Board 60: PeerLogic: Web Services for Peer Assessment

Dr. Edward F. Gehringer, North Carolina State University

Dr. Gehringer is an associate professor in the Departments of Computer Science, and Electrical & Computer Engineering. His research interests include computerized assessment systems, and the use of natural-language processing to improve the quality of reviewing. He teaches courses in the area of programming, computer architecture, object-oriented design, and ethics in computing.

PeerLogic: Web Services for Peer Assessment

Abstract

Peer assessment means students giving feedback on each other's work. Dozens of online systems have been developed for peer assessment. All of them face similar issues. The PeerLogic project is an effort to develop specialized features for peer-assessment systems so that they can be used by multiple systems without the need for re-implementation. In addition, the project maintains a "data warehouse," which includes anonymized peer reviews from different peer-assessment systems, which are freely made available to researchers.

Keywords: peer assessment, PeerLogic, sentiment analysis, reputation algorithm, topic assignment

1. Introduction

Peer assessment means students giving feedback to each other on their work. In the past 20 years, dozens of online systems have been developed to manage the process of peer feedback. Typically, students complete a homework assignment and then upload it to the online system. The system then assigns other class members as peer reviewers for the submitted work. Depending on the system, the authors may then have an opportunity to revise the work before a final, summative peer review.

Peer assessment[1] has many advantages. Students receive more copious feedback on their work than they would from an instructor or a TA. Rarely would an instructor have time to look over a first draft and then a final submission, but peer assessors, who have only a handful of submissions to review, can do so. Moreover, the feedback comes more quickly. An author can usually see the feedback as soon as the reviewer provides it, rather than having to wait until the instructor or TA is finished grading all the students. Finally, peer assessment forces students to write in a way that their peers can understand. They can't use shorthand that the instructor, with his/her superior knowledge, is expected to decipher. They learn to write for an audience of their peers, which is exactly the skill they need for later in their careers. Peer assessment has been shown to improve learning across the curriculum [1].

Online peer-assessment systems perform the same basic functions, though they often have features aimed at the types of courses taught by their designers, e.g., art critiques (Critviz [2]), case studies (Mobius SLIP [3]), or composition and writing (Eli Review [4] and MyReviewers [5]). These systems face many of the same issues. The PeerLogic project is an effort to develop specialized functionalities for peer-assessment systems so that they can be used by multiple systems without the need for re-implementation. It provides a JSON (Javascript Object Notation) interface so that a peer-assessment application can pass requests to the service and receive results back. In addition, the project maintains a "data warehouse," which includes

[1] "Peer assessment" is the term usually used for a class assignment where students assess each other's work. "Peer review" more often means the process of vetting academic work for merit (e.g., for publication or funding).

anonymized peer reviews from different peer-assessment systems, which are freely made available to researchers.

2. Reputation web service

First, we provide a reputation web service. A reputation algorithm is one means of vetting the credibility of a reviewer. In courses where peer-assessment scores can influence a student's grade, instructors want to have some level of trust in their peer assessors. A reputation algorithm compares scores given to the same artifact by different reviewers. A reviewer achieves a high reputation by both (i) awarding scores that are close to the scores given by other reviewers, and (ii) awarding substantially different scores to different artifacts. In other words, a good reviewer cannot be too far off from other reviewers' scores, and cannot blindly assign the same "average" score to all kinds of work. A peer-assessment system can calculate the grade for an artifact based on a weighted average of the scores assigned by different reviewers, with the weights being the reputation scores.

Several different reputation algorithms have been defined. Our web service implements the Hamer algorithm [6] and the Lauw algorithm [7]. Song et al. [8] studied the reputations computed by the two algorithms. In general, the Lauw algorithm delivers a narrower range between the lowest and highest reputation, though the two algorithms agree on which peer assessors are most and least reliable.

3. Sentiment analysis

Sentiment analysis [9] means analyzing written language in order to determine emotions and reactions to various kinds of phenomena. It has seen widespread application to product reviews, Twitter feeds, and management science, as well as other fields. It is important in peer assessment, because formative peer feedback is only useful if it is acted upon, and if the tone is too hostile or negative, it is likely to be disregarded.

The Peerlogic project provides a web service to determine how positive or negative a review comment is. It uses the VADER [10] model to assign a score to a review comment. The model is sensitive to both the polarity and intensity of sentiments expressed in social media contexts. When passed one or more text strings, the web service returns the overall sentiment of the text, as well as a vector indicating the amount of negative, neutral, and positive components.

We are currently extending our work to provide web services for recognizing peer-review comments that include suggestions or identify problems in the review text. Our goal is to enable a peer-assessment system to give feedback to a reviewer *before* a review is submitted—for example, reporting that a review seems too negative, that it does not say what needs to be improved, or does not give enough suggestions for improving the work. Then the reviewer will have an opportunity to improve the review before submitting it.

4. Summarization service

In some circumstances, authors can be overwhelmed by the amount of peer feedback they receive. Suppose that students work in teams of three to submit their work. Usually, each

student is asked to do about three reviews. This means that each team will receive about nine reviews—or even more, if students are allowed to earn extra credit by doing extra reviews. The review rubric may ask the student to comment on, say, a dozen aspects of the work. When each team has 100 or more review comments to sort through, it can be very helpful to see a summary.

Our summarization web service incorporates the *sumy* library, which has integrated seven different summarization algorithms. All seven are available to client systems. We performed a comparison [11] of the three most promising algorithms, Latent Semantic Analysis (LSA), TextRank, and KL-Sum, and asked students several questions about the efficacy of each. TextRank was rated most useful in all aspects except readability, where LSA received the highest rating.

5. Visualization of ranking

Peer-assessment systems can be divided into those that use *rating* or *ranking*. A system that uses rating asks peer assessors to rate each artifact on a numerical scale. In some systems, an artifact is rated holistically; other systems allow it to be rated on each of several rubric criteria. A system that uses ranking asks peer assessors to rank the reviewed artifacts against each other.

In a ranking system, an instructor will want to see how every class member fares in the composite ranking by all of their peers. The "rainbow graph" is a visualization that shows how each reviewer's rank contributed to the overall ranking. The graph is essentially a stacked bar chart. An artifact that received all "1"s would be at one side of the chart, and an artifact that received all "5"s would be at the other. Ratings of "1" are shown as taller bars than lower ratings, so that the stacked bar for a higher-rated artifact is taller than other bars. We provide the rainbow chart as a web service for ranking-based systems. Here is an example of the visualization.

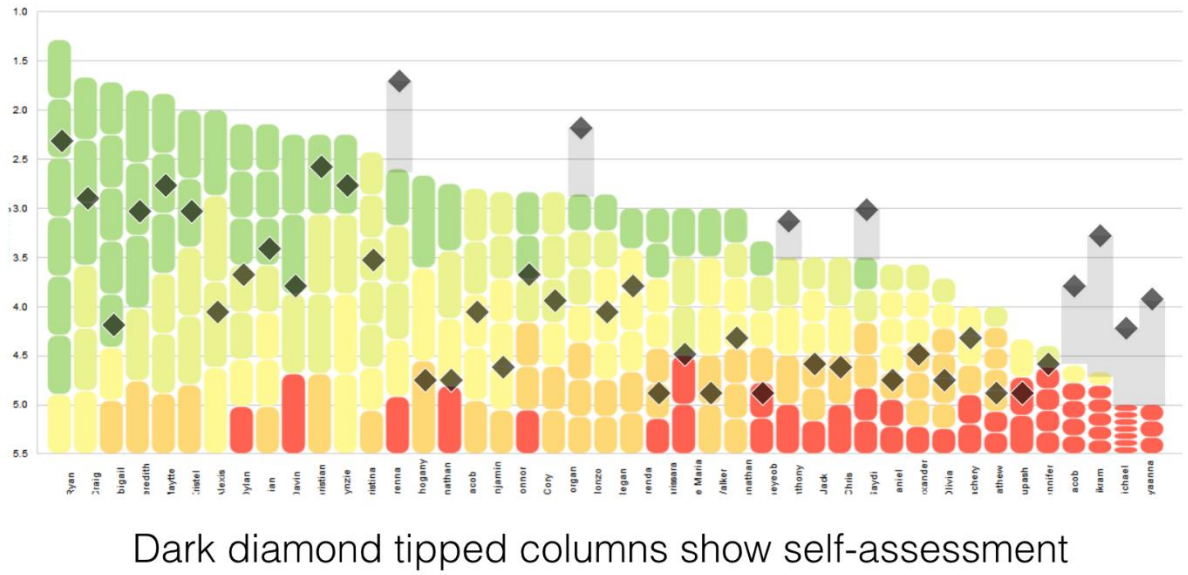

Figure 1. Example of visualization of rankings

The green bars represent a first-place ranking from one's peers. They are taller than the (yellow) second-place ranking bars, or the bars for any other rank. In this case, the highest-ranked artifact received six first-place rankings and one second-place rankings from its peer assessors. At the other end of the chart, someone who received all last-place rankings has the shortest composite bar. The service can also display self-assessment rankings on the same scale as peer-assessment rankings.

6. Team formation

Design projects are frequently assigned in courses (often capstone courses) where a limited number of student teams can pick each project topic. While many criteria can be used to form teams, one of the most important is that students are assigned topics that they have an interest in, and background for, a specific topic. The problem involves both aggregating students into teams and assigning teams to topics. The design space is very large, and an optimal solution is computationally intractable, so heuristics must be used.

The web service is based on a *k*-means clustering algorithm [12]. Students bid for their desired topic, either individually or in teams, by ranking the topics in order of preference. The bidding interface color-codes the topics across a spectrum, with the "hottest" (most in demand) topics colored red, and those that have rarely been requested colored green. Students drag the topics from the list at the left into a rank-ordered list of their preferences on the right. After coalescing students (who are not already on teams) into teams, the algorithm attempts to assign each team a topic from its preferred list. This approach has been used in courses for several years, and has been refined based on student feedback. The Peerlogic web service makes it available to other peer-assessment systems.

Topics

Topic #	Topic name(s)
E1815	Improvements to review grader
E1816	Visualizations for instructors
E1818	Role-based reviewing
E1819	Improve self-review Link peer review & self-review to derive grades
E1820	Review-comment tone-analysis report
M1800	Produce HTTP archive files based on Servo's network behavior
M1801	Implement the OffscreenCanvas API
M1802	Simplify the 2d canvas rendering implementation
M1803	Implement a web page fuzzer to find rendering mismatches

Selections

Topic #	Topic name(s)
E1822	Extend the functionality of badging
E1817	Student-generated questions added to rubric
E1824	Let course staff as well as students do reviews
E1823	Write integration tests for users_controller.rb
E1821	Regularize Expertiza DB schema

Figure 2. Bidding interface for team formation

7. Future work

There are several opportunities for applying natural language processing (NLP) to derive metrics on reviews. It can be used to count the suggestions [13] made in a review, or to count instances

of problems detected in the review. It is also useful to try to measure whether the feedback is localized [14] to refer to a specific portion of the text (which makes it easier for the author to act on), or to detect whether it gives a reason why the requested change should be carried out (which may heighten the author's motivation to make the change).

We can improve the usefulness of the data warehouse by encouraging originators of other peer-assessment systems to contribute data (so far, we have just reached out to the half-dozen systems involved with the Peerlogic project). This will be facilitated by further development of our Peer-Review Markup Language (PRML) [15] and the technology for transforming data into the schema used by the warehouse.

8. Summary

Peer-assessment systems perform the same basic functions and have the same needs. As new ways of improving assessments and visualizing data become available, our web-service approach makes it possible to implement them in one place, yet allow many systems to take advantage of them. The Peerlogic project (www.peerlogic.org) hosts several services, as well as a database of anonymized peer reviews that is available to any peer-assessment researcher.

References

[1] Topping, Keith J. "Peer assessment." *Theory into practice* 48, no. 1 (2009): 20-27.

[2] Tinapple, David, Loren Olson, and John Sadauskas. "CritViz: Web-based software supporting peer critique in large creative classrooms." *Bulletin of the IEEE Technical Committee on Learning Technology* 15, no. 1 (2013): 29.

[3] Palanski, M., D. Babik, and E. Ford. "Mobius SLIP: Anonymous, peer-reviewed student writing." *OBTC 2014 at Vanderbilt University* (2014).

[4] Hart-Davidson, William, Michael McLeod, Christopher Klerkx, and Michael Wojcik. "A method for measuring helpfulness in online peer review." In *Proceedings of the 28th ACM international conference on design of communication*, pp. 115-121. ACM, 2010.

[5] Branham, Cassandra, Joe Moxley, and Val Ross. "My reviewers: participatory design & crowd-sourced usability processes." In *Proceedings of the 33rd Annual International Conference on the Design of Communication*, p. 26. ACM, 2015.

[6] Hamer, John, Kenneth TK Ma, and Hugh HF Kwong. "A method of automatic grade calibration in peer assessment." In *Proceedings of the 7th Australasian Conference on Computing Education-Volume 42*, pp. 67-72. Australian Computer Society, Inc., 2005.

[7] Lauw, Hady W., Ee-Peng Lim, and Ke Wang. "Summarizing review scores of "unequal" reviewers." In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 539-544. Society for Industrial and Applied Mathematics, 2007.

[8] Song, Yang, Zhewei Hu, and Edward F. Gehringer. "Pluggable reputation systems for peer review: A web-service approach." In *Frontiers in Education Conference (FIE), 2015 IEEE*, pp. 1-5. IEEE, 2015.

[9] Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5, no. 1 (2012): 1-167.

[10] Gilbert, CJ Hutto Eric. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*. 2014.

[11] Pramudianto, Ferry, Tarun Chhabra, Edward F. Gehringer, and Christopher Maynard. "Assessing the Quality of Automatic Summarization for Peer Review in Education." In *EDM (Workshops)*. 2016.

[12] Akbar, Shoaib, Edward F. Gehringer, and Zhewei Hu. "Improving formation of student teams: a clustering approach." In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, pp. 147-148. ACM, 2018.

[13] Brun, Caroline, and Caroline Hagège. "Suggestion Mining: Detecting Suggestions for Improvement in Users' Comments." *Research in Computing Science* 70.79.7179 (2013): 5379-62.

[14] Nelson, Melissa M., and Christian D. Schunn. "The nature of feedback: How different types of peer feedback affect writing performance." *Instructional Science* 37.4 (2009): 375-401.

[15] Song, Yang, Ferry Pramudianto, and Edward F. Gehringer. "A markup language for building a data warehouse for educational peer-assessment research." *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2016.