Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading

Sumit Basu
Microsoft Research
One Microsoft Way
Redmond, WA
sumitb@microsoft.com

Chuck Jacobs
Microsoft Research
One Microsoft Way
Redmond, WA
cjacobs@microsoft.com

Lucy Vanderwende
Microsoft Research
One Microsoft Way
Redmond, WA
lucyv@microsoft.com

Abstract

We introduce a new approach to the machine-assisted grading of short answer questions. We follow past work in automated grading by first training a similarity metric between student responses, but then go on to use this metric to group responses into clusters and subclusters. The resulting groupings allow teachers to grade multiple responses with a single action, provide rich feedback to groups of similar answers, and discover modalities of misunderstanding among students; we refer to this amplification of grader effort as "powergrading." We develop the means to further reduce teacher effort by automatically performing actions when an answer key is available. We show results in terms of grading progress with a small "budget" of human actions, both from our method and an LDA-based approach, on a test corpus of 10 questions answered by 698 respondents.

1 Introduction

Increasing access to quality education is a global issue, and one of the most exciting developments in recent years has been the introduction of MOOCs—massively online open courses, in which hundreds or thousands of students take a course online. While this works wonderfully for lectures, assessment in the form of quizzes and exams presents some significant challenges. One straightforward solution is to use multiple choice questions, but it is well known that there is far greater educational benefit from fill-in-the-blank and essay questions (Anderson and Biddle, 1975). For many domains, then, there could be great value in using short answer questions; the problem is grading those answers without making the cost prohibitive. Even in relatively small classrooms of a few dozen to a few hundred, the ability to grade such answers more efficiently would be a great boon to teachers.

One approach to addressing this is to automatically grade such answers as right or wrong or with a numerical score—there have been significant past efforts in this space (Leacock and Chodorow, 2003; Mohler and Mihalcea, 2009; Jordan and Mitchell, 2009). However, in practice this path has some significant obstacles. The first is that while these approaches have made impressive progress, they are never 100% accurate—some number of misgraded answers are left on the table. The second is that assigning a score is not really sufficient—in a small classroom, a teacher would give feedback as to why the answer is wrong; ideally she would be able to do this in the MOOC scenario as well. A third problem, or at least a lost opportunity, is that there may be consistent patterns of misunderstanding amongst students that go beyond the fraction getting a question right or wrong. For instance many students might mistakenly believe that one of the rights afforded by the first amendment to the US constitution is the right to bear arms; a teacher would want to know this so that she could correct their misconception in class.

To address these issues, we propose looking at the problem in a different way. Instead of trying to grade answers completely automatically, we attempt to leverage the abilities of both the human and

Transactions of the Association for Computational Linguistics, 1 (2013) 391–402. Action Editor: Rada Mihalcea.
Submitted 5/2013; Published 10/2013. ©2013 Association for Computational Linguistics.

the machine. In particular, instead of classifying individual answers as being right or wrong, we propose to automatically find groupings and subgroupings of similar answers from a large set of answers to the same question, and let teachers apply their expertise to mark the groups. We found that answers for a particular question often cluster into groups around different modes of understanding or misunderstanding. Once identified, such groups would allow a teacher to quickly "divide and conquer" the grading task—she could mark entire groups as right or wrong, and give rich feedback to a whole group at once. This has the additional benefit of increasing a grader's self-consistency, found to be a problem in past studies (Jordan and Mitchell, 2009; Pulman and Sukkarieh, 2005). The groupings also allow the teacher to get an overview of the level of understanding of her students and the modes of misunderstanding. In the absence of an answer key, we propose to form these groupings automatically without any model of the question or its answers; however, if a simple text answer key is available, we can use it to automatically mark some groups.

While attractive in principle, the powergrading approach of dividing and conquering contains many questions and challenges of its own, as various students will express the same answer in many different ways. Popular approaches to clustering text, such as using LDA to group answers by inferred topics, do their best to explain these variations in terms of distributions over words, but are limited by their word-based representations of text. Ideally, we would like to learn how to group items together based on data, with an array of features that expand over time as our technologies grow more mature. We propose to model this distance function by training a classifier that predicts whether two answers should be grouped together, in the vein of past work which modeled the similarity between student answers and answer key entries (Mohler et al. 2011).

The notion of this distance function is subtle. We want answers that are paraphrases of each other, such as "the Congress" and "the houses of Cngress [*sic*]" to be close, but "the Senate and the House of Representatives" to be different from these so we can mark out this more precise mode. Because we are modeling the distance between answers as opposed to the answers themselves, we can use "between-item" features that measure semantic or spelling differences. We thus supply our classifier with the best available features that can account for misspellings, tenses, and other variations, with the hopes that we can add more sophisticated features in the future.

Finally, we would like to evaluate the benefit of our cluster-based approach. In general, evaluating how helpful a given clustering is for a particular task can be difficult, but this scenario offers a very specific criteria—we examine how far a grader can get with a given amount of effort. One take on this measure is "grading on a budget," where we want to maximize the progress from a fixed number of actions; another is "effort left for perfection," which is the number of additional user actions required to grade all items correctly. Under these criteria, we find that using clusters formed via the learned similarity metric leads to substantially better results than using those formed via LDA or individually classifying items.

2 Related Work

Decades of educational research have demonstrated the critical importance of assessment in learning. Testing contributes in multiple ways to the learning process: testing is characterized as formative when used to guide the learning, and summative when used to evaluate the student. Notably, testing has been shown to play a key role in the learning process as it assists retention (Anderson & Biddle, 1975), and answer construction for open response is shown to play a critical role in consolidating learning (Karpicke and Roediger, 2008).

Though multiple choice questions (MCQs) are the dominant method of assessment at present, there are drawbacks to the approach; see Conole and Warburton (2005) and Bull and McKenna (2004) for comprehensive surveys of Computer Assisted Assessment in general. MCQ is widely used primarily for the ease of grading, but while the summative value of MCQs may be obvious, the formative value of MCQs is dubious (Davies, 2002; Warburton and Conole, 2003). Additionally, answering an MCQ requires the recognition of the correct answer(s), which is known to be an easier task than the construction of the answer (Laufer and Goldstein, 2004). Essays are another form of assessment and have been shown to be amenable to automatic grading (Burstein et al., 2004), though the grading is not formative as it cannot provide feedback on essay quality.

Open response questions are challenging to grade, but testing with open response is both summative and formative. This challenge has attracted both the academic community as well as sponsored challenges such as the Automated Student Assessment Prize (Hewlett Foundation, 2013). The most widely used method for grading open response answers relies on careful authoring of the answer (Mitchell et al., 2002; Leacock and Chodorow, 2003; Jordan and Mitchell, 2009; Nielsen et al., 2009). C-rater (Leacock and Chodorow, 2003) is a paraphrase recognizer that identifies rephrasings of the answer key as correct answers. To recognize the paraphrases, c-rater uses sophisticated linguistic processing, in addition to automatic spelling correction. The authors describe an interface which guides the expert (i.e., teacher) in creating the model answers to the questions, which must represent a considerable time investment since only "if the teacher uses the same question for several classes or over several semesters, then the advantages of the initial effort are worthwhile." They report that c-rater agreed with human raters about 84% and a related work (Attali et al., 2008) reports agreement of 84% for the biology test and 93% for the Psychology test. Similarly, Jordan and Mitchell (2009) describe an authoring tool "which enables a question author with no knowledge of natural language processing (NLP) to use the software." These methods require that all of the various linguistically similar forms of the correct answer are encoded prior to grading, but cannot account for unanticipated student answers. Finally, Pulman and Sukkarieh (2005) compare hand-authored patterns with machine learned patterns based on simple word-based features, and find that the hand-crafted patterns perform better.

Short answer grading can also be formulated as a similarity task in which the score is assigned based on the similarity of the teacher and student answers (Mohler and Mihalcea, 2009; Mohler et al., 2011; Gomaa and Fahmy, 2012, i.a.). In Mohler and Mihalcea (2009), given a data set of 21 questions with 30 student answers each, the authors compare various measures of lexical similarity, including the use of knowledge-based resources (WordNet) and corpus-based metrics (Latent Semantic Analysis, or LSA); the best results (92% accuracy when the continuous grade scores are binarized and the threshold is set using binary labels on other held-out questions) are obtained using LSA trained on a corpus of Wikipedia articles topically related to the questions being graded (Mohler and Mihalcea, 2009). Mohler et al. (2011) explores how features can be included that encode the similarity of nodes based on a syntactic alignment of student answer and teacher answer; a system using these features does not on its own outperform the bag-of-words based metrics but in combination the improvements are measurable. Similarly, Meurers et al. (2011) and Hahn and Meurers (2012) show that using semantic analysis to align student and target answers, including functional roles such as subject/object, has an overall accuracy of 86.3%, improving on results that use alignment only at the surface level.

3 Data

While some datasets of student answers are publicly available, most have only a small number of students (e.g., 30 in Mohler and Mihalcea, 2009) or are multiple choice. To study the problem as we have posed it, we needed a large number of responses to open-ended questions. We thus selected twenty questions from the United States Citizenship Exam (USCIS, 2012) and offered them to two groups as a task on Amazon Mechanical Turk; we received 100 complete responses from the first group (for training) and 698 from the second (for test). A subset of these questions (1–8,13,20) were selected as they represented a range of answer lengths, from a few words to a sentence or two. The particular questions that were manually graded are listed in Table 1, as well as the average answer length and the number of case-independent unique answers. In addition to the train/test split, to further prevent any biasing from the target set, all training of classifiers and parameter settings was done on the complement of these questions (9–12 and 14–19) on the smaller set, so they were exclusive both in terms of answer content and responders.

For our proposed approach, we need two different types of labeling for our data. The first identifies groups of answers that are semantically equivalent; this is used to train the distance metric between items. This labeling was done by a single labeler (an author) on the complement set of questions described above, to ensure that we are learning general measures and not ones specific to particular questions or students. There is of course some subjectivity to this labeling, but rather than argue that we have the best possible labeling, we show that a learned model leads to improved performance.

Q#	Unique	Avg Len.	Question
1	57	3.3	What are the first ten amendments to the U.S. Constitution called?
2	132	3.2	What is one right or freedom from the First Amendment?
3	586	7.8	What did the Declaration of Independence do?
4	205	2.0	What is the economic system in the United States?
5	138	1.5	Name one of the three branches of the United States government.
6	219	2.8	Who or what makes federal (national) laws in the US?
7	395	5.2	Why do some states have more Representatives than other states?
8	157	4.0	If both the President and the Vice-President can no longer serve, who becomes President?
13	367	4.2	What is one reason the original colonists came to America?
20	276	4.8	Why does the flag have 13 stripes?

Table 1. Subset of questions used for evaluating our method and data characteristics for the 698 responses.

	# Marked Correct by Each Grader out of 698			
Q#	1	2	3	Kappa
1	651	652	651	0.992
2	609	617	613	0.946
3	587	587	492	0.574
4	567	574	541	0.864
5	655	668	658	0.831
6	568	582	548	0.838
7	645	649	652	0.854
8	416	425	409	0.966
13	613	535	557	0.659
20	643	674	678	0.449

Table 2. Differences in graders' judgments and inter-annotator agreement (Fleiss' Kappa).

The second type of labeling is the ground truth grading, i.e., the "correct" vs. "incorrect" for each student response for each question. Even though an answer key was available for our data, the open-ended nature of the questions means that some answers will be subject to interpretation. For instance, for the question "why does the flag have 13 stripes?" the answer key says "because there were 13 original colonies" and "because the stripes represent the original colonies," but when a student writes "13 states" as their answer, should that be considered correct, or does their fundamental confusion regarding states vs. colonies warrant a correction? As such, different teachers will have different grading patterns, and rather than attempting to optimize for the average labels, an effective system should help a teacher quickly converge to the grades they intend. We thus show separate results in terms of agreement with each of three graders.

Finally, in order to allow other researchers to benefit from this data, we have made the set of twenty questions, the answer key, and all responses, as well as the annotators' grades and groupings, available at http://research.microsoft.com/~sumitb/grading.

4 Learning a Similarity Metric Between Student Answers

Given the labeled groups of similar answers (and the remaining answers not assigned to any group), we wish to learn a distance metric between them. We frame this as the problem of learning a classifier of whether they are similar or not, where each data item will be based on two answers and a positive or negative label. The resulting classifier can return a score $sim(a_1, a_2)$ between 0 and 1; we can express the distance between items as $d(a_1, a_2) = 1 - sim(a_1, a_2)$. For each answer in a labeled group, then, we generate one positive and two negative examples: the positive example contains the current answer and one other answer from the group; the first negative example contains the current answer and an item from another group; the second pairs it with an item not placed in any group, for a total of 596 training examples.

4.1 Features for the Similarity Metric

For each labeled pair we generate an array of features expressing the relation between the items; we can then use these features and the labels to train the classifier we desire. These are "between-item" features: they concern the relationships between a_1 and a_2, as it is such features that we hope will be predictive of whether the items are similar. Note that all features below are computed after stopwords have been removed from both items. We also treat words that appear in the question as stopwords in a process termed as "question demoting" by Mohler et al. (2011), who found this resulted in noticeable improvements in measuring similarities between student answers and answer key entries.

Mohler and Mihalcea (2009) showed that a feature based on an LSA decomposition (Dumais, 2004) of Wikipedia was particularly powerful for grading. In a similar vein, we computed the LSA for all English Wikipedia articles (from 2012) using the

most frequent 100k words as a vocabulary; we then computed the similarity between answers using the top 100 singular vectors. In the descriptions below, we use "tf-idf vector similarity" to refer to the cosine similarity between standard tf-idf term-frequency vectors. These tf and idf scores are computed using the entire corpus of relevant answers, as this process does not make use of labeled data.

The full set of features is as follows:

a. *Difference in length*: the absolute difference between answer lengths in characters.
b. *Fraction of words with matching base forms*: we find the derivational base forms for all words (Quirk et al., 2012), count the words with matching bases in both answers, and normalize by the average answer length in words.
c. *Max idf of matching base form*: max idf of a word corresponding to a matching base.
d. *tf-idf vector similarity of* a_1 *and* a_2
e. *tf-idf vector similarity of letters*: the letter-based analogue to tf-idf with stopletters (punctuation, space, etc.) removed.
f. *Lowercase string match*: whether the lowercased versions of the strings match.
g. *Wikipedia-based LSA similarity*

While we have used these particular features for the experiments in this paper, we expect that over time, as more sophisticated features become available, the performance of this classifier and thus the technique as a whole will improve further.

4.2 Performance of the Similarity Metric

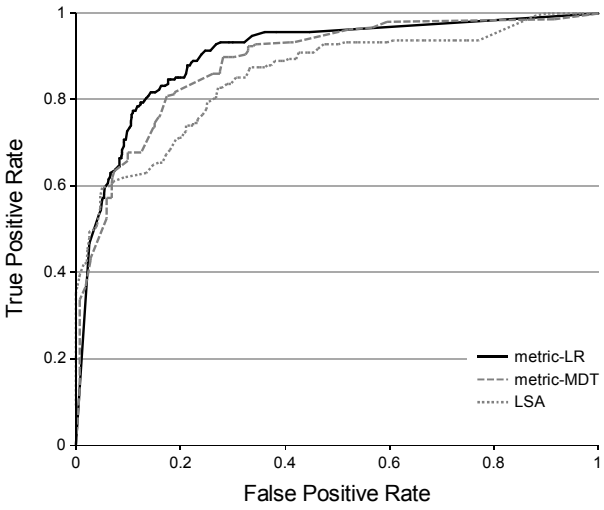

Figure 1. ROC for different similarity measures on the grouping task: trained metrics using logistic regression and a mixture of decision trees (MDT) as well as LSA.

With these features and labels, we can now train any of a number of classifiers to model the similarity function; in Figure 1 we show the performance characteristics as a receiver operating characteristic (ROC) curve for both logistic regression (maxent) and boosted decision trees, as well as the LSA metric for comparison. These were formed via ten-fold cross-validation in which we trained on grouping labels for 9 of the 10 training questions and tested on the 10th; we then swept over all threshold values.

We choose logistic regression due to its slightly stronger performance as well as the fact that its output is calibrated, i.e., the output value represents the probability that a_1 and a_2 are in the same group. This is important as we later use this value in our distance measure for clustering. While the threshold could be tuned for a particular task, the value of 0.5 is meaningful in terms of the probabilistic model, and is what we will use for judgments of similarity.

To understand the relative contributions of various features in the classifier, we trained a set of classifiers using each feature individually; the results are shown in Figure 2. As Mohler et al. (2011) found in their work, the tf-idf similarity feature is a powerful one, as is the letter-based similarity. Overall, though, the classifier trained on all features gives us the most robust performance.

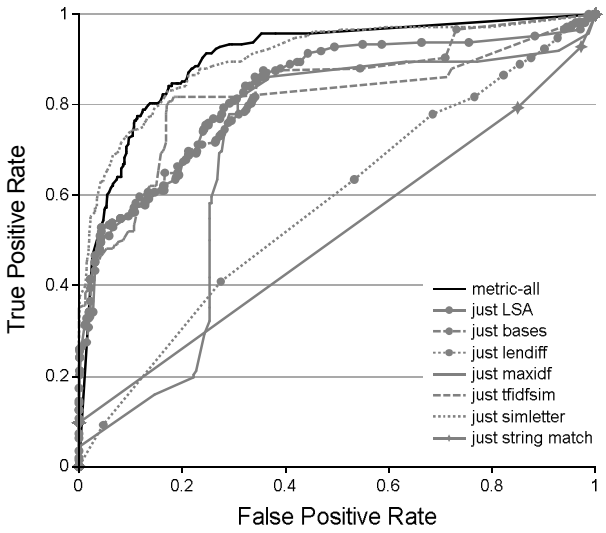

Figure 2. ROC on the grouping task for LR trained on individual features as well as all features.

5 Forming Clusters and Subclusters of Student Answers

To allow the teacher to "divide and conquer," we need to choose a strategy for the division. Initially

we only grouped the answers into a single layer of a small number of clusters (10), but found that this could lead to a great deal of variability within the clusters and made it difficult to get a sense of the range of answers. At the same time, with too many top-level clusters (e.g., 50), it would be necessary for a teacher to take at least 50 actions to grade the whole set, and she could not benefit from large clusters that were consistently correct or incorrect.

We found that forming subclusters within each cluster provided a good compromise and led to the most easily readable results. By having this two level hierarchy, we could maintain high-level groupings and still structure the content within them. Furthermore, teachers would be able to mark a cluster with a label if most items were correct or incorrect, then easily reach in and flip the label of an outlier subcluster. There remains the question of how many clusters and subclusters to use; based on the complement set of questions we found 10 clusters and 5 subclusters to be a good setting. In Section 6.3, we explore a range of values for these parameters to see how much this affects the results.

As for finding the clusters, given our learned similarity measure, we can use metric clustering to group the items into clusters and subclusters. We use the k-medoids algorithm with some small modifications. Alternatively, we could form these clusters and subclusters using the topic assignments from LDA, and explore this as a baseline. Furthermore, if an answer key is available, we can mark some clusters and subclusters automatically based on those answers, both for our metric clustering and for LDA, and develop approaches for each case. With or without these automatic labels, we then measure how much would be gained from each user action in terms of progress on the grading task.

5.1 Similarity-Based Clustering

As we only have distances between items as opposed to coordinates, we must turn to the domain of metric clustering; specifically, we make use of the k-medoids algorithm (Kaufman and Rousseeuw, 1987). We first use the distance metric learned above to form a matrix of all pairwise distances between items D:

$$D_{ij} = 1 - sim(a_i, a_j)$$

The canonical procedure for k-medoids, the Partitioning Around Medoids (PAM) algorithm (Kaufman and Rousseuw, 1990), is then straightforward—we pick a random set of indices to initialize as medoids, then for each iteration we assign all items to the cluster whose medoid it is closest to, and then recompute the medoid for each group by finding which item in each cluster has the smallest total distance to the other items. This process is iterated until the clusters converge.

However, there are a number of subtleties to this procedure. First, as items are generally closer to themselves than any other item, often clusters will "collapse" and end up with the centroid as a single item while other clusters become egregiously large. We address this issue with a redistribution step: if there are any empty or single item clusters, we examine the distribution of item distances to the medoids in the other clusters, and redistribute items from larger clusters if they have become unwieldy. We use the ratio of the mean to median distance to the medoid as a measure of cluster distortion; when it is greater than 1 it is likely that most items have a small distance (resulting in a small median) but there are outliers with large distances (causing the large mean) that could be moved. Second, as our classifier is trained to determine the probability of items being in the same group, we don't expect items with a value of less than 0.5 to be a good fit for the cluster—we thus reserve the final cluster for these "misfit" items. We implement this via an artificial item whose distance to all items is 0.5 and is the medoid of this last cluster; any item with no better match will fall there. These changes result in the modified PAM algorithm in Figure 3.

Modified PAM for k Medoids, N Items

1. Select $k-1$ points as medoids $c_{1..k-1}$
2. Create "artificial item" $N+1$ for last medoid c_k such that $D_{iN+1} = 0.5$
3. Until convergence:
   a. Assign each item to closest medoid
   b. For each cluster $C_s: |C_s| \leq 1$
      i. For each cluster C with medoid c find $r_C = mean(D_{jc})/median(D_{jc}) \ \forall j \in C$
      ii. If there is a cluster $C_m: r_{Cm} \geq r_{C\neg m} > 1$, move items $l: D_{lc} > 2 * mean(D_{jc})$ from C_m to C_s
   c. Recompute medoids for each cluster in $1..k-1$ as $c_q = \arg\min_j \sum_i D_{ij} \ \forall i,j \in C_q$

Figure 3. Modified algorithm for k-medoids clustering.

5.2 LDA Clustering

We make use of the LDA algorithm as our baseline (Blei 2003); clusters are formed by assigning answers to their most probable "topic." As we mentioned, despite its power, the LDA approach suffers from its sensitivity to individual words; the model depends on precisely the same words being used in multiple "documents" (in our case, answers). To reduce the effect of this sensitivity, we applied simple stemming (Porter, 1980) to the words; we found this greatly improved the performance.

5.3 Getting "User Labels" for Clusters

While a user-facing system based on this technology will involve an interactive experience leveraging the strengths of both the human and the algorithm, we wanted to measure how user actions might translate into grading progress. In our model of interaction, there are two "macro" actions the human can take in addition to labeling individual items—label all of the items in a cluster as correct/incorrect, or label all of the items in a subcluster as correct/incorrect. To choose between these actions, we model the human as always picking the next action which will maximally increase the number of correctly graded items. In intuitive terms, this amounts to the user taking an action when the majority of the items in the cluster or subcluster have the same label and are either unlabeled or labeled incorrectly. In order to prevent the undoing of earlier work, clusters must be labeled before subclusters contained within them can have their labels "flipped." When no actions will result in an increase in correct labels, the process is done; the remaining items must be labeled individually.

5.4 Automatic Labels from the Answer Key

When an answer key is available, we have devised mechanisms for both algorithms to automatically perform a subset of the available actions. In the case of our metric clustering method, we can determine the distance D_{ij} between any user answer and any answer key item. We compute the "correctness" of an answer as the maximum similarity to any answer key item. If the average correctness for a cluster or subcluster is greater than the classifier's threshold of 0.5, the set is marked as "correct" and otherwise "incorrect."

In the case of LDA, the model does not allow for computing distances to each item. Instead we add all the answer key items as additional items into the clustering and see what clusters they land in; we then label those clusters as correct. It would be possible to label the subclusters instead, but labeling the entire cluster gives the automatic actions the greatest chance for impact in the LDA setting.

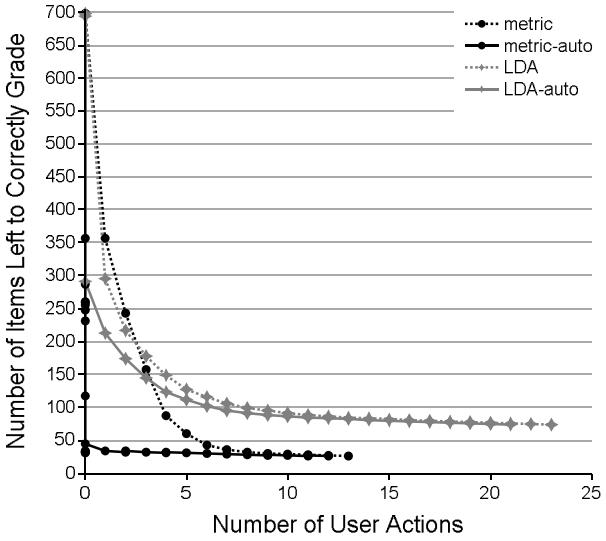

Figure 4. Number of items left to grade or correct out of 698 after each macro user action for G1, question 4.

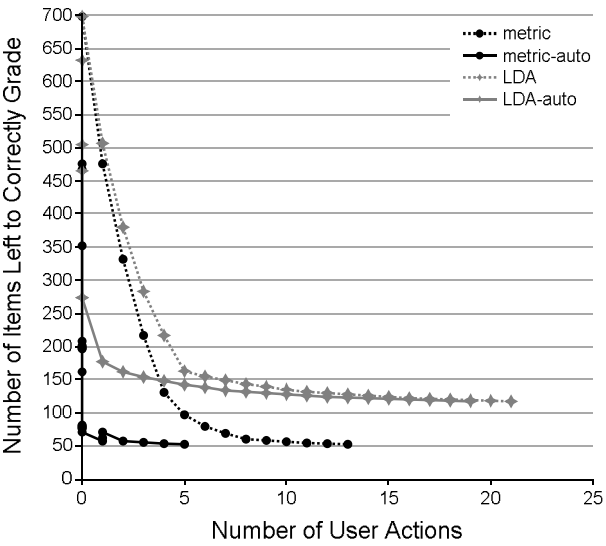

Figure 5. Number of items left to grade or correct out of 698 after each macro user action for G2, question 13.

5.5 Sample Results for Clustering Methods

In Figures 4–5, the benefits of the powergrading approach can be seen for two sample questions in terms of the reduction in work with each user action. Furthermore, when automatic actions are added (metric-auto and LDA-auto), the eventual reduction

	Metric Clustering			LDA Clustering			Metric Individual			LSA Individual			"Yes" Individual		
Q#	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**
1	2	1	2	11	12	11	**1**	**0**	**1**	11	10	11	44	43	44
2	**14**	12	**13**	42	34	38	15	**11**	**13**	38	38	38	86	78	82
3	**80**	**87**	145	110	110	184	103	107	**134**	247	249	242	108	108	203
4	**32**	**26**	**43**	92	80	106	56	59	48	67	62	53	128	121	154
5	**20**	**18**	26	21	19	23	32	23	35	24	19	**19**	40	27	37
6	**24**	**30**	**50**	65	58	71	108	120	138	130	136	146	127	113	147
7	**16**	**12**	**11**	54	50	47	61	57	58	515	517	518	50	46	43
8	**9**	**8**	**8**	207	212	204	14	11	21	**9**	12	16	279	270	286
13	**75**	**54**	50	82	133	121	99	67	**49**	126	64	76	82	160	138
20	**11**	**14**	**10**	47	26	22	37	18	**10**	35	84	70	52	21	17

Table 3. Number of actions left (smaller is better) for each question after automatic actions and three manual actions when an answer key exists, comparing various grading methods for each individual grader (G1–G3).

of work (the final y-axis value) is the same as the manual version, but far less human effort is required to get there. Another way to look at this is to consider a particular point on the x-axis, e.g. 3 user actions, and look at how far this amount of grader effort gets us. In our results for the entire corpus in the next section, we choose such an operating point to provide an overview of performance over all questions, graders, and methods.

6 Results and Discussion

We now turn to computing aggregate results over all of the test data, i.e., the 698 responses to each of the 10 questions from Table 1.

6.1 Grading with the Similarity Measure

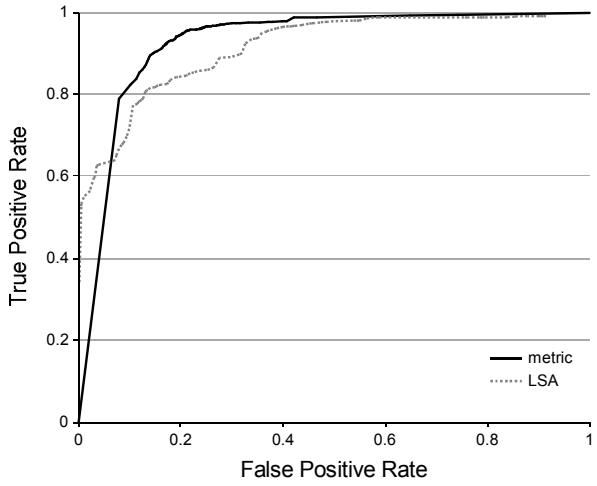

Figure 6. ROCs for the grading of individual items, learned similarity metric vs. the LSA metric.

We begin by considering the performance of our similarity metric for correctly grading individual responses via their similarity to answer key items; we show the ROC along with the performance of the LSA measure in Figure 6. The values were computed over all student responses and grades from all graders. As Mohler et al. (2011) found, the learned metric does improve over the powerful baseline of LSA. It is interesting to note, though, that in our case the classifier was not optimized to mark items as correct or incorrect, but to correctly predict whether a pair of items belongs to the same group.

We computed the overall accuracy on grading individual items with our learned similarity measure (92.9%), the LSA measure at its equal-error rate threshold of 0.7 (82.5%), and marking all answers as correct (84.6%). We also computed significance between these results with paired-t tests, and found all pairs significant at $p < 1e{-}8$.

6.2 Performance on a Budget in Terms of "Actions Left"

To now examine the overall potential of our method for the grading task, we need to think both about the appropriate metric to use as well as appropriate baselines. As we have designed our approach to work in concert with a human grader, maximizing the result of a small amount of human effort, we report results for our method and baselines in terms of the "number of actions left after N manual actions." One can think of this as how far we get if we are "grading on a budget"—after the algorithm has done what it can automatically, and the teacher takes the N best next actions (those resulting in maximal gain of correctly graded items), we compute how many actions (either cluster/subcluster flips or individual rescorings) would be required to complete the task to perfection, where perfection is defined as perfect agreement with a given grader.

The benefit of this measure is that given a set of student responses and corresponding grades (labels), it allows us to quantitatively compare any clustering (or non-clustering) method on this task.

In Table 3, we show these values for each grader (G1–G3) after three manual actions (N=3) for both clustering methods as well as using the individual classifiers—metric, the LSA value alone, and "always-yes," i.e., marking all answers as correct. For most of the questions, our metric-clustering based method requires fewer actions by a large margin—an average of 61% fewer actions than the LDA-based method and 36% fewer actions than the metric classifier operating on individual items. Note that there are some instances in which an individual classifier performs better, but typically only by a small number of additional actions.

	Metric Clustering			LDA Clustering		
Q#	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**
1	**10**	**9**	**10**	12	13	12
2	**22**	**20**	**20**	44	36	40
3	**87**	**96**	**154**	114	114	188
4	**36**	**32**	**46**	94	82	108
5	29	26	33	**24**	**24**	**26**
6	**29**	**30**	**50**	68	61	74
7	**27**	**23**	**22**	58	54	51
8	**16**	**18**	**15**	208	213	205
13	**79**	**62**	**58**	86	135	124
20	**18**	**23**	**21**	49	28	24

Table 4. Number of actions left for metric and LDA clustering after three user actions when no answer key is available.

In Table 4, we examine the case in which an answer key is not available, again seeing how much work is left after the budget of three manual actions. While the numbers are necessarily greater than in Table 3, they are still small compared to the full work of grading 698 answers. Note that the individual classifier or other automatic methods are not applicable here, as there are no known answers to compare against. We expect this is often the case the first time the teacher is giving and grading a test.

6.3 Choosing the Number of Clusters

As we did not wish to optimize on our test set, we based our choice of 10 clusters and 5 subclusters on data from the other questions not used for our evaluation. However, we wished to *post facto* examine what the effect of sweeping these parameters over a range of values would have been for our results. In Table 5 below we show the average number of actions left over all 10 questions and all three graders after auto and manual actions (N=3).

While there is a clear benefit to moving beyond one or two clusters/subclusters, the benefits of adding more clusters seem to plateau in the regime of our chosen settings. Note that we wish to minimize the value for each parameter, since more clusters/subclusters means a larger set of actions for the teacher to choose from. Furthermore, we prefer a smaller number of clusters, as that reduces the *minimum* number of actions, since at best the user could just mark the clusters to complete the task.

Num. Clus.	**Number of Subclusters**					
	1	**2**	**5**	**7**	**10**	**12**
1	107.5	67.3	41.7	39.5	35.4	34.2
5	41.8	40.8	31.6	31.7	30.6	29.8
10	35.5	34.2	30.1	30.4	29.6	30.0
15	33.7	32.8	29.8	29.7	29.9	29.7
20	32.7	32.3	30.5	30.6	31.5	31.1
25	33.8	32.2	31.5	32.1	31.8	31.6
50	33.3	33.4	33.1	33.6	34.4	34.4

Table 5. Average number of actions left across settings of the number of clusters/subclusters after 3 actions. The setting used for our other results is shaded in grey.

6.4 Finding Modes of Misunderstanding

One of the advantages of grouping items into clusters and subclusters is the ability for a teacher to detect modes of misunderstanding in their students, provide them with rich feedback, and potentially revise their teaching materials. We show an example from question 3 in Figure 7 below in which students have confused the Constitution with the Declaration of Independence. With a single short message, the teacher can explain the nature of the students' confusions instead of merely marking their answers as incorrect.

"What did the Declaration of Independence do?"

- Set rules/rights so that the people have rights to stand up too
- gaves everyone rights
- Gave everyone rights.
- Put our rights on paper.
- Give rights to americans.
- Entailed the Bill of Rights

Figure 7. A subcluster of student answers for question 3 exhibiting a consistent mode of misunderstanding.

6.5 Examining the Classification Errors

To better understand the patterns of errors made by the grading metric, we consider cases where all of the human graders agree and the learned similarity metric disagrees—this corresponds to 313 of the 6980 responses (4.5%). 89 of the 313 are false positives, where the metric labels the answer correct while all human graders agree that the answer is incorrect. Analysis of this set shows us examples where additional structure might be useful to the metric. For question 8, e.g., the correct answer is "speaker of the house", comprised of a head noun with a modifier. Looking at the errors suggests that a feature encoding the match between modifiers could provide enough information to label the answers "speaker", "speaker of the senate" and "senate speaker" as wrong because the modifiers, where there are any, do not match. For question 6, one of the correct answers is "Senate and House". Many of the answers labeled incorrectly mention either "senate" or "house" but not both. Taking the coordination of the answer key into account would provide more information to reject these answers as incorrect. While we see the need for modification structure and awareness of coordination, we did not observe any items where a predicate-argument analysis would improve results.

The other 224 out of 313 are false negatives, where the metric labels the answer as wrong while all human graders agree that the answer is correct. In question 3, e.g., the answer key mentions "independence" and "be free from" Great Britain. Student answers include paraphrases such as "separate from", "become our own country", "no longer one of their subjects". It's evident that human graders can recognize these phrases as being closely related in meaning, but it is more difficult to imagine which semantic features will be able to capture these. Incorporating a similarity metric which accesses dictionary definitions (see Lesk, 1986) was found to be informative in Mohler and Mihalcea (2011), and could connect "independent" and "not subject," but would not be sufficient to recognize many of the other paraphrases we observed.

7 Conclusions and Future Work

We have shown how the powergrading approach of dividing and conquering the short-answer grading task can greatly reduce the number of actions required; from another perspective it can greatly extend the impact of a small number of user actions when grading resources are limited. It offers teachers the opportunity to identify modes of misunderstanding among their students and provide rich feedback to groups of students whose incorrect answers have clustered together. Furthermore, this approach is effective when an answer key is not available, but is even more so when a simple list of text answers is provided. It is worth noting that the real power in powergrading is not in the specifics of the features we used, the choice of classifier, or the clustering algorithm. While all of these individual choices could be improved upon, it is the approach of dividing and conquering that leads to a substantial magnification of the progress a teacher can make with a given amount of manual effort.

This magnification is important because of the greater educational value of recall-oriented questions (Anderson and Biddle, 1975); thus far large-scale online courses have shied away from them because of the potential grading expense. We hope this technology may begin to change that view to the benefit of students everywhere.

Of course, in order to apply these results to MOOCs we need to move from 1k answers to 10k or more responses. In order to get a taste for how our methods might scale to such numbers, we examined how the answers and the clusters might evolve with increasing data size. When we considered the number of unique strings among the answer set, the fraction decreased modestly with increasing data, but plateaued at an average of around 40% (Figure 8 below). When we instead used our learned similarity metric and looked at the fraction of items that are classified as having no similar items, it dropped rapidly and flattened out to an average of 2–3% (Figure 9 below). In other words, we can expect 97% of responses to cluster nicely with others, and the rest to end up in our miscellaneous cluster k. As such, we expect even substantially larger datasets can be feasibly processed via our approach.

Finally, there is a clear direction forward from this point, and that is towards the teacher himself. While we have presented evidence for the workings of our algorithmic approach, there are many questions as to how to best surface these capabilities to a human operator, ranging from how clusters and subclusters are displayed to how users provide feedback. Furthermore, users could provide other kinds of feedback such as moving items between clusters,

allowing for the opportunity for relevance feedback; classifier uncertainty could be surfaced to the user, and active learning approaches could be used to ask the teacher for specific labels. We look forward to exploring this next set of human-computer interaction and interactive machine-learning challenges in our future work.

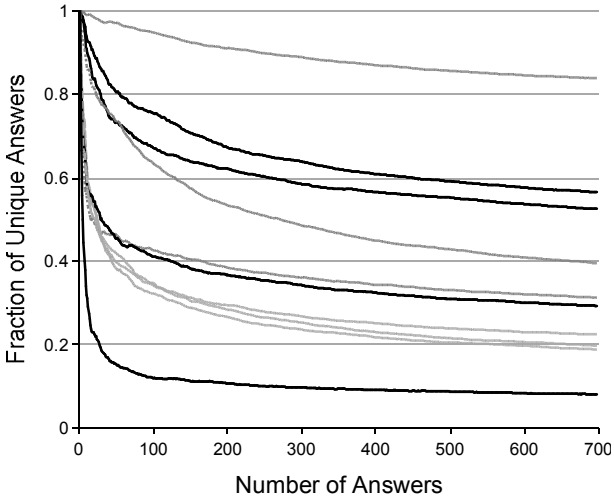

Figure 8. Fraction of answers with no exact match from another student; each curve represents one question, and the x-axis represents the size of the answer set (i.e., class size). Each curve is averaged over 30 reshufflings of the answer set.

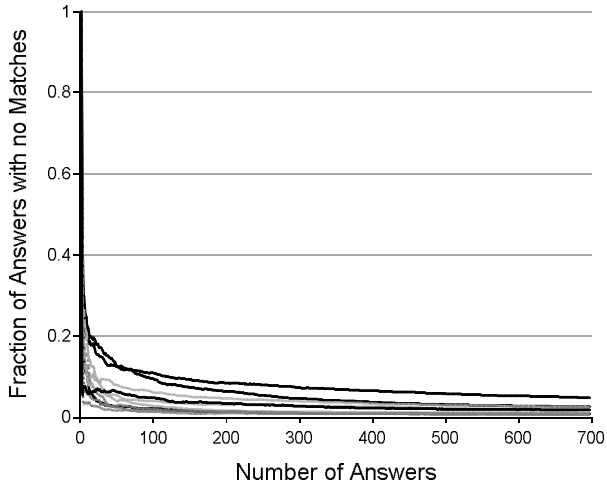

Figure 9. Fraction of answers with no match from another student using the learned similarity metric; each curve represents one question, and the x-axis represents the size of the answer set (i.e., class size). Each curve is averaged over 30 reshufflings of the answer set.

References

Richard C. Anderson and W. Barry Biddle. 1975. On asking people questions about what they are reading. In G. Bower (Ed.) *Psychology of Learning and Motivation*, 9: 90–132.

Yigal Attali, Don Powers, Marshall Freedman, Marissa Harrison, and Susan Obetz. 2008. Automated Scoring of Short-Answer Open-Ended GRE Subject Test Items. *ETS GRE Board Research Report No. 04-02*. ETS RR-08-20.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning*, 3(4–5): 993–1022.

Joanna Bull and Colleen McKenna. 2004. *Blueprint for Computer-Assisted Assessment*. London: Routledge-Falmer.

Jill Burstein, Martin Chodorow, and Claudia Leacock. 2004. Automated Essay Evaluation: The Criterion Online Writing Evaluation Service. AI Magazine, Vol. 25, No. 3, pp. 27–36

Gráinne Conole and Bill Warburton. 2005. A review of computer-assisted assessment. *ALT-J, Research in Learning Technology* 13(1): 17–31.

Phil Davies. 2002. There's no confidence in multiple-choice testing, in: M. Danson (Ed.) 6th International CAA Conference, Loughborough University, 4–5 July 2002.

Susan T. Dumais. 2004. Latent Semantic Analysis. In *Annual Review of Information Science and Technology (ARIST).* 38(4): 189–230.

Wael H. Gomaa and Aly A. Fahmy. 2012. Short Answer Grading Using String Similarity and Corpus-Based Similarity. In International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 3, No. 11, 2012.

Michael Hahn and Detmar Meurers. 2012. Evaluating the Meaning of Answers to Reading Comprehension Questions: A Semantics-Based Approach. In Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA7) . Association for Computational Linguistics. Montreal, Canada.

The Hewlett Foundation. 2012. The Automated Student Assessment Prize Phase 2: Short Answer Scoring. http://www.kaggle.com/c/asap-sas

Sally Jordan and Tom Mitchell. 2009, e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback. *British Journal of Educational Technology*, 40: 371–385.

Selcuk Karaman. Examining the effects of flexible online exams on students' engagement in e-learning. 2011. *Educational Research and Reviews* Vol. 6(3): 259–264, March 2011

Jeffrey D. Karpicke and Henry L. Roediger. 2008. The Critical Importance of Retrieval for Learning. *Science* 319: 966–988.

Leonard Kaufman and Peter J. Rousseeuw. 1987. "Clustering by Means of Medoids." In *Statistical Data Analysis Based on the L1-norm and Related Methods*, Y. Dodge, Ed., pp. 405–416. Elsevier, Amsterdam.

Leonard Kaufman and Peter J. Rousseeuw. 1990. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley and Sons, New York.

Batia Laufer and Zahava Goldstein. 2004. Testing Vocabulary Knowledge: Size, Strength, and Computer Adaptiveness. *Language Learning*, 54: 399–436.

Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated Scoring of Short-Answer questions. *Computers and the Humanities*, 37(4): 389–405.

Detmar Meurers, Ramon Ziai, Niels Ott, and Janina Kopp. 2011. Evaluating Answers to Reading Comprehension Questions in Context: Results for German and the Role of Information Structure. In *Proceedings of the TextInfer 2011 Workshop on Textual Entailment*, EMNLP 2011.

Tom Mitchell, Terry Russell, Peter Broomhead, and Nicola Aldridge 2002 Towards robust computerized marking of free-text responses. In *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough University, Loughborough, UK.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the European Association for Computational Linguistics* (EACL 2009). Athens, Greece.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. In *Proceedings of the Association for Computational Linguistics* (ACL). Portland, Oregon.

Rodney D. Nielsen, Wayne Ward and James H. Martin. 2009. Recognizing entailment in intelligent tutoring systems. *Natural Language Engineering*, 15(4):479–501.

Michael F. Porter. 1980. An Algorithm for Suffix Stripping. Program: Electronic Library and Information Systems, 14(3): 130–137.

Stephen.G. Pulman and Jana Z. Sukkarieh. 2005. Automatic Short Answer Marking. In *Proceedings of Building Educational Applications using NLP*, an ACL workshop.

Chris Quirk, Pallavi Choudhury, Jianfeng Gao, Hisami Suzuki, Kristina Toutanova, Michael Gamon, Wentau Yih, Lucy Vanderwende and Colin Cherry. 2012. MSR SPLAT, a language analysis toolkit. *In Proceedings of NAACL 2012*.

Benjamin Storm, Robert Bjork, and Jennifer Storm. 2010. Optimizing retrieval as a learning event: When and why expanding retrieval practice enhances long-term retention. *Memory & Cognition* 2010, 38(2): 244–253.

U.S. Citizenship and Immigration Services (USCIS). 2012. "100 Civics Questions and Answers (English version)." http://www.uscis.gov/USCIS/Office%20of%20Citizenship/Citizenship%20Resource%20Center%20Site/Publications/100q.pdf

Bill Warburton and Gráinne Conole. 2003. CAA in UK HEIs: the state of the art, in: J. Christie (Ed.) 7th International CAA Conference, University of Loughborough, 8–9 July 2003.