A Method to automatically choose Suggestions to Improve Perceived Quality of Peer Reviews based on Linguistic Features

Markus Krause
UCBerkeley, ICSI
public.markus.krause@gmail.com

Abstract

Peer reviews can be used to grade students in large online classes or to ensure response quality in crowdsourcing. Poorly written reviews can significantly harm student and contributor satisfaction. We present a method based on our natural language model to provide suggestions for reviewers to improve their writing style. We report the results of a user study that illustrates the positive effects of our method on perceived review quality. In our experiment our language based method increases perceived review quality by 20%.

Introduction

Peer review is a widely used method. Balfour illustrates its applicability in grading open text assignments in MOOCs (Balfour, 2013). Other peer assessment methods have demonstrated their utility in general grading online course submissions at scale (Piech et al., 2013). In crowdsourcing peer reviews can increase response quality and user satisfaction (Hansen, Schone, Corey, Reid, & Gehring, 2013; Luther et al., 2014). In this paper, we investigate the effect of different feedback methods on the perceived quality of peer reviews. In our study, we provided student *reviewers* with *suggestions* on their reviews of classmate's code submissions. We asked reviewers to revise and edit their review based on our *suggestions* and asked their peers to evaluate the quality of the revised reviews. We compare three different feedback methods 1) a generic feedback without suggestions on linguistic style (*none*) 2) a feedback with suggestions on all linguistic features (*all*) 3) a selective feedback with suggestions on only those linguistic features selected by our algorithm (*selective*). We calibrated our algorithm with reviews written by educators. This paper will demonstrate that reviewers using the selective feedback method received significantly higher ratings for their revised reviews than students using the other methods. We recruited 60 students in an undergraduate python-programming course (15 female). We randomly assigned individuals into one of the three conditions.

Language Model

We base our linguistic model on a feature set that has previously been used to investigate writing styles in educational settings (Kilian, Krause, Runge, & Smeddinck, 2012; Krause & Porzel, 2013; Krause, 2014). We use the following set of features: text length (average word length, average sentence length, number of sentences), emotional content (valence and arousal), language specificity, and sentence mood. We preprocessed all reviews with the NLTK part-of-speech (POS) tagger (Bird, Klein, & Loper, 2009). We then filtered stop words and words not in Wordnet (Miller, 1995). Wordnet is a natural language tool that provides linguistic information on more than 170,000 words in the English language. We also lemmatized the remaining words to account for different inflections. We calibrate our method using already rated reviews from previous instances of the course. We calculate mean and SD of the 25 most highly rated reviews out of a pool of 53 for each language feature. In the *selective* condition we gave feedback on a language feature only if the review differed more than 1.5 SD from the average of these reviews. In the *all* condition reviewers got suggestions on all features regardless of their divergence from the average in our calibration set.

Text length: the first three features we examined were the mean number of letters per word, mean number of words per sentence, and mean number of sentences per text. For the mean word length we considered only those words that have a *Wordnet* entry and are not stop words. The sentence length was measured including all words returned by the POS-tagger.

Emotionality: The next two features we looked at were valence and arousal. Valence refers to whether the review is positive, negative, or neutral, and arousal represents how strong the valence is. The normalized value of valence and arousal ranged from -1 to 1 and 0 to 1, respectively. Some examples, with normalized feature values, are provided below. We used *pattern.en*, a tool based on *NLTK*, to extract valence and arousal.

Valence=1.0 and arousal=1.0: *This is very good! I like the way the hierarchy is structured and it really helped me to understand my own code better!*

Valence=0.0 and arousal=0.0: *The structure is interesting but I understand why things are laid out this way.*

Specificity: Another feature we explored was specificity, which refers to how specific the words in the review were. We measured specificity by determining how deep each word appears in the *Wordnet* structure. Words that are closer to the root are more general (e.g. *dog*) and words deeper in the *Wordnet* structure are more specific (e.g.). Word depth ranges from 1 to 20 (20=most specific). To simplify the analysis and presentation, we normalize specificity to range from 0.0 to 1.0.

Specificity=1.0: *I like the consistency of the ontology the author applies to structure her code.*

Specificity=0.0: *I am not sure if this is a good idea.*

Sentence Mood: the last feature we considered involves looking at the moods of sentences in each review. Each sentence was classified as either indicative (written as if stating a fact), imperative (expressing a command or suggestion), or subjunctive (exploring hypothetical situations). The feature, which we refer to as *active*, corresponds to the ratio of non-indicative sentences in a review, with values falling between 0 and 1. See below for some examples. We again used *pattern.en* to extract sentence mood.

Active=1.0: *I would recommend structuring this class differently. I think you should have only one section with static variables.*

Active=0.0: *I am not sure about this structure. I think it could be structured differently.*

Results

As figure 1 shows our method has a positive impact on perceived review quality. We analyzed our results with an Analysis of Variance (ANOVA) that showed a significant influence of our three conditions on the perceived review quality F(2, 59)=3.948, p=0.021. We used Tukey's HSD test as post hoc test to find individual differences between all levels as shown in table 1. The results show that in our experiment reviewers using our selective method received better ratings on their revised reviews than reviewers in the other conditions. We see that the suggestions are not useful by themselves as there is no significant difference between the *all* (all suggestions available) and *none* (no linguistic suggestions available) condition.

	diff	*95% CI low*	*95% CI high*	*Adj. p*
none-all	0.35	-0.70	1.41	0.71
selective-none	0.87	0.10	1.91	0.04
selective-all	1.22	0.16	2.29	0.02

Table 1: Results of Tukey's HSD test. Our selective feedback method is significantly better than general feedback or feedback on all linguistic features.

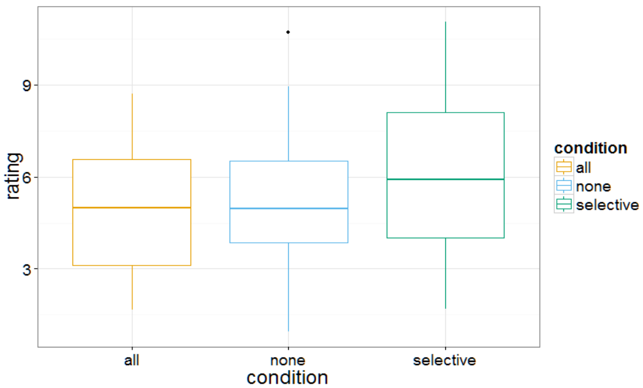

Figure 1: Average rating on the final review from reviewers from the different conditions. Showing only those improvement suggestions that differ more than 1SD from the calibration (selective condition, on right) significantly improves review ratings.

References

Balfour, S. (2013). Assessing writing in MOOCS: Automated essay scoring and Calibrated Peer Review. *Research & Practice in Assessment*, *8*, 40–48.

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*.

Hansen, D. L., Schone, P. J., Corey, D., Reid, M., & Gehring, J. (2013). Quality Control Mechanisms for Crowdsourcing. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work - CSCW '13* (p. 649).

Kilian, N., Krause, M., Runge, N., & Smeddinck, J. (2012). Predicting Crowd-based Translation Quality with Language-independent Feature Vectors. In *HComp'12 Proceedings of the AAAI Workshop on Human Computation* (pp. 114–115).

Krause, M. (2014). A behavioral biometrics based authentication method for MOOC's that is robust against imitation attempts. In *Proceedings of the first ACM conference on Learning @ scale conference - L@S '14* (pp. 201–202).

Krause, M., & Porzel, R. (2013). It is about time. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13* (p. 163).

Luther, K., Pavel, A., Wu, W., Tolentino, J., Agrawala, M., Hartmann, B., & Dow, S. P. (2014). CrowdCrit: crowdsourcing and aggregating visual design critique. In *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW Companion '14* (pp. 21–24).

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., & Koller, D. (2013). Tuned Models of Peer Assessment in MOOCs. In *Proceedings of the 6th International Conference on Educational Data Mining (EDM'13)* (pp. 153–160). Memphis, TN, USA.