A Method for Measuring Helpfulness in Online Peer Review

William Hart-Davidson
Michigan State University
East Lansing, MI 48824
hartdav2@msu.edu

Michael McLeod
Michigan State University
East Lansing, MI 48824
mcleodm3@msu.edu

Christopher Klerkx
Michigan State University
East Lansing, MI 48824
klerkxch@msu.edu

Michael Wojcik
Michigan State University
East Lansing, MI 48824
wojcikm4@msu.edu

ABSTRACT
This paper describes an original method for evaluating peer review in online systems by calculating the helpfulness of an individual reviewer's response. We focus on the development of specific and machine scoreable indicators for quality in online peer review.

Categories and Subject Descriptors
K.4.3 [**Organizational Impacts**] -- *Computer-supported collaborative work*

General Terms
Algorithms, Management, Measurement, Design, Human Factors, Theory

Keywords
Review, Education, Information Architecture, Assessment, Reputation, Workflow, Learning, Peer Review, Helpfulness

INTRODUCTION
Following the logic of Benkler **[1] in *The Wealth of Networks*,** a culture of massive scale peer production such as we see with Web 2.0 technologies creates a quality problem related to the inherent self-evaluation bias of content producers. To ensure quality in web scale peer production, we need something that approaches web scale peer review. Logistically, this is a difficult problem, and so we have seen the rise of machine-mediated systems for coordinating peer response and review such as Amazon.com's recommender system or Slashdot's peer review driven posting system**.**

While these systems address the quality of the product in peer-production - they make the content better - they do not do much to reveal, assess, or facilitate the development of review activity itself. In many systems and contexts, it is desirable to know who the best reviewers are and also, in educational settings, what kinds of instructional interventions may be useful for helping students become better reviewers.

Our review method takes input from a structured but flexible review workflow and stores the artifacts generated in the review process (e.g. comments, suggestions for revisions) along with other user-supplied descriptive and evaluative data that correspond with review metrics (such as whether or not a comment addressed a specific review criterion). The core method is designed as a web service that can receive this input from a variety of different sources and production environments, performing the analytics needed to calculate reviewer helpfulness and visualizations meant to offer both formative and summative feedback of reviewers' performance. Our method understands a review to be a group activity consisting of reviewers, review targets (documents), and criteria, all under the direction of a review coordinator.

There are a number of specific applications for this method, but we will focus on one application called "Eli" that has been developed for online peer review in classroom settings. Eli gives feedback to both teachers and students about reviewers' helpfulness in both a single review (a helpfulness score) and over time (a helpfulness index) so that students' ability to do reviews can be meaningfully evaluated.

SIGDOC 2010, October27-29, S.Carlos, SP, Brazil.ACM 978-1-4503-0403-0...$5.00..

1. Peer Production and Peer Review as Staples in an Information Economy
A distinctive feature of an information society, in general, and of Web 2.0 technologies as an outgrowth of an information society is the sheer volume of content produced by users in the course of doing everyday activities. Without being dismissive of the visual and interactive elements that users are producing, we might simply say that there is a lot of writing going on, by necessity, in the networked world. And by writing, we mean that process of constructing readable, usable, and valuable texts.

Following Benkler [1], we may apply an economic lens and view this writing, in the context of an information economy, as the equivalent of production work in a manufacturing economy, and just as in a manufacturing economy, we can see a need for processes of quality control:

> "To succeed, therefore, peer-production systems must also incorporate mechanisms for smoothing out incorrect self-assessments - as peer review does in traditional academic research or in the major sites like Wikipedia or Slashdot, or as redundancy and statistical averaging do in the case of NASA clickworkers. The prevalence of misperceptions that individual contributors have about their own ability and the cost of eliminating such errors will be part of the transaction costs associated with this form of organization. They parallel quality control problems faced by firms and markets." (Benkler, Ch. 4. p.15)

We can see that the value of a text, as Benkler argues, may be lessened when its quality is ensured only by self-review, an inherently biased proposition. We can see another problem in this formulation as well when we think about the reviewers themselves: how do good reviewers learn to be good reviewers? It

is this problem that we have focused on in developing a method for evaluating review activity so as to provide both summative and formative feedback to those who supervise and engage in review work.

1.1 Teaching & Learning Review in Formal & Informal Learning Environments

Though there have been some book-length studies of writing review in both academic and workplace settings, what these studies reveal is that review is something that is rarely learned in a formal way (i.e. as the subject of a course or an assignment in a course, with a set of pre-defined learning objectives and an assessment to determine how well a student meets these objectives after some intervention)[2,3]. Rather, review is practiced in the service of learning to become a better writer (e.g. peer review sessions in writing classrooms [2]) and is learned, in workplace settings, via a process of enculturation and, in the best of circumstances, via a mentor-mentee relationship (see Swarts [3]). This is not to say, though, that review activity is itself rare. On the contrary, most students in first-year and advanced writing courses conduct peer-review as a routine part of each writing assignment [4]; peer review, moreover, is the gold standard for judging quality in a wide range of professional and research-oriented contexts, comprising what Fitzpatrick calls the sine qua non of scholarly publishing [5].

2. Challenges of Measuring Review Quality

Just as review activity has not often been the focus of formal efforts to train or evaluate reviewers, it is correspondingly non-focal in most systems that are designed for writing. We would even suggest that review activity itself is not always modeled in review coordination systems in such a manner as to permit it to be evaluated. There are some promising features built into systems meant to build trust in online environments - reputation systems - that we will highlight futher below. But reputations systems take review activity as input and they do not seek to offer an assessment of either any one review or of an individual reviewers activity, specifically, over time. Below, we discuss each of these issues as research and design challenges we faced in developing a generalized model for evaluating writing review in online systems.

2.1 Challenge: Inadequate workflow & data model for capturing review activity in writing software

There are any number of ways current writing technologies might be leveraged to get feedback on an existing text. Some writing software include features specifically to facilitate "review" such as adding inline comments or allowing a collaborator to make changes that another author must approve. But in nearly all cases, writing systems value review only as a means of evolving the text to a "better" version, to a “draft 2.0,” an approach that makes it nearly impossible for these technologies to value review itself as a teachable and learnable activity.

Microsoft Word is a perfect example of writing software with dedicated review features. Word has a sophisticated “track changes” feature that allows a reviewer to directly alter the text and leave a report of that change for the author, who has the option to accept or reject those changes. Both Word and Adobe Acrobat have a set of commenting tools that allow reviewers to highlight selections of text and leave annotations, as well as digital versions of analogue activities like highlighting and “sticky notes” and drawing; Acrobat even has a set of features that allow reviewers access to proofreader marks and symbols. These features are incredibly powerful and make it easy to leave comments or suggestions for writers.

Where these tools fall short, however, is in capturing and making available for assessment the outcomes of a review. In Word, for example, when an author chooses to accept a change recommended by a reviewer, that change is hard-coded into the document and all traces of the suggestion are lost. The person who made the suggestion is not notified that their suggestion was accepted. In Acrobat, reviewers do not interact with the text directly (the software makes notes and edits over the top of the text, since it's text editing capabilities are limited) so the writer reviews comments and suggestions in Acrobat before returning to their text editor to make revisions, leaving no trace of the reviewer's work in the next version. This not only makes it impossible for reviewers to know when they have made helpful suggestions or comments, but it also makes it nearly impossible for instructors to assess the review work of their students.

There are two ways in which review as it is implemented in writing technologies like Word conflicts with review as it exists in practice. The first conflict is in the typical workflow of a review. Word does not see review as a distinct part of a workflow in which any number of other activities might be occurring. To put this another way, Word doesn't understand "a review" to be an event, a kind of social transaction as we understand it to be in the real world. Word sees review as a solitary experience done in relative isolation by a single person. A review is not something a group does, as far as Word knows. The second conflict lies in the way that the artifacts created during a review: comments, suggested changes, etc. are appended to a document. This means that there is no way for a reviewers comments on a document in Word to be seen as part of a larger activity in which the reviewer is engaged. There is also no way select a reviewer and see multiple comments on multiple documents.

2.2 Challenge: Lack of quality indicators & measures in review coordination systems

The two conflicts between review in the world and review in writing software are addressed by systems built to facilitated the coordination of reviews. There are many of these systems used by academic conference organizers, by publishers, and by government and other grant-making agencies to facilitate the submission, evaluation, publication and/or ranking of submissions for awards. The National Science Foundation's FastLane system is one such example, as is the JEMS system used to coordinate reviews for this conference.

While our project has included some design work meant to assist review coordination, we have not sought to replace any other system that does this kind of work. Rather, we have sought to enhance these types of systems by providing feedback for reviewers and review coordinators that these systems do not, to our knowledge, typically provide. We have worked on ways to provide feedback to reviewers, for example, on how helpful their review has been for the writer. We also have worked on ways to provide feedback to review coordinators about who their most

helpful reviewers for a given round of review were. In order to provide this feedback, we have created a method that requires input in conjunction with a model of the review process, associating quality indicators with key moments in a review workflow. Our method can be used in conjunction with any existing review coordination system that provides appropriate inputs (e.g. via an API). We discuss the review model in more detail below.

2.3 Challenge: Building a reviewer's profile

Another category of online systems that deals with both the scale and economics of peer production is web reputation systems. As Farmer & Glass [6] argue, web reputation systems are designed to help overcome problems associated with missing interactions in online social settings that help to build trust in similar processes face-to-face or in traditional organizations. Online reputation systems are often used in peer production systems to allow users to distinguish good content from bad, and good content providers from bad ones as well. Reputation systems take some review information as input.

Before we go any further in comparing our own work in evaluating writing review to that of online reputation systems, we want to make an important distinction between reviews of texts as done, for instance, during academic peer review and the kinds of product reviews or book reviews one might see on a website like Amazon.com. These are reviews as texts. That is, product reviews are the kinds of peer production that requires writing review, in some measure, to ensure the quality of. Amazon.com has a reputation system in place that helps to sort out helpful product reviews from not so helpful ones; over time, as you submit good product reviews, you acquire status in their system such that your reviews may be more readily found by users. Amazon's system famously involves a very simple feedback system in the form of a question posted to users: was this review helpful to you? An answer to that question amounts, then, to a review of the product review, the results of which add value to the product review and add to the reputation of the writer of that review.

Our work on writing review has focused on the evaluation of a text by a group of reviewers with an eye toward measuring which of the reviewers gave the most helpful feedback. And so while we have learned from online reputation systems, the main takeaway for us has been the way that actions taken by reviewers (or by writers, for that matter) accrue over time and stay affiliated with a person in some kind of online profile. What we have developed is a way for all of that activity to add up to a measure of one aspect of a reviewers' performance that we think helps both reviewers and those who supervise reviewers: helpfulness.

3. Helpfulness: A Proposed Measure of the Quality of Online Review

Helpfulness is one way we propose to measure the quality of an online review where a reviewer gives feedback to a writer under the direction of a review coordinator who sets some criteria. We do not mean for helpfulness to be a comprehensive measure of review quality. It is simply a desirable feature of both a review and a reviewer: both should be helpful. What makes for a helpful review? What we know from both reviewing literature on review [2] and from our own experience as teachers of writing and as recipients of reviewer feedback is that a helpful review describes where the writer has made specific moves in a text to achieve rhetorical aims (as reflected in review criteria), makes accurate and fair evaluative statements about the features of a text, and offers specific, actionable advice about how to improve the text. A helpful reviewer is someone whose reviews consistently display these features.

With such a complex definition of review, it is not enough to simply ask a writer: was this review helpful? The writers answer may be one data point to consider, but it falls short of providing enough quality evidence to say whether a reviewer met the high standard of providing helpful feedback described above. To do that, we combine human-supplied judgments and performance-based data gathered at various points in the review process in order to determine when a review meets the description above.

3.1 A Description of Review

As we mentioned earlier, our definition of review differs from an approach that understands reviews as consisting primarily of metadata attached to a single digital file or object. Reviews exist along a spectrum from strictly criterion-referenced in which the user must respond to specific, pre-determined standards, evaluating the documents according to these, to open reviews where reviewers may be given little or no structure for providing feedback.

At their most basic, reviews consist of an author asking a reviewer for feedback on a single document. More complicated reviews can involve a review coordinator gathering multiple documents and specifying criteria by which multiple reviewers will evaluate them. In our review evaluation method, we attempt to match review processes in specific systems with a generic model of review that is detailed in section 3.2.2.

3.2. Generalized Steps in the Workflow

Create Review

1. an author or review coordinator initiates a review by specifying a document or documents(review "targets")

2. next, a reviewer or reviewers are identified and a prompt is created to ask the reviewer(s) for feedback

3. optionally, a review coordinator may create specific criteria to guide reviewers

Conduct Review

4. reviewers are notified that an author or coordinator has solicited their feedback

5. reviewer examines the review prompt and documents and decides whether to accept the review or not

6. reviewer views all documents & criteria, makes comments, adds ratings and or selects relevant chunks of document(s) in order to link them to criteria and/or provide specific feedback

View Results

7. reviewer completes and submits review

8. author and/or coordinator is notified of the completed review

9. author and/or review coordinator can view and rate feedback submitted by reviewer(s)

10. Once review is complete, authors and review coordinators may access aggregated review information; review coordinators may also permit reviewers to see review results

3.3 Sample Performance-Based Indicators & System Events

This generalized workflow, regardless of the context in which it might be embedded, is built around a wide range of activities in which users are engaged. Some of these activities result in artifacts of the review process (comments or suggestions, ratings, responses to criteria, etc) while others can be observed and recorded by the infrastructure of the review system. Table 1 lists of observable data, either user-generated or system-observed, that could be utilized in a system that values review and review behavior.

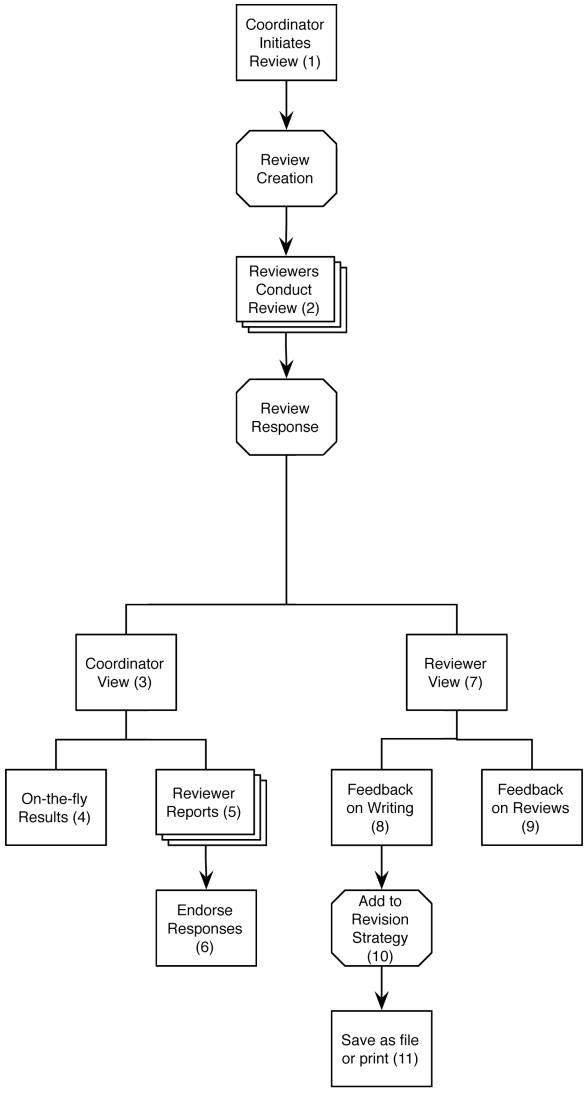


Table 1: Performance-based Indicators by Role with Corresponding System Events

Factor	User Roles	Event trigger
Document changes from version A to version B	Writer	Difference calculation
Document changes in the region of a comment	Writer	Difference + location
Document improves (gets a better rating) in relation to a criterion referenced in an earlier review	Writer	Difference + OnSubmit event
Comment by another reviewer appears in a similar area AND connected to similar criteria	Reviewers	Difference + OnSubmit
Document changes reflects a revision suggested explicitly by the reviewer	Writer, reviewer	Difference w/ parsing
Document changes reflect exact language suggested by the reviewer	Writer, reviewer	Difference w/ keyword matching
Reviewer accepts, rejects, or ignores a direct review request	Reviewer	OnClick event
Reviewer opts in to an open review request	Reviewer	OnClick event
Reviewer completes a review	Reviewer	OnSubmit or AJAX event
Reviewer provides a revised passage	Reviewer	revision tag in comment
Writer responds to a review	Writer	OnSubmit or AJAX
Writer moves comment to revision list	Writer	OnClick event
Writer inserts language from comment into draft (accept a proposed revision)	Writer	revision tagged content in body text
Review coordinator endorses a reviewer's comment	Coordinator	scaled response

Again, these are general types of behavior that can be observed and recorded inside a review system. Some of them, such as "writer responds to a review," are relatively simple observations by the system and can be recorded as discreet events. Others,

like "Document changes from version A to version B," involve much more complex difference comparison, evaluating the two versions of a text and comparing against the comments the writer received on the first version.

As individual observations, they are not particularly valuable; inside a system that assigns relative value and proportionate weights to different activities, however, they can be utilized to make observations about the review practices of users. In the next section we will detail a system designed to teach writing review and utilizing this method.

4. Eli: Designing a Helpfulness Algorithm for a Classroom-based Peer Review System

Building on our generalized framework for review, Eli is our response to a problem we experience regularly as writing teachers. Specifically, we as teachers value review, but the demands of collecting, assessing, and tracking the helpfuness of our student's work as reviewers, in addition to the already daunting task of responding to student work as writers, are far too overwhelming. Eli is intended to be a system that not only makes the peer review process visible and assessable but also to help teach students to be better reviewers.

4.1 Helpfulness Algorithm

Our approach to building a system to meet these demands was to start small and scale upward over time. Because of our limited development resources, our approach was to identify machine-observable behaviors that didn't involve complex versioning or difference comparison to calculate and to build a relatively simple but useful metric around those. With that development model in mind, we identified the following behaviors that we felt best represent the helpfulness of reviewers:

- reviewer responses to instructor-specified criteria (whether or not they responded)
- rating of suggestions and comments by the writer of texts being reviewed
- number of suggestions and comments the writer adds to their revision strategy
- placement of suggestions in writer's revision strategy (ordinal positioning)
- instructor endorsement of criteria responses
- instructor endorsement of comments and suggestions
- use of comments or suggestions by a writer in a new version of a text (as specified by the writer)

These factors are all either binaries (yes/no or present/absent) or simple calculations (ratings of suggestions) which fit nicely into our limited development framework. Given greater resources we might elect for more complex methods, but in the absence of any system at all we felt these metrics would still yield powerful data and transform our pedagogy.

The next step, then, was to assign relative weights to each of the observed data points. This was necessary because all data points are not necessarily created equal; the fact that a user responds to all criteria, for example, doesn't imply helpfulness nearly as much as if the instructor endorsed one of those responses. To that end, we developed a weighting that we felt valued the data points in a way that more accurately reflected their individual degree of helpfulness:

Table 2: Helpfulness Factors with Associated Weights

Data Points	Weight
reviewer responses to instructor-specified criteria	1
rating of suggestions and comments by the writer	2
number of suggestions and comments added to revision strategy	2
placement of suggestion on writer's revision strategy	3
instructor endorsement of criteria responses	4
instructor endorsement of comments and suggestions	4
use of comments or suggestions in a new version of the text	5

In our very initial testing of the algorithm with these weights applied, the algorithm resulted in a number that, while accurately describing the relative helpfulness of each student, did not yield a number that we as instructors felt would be beneficial for students. For example, the algorithm assessed a student who performed poorly on a review as having a "12" while a student who performed exceptionally well only received a "68." We applied a corrective multiplier to the result of the algorithm to bring the result into line with the 100% scale that students and instructors are more familiar with.

4.3 Testing & Tuning the Algorithm with Sample Review Values

Once the algorithm was established we needed a way to test our assumptions not only about the relative weights of each metric but also the general findings of the algorithm - we needed to see if its computations would match up with how instructors would assess the review work of students. To accomplish that, we fabricated a set of review data that we believed would exemplify one of the harder distinctions the algorithm would need to make. Specifically, we created a review with the following context:

- Students divided into groups of 5 (each writer responds to 4 classmates)
- The instructor asked for responses to 3 criteria
- Writers can rate each suggestion made by their classmates on a 5-point scale
- The writer in this scenario created a Revision Strategy using 6 classmate comments

Table 3 provides additional detail to flesh out the values used in this test.

Table 3: Fabricated Student Reviewer Data Created to Test Weighting of Factors in Helpfulness Score Calculation

Observable Data Point	Student #1	Student #2	Student #3	Student #4

# of responses to instructor-specified criteria (*out of 3 possible criteria in this review*)	3	2	3	1
Criteria responses ratified by instructor	3	2	2	1
Number of suggestions made	5	3	2	1
Rating of suggestions (*out of 5 points per suggestion*)	22 of 25	10 of 15	10 of 10	1 of 5
# of suggestions utilized on Revision Strategy	3 of 5 made	1 of 3 made	2 of 2 made	0 of 1 made
Placement of suggestion on writer's strategy (*of the 6 total comments on the strategy*)	#1, #3, #4	#6	#2, #5	-
Suggestions utilized in revision	2	0	1	0

There are two clear outliers in this trial data set. Student #4 is clearly the least helpful reviewer; the student did not respond to all of the instructor's criteria and only made one suggestion which was rated poorly by the writer, was not ratified by the instructor and was not utilized in the revision strategy. Student #2 did better than Student #4 but was likewise not very helpful; this student also did not respond to all of the criteria, made 3 comments, only one of which was ratified by the instructor and used on the writer's revision strategy, but it was not utilized in the final draft.

The challenge for the algorithm was in differentiating between Student #1 and Student #3, both of whom represent what we believed to be helpful reviews. When we presented this data to a group of writing instructors and asked them to determine the most helpful reviewer, after lengthy conversation, the nearly unanimous verdict was that Student #1 was the most helpful. While the comments made by Student #3 were better rated on average, Student #1 made more comments that were less highly rated but were a higher priority on the writer's revision strategy and contributed more to the next version of the document. While both students were helpful, in terms of contributions toward a new text, the human raters thought Student #1 was more helpful. Thankfully, the algorithm agreed - each of the data points and their weights had been chosen carefully enough that the results of the data crunching resulted in an assessment that matched the vast majority of our human raters.

5. Displaying Formative Feedback to Teachers & Students in Order to Foster Learning of Review

One of the many affordances of a system designed to track and respond to discreet writing and review events is the extraordinary potential for visualizing writing behavior that have previously been impossible. With that in mind, in this section we will detail some of our early work developing graphic representation of review behavior inside Eli.

One of the most straightforward approaches is to utilize the raw numbers. As designers and information architects we were always keenly aware of the possibilities, particularly the most complex and challenging visualizations; as teachers, however, we were perhaps most excited for a simple numerical report of review data that had never before existed (fig. 1.).

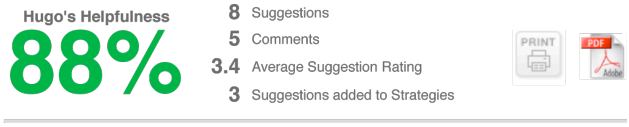


Figure 1: Helpfulness Numerical Display

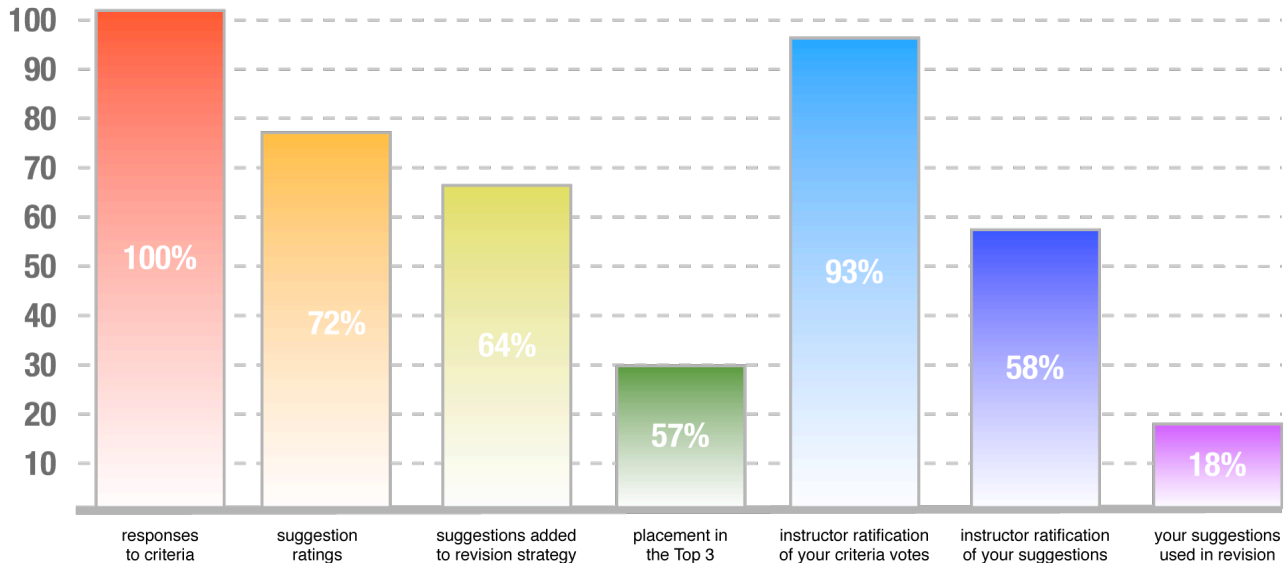
In this simple illustration, we make visible raw data that all but the most ambitious teachers have never seen - statistics about individual student writing behavior. Here, with a single glance, we can get a heads up of the work our student do as reviewers, as well as the perceived helpfulness of that work. We know the quantity of this student's work as well a quick insight into the perceived usefulness by the writers to whom this student responded.

A far more complex visualization of a student's review data is what we call the "Helpfulness Index" (fig 2). While students receive a Helpfulness Score after each review (each review is scored individually), their responses over time are compiled into their Helpfulness Index. Like the Helpfulness Score, the Index is based on a computation of each of the observable data points in a review, but compiled from the student's entire review history. Each individual data point becomes a statistic and, based on a student's performance in one of those areas, the instructor can offer guided feedback on that specific area of a user's review work. Also, because these scores are computed by the system, the system could be pre-loaded with sets of formative feedback that would guide a user on how to improve that aspect of their score. For example, in the bar chart here used to represent a student's review history, the influence of their suggestions on the writer's revised texts is quite low. The system, knowing this, could guide the user to sets of

instructional material or tips showing them suggestions by other reviewers that have been used in revisions.

Another potentially powerful representation of Helpfulness data is to plot the progression of a student over time (fig 3).

Figure 2: Helpfulness Index



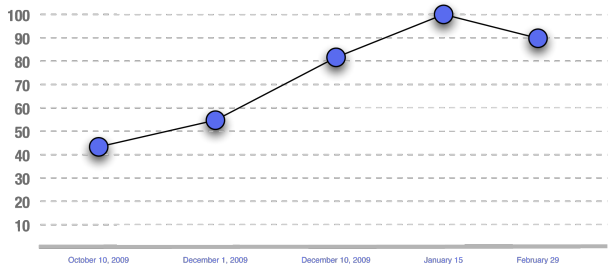


Figure 3: Helpfulness Scores Over Time

Even if students are assigned only five reviews over the course of a school year or a semester, the five Helpfulness Scores they receive from those reviews can still create a compelling representation of student learning. In the example below, the perceived helpfulness of a student's review work can be represented in a sparkline diagram that clearly shows that the student learned how to respond to classmates with more helpful feedback. For specifics, this student's teacher could drill down into individual reviews and see which factors improved over time.

Again, because these numbers are simple machine observations and computations, the exact same representations could be used to show the progress of all the students in a course. With a little more computational power, these statistics could be generated for an entire writing program. If our algorithm is capable of matching instructor's perceptions about review quality, the results of that algorithm over time should be an accurate representation of student learning.

6. Future Work

Eli is currently in development and will be put in front of students in beta form in late summer 2010. Once students have used the system and we have a robust data set we can begin larger-scale fine tuning of the Helpfulness algorithm to better align it with instructor expectations. From there, depending on the success of the beta, we could begin scaling up and implementing some of the machine-derived data (particularly difference calculations between two versions of a text).

1. REFERENCES

[1] Benkler. Y. (2007). *The wealth of networks: How social production transforms markets and freedom*. New Haven, CT: Yale UP.

[2] Kastman-Breuch, L. (2004). *Virtual peer review: teaching and learning about writing in online environments.* Albany: SUNY Press.

[3] Swarts, J. (2008). *Together with technology: writing review, enculturation, and technological mediation*. Amityville, NY: Baywood.

[4] Fitzpatrick, K. "From the Crisis to the Commons." In "In Focus: The Crisis in Publishing." *Cinema Journal* 44.3 (Spring 2005): 92-95.

[5] Fitzpatrick, K. (2009). Planned obsolescence: Publishing, technology, and the future of the academy. NYU Press. http://mediacommons.futureofthebook.org/mcpress/plannedobsolescence/one/

[6] Farmer, F.R. & Glass, B. (2010). *Building web reputation systems*. Cambridge, MA: O'Reilly.