Natural Language Processing techniques for researching and improving peer feedback

Wenting Xiong, Diane Litman & Christian Schunn

University of Pittsburgh | United States

Abstract: Peer review has been viewed as a promising solution for improving students' writing, which still remains a great challenge for educators. However, one core problem with peer review of writing is that potentially useful feedback from peers is not always presented in ways that lead to revision. Our prior investigations found that whether students implement feedback is significantly correlated with two feedback features: localization information and concrete solutions. But focusing on feedback features is time-intensive for researchers and instructors. We apply data mining and Natural Language Processing techniques to automatically code reviews for these feedback features. Our results show that it is feasible to provide intelligent support to peer review systems to automatically assess students' reviewing performance with respect to problem localization and solution. We also show that similar research conclusions about helpfulness perceptions of feedback across students and different expert types can be drawn from automatically coded data and from hand-coded data.

Keywords: peer review, artificial intelligence, feedback features, coding

journal of
WRITING RESEARCH

Xiong, W., Litman D., & Schunn, C. (2012). Natural Language Processing techniques for researching and improving peer feedback. *Journal of Writing Research, 4*(2), 155-176.

Contact: Wenting Xiong, University of Pittsburgh, Department of Computer Science, 210 S. Bouquet St., Pittsburgh, PA, 15260 | USA – wex12@cs.pitt.edu

Poor achievement in high school writing has been a salient problem in the US for a number of years and some large-scale efforts have been successfully launched, such as the National Writing Project. However, overall performance remains poor. What is the problem? A number of factors have been discussed, but a commonly mentioned and obvious bottleneck in writing instruction is the large amount of resources it requires. Writing improvement, as in all other areas, greatly benefits from regular feedback on performance, and feedback on writing is incredibly resource intensive. As a result of increasing workload in all teaching settings, teachers of content areas (e.g., social studies or science) provide limited feedback on the writing per se, do not require multiple drafts (a key feature of improving writing skills), or avoid writing assignments entirely. As class sizes increase, English/communications teachers also limit the workload that arises from the demands of providing feedback on writing. For example, typically, high school students have only one or two opportunities a semester to practice evidence-based writing or to write papers of five or more paragraphs (Kiuhara, Graham, & Hawken, 2009).

One path to improvement involves technology that provides students direct feedback on their writing, from complex grammar checkers to more sophisticated computational linguistics methods that can identify argument structure problems or other content problems (Attaliand & Burstein, 2006; Graesser & McNamara, 2012; McNamara, Crossley, & McCarthy, 2010; McNamara, Louwerse, McCarthy, & Graesser, 2010) to wizards that step students through a more effective writing process (Proske, Narciss, & McNamara, 2010). The generalizability of these approaches across settings and writing genres is yet to be established (Ericsson & Haswell, 2006), but it is likely that these approaches will be part of the solution.

Another possible path involves peer review. In general, peer review is consistent with learning theories that promote active learning, including collaborative and cooperative learning, provision of feedback, repeated opportunities to practice, and relevant domain-specific tasks (Ashbaugh, Johnstone, & Warfield, 2002; Cornelius-White, 2007; Palincsar & Brown, 1984; van den Berg, Admiraal, & Pilot, 2006; Vygotsky, 1978). Further, peer review of writing is a commonly recommended technique to include in good writing instruction (Graham & Perin, 2007; Topping, 1998, 2008). Conceptually, peer review makes more salient the communicative and rhetorical aspects of writing—students receive feedback from audiences who do not already know the content being conveyed (Cohen & Riel, 1989) and experience firsthand the consequences of poor writing strategies. Peer review can also improve high school student attitudes towards writing (Katstra, Tollefson, & Gilbert, 1987). In addition, as a source of feedback, a number of studies have found that feedback from a group of peers can be at least as useful as that of teachers (Cho & MacArthur, 2010; Patchan, Charney, & Schunn, 2009), especially when good rubrics and incentives for reviewing are included. Surprising to many, even weaker writers can provide feedback that is useful to stronger writers (Nelson, Melot, Stevens, & Schunn, 2008; Patchan & Schunn, 2010). Further, several studies have also found that the process of providing feedback leads to improvements in the feedback-providers' own writing (Sadler & Good, 2006), especially when the students provide constructive feedback (Wooley, Was, Schunn, & Dalton, 2008), and put effort into the process (Cho & Schunn, 2010). One experiment with second language learners found that writing improved more from regularly providing feedback to peers than from regularly receiving feedback from peers (Lundstrom & Baker, 2009).

However, there are a number of challenges to effective and broad implementation of peer review. Often there are logistical challenges: distributing documents to multiple reviewers and multiple reviews back to authors; insuring completion of so many reviewing tasks; and monitoring quality of feedback. However, just as editors of journals and chairs of conferences have discovered, the advent of the web, web-forms, and simple databases has made it relatively easy to address these logistical challenges with a web-based peer review system. Many peer-review systems specifically for writing instruction have been created that greatly reduce the logistical challenges and allow for peer review to easily proceed even with hundreds of students in a class, such as PeerMark in turnitin.com, SWoRD (Cho & Schunn, 2007), and Calibrated Peer Review (Chapman & Fiore, 2000; Prichard, 2005).

1. Problems for Research and Practice of Peer Review of Writing

While web-based peer review of writing holds much promise for writing instruction (Goldin, Ashley, & Schunn, 2012) and has produced some interesting learning outcomes, it is not optimal in its current form. We focus in this paper on one core problem in peer review of writing that relates to the nature of the feedback that authors receive. In particular, potentially useful feedback from others is not always presented in ways that lead to revisions. Nelson and Schunn (2009) systematically examined a large dataset of peer reviews and coded changes across drafts for whether the provided feedback had resulted in a revision that attempted to address the changes. Then the large quantity of feedback was systematically hand-coded for many different features that could influence implementation. From statistical analyses, two features were found to predict implementation: providing localization information for the problem (through explicit page/paragraph numbers, paraphrasing, direct quotations, or other location details) and providing a concrete solution (i.e., a way to address some identified problem rather than just noting what that problem was). In other words, feedback that included this information was more likely to be implemented by the author than feedback that did not include the information. Both of these effects on implementation appeared to be at least partially mediated by influencing whether the author understood the problem being described.

From this prior line of work, we wish to point out two types of problems, one for research and one for practice. On the research side, this kind of research is very time-consuming. The positive AND negative feature of peer review systems is that they easily produce large quantities of data for analysis: positive because that enables statistical analyses to tease apart related factors, confounding variables, and moderating variables; negative because there is a mountain of feedback that must be hand-coded. If technology could be introduced to automate or partially automate the coding of feedback into types, research would be greatly accelerated to address many looming and exciting research questions beyond the basic point established by Nelson and Schunn: Does the type of feedback generated or helpfulness of the feedback vary systematically by genre, prior experience with peer review, incentives to do peer review, cultural norms regarding directness of critical comments, familiarity with the topic of the paper being reviewed, …?

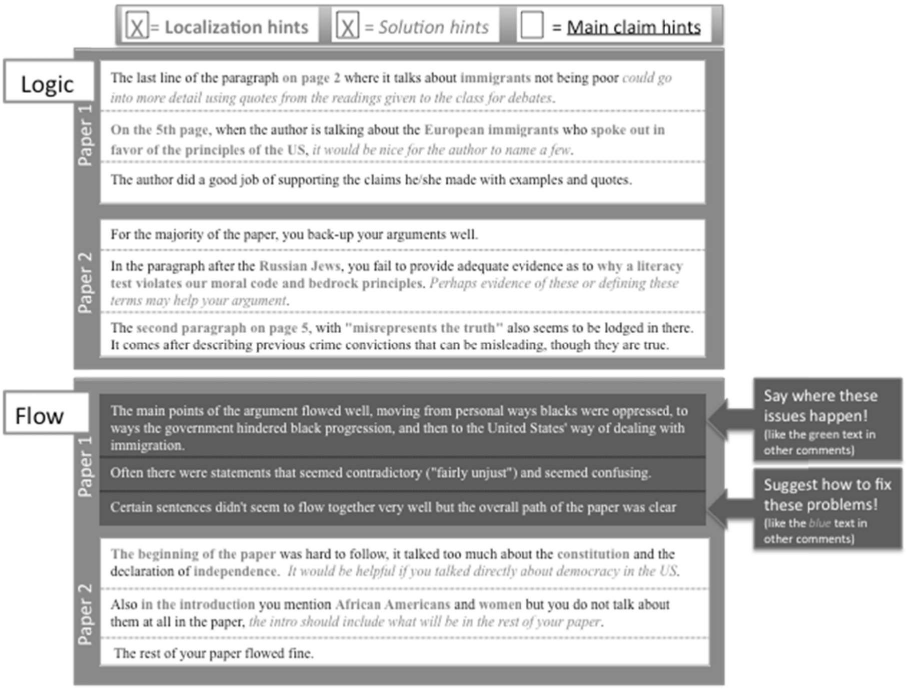
On the practice side, knowing what kind of feedback is more helpful to authors should lead to simple effective interventions, but it turns out not to be so simple. We have conducted experiments in which we provide direct and detailed instructions to students, including interface prompts, to always include localization information and solutions. However, the impact was minimal, with students ignoring instructions or providing only partial localization information or non-solutions (e.g., "fix that problem"), perhaps because they misunderstand what is being requested or perhaps because they are having trouble monitoring compliance with these instructions while also dealing with the complexities of diagnosing issues in papers and composing comments. In either case, we believe students required more intense training in which they receive explicit, detailed feedback about these elements of their feedback if they are to make reliable and systematic changes on those elements. We doubt there are enough people resources (peers, instructors, teaching assistants) to provide this kind of training feedback on feedback. Instead, we suggest technology could be used to automate that kind of training feedback.

2. A Possible Solution: Automatically Processing Peer Feedback

As a solution to both the problem for researchers studying peer review and for instructors wishing to improve the helpfulness of peer reviews, we suggest that computational linguistics techniques can be used to automatically detect the presence or absence of key feedback features (e.g., the presence of localization information, or the presence of explicit solutions to problems). For researchers, they could then use this technique to quickly code large volumes of peer feedback data by computer rather than by hand. In a later section of this paper (Study 3), we compare analyses of a peer feedback corpus using computer-coded feedback and hand-coded feedback to see whether similar conclusions would be found.

For instructional purposes, we propose to automatically process peer reviews *as they are submitted* to explicitly prompt reviewers for localization and explicit solutions when their reviews are missing those features, as well as to highlight aspects of their reviews that do include these features. That is, reviewers would submit their review comments online. A server (likely the one hosting the peer review website) would apply computational linguistics techniques to determine whether localization information and explicit solutions were generally included in the comments. Reviewers not providing a sufficient number of localization details or explicit solutions would be prompted to revise their reviews. Examples from their own reviews in which localization details or explicit solutions are included could be highlighted, so the students would have direct, very concrete feedback on these aspects of their overall feedback approach.

Figure 1 presents a prototype of an interface of this type. Note that successful examples of localization and solutions from the reviewers' own review are highlighted as well as indicating where localization and solution information might be absent. We believe that this kind of immediate feedback is more likely to produce changes in reviewing behavior than more general prompts to be specific and provide constructive comments. Also note that these hints are suggestions rather than issues requiring repair, and that the hints can be turned on and off; this approach to feedback is less problematic when feedback has some errors in it as computer-based feedback on writing is always likely to have.



The last line of the paragraph on page 2 where it talks about immigrants not being poor *could go into more detail using quotes from the readings given to the class for debates.*

On the 5th page, when the author is talking about the European immigrants who spoke out in favor of the principles of the US, *it would be nice for the author to name a few.*

The author did a good job of supporting the claims he/she made with examples and quotes.

For the majority of the paper, you back-up your arguments well.

In the paragraph after the Russian Jews, you fail to provide adequate evidence as to why a literacy test violates our moral code and bedrock principles. *Perhaps evidence of these or defining these terms may help your argument.*

The second paragraph on page 5, with "misrepresents the truth" also seems to be lodged in there. It comes after describing previous crime convictions that can be misleading, though they are true.

The main points of the argument flowed well, moving from personal ways blacks were oppressed, to ways the government hindered black progression, and then to the United States' way of dealing with immigration.

Often there were statements that seemed contradictory ("fairly unjust") and seemed confusing.

Certain sentences didn't seem to flow together very well but the overall path of the paper was clear

Say where these issues happen!
(like the green text in other comments)
Suggest how to fix these problems!
(like the *blue* text in other comments)
Paper 2

The beginning of the paper was hard to follow, it talked too much about the constitution and the declaration of independence. *It would be helpful if you talked directly about democracy in the US.*

Also in the introduction you mention African Americans and women but you do not talk about them at all in the paper, *the intro should include what will be in the rest of your paper.*

The rest of your paper flowed fine.

Figure 1: Prototype of interface for feedback to reviewer pre-review submission regarding the inclusion of localization and solution features in their comments. Localization hints are in bold text. Solution hints are in italics text. Feedback with either no localization hints or no solution hints are flagged in dark boxes with call-out boxes to the right. Checkboxes at the top of the screen turn hints on and off.

2.1 The Technical Challenge

Before implementing this solution to explore benefits of automatic coding for researchers or effects of immediate feedback on comments on learners, there is a technical challenge to solve: is it even possible to automatically detect the presence/absence of localization and solutions in peer feedback free text comments, or at least at levels of reliability that are useful for research or for instruction? In this paper, as in Leijen and Leontjeva (2012) we report early attempts that show promising results. This work builds on standard techniques from a branch of Artificial Intelligence called Natural Language Processing (NLP).

At a high level, our approach for detecting localization or explicit solutions involves a three-step process. First, we build a domain lexicon using student papers: what are the students writing about? The presence or absence of domain-specific words can be a clue to whether the feedback is about generic constructs or likely specific to ideas raised in the paper. Each paper topic raises different ideas, and so there needs to be a simple method for generating that list. We use the papers themselves in creating this list, looking for both common words in isolation (called unigrams) as well as common pairings of words (called bigrams).

The next step is counting up basic features in each piece of feedback: how many domain words, how many modals (should, could, would), how many negations, how many total words, how large is the overlap (in words) between the comment and something in the paper it is commenting about, … . The methods section will present the full set of features that were detected. Here we simply mention this basic step in Natural Language Processing. It is no more complex than the search/replace function in MSWord, and we expect this step to be highly reliable.

The last step involves a logic model that uses the basic features to classify particular pieces of feedback as being of one type or another (e.g., having localization information or not; having explicit solution information or not). The logic models can take a variety of different forms; we select whichever form is most accurate for the given context. In this paper, we will use decision trees and regression models. Regardless of the form, the logic models can be quite complex, potentially involving elaborate combinations of many different features. As a fictitious example, a decision tree rule might be: if the feedback has more than 30 words AND the feedback involves more than 4 domain bigrams AND the feedback involves the word "should" then classify it as having an explicit solution. There would be many such rules that together classify any particular piece of feedback. Note that this model is not really based on logic *per se*, because it is not logically necessary that every piece of feedback with those characteristics has an explicit solution. Rather, this model is based on empirically observed relationships: it just happens that feedback with those characteristics *tends* to have an explicit solution. This is the real technical challenge: to find a combination of simple features that can account for complex, abstract concepts like 'having localization information' or 'having solution information' as expressed in free text peer feedback.

The challenge has two parts: 1) does there exist any model that can reliably classify most of the feedback appropriately using combinations of simple features detectable by computer algorithms?; and 2) can we figure out what that model is, from a nearly infinite space of possible models? We cannot answer the first question definitively in the absence of answering the second question: we may have a poorly performing current model but we might not yet have found a good possible model that does exist. In general, the space of possible models can be searched to make suggestions about whether a good model likely exists. We use standard techniques from a field called Machine Learning, which has highly effective methods for developing classification models from a given training corpus by intelligent searches through the space of possible models. These Machine Learning techniques present diagnostic information about the goodness of fit of the model to the given data. Further, the procedures often apply the training procedure to part of the corpus of data and then test the resulting models on the remaining data. This mixed training/testing procedure helps to prevent the models from being overly fragile, bizarrely specific models that only apply to the provided training examples.

2.2 Related Work

In general, NLP researchers have explored various ways in which automated text analysis tools (e.g., TagHelper) and methods can support researchers studying complex learning settings, either to replace hand coding or supplement it (Dönmez, Rose, Stegmann, Weinberger, & Fischer, 2005; Wang et al., 2007). Based on previous theoretical discoveries, researchers from the data mining community have tried to predict feedback helpfulness fully automatically. With the help of peer review software such as SWoRD, peer-review corpora are being collected and can be used for data mining and machine learning. Cho (2008) applied machine learning to a corpus collected with SWoRD to classify peer-review feedback as helpful or not helpful based on simple tags that were automatically generated by existing tagging software, TagHelper 2.0. Here machine learning algorithm performance was compared to author evaluations of the helpfulness of comments. Cho found that the performance of the classifier was limited by errors from the tagging software, which could not distinguish problem detection and solution suggestions — the feature that we are trying to detect automatically in this project.

With respect to using NLP to identify elements of critical feedback, we drew on prior work on sentiment analysis from product reviews (Wilson, Wiebe, & Hoffmann, 2005) and work on automatic detection of paraphrasing (Malakasiotis, 2009), because paraphrasing is one method that reviewers can localize a comment (i.e., "in the section in which you talk about …"). The prior work on automatic detection of plagiarism (i.e., overlap between an essay and libraries of text found online) was similarly relevant to automatic coding for comment localization (Ernst-Gerlach & Crane, 2008). They proposed an overlapping-window algorithm that searches for the most likely referred window of words through all possible primary materials to match a possible citation in a reference work. We applied this algorithm for our purpose, and developed features from the information of the window that the algorithm retrieved (e.g. window size, the number of overlapped items).

3. Study 1: Localization Detection

3.1 Overview

We selected the dataset used by Nelson and Schunn (2009) in which peer review comments submitted using SWoRD on a writing assignment from a large undergraduate history class were hand-coded for the presence and absence of several features, including localization and solution information. Next we use Natural Language Processing to determine the presence of simple text attributes (e.g., the length of the comment, the presence of search domain words, etc.). Then we use the open source machine learning software Weka (Witten & Frank, 2005) to automatically learn models that match, as closely as possible, the hand-coded localization information about each comment. Finally, we evaluate the accuracy of the best model.

3.2 Course Context

The course was selected to be a typical undergraduate survey course that is a common experience in US university settings in which large numbers of students are exposed to disciplinary content and reasoning skills. The course was entitled History of the United States, 1865-present. The students were heterogeneous, involving both students taking the course to fulfill a general education requirement, as well as students taking the course for interest. The majority of the students were native English speakers, and most students had taken a general writing class involving persuasive writing genres in their first year at the university, as is typically done in the US. As an introductory class, the participants can be generally considered novices in history writing and likely in argumentative essays in general. The selected essay assignment required students to write a six-to-eight page argument-driven essay answering one of the following questions: (1) whether the United States became more democratic, stayed the same, or became less democratic between 1865 and 1924, or (2) examine the meaning of the statement ''wars always produce unforeseen consequences'' in terms of the Spanish-American-Cuban-Filipino War and/or World War I.

To guide the reviewing activity, students were given general guidelines for how to provide useful comments (be constructive, be specific, be respectful) and then three different reviewing prompts for generating end-comments regarding the general flow of the paper and the logic of the arguments (see Nelson & Schunn for details). For example, for the prose flow dimension the detailed problem was: "Did the writing flow smoothly so you could follow the main argument? This dimension is not about low level writing problems, like typos and simple grammar problems, unless those problems are so bad that it makes it hard to follow the argument. Instead this dimension is about whether you easily understood what each of the arguments was and the ordering of the points made sense to you. Can you find the main points? Are the transitions from one point to the next harsh, or do they transition naturally?”

Thus, our analyses focus on detecting key features in end-comments provided by relative novices under loosely focused reviewing prompts. Note that these comments are provided with some accountability pressure to provide useful comments; see SWoRD system details below.

3.3 SWoRD System

SWoRD (Scaffolded Writing and Rewriting in the Disciplines) is a Web-based system that implements reciprocal peer reviewing of writing (Cho & Schunn, 2007). SWoRD shares many features in common with a variety of existing web-based peer review systems that have emerged over recent years (e.g., Calibrated Peer Review, the peer review add-on within Turnitin.com). SWoRD was initially developed for use in large undergraduate courses in academic disciplines in which writing is rarely assigned. Since 2002, SWoRD has been used by about 7,000 students from over 150 classes at twenty universities. SWoRD enables instructors to implement a range of reciprocal peer review activities:

1. Students write first drafts on a writing task and submit them online.
2. Students then are randomly assigned a set of these drafts to review (3-6 drafts).
3. As reviewers, students analyze the drafts along several evaluative dimensions, using prompts that incorporate explicit rubrics (determined by the instructor).
4. Students submit written comments online for a set of comment dimensions determined by the teacher.
5. Students rate papers on a set of rating dimensions determined by the teacher.
6. Students receive the (anonymous) feedback from all reviewers together.
7. Students revise their drafts, and re-submit them.
8. Authors also provide back-comments to the reviewers regarding the helpfulness of the written comments that they provided.
9. The revised drafts are made available to the same first draft reviewers.
10. The reviewers then observe how the revised drafts differ from the first drafts.
11. The reviewers rate and comment on the revised draft along the same dimensions.

For the current analyses, we focus on the comments submitted at step 4. The use of the back-comments in SWorD may have led students to produce more helpful comments overall than if there had been no pressure to provide useful comments.

3.4 Natural Language Processing Methods

First, we use Natural Language Processing to automatically represent each feedback comment as a vector of text attribute values. The attribute representation used in our history localization studies is shown in Table 1. This representation incorporates four types of attributes, motivated by our intuitions as well as by research in related areas such as quotation detection (Ernst-Gerlach & Crane, 2008).

Table 1. Summary of key text attributes of the feedback comments that serve as input into the decision tree for detecting localization, and how they were obtained through Natural Language Processing techniques.

Text attribute	How obtained
Regular expression attributes: regularExpressionTag	Simple regular expressions were employed to recognize common phrases of location (e.g., "on page 5", "the section about"). If any regular expression was matched, the value of the Boolean feature regularExpressionTag was True.
Domain lexicon attributes: #domainWord	Using standard statistical Natural Language Processing methods (e.g., to extract frequent lexical bigrams from text), a dictionary of domain words was generated automatically from the student papers. #domainWords then counted the number of domain words in each feedback comment.
Syntactic attributes: sub-domain-obj, deDeterminer	Using parsing software, we extracted information from the underlying syntactic structure of each feedback comment. We computed whether there were any domain words between the subject and the object (sub-domain-obj), and counted the number of demonstrative determiners (this, that, these and those).
Overlapping-window attributes: windowSize, #overlaps	To match a possible quotation from the paper being commented on, we searched for the most likely referred window of words in the author's paper. We extracted the length of the matching window (windowSize), plus the number of overlapped words in the window (#overlaps).

To illustrate the usage of this representation, consider the feedback values that were extracted from the following feedback comment (hand-coded as "Localized=True"):

> The section of the essay on African Americans needs more careful attention to the timing and reasons for the federal governments decision to stop protecting African American civil and political rights.

In the attribute vector, regularExpressionTag=True because one regular expression was matched with "the section of"; #domainWord = 8 because the sentence contained “African” (2), “American”, “Americans”, “federal”, “governments”, “civil”, “political.” There were no demonstrative determiners, thus deDeterminer=0. “African Americans” was between the subject “section” and the object “attention”, so “sub-domain-obj=True.” We constructed our domain lexicons in a data-driven manner by looking for common terms in the student papers themselves (Xiong & Litman, 2010; Xiong, Litman, & Schunn, 2010).

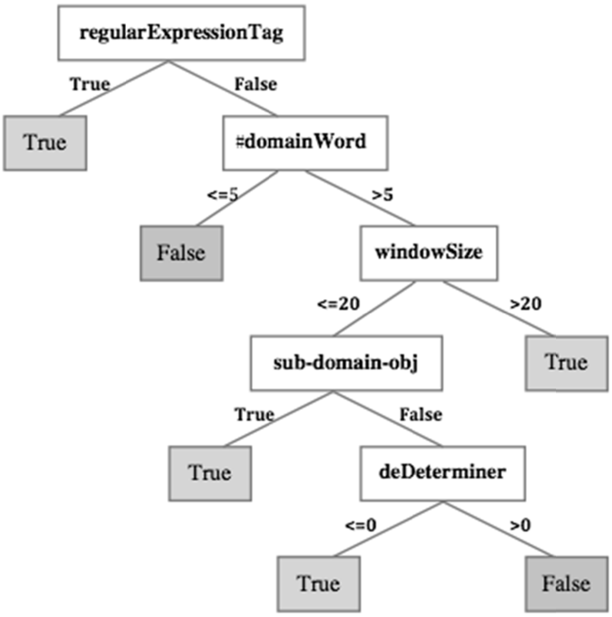


Figure 2: Decision tree for deciding whether a feedback comment is localized or not. The input to the tree is a feedback comment that has already been represented as a set of attribute values using Natural Language Processing. Each internal node corresponds to a test of the value of one of the attributes, while the branches from the nodes are labeled with the possible values of the test. If the branch leads to an internal node, another test is performed. When the branch reaches a leaf node, the decision regarding the localization value (True or False) is returned.

Once each of our annotated feedback comments is represented as an attribute vector, we next use Weka to automatically construct a model for detecting the target feedback feature (e.g. localization). While Weka supports many machine-learning algorithms, decision trees such as shown in Figure 2 have performed well in our pilot studies, and are easy for humans to interpret. For example, the learned tree for detecting localization first uses a match by a regular expression to detect localized feedback. For feedback whose regularExpressionTag is False, it then looks at the count of domain words. For domain word counts greater than 5, the overlapped content between feedback and its targeted paper is then considered, and so on. Note that Weka automatically selects the most predictive features (e.g. regularExpressionTag, #domainWord, etc.) and ignores the less useful ones (e.g. #overlaps).

3.5 Results

Results of classification models are typically compared against a baseline model that always predicts the more common category. Our model performs significantly better than the baseline. Overall accuracy is 77% in comparison to an accuracy of 53% for a baseline model, which is a statistically significant difference (*p*<.05). Models are often also examined in greater detail in terms of recall and precision metrics. Because we are primarily interested in flagging lack of localization, we are more interested in these metrics when predicting no localization. In this case, *recall* is the percentage of feedback comments that were actually not localized that were detected as not localized by the model; here the model rate was 82% whereas the baseline model was significantly lower, 0% (*p*<.05). *Precision* is the percentage of feedback comments labeled as not localized by the model that were in fact not localized; here the model rate was also 73% where was the baseline model was significantly lower again, 0%, (*p*<.05). Overall, the model can find a majority of non-localized comments while not suffering from a high rate of false alarms.

4. Study 2: Solution Detection

4.1 Overview

We applied the same general procedure to the same history course dataset, but this time trying to predict the presence/absence of explicit solutions in the comments rather than localization information. There were a different set of a relevant text attributes to use. Further, in this case, a linear regression model was the most successful model type.

4.2 Methods

Similarly as for detecting the presence of localization, each feedback comment is represented as a vector of attributes. Based on our intuition from our brief exploration of students' comments, we developed three groups of attributes that can be automatically derived from the surface of sentences, as shown in Table 2.

To illustrate the usage of this representation, consider the following feedback comment again (hand-coded as "solution-provided=True"):

> The section of the essay on African Americans needs more careful attention to the timing and reasons for the federal governments decision to stop protecting African American civil and political rights.

This feedback has 31 words (wordCount=31) and its index in the review is 2 (feedbackOrder=2). It has 2 collocation domain-topics ("African American" 2), 9 single-word domain-topics ("African" 2, "American", "Americans", "federal", "governments", "civil", "political" and "rights"), and 2 collocation essay-topics ("African American" 2) plus 8 single-word essay-topics (same as the unigram domain-topics except the "rights"). These four numbers are then normalized by the count of words in this feedback. As for Keyword, it contains 1 Suggestion ("need") and 2 Negatives ("more", "careful").

Given the attribute vector representation of feedback, we use the logistic regression model provided by Weka to learn the solution classifier. Based on our pilot study, logistic regression performs significantly better than the decision trees for detecting solutions. The algorithm constructs a linear relationship between the logistically transformed sum of weighted attributes and possible class values (true and false). To build the model, Weka automatically learned the weights and parameters from the provided training data.

Table 2. Summary of key text attributes of the feedback comments that serve as input into the regression model for detecting presence of explicit solutions, and how they were obtained through Natural Language Processing techniques.

Text attribute	How obtained
Simple features: wordCount, feedbackOrder	wordCount is the length of the comment and feedbackOrder its location in the overall review text (i.e., 1st comment, 2nd comment, … in one reviewer's set of comments).
Essay attributes: Domain-topic single words, Domain-topic combination words, Essay-topics single words, Essay-topics combination words	Four attributes that capture how closely the feedback comment is related to the domain topic and the specific essay topic. Domain/essay topics are represented as a set of single/combination words that are automatically computed based on their frequencies regarding all students' essays (to get domain topic words) or the feedback comments and the associated essay (to get the essay topic words). The specific attributes are the counts of each set of single/combination words that appeared in the feedback comment, which are normalized by the length of the comment length.
Keyword attributes: suggestion, location, problem, idea verb, transition, negative, positive, negation, summarization	Count attributes of nine categories of words. The words were semi-automatically learned in our pilot study and grouped into nine sets based on their syntactic and semantic functions. For example, location keywords are [page, paragraph, sentence], and negative keywords are [fail, hard, difficult, bad, short, little, bit, poor, few, unclear, only, more, stronger, careful, sure, full].

4.3 Results

Table 3 presents the best model, first listing the attributes in that model that predict the presence of an explicit solution (from most important to least important), and second listing the attributes in that model that predict the absence of an explicit solution (from most important to least important). Overall, domain/essay topic lexicons appear to matter a lot, but in a mixed way: domain topic words in combination appear to predict the presence of solutions, but domain topic words and essay topic words (alone or in combinations) appear to predict the absence of solutions. Not surprisingly, suggestion keywords (should, must, could, …) are highly predictive of solution presence. Also idea verbs (consider, mention) are strongly associated with solutions. Later comments appear also to be more likely to include suggestions. Perhaps more surprisingly, negations (not, doesn't, don't) are also predictive of solution presence. But this is not about explicitly naming problems, because negative keywords (fail, hard, difficult, bad, short, little, bit, poor, few, …) are mildly associated with the absence of solutions (presumably because the comment presents a problem rather than a solution).

Table 3. Coefficients and odds ratios for the best logistic regression model predicting presence of explicit solutions in comments.

Text attribute	Coefficient	Odds ratio
Predictors of presence		
Domain-topic combination words	26.29	2.61 x 1011
Suggestion keyword	2.44	11.45
Negation keyword	1.77	5.85
Idea verb keyword	0.82	2.29
Location keyword	0.4	1.49
feedbackOrder	0.31	1.36
wordCount	0.05	1.05
Predictors of absence		
Essay-topic combination words	-34.41	
Domain-topic single words	-3.65	38.46
Summarization keyword	-0.63	1.89
Problem keyword	-0.55	1.72
Positive keyword	-0.45	1.56
Essay-topic single words	-0.29	1.33
Negative keyword	-0.24	1.27
Transition keyword	-0.16	1.18

In line with this last point, problem keywords (error, mistakes, typo, problem, difficulties, conclusion) are also associated with the absence of solutions. Finally, summarization and positive keywords are associated with the absence of solutions, presumably because more summary oriented comments and praise oriented comments (even though they include an explicit problem) may not seem to critically require a provided solution.

Our solution model performs significantly better than the baseline model that always predicts the more common category (*p*<.05); it achieves an overall accuracy of 83% while the accuracy of the baseline is 63%. When our model is used to detect the false case (when the feedback provides no solution), the precision is 83% and the recall is 91%. In comparison, though the baseline has perfect recall (since it always predicts false), its precision is only 61%, which is significantly lower (*p*<.05) than our solution model. In other words, we find the great majority of no solution feedback cases while having a relatively low false alarm rate.

5. Study 3: Can Research Rely on Automatic Coding?

5.1 Overview

Having established that peer feedback can be automatically processed for localization and solution information to at least intermediate levels of accuracy, we can now explore whether those accuracy levels are sufficiently high for practical purposes. In particular, we explore whether researchers using the automatically coded data would come to the same research conclusions as researchers using hand coded data.

5.2 Methods

We used the same peer review dataset and computational linguistic model described in Study 1 for predicting localization as a test case. But we needed a new research question to examine; the model was developed to match hand coding of a feature that predicted student implementation rates, and thus it would not be very surprising if the models could also predict student implementation rates in this very dataset. So, instead we sought to predict helpfulness ratings generated by the peers and two different kinds of experts on the peer comments. The helpfulness ratings of the peers were collected via SWoRD as part of the normal peer review process—each author rates the set of comments on a given dimension from a given peer for helpfulness. The helpfulness ratings of the experts were collected afterwards (Cho, Schunn, & Charney, 2006). One of the experts was the history instructor of the course, which we call a content expert. The other expert was the director of the Writing Center and had a PhD in English, which we call a writing expert. The research questions we asked here were: 1) what features predict expert judgments of helpfulness; 2) do students, the content expert, and the writing expert each value the same types of feedback features?

Note that the helpfulness data is at the review level rather than the individual feedback element level. Thus, for each review we calculate the proportion of comments that were deemed to be localized, either by hand coding or by automatic coding. Similarly so for other features included in the analyses that were hand-coded:

proportion of comments that had solutions, proportion of comments that were praise only, etc.

5.3 Results

The first step is to determine whether we can account for expert helpfulness ratings (averaged across the two experts) using the feedback features that were hand-coded. We conducted a regression analysis using feedback type proportions (praise only comments, summary only comments, problem/solution containing comments), proportion localized critical comments, and proportion solution providing comments as possible predictors. To establish the robustness of the pattern across the data, we used a 10-fold cross-validation technique. The data is divided into 10 subsets. A regression model is built from 9 of the 10 subsets, and then tested on the 10^{th} held-out subset as a true prediction test. This approach is repeated for all 10 held-out subsets. Also, we found that support vector machines (Burges, 1998) produced better fits to the data than linear regressions, and so we report those SVM results.

Across the 10 cross-validation tests, the expert helpfulness data in the held-out set could be predicted from the SVM models with a mean Pearson correlation of *r*=.43 (95% CI of the correlations of ±.09), and localization was regularly an important part of these predictions (more on this below). Thus, these simple feedback features could predict significant elements of the expert judgments. When we replaced the localization hand-coded data with automatically coded data (for both building the models and testing them on the held-out data), the SVM models performed equally well, with a mean correlation of *r*=.46 (95%CI of the correlations of ±.11). Thus, there was no additional noise to our ability to model expert judgments by switching to automatically coded data**.**

More important, though, are the details of the models: are the same models built when human-coded or machine-coded data are used? Because the structure of SVMs are a little complex to parse, we conducted a simpler experiment in which we did simple stepwise regression again with 10-fold cross validation, and asked whether the stepwise regressions built models with the same features. Further, with only a few possible predictive features, there was a high likelihood of finding 'similar' predictive features. Therefore, we added a number of other possible predictors from the hand-coding and derived from the quantitative paper ratings: overall positivity of the review (pRating), and the difference between the rating of the given review and ratings generated by other students. For additional technical details, see Xiong and Litman (2011).

We repeated this approach in predicting helpfulness ratings generated by students, the writing expert, the content expert, and the average of the two experts. Table 4 presents which features were found in more than half of the 10 stepwise regressions in each case (and the number of times, out of 10, that given feature was found). Focusing on the hand-coded results first (middle column), we see that different features were important to different raters. Students were very influenced by the overall positivity of the review, whether explicit solutions were presented, and whether more than just summary or praise was included. By contrast, the writing expert focused heavily on whether a solution was included or not, the content expert focused on localization information, explicit solutions, and positivity of the review, and the experts combined appeared to be best modeled with solutions, positivity, and localization. Note that localization information was very much important in two of the cases (included in all or almost all of the best-fitting models) and very much unimportant in the other two cases (included in fewer than half of the best-fitting models if at all).

Table 4Feedback features commonly included (and frequency of inclusion) in 10 cross-validation stepwise regressions predicting feedback helpfulness ratings generated by peers and experts, using either hand-coded localization information or automatically generated localization information.

Helpfulness rater	With Hand-coded Localization information	With Automatically-coded Localization information
Students	pRating 10 Solution% 10 Problem% 9	pRating 10 Solution% 10 Problem% 10
Writing expert	Solution% 10	Solution% 10
Content expert	Localization% 10 Solution% 10 pRating 10	Localization% 10 Solution% 10 pRating 10
Expert average	Solution% 10 pRating 10 Localization% 9	Solution% 10 pRating 10 Localization% 9

Would the same conclusions be drawn with automatically coded localization data? The rightmost column of Table 4 presents these results. Overall, the regression model results are remarkably similar to those obtained from the hand-coded localization data, and 100% identical with respect to results involving localization. Thus, a researcher using the automatically coded localization data would have come to the same conclusions as a researcher using hand-coded data: the content expert values localization information but students and the writing expert do not.

6. General Discussion

The purpose of this paper was to explore the viability of automatically detecting lack of helpful elements in peer feedback using a combination of Natural Language Processing and Machine Learning techniques. On this front, we found that it could be done with some level of success. As researchers interested in studying peer feedback on writing, we can explore the use of these models to test the ways in which localization and solution frequency in feedback changes across individuals and situations, and the ways in which it influences the writing process. As writing instructors using web-based peer review, we are now well positioned to insert this technology into the SWoRD peer review system (with an interface similar to that in Figure 1), and then conduct an experiment that evaluates the impact of this interface on peer review comments and author implementation of comments.

The predictive models of the presence and absence of localization and solution information in comments were simply meant to be statistical tools rather than to provide deep insights into the content of localization information and solutions. However, the models do provide some interesting details on the kinds of attributes that tend to correlate with localization information and solutions even if those attributes are not necessarily part of localization information or solutions *per se*. For example, the model of solutions appears to be based on a slight opposition between providing descriptions of what the problem is and descriptions of what the solution is. Similarly, localization information was less likely to occur when there were few domain words in the feedback, possibly a sign of more superficial feedback thought by the reviewers to not require localization information.

One technical point worth mentioning is that the comments that were analyzed here were already pre-processed in two important ways. First, comments were segmented into idea units. In the old SWoRD interface that was used to collect this data, reviewers submit long comments to three different commenting prompts. We then split these comments into separate idea units by hand (see Nelson and Schunn (2009) for details). It would be a more difficult task to use NLP techniques to identify idea units in the larger comments. However, in the new SWoRD interface (and in some other web-based peer review interfaces like PeerMark), comments for a given commenting prompt are submitted as a set of separate idea units, so we believe this issue need not be a significant limitation.

Related to this point, we note that it is possible to examine the presence or absence of localization or solution information in feedback at the idea unit (as in Studies 1 and 2) or proportion of comments with localization and solution information at the whole comment (as in Study 3). Detection could also be done at higher levels of aggregation like at the complete review (across commenting dimensions) or reviewer levels (across multiple reviews by one individual). For example, we could develop a system that simply notes which students generally fail to include localization information and suggest they generally do so more often. We have explored more aggregate detection of feedback features (Xiong, Litman, & Schunn, 2010). The advantage of more aggregate feedback is that it might be more accurate (e.g., be less likely to give students incorrect feedback). The disadvantage of more aggregate feedback is that students may not be sure where specifically to improve—ironically similar to our point about the importance of localization information in feedback.

A second technical point is that the data was also further separated by hand into summary only comments, praise only comments, and critical comments that included problems and/or solutions. Our current work focused only on those critical comments, leaving aside summary only or praise only comments. Again, it is an additional technical challenge to automatically split comments into overall comment type (summary, praise, criticism) using Natural Language Processing techniques. However, we have begun work on this issue and have already found that it is possible to do with moderate levels of accuracy (Xiong, Litman, & Schunn, 2010). Another approach is to have students explicitly list summary, praise, and criticism in different review form areas. Pilot work on this approach suggests students are able to follow such instructions with high reliability.

7. Future Work

In terms of researcher use of these tools, we note an intermediate approach for those not yet willing to trust full automated coding of feedback data. Rather than coding from scratch, data could be first automatically coded through our algorithms, and then human coders could verify these codes. We have explored this approach and it appears to speed-up the coding process.

Clearly a next step in terms of applicability to writing practice is to implement the model in a peer review system to examine impact on peer comments and author implementation of feedback. Will reviewers significantly improve on both dimensions with this feedback? Will those improvements continue in later peer reviewing when such direct feedback is removed? If the feedback now begins to include these features more regularly, will this change in peer feedback translate into better papers, or is attention being drawn away from other key aspects of good feedback on writing?

Another question for both research and writing practice involves the generalizability of this approach across courses, which will vary by genre, complexity of writing, complexity of feedback, complexity and variety of domain topics, and presence of many second language learners, and many other issues which will influence the content of peer reviews. We have begun to explore SWoRD data from other courses and find that our techniques do appear to generalize to other courses.

There are also a number of important next steps towards further improvements to the model itself because the accuracy of prediction is still far from perfect. It is worth noting, however, that it is not necessarily the case that nearly perfect feedback is required to provide a strong push towards better feedback. As one idea on technical improvements, in the future, we hope to construct a more complete dictionary of domain vocabulary, which might provide us with better results for both localization and solution detection.

Acknowledgements

Work on this project was supported by a grant from LRDC to the second and third authors.

References

Ashbaugh, H., Johnstone, K. M., & Warfield, T. D. (2002). Outcome assessment of a writing-skill improvement initiative: Results and methodological implications. *Issues in Accounting Education, 17*(2), 124-148.

Attaliand, Y., & Burstein, J. (2006). Automated essay scoring with e-rater v.2. *Journal of Technology, Learning, and Assessment, 26*.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. Da*ta mining and knowledge discovery, 2*(2), 121-167. doi: 10.1023/A:1009715923555

Chapman, O. L., & Fiore, M. A. (2000). Calibrated peer review. Jo*urnal of Interactive Instruction Development, Winter,* 11-15.

Cho, K. (2008). Machine classification of peer comments in physics. Paper presented at the 1st Inte*rnational Conference on Educational Data Mining*, Montreal, Quebec, Canada.

Cho, K., & MacArthur, C. (2010). Student revision with peer and expert reviewing. Learning and Instruction, 20(4), 328-338. doi: 10.1016/j.learninstruc.2009.08.006

Cho, K., & Schunn, C. D. (2007). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. Compu*ters & Education, 48*(3), 409-426.

Cho, K., & Schunn, C. D. (2010). Developing writing skills through students giving instructional explanations. In M. K. Stein & L. Kucan (Eds.), I*nstructional Explanations in the Disciplines: Talk, Texts and Technology.* New York: Springer.

Cho, K., Schunn, C. D., & Charney, D. (2006). Commenting on writing - Typology and perceived helpfulness of comments from novice peer reviewers and subject matter experts. *Written Communication, 23*(3), 260-294.

Cohen, M., & Riel, M. (1989). The Effect of Distant Audiences on Students' Writing. American Edu*cational Research Journal, 26*(2), 143-159.

Cornelius-White, J. (2007). Learner-centered teacher-student relationships are effective: A meta-analysis. Re*view of Educational Research, 77*(1), 113-143.

Dönmez, P., Rose, C. P., Stegmann, K., Weinberger, A., & Fischer, F. (2005). Supporting CSCL with automatic corpus analysis technology. Paper presented at the 2005 conference on *Computer Support for Collaborative Learning.*

Ericsson, P. F., & Haswell, R. H. (Eds.). (2006). *Machine scoring of student essays: Truth and consequences.* Logan, UT: Utah State University Press.

Ernst-Gerlach, A., & Crane, G. (2008). Identifying Quotations in Reference Works and Primary Materials. *Research and Advanced Technology for Digital Libraries, 5173*, 78-87.

Goldin, I. M., Ashley, K., & Schunn, C. D. (2012). Redesigning Educational Peer Review Interactions Using Computer Tools: An Introduction. *Journal of Writing Research, 4*(2), 111-119.

Graesser, A. C., & McNamara, D. S. (2012). Use of computers to analyze and score essays and open-ended verbal responses. In H. Cooper, P. Camic, R. Gonzalez, D. Long & A. Panter (Eds.), *APA handbook of research methods in psychology*. Washington, DC: American Psychological Association.

Graham, S., & Perin, D. (2007). *Writing Next: Effective Strategies to Improve Writing of Adolescents in Middle and High School* - A Report to the Carnegie Corporation of New York. Washington DC: Alliance for Excellent Education.

Katstra, J., Tollefson, N., & Gilbert, E. (1987). The Effects of Peer Evaluation on Attitude Toward Writing and Writing Fluency of Ninth Grade Students. *Journal of Educational Research, 80*(3), 168-172.

Kiuhara, S. A., Graham, S., & Hawken, L. S. (2009). Teaching writing to high school students: A national survey. *Journal of Educational Psychology, 101*(1), 136-160.

Leijen, D. A. J., & Leontjeva, A. (2012). Linguistic and review features of peer feedback and their effect on the implementation of changes in academic writing: A corpus based investigation. *Journal of Writing Research, 4*(2), 178-202.

Lundstrom, K., & Baker, W. (2009). To give is better than to receive: The benefits of peer review to the reviewer's own writing. *Journal of Second Language Writing, 18*, 30-43.

Malakasiotis, P. (2009). Paraphrase recognition using machine learning to combine similarity measures. Paper presented at the 47th *Annual Meeting of ACL* and the 4th *Int. Joint Conf. on Natural Language Processing of AFNLP*.

McNamara, D. S., Crossley, S. A., & McCarthy, P. M. (2010). Linguistic features of writing quality. *Written Communication, 27*, 57-86.

McNamara, D. S., Louwerse, M. M., McCarthy, P. M., & Graesser, A. C. (2010). Coh-Metrix: Capturing linguistic features of cohesion. *Discourse Processes, 47*, 292-330.

Nelson, M. M., Melot, B., Stevens, C., & Schunn, C. D. (2008). The effects of skill diversity in peer feedback: It's what you don't know. Paper presented at the 30th *Annual Meeting of the Cognitive Science Society*.

Nelson, M. M., & Schunn, C. D. (2009). The nature of feedback: how different types of peer feedback affect writing performance. *Instructional Science, 37*(4), 375-401.

Palincsar, A., & Brown, A. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1*, 117-175.

Patchan, M. M., Charney, D., & Schunn, C. D. (2009). A validation study of students' end comments: Comparing comments by students, a writing instructor, and a content instructor. *Journal of Writing Research, 1*(2), 124-152.

Patchan, M. M., & Schunn, C. D. (2010). Impact of Diverse Abilities on Learning to Write through Peer-Review. Paper presented at the 32nd annual meeting of the *Cognitive Science Society*, Portland, OR.

Prichard, J. R. (2005). Writing to learn: An evaluation of the calibrated peer review(tm)program in two neuroscience courses. *Journal of Undergraduate Neuroscience Education, 4*(1), A34-A39.

Proske, A., Narciss, S., & McNamara, D. S. (2010). Computer-based scaffolding to facilitate students' development of expertise in academic writing. *Journal of Research in Reading, 35*(2), 136-152.

Sadler, P., & Good, E. (2006). The impact of self-and peer-grading on student learning. *Educational Assessment, 11*(1), 1-31.

Topping, K. J. (1998). Peer Assessment between Students in Colleges and Universities. *Review of Educational Research, 68*(3), 249-276.

Topping, K. J. (2008). Peer Assessment. *Theory Into Practice, 48*(1), 20-27.

van den Berg, I., Admiraal, W., & Pilot, A. (2006). Peer Assessment in University Teaching: Evaluating Seven Course Designs. *Assessment and Evaluation in Higher Education, 31*(1), 19-36.

Vygotsky, L. S. (1978). *Mind in Society*. Cambridge: Harvard University Press.

Wang, Y.-C., Joshi, M., Rose, C. P., Fischer, F., Weinberger, A., & Stegmann, K. (2007). Context Based Classification for Automatic Collaborative Learning Process Analysis. Paper presented at the 2007 Conference on Artificial Intelligence in Education.

Wilson, T., Wiebe, J. M., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. Paper presented at the *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Wooley, R., Was, C., Schunn, C., & Dalton, D. (2008). The effects of feedback elaboration on the giver of feedback. Paper presented at the 30th *Annual Meeting of the Cognitive Science Society*.

Xiong, W., & Litman, D. (2010). Identifying Problem Localization in Peer-Review Feedback. Paper presented at the 10th *International Conference on Intelligent Tutoring Systems*, Pittsburgh, PA.

Xiong, W., & Litman, D. (2011). Understanding Differences in Perceived Peer-Review Helpfulness using Natural Language Processing. Paper presented at the 6th *Workshop on Innovative Use of NLP for Building Educational Applications* (ACL-HLT Workshop). Portland, OR.

Xiong, W., Litman, D., & Schunn, C. D. (2010). Assessing Reviewers' Performance Based on Mining Problem Localization in Peer-Review Data. Paper presented at the *Third International Conference on Educational Data Mining,* Pittsburgh, PA.