, “however”, and “either ... or”, which change the meaning of the portion of the sentence that follow them. Since reviews are subjected to text classification, for content and tone identification, we have to make sure that the text does not contain opposing ideas, which could possibly mislead the classifier. For example the review, “The wiki is clear, but has a few grammatical errors.” conveys multiple thoughts, i.e., while “The wiki is clear” is a praise, the segment “ but has a few grammatical errors” identifies a problem in the author’s work. Therefore this pre-processing step helps avoid mis-classifying the complete review as either just a praise or as a criticism of the author’s work.

However, this pre-processing step works only when the reviews contain transition keywords. But there are times when reviewers provide feedback that contain multiple opposing ideas that may not be easily distinguishable by the system. Consider the review, “ There are plenty of links, and the citations seem done appropriately (with exception for the pictures) . I’ve already counted off a point for links for the source of the pictures”. This review is similar to the previous example in

that it provides a positive criticism “There are plenty of links, and the citations seem done appropriately”, but provides a negative comment within the braces “with exception for the pictures” and which continues into the next sentence “I’ve already counted off a point for links for the source of the pictures”. In the case of such reviews it is hard to identify a demarcation between the different idea segments. Therefore the supervised classification technique (discussed in Section 2.3.) identifies the category this review is *most likely* to belong to.

The reviews that are broken at transition keywords are then subjected to another pre-processing step. In this step reviews containing the word “not” are selected and the phrase “not” + word is replaced by the antonym of the word. Reviews containing phrases “is a good book” and “not a good movie” could be deemed similar due to the presence of the word “good”, although the phrases contain opposite meanings. This pre-processing therefore helps avoid such confusion.

2.2. Automated Review-Quality Assessment Metrics

In order to assess quality, reviews have to be first represented using metrics that capture their most important features. In general a good quality review contains:

- (i) coherent and well-formed sentences, which can be easily comprehended by the author, as well as,
- (ii) sufficient amount of feedback.

In this section we discuss in detail the steps that we take to calculate each of the following metrics:

- (1) Content
- (2) Tone and
- (3) Quantity of feedback

2.2.1. Content

A review is expected to provide an assessment of the kind of work that was done - praising the submission’s positive points, identifying problems, if any, and offering suggestions on ways of improving the submission. Based on this we classify content of a review into the following three classes:

Summation – Reviews that fall into this class are those that contain either a positive or a neutral assessment of the author’s submission. These type of reviews tend to be little more than *only* summaries of the author’s work, with no additional information provided. The reviewer does not point out any problems in the work, or does not offer a suggestion for improvement.

Example of a summative review: “I guess a good study has been done on the tools as the contents looks very good in terms of understanding and also originality. Posting reads well and appears to be largely original with appropriate citation of other sources.”

Problem detection – Reviews in this category are critical of the author’s submission and point out problems in the submission. However, they do not offer any suggestions to improve the work.

Example of a problem-detecting review: “There are few references used and there are sections of text quoted that appear to come from a multitude of web sites. Less dependence on wikipedia would be good.”

Advisory – Reviews that offer the author suggestions on ways of improving his/her work fall into this category. Reviews of this type also display an understanding of the author’s work.

Example of an advisory review: “Although the article makes use of inline citations which is a plus, there are only a few references. Additional references could help support the content and potentially provide the examples needed.”

Based on the type of content a review contains we decide the content quality of a review. For instance summative reviews provide only summaries of the author’s work and are less useful to the author, whereas reviews that identify problems in the author’s work or provide possible suggestions for improvement can be used by authors to improve their work and are hence considered more important.

2.2.2. Tone of Feedback

Tone pertains to the semantic orientation of the text. Semantic orientation depends on the reviewer’s choice of words and the presentation of the review. Semantic orientation or tone of the text can be classified into one of the following categories:

Positive – A review is said to have a positive tone if it predominantly contains positive feedback, i.e., it uses words or phrases that have a positive semantic orientation.

Example of a review containing a positive tone: “The page is very well-organized and the information under corresponding titles is complete and accurate.”

Adjectives such as *well-organized*, *complete* and *accurate* are good indicators of a positive semantic orientation.

Negative – Into this category are placed reviews that predominantly contain words or phrases that have a negative semantic orientation. Reviews that provide negative criticism to the author’s work are most likely to fall under this category, since while providing negative remarks reviewers tend to use language or words that are likely to offend the authors. (Such reviews could be morphed or written in such a way that is less offensive to the author of the submission.)

Example of a review containing negative tone: “The examples are not so easy to understand and have been borrowed from other sources. Although the topic is Design Patterns in Ruby, no examples in Ruby have been provided for Singleton and Adapter Pattern.”

Although this review does not contain explicit negatively oriented words we notice that it does have a negative orientation. Review segments such as *not so easy to understand, have been borrowed from other source* and *no examples in Ruby* are indicators of the same. Our text classification technique is trained on reviews containing similar negatively oriented reviews in order to identify the orientation of such reviews.

Neutral – Reviews that do not contain either positively or negatively oriented words or phrases or contain a mixture of both (positively and negatively oriented words or phrases) are classified into this category.

Example of a review containing a neutral tone: “The organization looks good overall. But lots of IDEs are mentioned in the first part and only a few of them are compared with each other. I did not understand the reason for that.”

This review contains both positively and negatively oriented segments, i.e., “The organization looks good overall” is positively oriented, while “But lots of IDEs are mentioned in the first part and only a few of them are compared with each other. I did not understand the reason for that.” is negatively oriented. Hence, it is classified as a neutral review.

In case of both content and tone, a single review is likely to belong to multiple categories, which are not easily distinguishable (by the system) (as explained in Section 2.1. on Text Pre-processing) due to the nature of the text. For instance consider the review,

“Examples provided are good; a few other block structured languages could have been talked about with some examples as that would have been pretty helpful and useful to give a broader pool of languages that are block structured.”

While classifying for content, we see that the first part of the review, “Examples provided are good” praises the submission, while the second part, “a few other block structured languages could have been talked about with some examples as that would have been pretty helpful and useful to give a broader pool of languages that are block structured,” provides advice to the author. Our text categorization technique (LSA and cosine, explained in Section 2.3.) identifies the class that the review has the highest probability of belonging to. In the case of this review we see that the advisory part is more pronounced than the summative part and therefore such a review gets classified as an advisory review.

2.2.3. Quantity of Feedback

Text quantity is important in determining review quality since a good review must provide the author with sufficient feedback. We plan on using this metric to indicate to the reviewer the amount of feedback he/she has provided in comparison to the average review quantity (from other reviewers of the system), thus motivating them to provide more feedback to the authors. Quantity of feedback is identified by taking a count of all the unique tokens in a piece of review

text. For instance, consider the following review, “The article clearly describes its intentions. I felt that section 3 could have been elaborated a little more.” The number of unique tokens in this review is 15 (excluding articles and pronouns).

In the next section we describe the text classification technique employed to identify the content and tone of new reviews.

2.3. Latent Semantic Analysis and Cosine Similarity

Latent semantic analysis is a widely used text classification technique [Landauer1998]. Latent semantic analysis transforms a matrix containing relationships between documents (reviews) and its terms (tokens in a review) from a high-dimensional space into one with a reduced number of dimensions. LSA uses truncated singular value decomposition to accomplish this. The resultant matrix contains values that help identify the degree to which tokens across different reviews belong to a certain review, instead of the 0s or 1s, which is what a simple term-frequency matrix contains.

Cosine is a similarity measure that helps identify the degree of similarity there exists between reviews (in terms of their tokens). Cosine produces a similarity value in $[0, 1]$, where a 0 indicates no similarity and a 1 indicates an exact match between the reviews.

Reviews from the past that have been annotated with content and tone information are used to build a model using LSA. The model is used to identify the content and tone of new reviews by identifying the similarity values, using cosine, of the new reviews with existing reviews. The content and tone of existing reviews that are *closest* to the new reviews are identified as the content and tone of the new review.

2.4. Review Vector Representation

Reviews are represented using the above-mentioned metrics, which form review vectors:

review_vector (content, tone, quantity)

In order to predict metareview scores for new reviews, review vectors of new reviews are compared with those of existing reviews, which have been manually metareviewed. This is done because reviews containing similar review vectors are likely to have similar metareview scores. We use cosine similarity to compare new and existing review vectors.

3. Evaluation

In this section we discuss some of experiments that we conducted to test our automated metareviewing technique. The statistical analysis tool R is used to carry out the text analysis [7]. The *lsa* package available for R is used to perform classification using LSA and cosine. For text pre-processing, the text mining and natural language packages such as *tm*, *openNLP* and *Wordnet* are used.

3.1. Expertiza

Review data consisting of textual feedback provided by students was collected from courses at North Carolina State University that used the Expertiza system [2, 3]. Expertiza is a collaborative web-based learning application that helps students work together on projects and critique each other's work using peer reviews. Figure 1 shows a screenshot of a review questionnaire presented to students. Similar questionnaires are used to collect (manual) metareview scores for reviews. As a response to each question a student provides a textual response as well as a

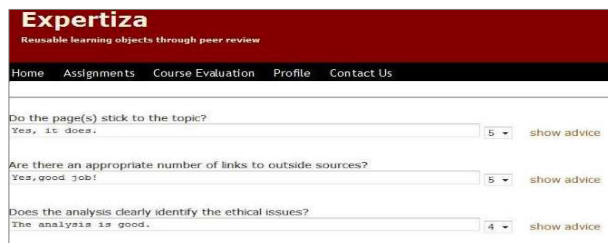


Figure 1: Screenshot of a review questionnaire from Expertiza, which contains textboxes for students to write out textual reviews and dropdown boxes for scores.

numeric score.

Expertiza frequently hosts courses related to software engineering, and the assignments, reviews and metareviews completed using Expertiza provide us with the required data to study review quality and predict metareview scores.

3.2. Experiments and Results

The aim of this experiment is twofold:

1. To identify the performance of our approach in identifying the content quality and tone of reviews.
2. To identify the extent to which metareview scores predicted by our approach agree with human-given metareview scores.

3.2.1. Identifying the Content Quality and Tone of Reviews

Around 993 reviews with known content and tone values were used as the training data set, to build a model that makes predictions for reviews in the test set. The test set consisted of 361 reviews that were collected from wiki assignments in which teams of students work together to write or edit a wiki page on a certain topic.

The reviews used for training the model were annotated by a single human annotator (referred to as X). In order to estimate the correctness of these labels provided by the human annotator, we randomly selected 10% of the reviews and got three other annotators to classify the data. We found an average agreement of 83.5% between each of the annotators and X.

In order to identify the accuracy of our approach to predict the content and tone metrics, we compare the values generated by our technique with those provided by a human annotator. The accuracies are listed in Table 2. Our approach has a good degree of agreement with the human annotations producing greater than 60% accuracy for both content and tone. Our accuracies are better than values a random assignment of classes would produce 33% for content and tone (three classes in each category).

Table 2: Accuracy of our approach in predicting content and tone of reviews.

Type of classification	Accuracy
Content	62.3%
Tone	62.8%

3.2.2. Metareview Score Prediction

In order to test the performance of our model in predicting metareview scores, we select 361 reviews, each of which has been metareviewed by a human metareviewer. Each of these reviews is represented by metrics content, tone and quantity of feedback.

The set of 361 reviews is then divided into 253 for training and 108 for testing. Our system predicts metareview scores for each of the 108 reviews in the test set by identifying the review vector in the training set that is closest to it and using that training review vector’s score as the test review vector’s metareview score.

The performance is measured by identifying the number of scores predicted by the system that were closest to the actual scores for each of the test reviews. We use 0.5 as the threshold, i.e., if the difference between the actual and predicted scores is less than or equal to 0.5, the predicted scores are considered to be *correct* predictions. The results are described as follows.

For 62.9% of the test reviews the difference between the predicted and actual scores was less than or equal to 0.5, meaning that the prediction was correct.

12% of the reviews did not contain any textual feedback, yet the metareview scores given to these reviews were greater than 1 (the lowest score a review can get). Our system predicts metareview score as 1 for reviews with no feedback. As a result, for 12% of the reviews the scores given by the system did not agree with those given by metareviewers. We noticed that metareviewers tend to be generous and give reviewers high scores even when they did not fully deserve it. Since our system judges reviews based on the contents of the textual feedback, these reviews are given the lowest score, due the absence of any feedback. Our new accuracy after adjustment for scores predicted for empty reviews is 74.9%.

The accuracy value indicates that reviews with similar content, tone and quantity metrics tend to have similar metareview scores. Thus these metrics could be useful in predicting metareview scores and thus the review quality of new reviews that come into the system.

One of the reasons for the metareview scores to be so high could be that metareviewers are not sufficiently informed and need better guidance on the process of metareviewing. A more specific rubric or questionnaire is likely to guide students to provide better metareviews.

4. Future Work

In the future we plan to investigate the usefulness of our approach in assessing reviews written for scientific articles and peer-reviewed journal or conference papers. One of our goals is to obtain more review data to further evaluate our approach and establish its usefulness. We also plan to integrate a review relevance identification component into our system. The review relevance identification component will check the review for its relevance to the content in the author's submission. This will help us identify reviews that appear to be generic as well as reviews that do not provide justifications for their criticisms.

We also plan to conduct user studies to evaluate the effectiveness of the automated review quality assessment technique. We plan on investigating the effect of such an automated approach on reviewers and identify the extent to which it helps them improve their reviewing skills. We are also interested in studying authors' perspectives on improved reviews and identify if better reviews motivate them to improve their submissions.

5. Summary

In this paper we introduce the process of automated metareviewing, which aims to provide students with the guidance they need to improve their reviews. Automated metareviewing provides students with immediate feedback on the quality of their reviews. Our aim with this approach is to guide reviewers to provide better quality reviews to the authors. Better quality reviews are more likely to inspire authors to improve their work by incorporating these review comments.

Our approach to review-quality assessment is unique in that it uses metrics such as content, tone and quantity of textual feedback to study review quality. We use a supervised text classification approach – latent semantic analysis, to identify the content and tone of reviews. We use cosine similarity to determine the metareview scores of reviews by comparing their review vectors with those of previously metareviewed reviews. Metareview scores given on a Likert scale of values 1 to 5 along with the different review metrics act as good indicators of the review's quality. Our approach has an accuracy of 62.3% in predicting content and 62.8% in predicting tone and has an accuracy of 62.9% in predicting metareview scores for new reviews.

References

- [1] K. Cho, “Machine classification of peer comments in physics,” in *Educational Data Mining*, 2008, pp. 192– 196.
- [2] E. F. Gehringer, L. M. Ehresman, and W. P. Conger, S.G., “Reusable learning objects through peer review: The Expertiza approach,” in *Innovate: Journal of On- line Education*, 2007.
- [3] E. F. Gehringer, “Expertiza: information management for collaborative learning,” in *In Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-Learning Support*. IGI Global Press, 2009.
- [4] F. P. W. Landauer, T. K. and D. Laham. An introduction to latent semantic analysis. In *Discourse Processes. Special Issue on Quantitative Approaches to Semantic Knowledge Representations*, 259–284. Volume 25. 1998.
- [5] T. A. S. Pardo, L. H. M. Rino and Maria das Graças Volpe Nunes, “Extractive summarization: How to identify the gist of a text,” in *Proceedings of the 1st International Information Technology Symposium - I2TS*, Florianopolis-SC, Brazil, October 1-5 2002, pp. 1–6.
- [6] Lakshmi Ramachandran and Edward F. Gehringer. 2011. Automated assessment of review quality using latent semantic analysis. *11th IEEE International Conference on Advanced Learning Technologies*. 136-138 July.
- [7] R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- [8] P. D. Turney, “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews,” in *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.