An Exploration of Automated Grading of Complex Assignments

Chase Geigle, ChengXiang Zhai, Duncan Ferguson
University of Illinois at Urbana-Champaign
{geigle1,czhai,dcf}@illinois.edu

ABSTRACT

Automated grading is essential for scaling up learning. In this paper, we conduct the first systematic study of how to automate grading of a complex assignment using a medical case assessment as a test case. We propose to solve this problem using a supervised learning approach and introduce three general complementary types of feature representations of such complex assignments for use in supervised learning. We first show with empirical experiments that it is feasible to automate grading of such assignments provided that the instructor can grade a number of examples. We further study how to integrate an automated grader with human grading and propose to frame the problem as learning to rank assignments to exploit pairwise preference judgments and use NDPM as a measure for evaluation of the accuracy of ranking. We then propose a sequential pairwise online active learning strategy to minimize the effort of human grading and optimize the collaboration of human graders and an automated grader. Experiment results show that this strategy is indeed effective and can substantially reduce human effort as compared with randomly sampling assignments for manual grading.

Author Keywords

Automatic grading; ordinal regression; supervised learning; learning to rank; active learning; text mining

INTRODUCTION

Information Technologies have been transforming education dramatically recently, leading to the rapid growth of Massive Open Online Courses (MOOCs), which have not only made education more affordable and scalable, but also have huge potential to enable more effective personalized learning. Automatic grading technology has been a key component enabling the success of MOOCs. Unfortunately, the current technology for automatic grading is mostly limited to multiple-choice questions, short answers [3, 16, 20, 26, 22], and simple essay scoring [2], which makes it quite challenging for the current MOOCs to provide sophisticated assignments for teaching complex concepts or skills (e.g., critical thinking skills) since they cannot be easily graded in a scalable way. A solution currently adopted to bypass this difficulty is to use the calibrated peer review [2, 30, 24]. While there are encouraging findings about peer assessment and methods proposed to improve it [14, 24], there are still systematic problems with this approach: discrepancy between peer and instructor ratings, variation in ratings over time by the same peer rater, inconsistency across exercises for rating two works of similar quality, differences in rater stringency, and random fluctuation of ratings of the same work under varied conditions [30]. Preliminary data from a recent attempt to use this technique with veterinary students has also shown that peer reviews have a distinct positive bias (vide infra) relative to an expert instructor analysis [9]. Thus, it is important to develop more powerful automatic grading technology that can be applied to more sophisticated exercises than those provided by the current MOOCs, which are necessary in many education scenarios.

To automate grading of such a complex assignment, a natural idea is to use supervised machine learning to learn from graded examples for automatically assigning grades to ungraded assignments. As in other machine learning applications, the general idea here is that if we can extract those features from the assignments that can indicate the quality of an assignment, a machine learning program would be able to pick up the patterns of the features that can distinguish high-quality work from low-quality work from a sample of graded assignments (i.e., "training data"), thus potentially assigning a grade automatically to an ungraded assignment.

Although this idea is natural and appealing, there are many challenges and questions that we must address before we can effectively deploy such a technology in a real education environment, and a main goal of this paper is to take a first step toward systematically addressing these questions.

1. **Feasibility:** How feasible is it to use machine learning to automate grading of a complex assignment? What general features can we extract from assignments for automated grading? How effective are the state of the art machine learning approaches for automated grading? Are they sufficiently effective to be immediately useful in practice?

2. **Problem Formulation and Evaluation:** How should we formulate the grading problem as a machine learning problem? There are at least two options. One is to frame it as a classification problem with the goal of classifying an assignment into one of the finite number of pre-defined grade levels based on a rubric. The other is to frame it as a

L@S 2016, April 25 - 26, 2016, Edinburgh, Scotland UK

ACM 978-1-4503-3726-7/16/04 $15.00
DOI: http://dx.doi.org/10.1145/2876034.2876049

ranking problem where the goal is to rank the assignments based on the quality without necessarily assigning a specific grade—human graders can then go through the ranked list to segment the assignments into different grade levels. How should we design evaluation metrics to measure the quality of the results of automated grading?

3. **Integration of Automated Grading and Human Grading:** How exactly should such an automated grader be integrated with instructor or TA grading? A more general question is: how can we optimize the collaboration of an imperfect automated grader with more reliable human graders? Intuitively, the optimization depends on a trade-off between the quality/reliability of grading and the amount of human effort required. But given an expected amount of human effort, what is the best way to have the automated grader to assist a person in grading? What is the best way to have a human grader help train the machine-learning based automated grader?

While some of these questions have been studied for non-complex assignments, most of them are open new questions that have not been addressed in the existing work (see Section 2 for a detailed discussion of related work). In this paper, we will systematically study these questions using a particular type of complex assignment that requires sophisticated critical thinking skills, i.e., medical case assessment. This kind of assignment is very important for medical professional education. By studying how to automate grading for medical assessment assignments, we can potentially enable medical professional education to scale up—a much needed effort. Not teaching clinicians about clinical uncertainty has been referred to as "the greatest deficiency of medical education throughout the twentieth century" [7, 11]. However, implementing an instruction plan with an online education system at large scale to teach clinical uncertainty in decision making raises many significant challenges that must be solved, particularly challenges in automatic evaluation of the case studies completed by the students, which we address in this paper by leveraging information retrieval and machine learning techniques.

To study the feasibility questions, we propose a general methodology for designing three complementary types of feature representations of such complex assignments, including token features, similarity features, and selection features, and experiment with these features using ordinal regression for predicting the grade levels in multiple dimensions of rubrics. The token features are based on the term tokens extracted from an assignment and they offer the most general representation and are robust in practice. The similarity features are to capture the similarity between an assignment and the solution provided by an instructor; the intuition is that the higher the similarity is, the higher the grade should be. Finally, the selection features are to quantify the accuracy of the selection of relevant parts in a case description based on how well the selected parts match the solutions (e.g., choosing to run the right lab tests in a clinical case). While it is generally beneficial to manually design assignment-specific features, such features cannot be generalized to work on other assignments; in this paper, we focus on studying *general* features that can be *automatically computed* on any semi-structured complex assignment, and aim at understanding their effectiveness.

A practical challenge in studying our problem is the lack of a large set of graded assignments which is needed both for training a machine learning program and for validating the results of automated grading. This is partly due to the fact that grading such assignments takes much human effort: the very reason why we need to study automated grading for such assignments. In our experiments, we used a data set of 107 student submissions for one medical case assessment assignment that is available to us. While the data set is small, we are able to observe statistically significant differences in our experiments, thus it still allows us to draw meaningful conclusions about different approaches to automated grading.

Our study with this data set shows that it is feasible to automate grading of a complex assignment such as a medical case assessment using standard machine learning approaches and the proposed three kinds of general features provided the instructor can grade a small number of examples, but the grading accuracy on different rubric categories varies substantially.

The results of our feasibility study reveal that there is a great deal of variation in the grades given by instructors due to the inevitable subjectivity of the rubrics. This suggests that it might be less effort and more reliable for an instructor to make pairwise judgments between a pair of assignments as opposed to assigning an exact numerical or letter grade. Working on such pairwise preference judgments also makes it easy to integrate non-expert judgments (such as peer grading) that might not be reliable in the exact grades assigned but may include relatively reliable preference judgments. Moreover, working on pairwise preferences naturally "eliminates" the need for normalizing numerical grades which might be biased (e.g., some graders may be overly generous).

Given that we will attempt to obtain pairwise preferences as training examples, it follows that we should frame the problem of automated grading as to rank the ungraded assignments, as opposed to predict the exact grade of an assignment. The ranking would be in descending order of quality (in any rubric dimension or overall quality with consideration of multiple dimensions), and a human grader can then easily segment the list into any desired grade levels. In comparison with predicting exact grades, such a ranking formulation also offers a natural way to engage humans in validating and finalizing the grades. For evaluation, although retrieval measures such as Mean Average Precision (MAP) or normalized Discounted Cumulative Gain (nDCG) are commonly used for evaluating a ranked list, we suggest that the Normalized Distance-based Performance Measure (NDPM) [31] is a better measure for our ranking problem since it can better handle the many inevitable ties that occur in our case.

In practice, an automated grader must be integrated with a human grader so as to minimize the overall effort of the human grader while ensuring a certain level of grading accuracy. There is an inherent trade-off here since to increase the grading accuracy we would like to have as many training examples (i.e., manually graded assignments) as possible, which, however, would incur more human effort. To optimize human-machine collaboration and enable a flexible trade-off between human effort and grading accuracy, we propose the following sequential training process based on active machine learning: *(1)* a human grader first grades a small number of assignments as the initial training set (this could be either numeric or letter grades, or pairwise judgments); *(2)* the machine would learn from the initial set, and identify the next "best" example (i.e., assignment) to label and present it for human to grade (where "best" here means that the example is most valuable to help the automated grader improve its accuracy); *(3)* a human would grade the nominated example to increase the size of the training set by one; *(4)* the machine would learn from the augmented training set and repeatedly present a new example for the human to grade until it reaches a desired level of accuracy, at which point, the process stops and the human grader would segment the final ranked list to generate grades for all the assignments. Our experiment results show that this online active learning process is much more effective than batch training.

RELATED WORK

To the best of our knowledge, no previous work has studied how to automatically grade a complex assignment such as a medical case assessment. However, our work is related to multiple lines of existing work, which we briefly review below.

Automated grading has been explored mostly for constrained question types where the correct answer has a certain, well known form. Programming assignments, for example, have long been a target for automatic grading [10, 12] as their very medium can easily be leveraged for providing "yes" or "no" feedback with respect to programmatic correctness. For specific assignment types, more sophisticated techniques like edit-distance of canonical representations has been explored [1]. Recent efforts have focused on providing feedback to students about their programs by leveraging structural similarities in the code itself to allow feedback to be provided to many assignments at once that share particular features [23, 25].

In this vein, clustering-based techniques have been applied to tools designed to help instructors manually grade short-answer MOOC assignments at scale by allowing them to assign grades to entire clusters of students at once [3]. Hierarchical clustering methods were applied in this work to allow the instructor to "drill down" as far as he/she would like to assign grades and feedback to students. Their method, PowerGrading, can be regarded as optimizing the collaboration of humans and machines heuristically, but the approach does not take advantage of supervised learning from graded examples of instructors, which we explore. Furthermore, if a cluster is poorly formed, the grading error can be serious no matter how an instructor optimizes the grade assignment to a cluster.

Mitros et. al. [21] give a brief overview of different strategies for grading and proposed a heuristic workflow to optimize the collaboration of assessors in consideration of different costs associated with different graders. However, it does not address the question of how to optimize the recommendation of assignments for graders to grade in order to maximize the effectiveness of the machine learning component of their framework, a goal we seek to achieve in this paper. We could deploy our technology in their framework by modifying the threshold strategy (e.g., for cutoffs on a ranked list). Both of these methods [3, 21] only explored the short-answer question space, leaving semi-automated grading of more complex assignment outputs (like the outline-form case assessments we study here) unexplored.

Another approach would be to attempt to predict the grades explicitly. One branch of work in this direction based on information extraction techniques focuses on matching expected patterns in the answer; Many methods require the manual construction of these patterns [20, 16], while others attempt to learn them from large training datasets [26]. In either case, the methods require strong supervision support to be effective. Other works take an unsupervised text-similarity approach and compare the student answers with a gold standard answer using a wide variety of similarity functions [22].

Grading of long-form student answers has also been explored [2]. In CarmelTC [27] a combination of topic modeling and text classification approaches are taken to score student essays. The system attempts to determine which "key components" have been mentioned in each essay and uses this information to suggest to students what components they may be missing. Approaches that purely use document similarity metrics [8] or purely supervised classifiers [15] have been used for grading as well, but the rubrics are not as complex as those required for medical case assessment.

The task of predicting categorical labels with an implicit ranking (ordinal variables, often the result of surveys on a Likert scale) is often solved via ordinal regression methods [19]; our work adds yet another application of ordinal regression to the many already explored. Using machine learning for optimizing ranking has been extensively in information retrieval [18]; our work explores an interesting novel application of online active learning to automated grading where we are interested in minimizing the training sample to label while maximizing the ranking accuracy over a finite number of known test cases.

MEDICAL CASE ASSIGNMENT

Complex assignments inevitably vary across courses. As a first step toward studying how to automate grading of such assignments, we use a medical case assignment in the veterinary medicine domain for our study. At a high level, such an assignment represents a typical type of analysis assignment where the students are given a case description with both an unstructured text description as well as some structured data (e.g., lab test results), and are asked to perform an analysis of the case. The analysis generally involves 1) selecting relevant content from the case description, which can be selected from both the text part and the structured data, 2) answering questions with short textual answers, and 3) writing assessments in natural language text.

More specifically, the case exercises were developed using the WhenKnowingMatters (WKM) web-based case formulation software[1] which facilitates development and exchange of text-based cases while allowing students to objectify their

[1] http://www.whenknowingmatters.com

analysis	2.6355 ± 0.7660
answers	3.0280 ± 0.7668
application	2.8692 ± 0.5651
clarity	3.3832 ± 0.9437
quality	3.1121 ± 0.9795
questions	2.8224 ± 0.6810

Table 1. Mean and standard deviation of ranks in each of the rubric dimensions we study.

observations from a case and manipulate them in an outline format around a suggested scaffold provided by the instructor. The student's analysis is then rendered into a structured text format to facilitate automatic grading.

Due to the lack of automated grading tools, the assignments are currently graded manually. An assessment rubric designed prior to instruction was used by the instructor to evaluate student performance on a subjective, 5-point scale (listed here in increasing order): novice, beginner, competent, proficient, and expert. Rubric categories were related to elements of critical thinking and communication: *(1)* **Questions:** developing relevant refining (or clarifying) questions to answer based upon an honest assessment of current knowledge base; *(2)* **Answers:** approach to seeking answers to developed questions; literature search, etc.; *(3)* **Quality:** judgment of quality of information; awareness and application of standards of a discipline, bias detection including appropriate humility to detect one's own potential bias, application of statistical concepts; *(4)* **Analysis:** analysis of an argument; *(5)* **Clarity:** clarity and communication of thought; conciseness, grammar, spelling, elocution; and *(6)* **Application:** application and understanding of appropriate disciplinary content.

For our experiments, we used a data set consisting of $n = 107$ student submissions for one medical case analysis assignment in a veterinary medicine course at UIUC. Each was graded according to the rubric detailed above. We report the mean rank and standard deviation for each of these six labels in Table 1; where 1 corresponds to novice and 5 to expert. The instructor also created a "gold standard" assessment for the assignment case, which is available for the automated grading tool to use. We wish we could use a much larger data set, but the size of the data set is limited by the amount of manual work needed for grading, which is precisely our motivation for studying how to automate grading.

Figure 1 shows an example of a very simple case and a typical student answer. In the case description, the student can see a text description of the case and a number of lab test results in the form of structured data. The student assessment is seen to be a semi-structured text with indented structures based on a scaffold provided by the instructor. Multiple tags indicate different kinds of answers, including, e.g., selected content from the original case description, selected lab tests (both are "observations"), and text input by the student reflecting his/her assessment (called "analysis").

Because of the complexity, automated grading of such an assignment is very challenging. First, due to variations across different assignments, it is almost impossible to learn from the grading results of one assignment to automate grading of another (often called "transfer learning"), even though such an "inter-assignment" automated grading is ideal. We thus focus on a more realistic setting of attempting to automate the grading after the instructor has already graded some assignments, which we may refer to as "intra-assignment" automated grading, which, strictly speaking, is actually semi-automatic grading. Our goal is thus to study whether and how we can leverage machine learning to learn from the graded assignments to reduce the grading burden on an instructor, either by directly predicting grades or by providing a ranking as a scaffolding for assigning grades.

FEASIBILITY OF AUTOMATED GRADIING

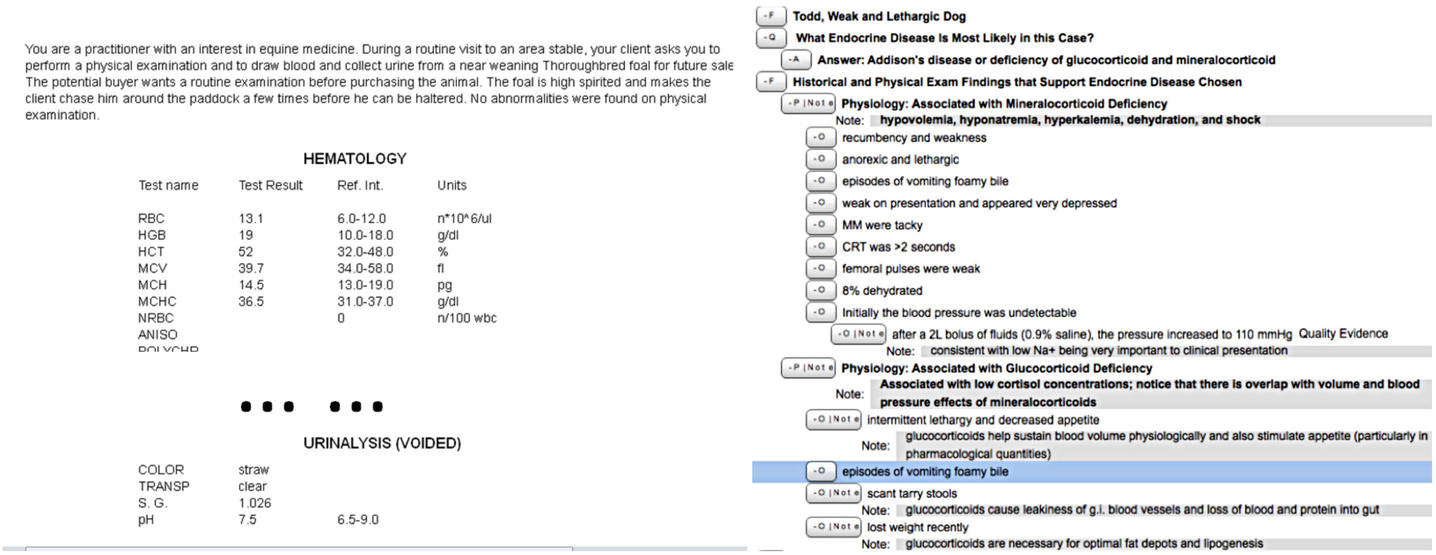
In this section, we discuss and study the feasibility of using machine learning methods, particularly supervised learning, for automating the grading of complex assignments. We first present the general idea of supervised learning, then propose a general methodology for designing three complementary types of features for representing assignments, which are needed for supervised learning, and finally present experiment results.

Supervised Learning

In supervised learning, a model is built to predict the outcome (or label) of a new data example based on previous examples it has seen before (called the training data). Thus a natural way to use supervised learning for grading is to have a human (e.g., instructor) to grade a set of assignments to be used as training data to learn a model to predict the grade of each ungraded assignment. A critical component of this infrastructure is the decomposition of examples into feature vectors—this decomposition enables the use of algorithms for learning functions from these vectors to the output labels desired. Typically, these feature vectors are either binary or real-valued, and are often (but not always) in a high-dimensional space. The performance of the learned function is directly tied to the features used in the vector representation of the examples—poor features result in low predictive capability due to the algorithm being unable to find meaningful patterns in the examples. As such, these features are a critical component of any supervised learning approach. With a properly defined set of features that are capable of capturing the salient patterns in the training examples, the task can be given to any of a number of state-of-the-art algorithms for learning appropriate predictive functions that can be applied to yet-unseen data (the test data). Another factor affecting the accuracy of prediction is the number of training examples, with more training examples leading to higher accuracy. However, since creating training examples generally requires manual work, we tend to have only a limited number of training examples to work with. How to define general features that we can automatically compute based on a complex assignment and how to minimize manual effort in creating training examples are two major questions that we study in this paper.

Defining Features of a Student Assignment

The performance of a supervised learning approach is highly dependent on the effectiveness of the features fed into the learning program. Thus a main technical challenge we need



You are a practitioner with an interest in equine medicine. During a routine visit to an area stable, your client asks you to perform a physical examination and to draw blood and collect urine from a near weaning Thoroughbred foal for future sale The potential buyer wants a routine examination before purchasing the animal. The foal is high spirited and makes the client chase him around the paddock a few times before he can be haltered. No abnormalities were found on physical examination.

HEMATOLOGY

Test name	Test Result	Ref. Int.	Units
RBC	13.1	6.0-12.0	n*10^6/ul
HGB	19	10.0-18.0	g/dl
HCT	52	32.0-48.0	%
MCV	39.7	34.0-58.0	fl
MCH	14.5	13.0-19.0	pg
MCHC	36.5	31.0-37.0	g/dl
NRBC		0	n/100 wbc
ANISO			
POLYCHR			

● ● ● ● ● ●

URINALYSIS (VOIDED)

COLOR	straw	
TRANSP	clear	
S. G.	1.026	
pH	7.5	6.5-9.0

- -F **Todd, Weak and Lethargic Dog**
- -Q **What Endocrine Disease Is Most Likely in this Case?**
    - -A **Answer: Addison's disease or deficiency of glucocorticoid and mineralocorticoid**
- -F **Historical and Physical Exam Findings that Support Endocrine Disease Chosen**
    - -P |Note **Physiology: Associated with Mineralocorticoid Deficiency**
      Note: **hypovolemia, hyponatremia, hyperkalemia, dehydration, and shock**
        - -O recumbency and weakness
        - -O anorexic and lethargic
        - -O episodes of vomiting foamy bile
        - -O weak on presentation and appeared very depressed
        - -O MM were tacky
        - -O CRT was >2 seconds
        - -O femoral pulses were weak
        - -O 8% dehydrated
        - -O Initially the blood pressure was undetectable
            - -O |Note after a 2L bolus of fluids (0.9% saline), the pressure increased to 110 mmHg Quality Evidence
              Note: consistent with low Na+ being very important to clinical presentation
    - -P |Note **Physiology: Associated with Glucocorticoid Deficiency**
      Note: **Associated with low cortisol concentrations; notice that there is overlap with volume and blood pressure effects of mineralocorticoids**
        - -O |Note intermittent lethargy and decreased appetite
          Note: glucocorticoids help sustain blood volume physiologically and also stimulate appetite (particularly in pharmacological quantities)
        - -O episodes of vomiting foamy bile
        - -O |Note scant tarry stools
          Note: glucocorticoids cause leakiness of g.i. blood vessels and loss of blood and protein into gut
        - -O |Note lost weight recently
          Note: glucocorticoids are necessary for optimal fat depots and lipogenesis

Figure 1. An example of a case description (left) and a reference assessment (right). Assessment bullets are labeled: "F" is part of the instructor provided framework, "Q" is a question posed by the instructor, and "P" is a physiological point made by the student.

to solve is how to design effective features for representing an assignment.

To address this challenge, we propose a general framework for defining features for complex assignments such as the one we explore in this paper. The features we propose are general in nature and thus should be applicable to any assignment that is presented in a text-based, semi-structured response form. We describe a set of feature classes and evaluate the performance of these features on an example autograding task to evaluate their predictive capacity. Our framework consists first of constructing a "view" of an assignment and then defining features based on this view. The view chosen for the assignment is critical in that it changes the way we may naturally describe it and thus leads to the definition of distinct classes of features distinguished by the view taken to derive them. We will explore features by progressively taking views that make stronger assignment design assumptions: while the features are still general, each view progressively narrows the space of possible student response types.

The first class of features, which we call **token features**, are generated by taking a view of the student response consistent with the traditional "bag of words" approach common in information retrieval contexts [18]. In this view a document is decomposed into a vector of count data that indicates the frequency of words within the document. Two features are thus natural. The first type of feature would indicate the number of occurrences of a given word in a student submission (and is thus real-valued), and the second would indicate the presence or absence of a word (and is thus binary-valued). These features would both create a high-dimensional representation of the student submission, and are motivated by an attempt to capture the difference in vocabulary between assignments. This is often enough to capture whether the correct ideas are mentioned without requiring extensive computation (features from this class are trivial to compute for every student submission), though more discriminative units such as n-grams (a sequence of *n* words) may also be easily used to replace single words if necessary. Document classification techniques typically operate in this kind of space.

The second class of features, which we call **similarity features**, are generated by characterizing a student submission by the "distance" from a gold standard (e.g., an assignment submission generated by the instructor). With this view, features can be derived that strongly utilize the structure of the assignment (e.g., how closely does the outline structure of the student assignment match the outline structure of the instructor assignment?) as well as features that loosely utilize or completely ignore the structure of the assignment. Examples of features that loosely utilize the assignment structure would be the similarity of certain outline bullet types with the gold standard bullet types of the same category. A bullet type in our examples could be "observation" (indicating something selected from the assignment text directly) or "analysis" (indicating original thoughts from the student). These features require the assignment to be structured in such a way that this information is easily extracted, but do not look so closely at the overall structure of the outline itself. Ignoring the structure of the assignment, features can be generated that indicate the overall similarity with the gold standard. Document clustering techniques typically operate in this kind of space, as well as retrieval functions in search systems [18].

The third class of features, which we call **selection features**, are generated by measuring concrete statistics about the selection of bullet points compared to a gold standard. In some sense, these are similar to the similarity features, but they differ in that they make a stronger assumption about the assignment structure—namely, that students are producing the exact same text that should occur in a similar section of the gold standard. Examples of selection features would be precision (what fraction of the bullets selected by the student also appear in the gold standard?) and recall (what fraction of bullets selected in the gold standard were also selected by the student?) [18].

analysis	0.5642 ± 0.0733
answers	0.5491 ± 0.0634
application	0.3321 ± 0.0580
clarity	0.7604 ± 0.0659
quality	0.7868 ± 0.0810
questions	0.4321 ± 0.0640

Table 2. Difficulty of grading each rubric dimension, characterized by MAE of a SVOR model learned on 50% of the data. 10 randomized experiments were run; reported is the average and standard deviation.

Ordinal Regression for Grade Prediction

Because of the ordinal nature of our grade labels (categorical with an implicit ranking), it is natural to apply ordinal regression techniques to our automated grading setup. In particular, we will utilize support vector ordinal regression (SVOR) [5], a generalization of the popular support vector machine (SVM) [6] for classification to the case of ordinal labels in the study of feasibility of grade prediction.

Experiment Results

We now present the results of ordinal regression on our medical assignment data set to assess the effectiveness of the proposed features and examine how effective such a state of the art learning method is for solving the grading problem.

We first explored using only the most general of our feature types—token features—to attempt to understand the differences in grading difficulty across our different rubric dimensions. Frequency-based token features were extracted: we used the META toolkit[2] at version 1.1 with its default tokenizer, stemmer, and stopword list. For regression, we used a modified version of LIBSVM[3] for ordinal regression [17].

In an actual grading scenario, the instructor would manually grade a certain number of the submissions, learn the regression function from these labeled examples, and then apply the learned model to the remaining unlabeled examples. To simulate this, we ran the following experiments: for each rubric dimension, we took the collection of student documents and randomly split it into two groups (the training and test sets) each containing 50% of the data[4]. A function is learned based on the labeled training set which is then used to label the examples in the test set. We compute the **mean absolute error** (MAE), defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |r(f(x_i)) - r(y_i)|$$

where $f(x_i)$ is the predicted label of the example x_i, $r(\cdot)$ is the rank of a given label, and y_i is the gold standard label for the example x_i. This experiment is repeated for ten different randomized splits, and we report the average and standard deviation of the test set MAE in Table 2.

We can observe that the rubric labels with the least variation are the easiest to predict (e.g., "application" and "questions"), whereas rubric dimensions with higher data variance (e.g., "quality") are more difficult.

The Impact of Different Feature Types

Moving beyond simple token features, we extracted both similarity and selection features from our assignments and incorporated them incrementally into our model to measure the predictive capacity of different feature types.

Our token features were generated using the same process detailed previously (frequency-based features extracted using the META toolkit). Our similarity features (compared against an instructor-generated assignment submission) were overall similarity, similarity of only "observation" bullets, and similarity of only "analysis" bullets. These were computed using the Okapi BM25 similarity function often used in information retrieval as a scoring function [18], treating the instructor submission as a query and the student submissions as documents to be scored. Finally, our selection features were precision and recall [18] of the selected lab data in the student case analysis when compared against the instructor's assignment.

We investigate the predictive capacity of these features by exploring the improvement of the model when predicting our most challenging labels ("quality" and "clarity"). We ran ten separate experiments with different randomized training sets consisting of 50% of the data when using different feature combinations to represent the student submissions. We again report the average and standard deviation of the MAE on the test set across the ten runs. To further explore whether the regression method is truly capturing patterns relevant for grading, we compare its MAE against the MAE obtained by using a naïve baseline: compute the most frequent label in the training data, and then assign this label to all examples in the test set. Intuitively, this is a very reasonable baseline when comparing MAE—if the labels are normally distributed, picking the most frequent one will ensure an absolute error of zero for the majority of the examples—while simultaneously being unhelpful for discriminative grading (which the regression method hopes to capture). Our results are summarized in Tables 3 and 4.

We see that for the "quality" dimension, the model is able to successfully learn generalizable patterns in our features to predict the label with errors that are statistically significantly less than the baseline method. In general, the token features dominate the performance, but it would seem as though the similarity and selection features have lower variability in the MAE. Again, this result suggests that there are likely gains to be had by utilizing a more sophisticated feature selection method to remove some of the noise introduced by extraneous token features.

However, the "clarity" label shows us that the problem is far from being solved in a general sense. Here, we see that our method consistently fails to beat the baseline method, with the winning method being seemingly random. This indicates that the features we have selected thus far are more tailored toward discrimination along certain dimensions of the grading rubric than others. More work must be placed into developing features that truly capture the "clarity" dimension to allow

[2] https://meta-toolkit.org

[3] http://www.work.caltech.edu/~htlin/program/libsvm/

[4] We do not use something like 10-fold cross validation due to the small size of the available labeled data to ensure that the training and test sets can be as representative of the actual data as possible.

	Baseline	**SVOR**
sim (3)	0.9358 ± 0.0882	**0.8811 ± 0.0940**
sim + sel (5)	0.9566 ± 0.1677	**0.8642 ± 0.0325**
toks (2646)	0.9075 ± 0.0789	**0.7660 ± 0.0910**†
all (2651)	0.9792 ± 0.1568	**0.7566 ± 0.0738**†

†: statistically signifigant using an unpaired t-test with $p \leq 0.05$.

Table 3. Effectiveness (in terms of MAE) of incorporating additional features in grade prediction for "quality" dimensions using SVOR methods compared to the mode-assigning baseline. Number of features is given in parenthesis.

	Baseline	**SVOR**
sim (3)	0.7906 ± 0.0771	**0.7830 ± 0.0836**
sim + sel (5)	**0.7623 ± 0.0649**	0.7811 ± 0.0561
toks (2646)	0.7528 ± 0.0550	**0.7415 ± 0.0597**
all (2651)	**0.7189 ± 0.0617**	0.7226 ± 0.0527

Table 4. Similar experiment to Table 3, but for "clarity" dimension.

the model to extract the patterns the instructor observes when grading along this dimension.

What this demonstrates is that automatic grading of complex assignments is currently feasible, but perhaps only in a limited fashion. Careful feature generation is required, but in some cases a model can be learned to effectively grade assignments. We suspect that significant gains in grading performance can be obtained in other dimensions with better features.

AUTOMATED GRADING AS RANKING ASSIGNMENTS

Some of the results from our feasibility study using ordinal regression raise the question whether framing the problem of automated grading as ordinal regression is appropriate. Indeed, as we will discuss, it appears to be more advantageous to frame the problem as one of ranking the ungraded assignments, which a human grader can segment into desired grade levels.

Specifically, as we observed in Tables 3 and 4, outright prediction of an ordinal grade can be very challenging due to the highly concentrated nature of the dataset labels (see Table 1). The vast majority of grade information available for the grade prediction task is centered around the mean, leaving very little information in the tails for a supervised learner to extract patterns from. (In some cases, for example, there are as few as one example for the highest and lowest ordinal grade values). The result is noisy output that may be inappropriate for using directly. However, it is worth noting that ordinal grade prediction is a hard problem, even for humans: a previous study suggests disagreement rates around 44% for short answer grading [22]. We suspect that this only becomes larger as assignments become more complex and difficult to grade, which makes the task of outright label prediction much more difficult for the machine as well.

Thus an alternative, and more reasonable approach may be to produce a ranked list of assignments from best to worst. Annotators are typically more consistent at providing judgments of the form "is *a* better than *b*?" than "on a scale from 1–5, how good is *a*?" [4], so it is reasonable to suspect that a machine learning model could achieve better results when trained using such pairwise judgments. If a system can provide a good ranking of assignments, an instructor simply needs to assign "cutoff" points in this ranking to determine grades. This simplifies the learning problem from attempting to predict an ordinal label for a specific assignment to assigning a ranking to a set of assignments. This is a well studied area in information retrieval called "learning to rank" [13, 18], and there are a wide variety of methods available that one can use to learn a ranking function for documents given a set of features.

One particular method that we will explore is a pairwise solution called a Ranking SVM [13], where the problem of learning to create ranked lists is decomposed into the problem of determining preferences for pairs of items (i.e., whether *a* should appear before *b*). A traditional SVM model is learned on this decomposition, and its weight vector is used to define a retrieval function that is the dot product with a document's feature vector.

Before we explore the efficacy of such an approach, however, we must first redefine some measure by which we can measure performance. Because the system is no longer predicting a rating for each assignment, we cannot use MAE as before.

Evaluating Ranking-based Grading Systems

Our goal is to produce a ranking of student assignments that is consistent with instructor evaluation. One way of framing this problem is to compare the ranking produced by the system to the ranking produced by the instructor (which we'll call the "reference ranking"). A system's ranking can then be evaluated using some measure of correlation between the two rankings. We note a preference for metrics that take into account the *entire* ranked list—this contrasts with most of the preferred measures in information retrieval evaluation which typically place heavier emphasis on the top-ranked elements. While this makes sense in a search context, our goal is to produce an exhaustive ranking of the assignments, so we focus on these types of measures.

Measures for rank correlation are plentiful. Perhaps the most commonly used metrics are Kendall's τ or Spearman's ρ (which have been found to be highly correlated in practice [29]; thus, we present only one for illustration). Kendall's τ can be formulated as

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)},$$

where n_c is the number of *concordant pairs*, and n_d is the number of *discordant pairs*, and n is the number of items ranked. To compute n_c and n_d, one considers all pairs (x_i, y_i) and (x_j, y_j) (that is, pairs of tuples) of assigned rankings in the system ranking X and the reference ranking Y (the denominator is simply the number of such pairs). A pair is *concordant* if the ordering of the items i and j in X and Y is consistent—in other words, if $(x_i < x_j) \wedge (y_i < y_j)$ or $(x_i > x_j) \wedge (y_i > y_j)$. A pair is *discordant* if the ordering of items in the two lists is inconsistent—in other words, if $(x_i < x_j) \wedge (y_i > y_j)$ or $(x_i > x_j) \wedge (y_i < y_j)$. This is then a correlation measure, with values bounded in $[-1, 1]$, with 1 indicating a perfect correlation and -1 indicating inverse correlation.

One of the assumptions Kendall's τ makes is that there are no ties in ranks. However, in a realistic grading scenario based on rubrics we expect many ties. Fortunately, there is a variation of Kendall's τ, denoted as τ_b, that accounts for ties in the rankings. This is formulated as

$$\tau_b = \frac{n_c - n_d}{\sqrt{(n_c + n_d + t_x)(n_c + n_d + t_y)}}$$

where t_x is the number of pairs that were tied on *only* their ranking from X, and t_y is the number of pairs that were tied on *only* their ranking from Y.

This may, at first glance then, seem like a good measure to use, but it is not without its problems. Despite taking into account ties in the rankings, it may still penalize a system for re-ordering items that were tied in the reference ranking—in other words, we may be penalized for not correctly identifying elements who are tied in the reference ranking. Consider a simple example: suppose the ranking proposed by a system is $X = (1,2,3,4,5,6)$ but the reference ranking is $Y = (1,1,2,2,3,4)$. Intuitively, the system made no real mistakes in that no pair where the reference ranking asserted an is in the wrong order in X. However, we'll see that $\tau_b \approx 0.9309$, indicating that the system did not achieve perfect correlation.

To address this issue, Yao [31] proposed the normalized distance-based performance measure (NDPM), which computes a distance between two rankings that is insensitive to a system's reordering of tied elements in the reference ranking. NDPM is computed as

$$NDPM = \frac{2n_d + t_x}{2(n_c + n_d + t_x)}.$$

This can also be described as the distance between the system ranking and the reference ranking divided by the maximum achievable distance any ranking could have from the reference ranking. Thus, a value of 0.3 would indicate that the system ranking was 30% of the distance away from the reference ranking than the reverse of the reference ranking. Since this is a normalized *distance* measure, a value of 0 would indicate a perfect ranking. Indeed, if we compute NDPM for the example rankings above, we achieve this result. Thus, we feel that NDPM is perhaps the most appropriate measure for evaluating automatic grading systems that produce an ordering of assignments as their output.

EFFICIENTLY UTILIZING HUMAN JUDGMENTS WITH ACTIVE LEARNING

As in all supervised learning approaches, the accuracy of the automated grader based on learning to rank. depends on the quantity and quality of the training examples available for the learner to use. Ideally, we would like human graders to provide as many graded examples as possible, but this would reduce the benefit of an automated grader. Indeed, if a human grader completes grading all the assignments, there would be no need for the automated grader! However, if there are insufficient training examples to learn from, the automated grader migh have a low accuracy, which would further require more human effort on "post-editing" the grading results of the automated grader. Thus there is clearly a complicated tradeoff between the effort of manual grading and the utility of the trained grader that may have to be empirically optimized in an application-specific way. However, it is very clear that if we ask human graders to grade a certain amount of assignments, we would like the graded assignments to be as useful to the automated grader as possible. Just randomly selecting a sample of assignments for manual grading is not the best way. A natural solution to this problem is to employ active learning to allow the machine learning model to guide the instructor in providing the supervision to make the most effective use of his/her effort.

Building on these observations, we thus propose the following "pairwise active learning to rank" model for automatic grading, which will employ the following process where k_1 is a parameter that can be empirically set: *(1)* Ask the instructor for comparative judgments on k_1 pairs of assignments, *(2)* Learn a model using a learning-to-rank approach on the available pairwise judgments, *(3)* Apply the model to all remaining unjudged pairs, *(4)* Select an unjudged pair to present to the instructor for judgment, and *(5)* Go to step *(2)*. Instantiations of this general approach will differ mainly in steps *(2)* and *(4)*.

To study whether our proposed active learning approach better utilizes human judgments during the grading process, we performed the following experiment. We took our assignments and assigned each a "composite score", computed as the average of their ordinal score for each of the six rubric dimensions. Our task is then to learn a ranking that is consistent with the ranking produced by these composite scores while simultaneously *minimizing instructor effort* in labeling.

We first transform the $n = 107$ assignments into $\frac{1}{2}n(n-1) = 5671$ assignment *pairs* (x_i, x_j) with corresponding labels $y_{ij} \in \{+1, -1\}$ indicating whether x_i should be ranked above or below x_j in the ranking. Ties were broken arbitrarily by assignment id. The supervision given by the instructor is then to indicate a preference for ranking x_i relative to x_j.

Following the process laid out in the beginning of the section, we first start with $k_1 = 10$ random pairs selected from the transformed data and ask for labels from the instructor. We then learn the model, compute the NDPM for the ranking produced by the model for all n assignments, and then ask for additional supervision by selecting the unlabeled assignment pair whose distance from the decision boundary for the model is lowest (this is a known, simple approach to uncertainty sampling [28]) and repeat the training/evaluation loop. Our particular model choice was a linear SVM provided through the META toolkit.

We compare this active learning scenario with a random learning baseline, which is the exact same process as above, but instead of selecting the most uncertain pair in the unlabeled data we select one uniformly at random. This will allow us to see whether the uncertainty sampling approach is truly helping to guide the learning process to make more efficient supervision choices or not.

Our results are summarized in Figure 2. Recall that a NDPM value of 0.3 indicates that a system ranking was 30% of the maximal achievable distance away from the reference ranking. We can see that even at a small fraction of all of the assignment pairs, the active learning approach (blue line) is able to achieve better NDPM than simply learning at random (red line). This

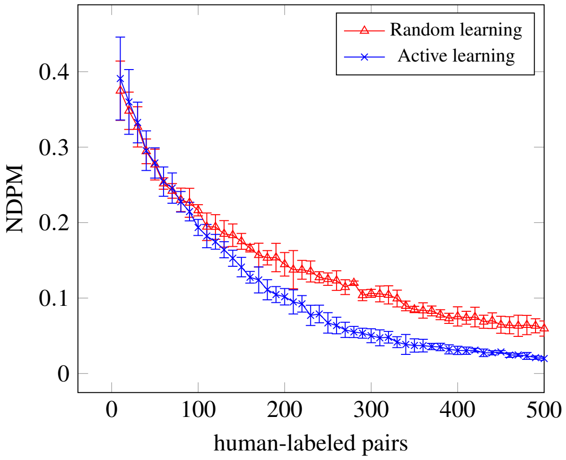


Figure 2. A comparison between a randomized learning solution and an active learning solution to the grading-as-ranking problem. Reported is the average NDPM (lower is better) over 5 runs, with error bars indicating one standard deviation.

is consistent with our hypothesis that active learning as part of an automatic grading system can make more effective use of an instructor's time than a purely passive supervised approach.

How much instructor effort goes in to judging 200 assignment pairs? This may initially seem like a lot, but each pair is not labeled in isolation—labeling many pairs will inevitably include assignments that have already been seen before. These familiar assignments make providing a pairwise judgment faster than it would be if done "cold". In general, it is also reasonable to assume that the effort involved in simply saying whether assignment a is better than assignment b is lower than having to consult a rubric to assign an actual point (or letter) value. It is important, however, to ensure that providing a pairwise judgment takes as little effort as possible relative to assigning a numeric or letter grade. An interesting future direction is then to design an interactive system that attempts to further drive the cost of providing judgments down.

DISCUSSION AND FUTURE WORK

In the previous experiments we formulated the automated grading problem as a ranking problem, and introduced a rank distance measure (NDPM) as a form of evaluating the quality of a ranked list generated by an automated (or semi-automated, in our case) system. Under a ranking-based problem formulation, we argue that this is the most sensible metric for evaluating the ranking accuracy relative to a gold standard.

However, the value of NDPM cannot be easily compared with the values of existing metrics (such as MAE) that have been traditionally used in evaluating automated grading systems in the past. There is a need to evaluate a ranked list from the perspective of its impact on the eventual grades assigned to student work. Unfortunately, how to evaluate the utility of a ranked list appropriately remains a challenge partly due to the difficulty in choosing the cutoffs, which may depend on the desired tradeoff that an instructor wants (e.g., a desired distribution of grades in different brackets). In practice, we envision that the instructor would visit points in the ranked list and choose cutoffs based on the tradeoff between the different types of grading errors. Exploring grade cutoff assignment strategies remains an important future direction, and our framework coupled with such a cutoff strategy would enable evaluation based on the traditional grade prediction task.

While we believe the results here show that the methods employed are feasible for grading complex assignments, more work remains to be done to understand just how well our system performs relative to human judgments. Future work should explore this by measuring human consensus in grading these complex assignments, similar to what has done for short answers [22]. Furthermore, we only investigated very simplistic features—such as the bag-of-words model—which are very general but not very sophisticated. Exploring the feature space further to find more sophisticated features that perform well in practice and are more tailored to the goals of medical case assessments remains as future work.

Another major limitation of our study is the limited size of the data set. This is partly due to the fact that such complex assignments currently can only be graded by human graders. In the future, we hope to deploy our automated grading tools to help scale up such courses to enable more students to participate, which in turn, would help collecting more data for further verification of our observations and conclusions.

Finally, a crucial direction that remains unexplored is feedback: how could such a system give more detailed feedback to students beyond just their ordinal rating along a rubric dimension? Currently, peer grading approaches have an advantage in this sense, as your peers can suggest to you corrections or point out specific mistakes that you made. It is worth investigating whether or not we can generate "explanatory reports" of grading results when using a supervised learning approach.

CONCLUSIONS

Automated grading of complex assignments is necessary for scaling up learning without compromising effectiveness of learning. Using a data set of medical case assessment assignments, we conducted the first systematic study of how to leverage machine learning to automate grading of such a complex assignment. Our study has led to several contributions.

First, we have experimentally shown the feasibility of using supervised learning techniques for automated grading of medical case assignments under certain conditions provided that the instructor can manually label a number of the assignments to serve as a training set. In particular, an ordinal regression method can be applied to the data with results that consistently outperform the majority-label baseline in terms of MAE.

Second, we proposed a general framework for the development of three complementary types of representative features for student submissions (i.e., token features, similarity features, and selection features) —while we applied these features to our specific task of medical case analysis grading, these feature types (and generation framework) are general and should apply to the grading of any complex assignment.

Third, we proposed to frame the problem of automated grading as a ranking problem, which can more naturally assist human graders to validate and finalize grades of ungraded assignments and learn from pairwise preference judgments that can be potentially created more reliably by human graders including through peer grading. We also suggested NDPM as potentially a better measure for this ranking task than other measures due to its superiority in handling many tied cases.

Finally, we proposed an iterative procedure of online active learning to rank to efficiently utilize human judgments, and thus optimizing the collaboration between human graders and the automated grader. Experiment results confirm the efficiency of this procedure which can substantially save human effort as compared with randomly choosing sample assignments for humans to grade.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant Number DGE-1144245.

REFERENCES

1. R. Alur, L. D'Antoni, S. Gulwani, D. Kini, and M. Viswanathan. 2013. Automated Grading of DFA Constructions *(IJCAI)*. 1976–1982.
2. S. P. Balfour. 2013. Assessing writing in MOOCs: Automated essay scoring and calibrated peer review. *Research and Practice in Assessment* 8, 1 (2013), 40–48.
3. M. Brooks, S. Basu, C. Jacobs, and L. Vanderwende. 2014. Divide and Correct: Using Clusters to Grade Short Answers at Scale. In *ACM L@S*. 89–98.
4. C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder. 2007. (Meta-) Evaluation of Machine Translation. In *WMT*. 136–158.
5. W. Chu and S. S. Keerthi. 2007. Support Vector Ordinal Regression. *Neural Comput.* 19, 3 (2007), 792–815.
6. C. Cortes and V. Vapnik. 1995. Support-Vector Networks. *Mach. Learn.* 20, 3 (1995), 273–297.
7. B. Djulbegovic. 2004. Lifting the fog of uncertainty from the practice of medicine. *British Medical Journal* 329, 7480 (2004), 1419–1420A.
8. R. M. Duwairi. 2006. A Framework for the Computerized Assessment of University Student Essays. *Comput. Hum. Behav.* 22, 3 (2006), 381–388.
9. DC Ferguson, LK McNeil, EM Mills, and JE Ehlers. 2014. International efforts to encourage critical clinical thinking (CCT) skills in veterinary students.. In *Veterinary Educational Collaborative biannual meeting, Ames, IA.* (abstract).
10. G. E. Forsythe and N. Wirth. 1965. Automatic Grading Programs. *Commun. ACM* 8, 5 (1965), 275–278.
11. E. Gambrill. 2006. *Critical Thinking in Clinical Practice: Improving the Quality of Judgments and Decisions* (2nd ed.). Wiley and Sons. 648 pages.
12. M. T. Helmick. 2007. Interface-based Programming Assignments and Automatic Grading of Java Programs. In *SIGCSE*. 63–67.
13. T. Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *KDD*. 133–142.
14. C. Kulkarni, K. P. Wei, H. Le, D. Chia, K. Papadopoulos, J. Cheng, D. Koller, and S. R. Klemmer. 2013. Peer and Self Assessment in Massive Online Classes. *ACM Trans. Comput.-Hum. Interact.* 20, 6 (2013), 33:1–33:31.
15. L. S. Larkey. 1998. Automatic Essay Grading Using Text Categorization Techniques. In *SIGIR*. 90–95.
16. C. Leacock and M. Chodorow. 2003. C-rater: Automated Scoring of Short-Answer Questions. *Computers and the Humanities* 37, 4 (2003), pp. 389–405.
17. L. Li and H. Lin. 2007. Ordinal Regression by Extended Binary Classification. In *NIPS*. MIT Press, 865–872.
18. C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
19. P. McCullagh. 1980. *Regression models for ordinal data*.
20. T. Mitchell, T. Russell, P. Broomhead, and N. Aldridge. 2002. Towards robust computerised marking of free-text responses. In *ICAAC*.
21. P. Mitros, V. Paruchuri, J. Rogosic, and D. Huang. 2013. An Integrated Framework for the Grading of Freeform Responses. In *MIT LINC*.
22. M. Mohler and R. Mihalcea. 2009. Text-to-text Semantic Similarity for Automatic Short Answer Grading. In *EACL*. 567–575.
23. A. Nguyen, C. Piech, J. Huang, and L. Guibas. 2014. Codewebs: Scalable Homework Search for Massive Open Online Programming Courses. In *WWW*. 491–502.
24. C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. 2013. Tuned Models of Peer Assessment in MOOCs. In *EDM*.
25. C. Piech, J. Huang, A. Nguyen, M. Phulsuksombati, M. Sahami, and L. J. Guibas. 2015. Learning Program Embeddings to Propagate Feedback on Student Code. In *ICML*. 1093–1102.
26. S. G. Pulman and J. Z. Sukkarieh. 2005. Automatic Short Answer Marking. In *BEA*. ACL, 9–16.
27. C. P. Rosé, A. Roque, D. Bhembe, and K. Vanlehn. 2003. A Hybrid Text Classification Approach for Analysis of Student Essays. In *BEA*. ACL, 68–75.
28. B. Settles. 2012. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6, 1 (2012), 1–114.
29. G. Shani and A. Gunawardana. 2011. *Evaluating Recommendation Systems*. Springer, Chapter 8.
30. H. Suen. 2014. Peer assessment for massive open online courses (MOOCs). *The International Review of Research in Open and Distance Learning* 15, 3 (2014).
31. Y. Y. Yao. 1995. Measuring Retrieval Effectiveness Based on User Preference of Documents. *J. Am. Soc. Inf. Sci.* 46, 2 (1995), 133–145.