Classification of open-ended responses to a research-based assessment using natural language processing

Joseph Wilson,[1] Benjamin Pollard,[1,2,3] John M. Aiken,[4,5] Marcos D. Caballero,[5,6] and H. J. Lewandowski[1,2,*]

[1]*Department of Physics, University of Colorado Boulder, Boulder, Colorado 80309, USA*
[2]*JILA, National Institute of Standards and Technology and the University of Colorado, Boulder, Colorado 80309, USA*
[3]*Department of Physics, Worcester Polytechnic Institute, Worcester, Massachusetts 01609, USA*
[4]*The Njord Centre, Departments of Geosciences and Physics, University of Oslo, 0316 Oslo, Norway*
[5]*Center for Computing in Science Education and Department of Physics, University of Oslo, 0316 Oslo, Norway*
[6]*Department of Physics and Astronomy, Department of Computational Mathematics, Sciences, and Engineering and CREATE for STEM Institute, Michigan State University, East Lansing, Michigan 48824, USA*

(Received 16 June 2021; accepted 14 April 2022; published 2 June 2022)

Surveys have long been used in physics education research to understand student reasoning and inform course improvements. However, to make analysis of large sets of responses practical, most surveys use a closed-response format with a small set of potential responses. Open-ended formats, such as written free response, can provide deeper insights into student thinking, but take much longer to analyze, especially with a large number of responses. Here, we explore natural language processing as a computational solution to this problem. We create a machine learning model that can take student responses from the Physics Measurement Questionnaire as input, and output a categorization of student reasoning based on different reasoning paradigms. Our model yields classifications with the same level of agreement as that between two humans categorizing the data, but can be done by a computer, and thus can be scaled for large datasets. In this work, we describe the algorithms and methodologies used to create, train, and test our natural language processing system. We also present the results of the analysis and discuss the utility of these approaches for analyzing open-response data in education research.

DOI: 10.1103/PhysRevPhysEducRes.18.010141

I. INTRODUCTION

Surveys and questionnaires have long been an integral tool for physics education researchers to improve physics courses and understand student thinking. They offer a way to quantify or categorize aspects of students' reasoning, their proficiency at a task, and their attitudes, beliefs, or epistemologies around a topic among other dimensions important for student learning [1,2]. By doing so in a standardized way, surveys allow researchers to measure proportions of students within a group, track how these proportions change over time, and observe correlations that can suggest relationships between student characteristics. Because their standardized assessments aim to be representative and generalizable, survey studies usually involve collecting a large number of responses. To make analysis practical, surveys used in this way are typically designed using a closed-response format, with a predetermined and manageably small set of potential responses. However, closed-response surveys are limited in the fullness, depth, and subtlety of student understanding they can measure. A comprehensive overview of the role of surveys in physics education research, as one aspect of quantitative methods more generally, can be found in Ref. [3].

In contrast, open-ended formats (such as written text responses) can provide deeper insights than closed-response data, as students are not limited to a given set of responses. For example, by analyzing students' own words, we can potentially achieve a greater sense of not only what their answer is, but how they arrived at it. Unfortunately, though, open-response data is difficult and time consuming to analyze with the large number of responses that are often necessary to make representative and generalizable claims.

*lewandoh@colorado.edu

Natural language processing (NLP) and machine learning could be a computational solution to this problem in some cases. Statistical models built around the words students use in their written open responses can be used to quantify or categorize these responses, achieving the same outcome as a closed-response survey with a still-manageable level of effort. Language processing and analysis tools like NLP are increasingly being used in educational settings to analyze student work [4–11]. A machine learning system can find patterns in different types of responses, converting students' words into lists of numerical values that represent different characteristics (features) of a response. Subsequently, a computational model could be created and fit to those features, known as training the model, to produce an overall quantification or categorization of the response. These techniques represent a systematic and computational approach to classifying a student's written response into one of several relevant categories. Once a well performing system has been created based on a set of precategorized data, one can predict classifications for any sufficiently similar response using the same model.

In this work, we use a large dataset of precategorized written responses to open-ended questions from a research-based assessment survey to create and test a NLP system for subsequent quantitative data analysis. We use data from students in the large introductory physics lab course at the University of Colorado Boulder (CU) who completed the Physics Measurement Questionnaire (PMQ), an assessment tool for studying student reasoning around measurement uncertainty [12]. Our goal of this work is not to evaluate student learning in this environment or test how machine learning methods can be used generally for open-ended assessments. Rather, we want to explore if we are able to automate the scoring process for the PMQ survey as a proof-of-principle study using NLP techniques. To be able to do this, we aim to answer the following research question:

- What level of performance can be achieved using natural language processing on open-ended assessment responses for the PMQ?
- How does our NLP system compare to a human performing the same task?

In this paper, we present the background needed to put this work in context, including the course context and the PMQ survey itself, as well as the process of human coding the data. We then discuss, in detail, the methods we used for creating and testing the NLP models and the outcomes of those models applied to our data. Finally, we present a discussion of the outlook for using NLP methods for analysis in physics education research.

II. BACKGROUND

A. Course context

The data for this work came from students in the introductory physics lab course at CU. This course is a stand-alone, large-enrollment class that students typically take during their second semester of study, and is almost always the first physics lab course they have taken at CU. The course is typically taken concurrently with an introductory physics theory course on electricity and magnetism, with students having already taken an introductory course on mechanics. The lab course consists of weekly two-hour lab activities involving these topics at an introductory level. Students are graded based on work produced from these activities and on participation in 5–6 supplemental lecture sessions; there is no midterm or final exam. Self-reported information regarding students' gender, race, ethnicity, and major from Spring 2018, a representative semester of the course, is shown in Table I (reproduced from Ref. [13]).

TABLE I. Self-reported gender, race, ethnicity, and major of students enrolled in the course in Spring 2018. There are around 1200 students in total included in this work.

Women	25.6%
Men	72.7%
Gender nonconforming	1.7%
American Indian or Alaska Native	0.7%
Asian American	15.3%
Black or African American	2.3%
Hispanic or Latino	9.8%
Native Hawaiian or other Pacific Islander	0.4%
White	67.1%
Other race or ethnicity	4.3%
Physics or Engineering Physics	17.9%
Other Engineering	43.0%
Other STEM	35.1%
Other disciplines	4.0%

Between 2016 and 2018, the course underwent a transformation led by H. J. L., the lead instructor throughout that period. In many respects, the course before and after the transformation can be thought of as two completely different courses. Prior to the transformation, the lab activities involved mostly measuring a value that was already known to students from their theory courses, and the supplemental lectures focused largely on error propagation. After the transformation, the lab activities involved making predictions of unknown parameters based on preliminary measurements and then testing those predictions. Lectures after the transformation de-emphasized error propagation, focusing instead on basic concepts of measurement uncertainty, such as distributions, standard deviation, and standard error of the mean. More details about the transformation can be found in Refs. [13–15].

Throughout the transformation process, researchers collected data to measure the impact of the course and the transformation. In addition to focus group interviews and written responses [14,15], research-based assessment instruments were used to quantify learning before and after the transformation [13,16–18]. They were administered as

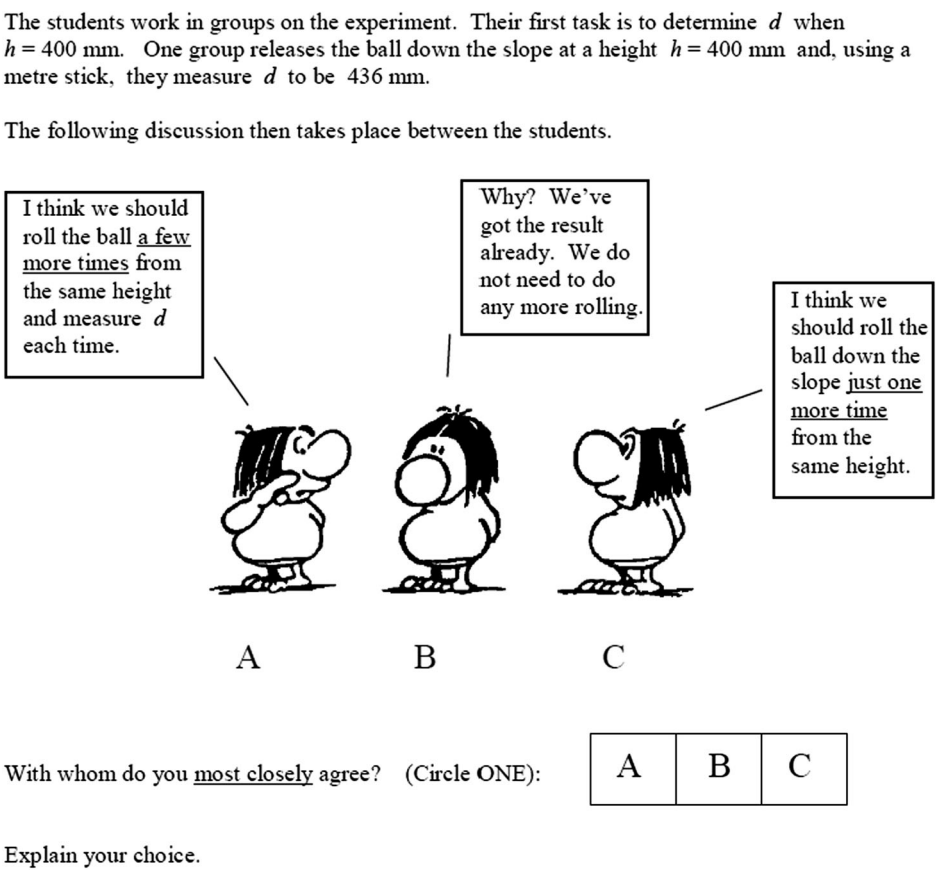

FIG. 1. The RD probe of the PMQ. Reproduced from Ref. [23].

pretest and post-test in semesters in which the original course and the transformed course were taught. The students completed them online either during a lab session or outside of lab, depending on the semester, and received a small amount of course credit for completion.

Here, we focus on data from one of these surveys, the Physics Measurement Questionnaire, which is described in the next section. We use a combined dataset from the pre- and post-tests from Spring 2017 and Spring 2018. In this work, we do not distinguish between data from pretest vs post-test, or from the original vs the transformed course, as we are focused on creating an automated system to categorize responses from students more generally.

B. Physics Measurement Questionnaire

The PMQ is a research-based assessment tool for measuring student reasoning around statistical measurement uncertainty in introductory physics lab courses. It was developed at the University of Cape Town, ZA [18–22] for first-year university students at that institution. Subsequent work by some of us established its utility for students at CU [16,17].

The PMQ consists of a set of survey questions, or probes, relating to an experiment in which a ball is rolled down a slope, travels in free-fall, and lands a measured distance away from the end of the slope. Each probe concerns a different aspect of measurement uncertainty, such as data collection, data processing, or data comparison. In this work, we focus on four of these probes: RD (repeating distance, concerning data collection), UR (using repeats, concerning data processing), SMDS (same mean different spread, concerning data comparison), and DMSS (different mean same spread, also concerning data comparison). As an example, the RD probe is shown in Fig. 1. For each probe, students are presented with a choice and asked to select one of several possible responses. They are then asked to explain their choice in an open-response format. Their choices and the text of their explanations, together for each probe, compose the response data collected by the PMQ.

In addition to the probes themselves, the creators of the PMQ developed two paradigms for interpreting data from the PMQ [20,21]. They are the pointlike paradigm and the setlike paradigm, with the setlike paradigm tending to be more aligned with a probabilistic approach to measurement uncertainty and characteristic of expertlike reasoning. In the setlike paradigm, multiple measurements form a distribution, with each data point yielding more information about the underlying measurand, but never yielding its true value without uncertainty. On the other hand, the pointlike

TABLE II. Human-to-human comparison of Cohen's kappa for IRR on a sampling of data from each of the four probes of the PMQ, where RD stands for "repeating distance," UR stands for "using repeats," SMDS stands for "same mean different spread," and DMSS stands for "different mean same spread".

RD	UR	SMDS	DMSS
0.65	0.90	0.67	0.71

paradigm attributes variation between data points to errors or mistakes. It maintains that if all of these influences are eliminated, a single measurement could yield the true value of the measurand.

To analyze responses collected at CU to the four probes in the PMQ identified above, several of us and others developed a new coding scheme based around the point and set paradigms. It was initially based on the coding scheme developed by the PMQ creators using responses from students in Cape Town, but was expanded and then consolidated to better represent responses from CU students. That process is described in detail in Ref. [18]. The coding scheme consists of a separate set of 12–16 codes for each of the four probes [24]. Each code is associated with either the point paradigm, the set paradigm, or an undefined classification. The undefined code is used if the response could not be classified as either set or point in its entirety or if there was not sufficient information to characterize the underlying reasoning into point or set. In this work, we analyze results only at the level of these three paradigm classifications, losing the finer distinctions made by the individual codes themselves.

As a final step in developing the coding scheme for PMQ responses at CU, two researchers independently coded a subset of data from a semester of the introductory lab course at CU (Fall 2017) and compared their results as a measure of interrater reliability (IRR). Cohen's kappa [25] was used to compare the categories assigned by the two researchers. The resulting values are presented in the Table II. These values act as a benchmark for the automated system developed here.

After developing the coding scheme and establishing its reliability with a subset of the data, one of the researchers involved in the IRR process used the codebook to categorize responses from two semesters of data from the introductory lab course at CU, one before the course was transformed and one after the transformation. For each of these semesters, Spring 2017 and Spring 2018, the PMQ was administered twice, as a pretest and as a post-test. These four administrations were combined into a single human-coded dataset of about 2450 responses. (Descriptive statistics on the responses can be found in the Appendix.) That dataset, and the corresponding human-assigned coding classifications, forms the data used in this work.

The results of the human coding are presented in Fig. 2, where the distribution of responses categorized as setlike, pointlike, or undefined is shown for each probe. For all of the probes, the responses are not distributed evenly across the categories. Instead, we see a smaller number of pointlike responses on the RD probe, while setlike and undefined responses both make up about 40% of the data each for that probe. The UR probe has so little variance in the paradigm of the response that we did not consider it moving forward. The SMDS and DMSS probes have a majority setlike response, and neither have very many undefined responses, however, there is enough variance that we can build a classifier for those probes.

The dataset we are using comes from the RD, SMDS, and DMSS probes, as described above. Since each probe is measuring different measurement concepts, the pointlike and setlike paradigms manifest themselves via different words and phrases for each probe. For example, a student employing setlike reasoning would likely mention the *spread* of the data on the SMDS probe, but that same student could reasonably not use that word on the RD probe, and still have setlike reasoning. Therefore, each probe has its own patterns in the data, and requires a separate classification system to be built for it. The systems

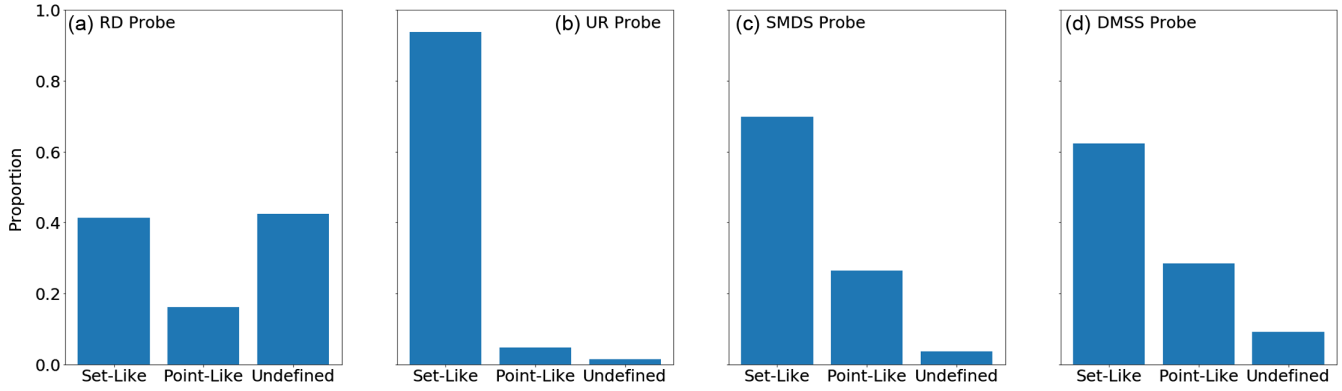

FIG. 2. Distribution of student responses classified as set, point, and undefined for each probe for the entire human-coded dataset. The UR probe responses are almost all setlike and therefore this probe has been omitted from the NLP study.

for each probe are built using the same methods, but are trained on data specific to that probe. This is consistent with how the original human coder classified the data.

C. Automated review of student responses

Until fairly recently, reviewing student responses to open-ended questions required dedicated instructors to read and to mark each response. The automated review of students' responses has grown as the technology to process those responses has advanced [26,27]. PYTHON's Natural Language Toolkit [28] and R's tm package [29] are widely used open source tools for text mining and analysis. Now, we can use a variety of computer algorithms to process answers that students produce in written (typed) form [4–8,11], with more recent work in science education making use of types of the open-source tools presented here [9,10].

More recently, questions have been raised around the use of approaches like NLP on student work. In their comprehensive review on the subject of artificial intelligence in higher education, Zawacki-Richter *et al.* [30] discussed several issues that need to be addressed for future work using AI including critical reflection of challenges and risks of using AI in education and a weak connection of the use of AI to theoretical educational perspectives. They called for further exploration of the use of AI in which these issues are addressed directly.

There is evidence of wider use of text mining and NLP in education, which include analysis of both student work [4,7,11] and the research literature [9,10]. Importantly, these examples use both supervised (i.e., known coded data) and unsupervised (i.e., emergent analysis) approaches. We limit our discussion to work that focuses on classifying text into themes—fully acknowledging that there are other uses of text mining in educational spaces. Nehm and colleagues analyzed written essays by biology students on issues of natural selection [4]. Here, Nehm built from their prior work including an instrument that elicited a collection of student ideas on natural selection [31,32]. In this work, Nehm *et al.* used SPSS's text extraction and modeling tools. Ullmann analyzed essays written by students using a framework for reflective writing [7]. Students' coded written work was mined to determine the presence of eight dimensions of reflective writing. Here, Ullman had each sentence coded by a human for the presence of the eight dimensions and, later, compared the automated analysis to the human coding. Ullman was able to reliably find five of the eight dimensions in student writing and, with less confidence, the remaining three. Wulff and collaborators built off the work of Ullman to analyze the reflections of preservice physics teachers [11]. They adapted Ullman's framework and searched for reflective elements in preservice physics teachers' essays—similarly to Ullman—with reasonable success. In contrast to the focus on student work above, Odden and colleagues mined the text from the last decade's worth of the Proceedings of the Physics Education Research Conference [9]. In their unsupervised approach, they were able to extract 10 themes and trace their prevalence over time. Their work was extended to the journal Science Education where they mined the last century's worth of articles to extract over a dozen themes in three topical areas [10]. They then discussed their historical prevalence of various themes as they related to different movements in science education.

Our work represents these kinds of explorations where we have used NLP in a physics laboratory context. In each of the prior studies, the specific context and questions were critical to forming the approach and discussing the resulting evidence—our work is no different in that respect. In addition, the assessment that we have used (Sec. II B) is grounded in physics education theory, in particular, point-like and setlike reasoning [19]. Finally, as conscientious educators ourselves, we grappled with the educational and ethical concerns that Zawacki-Richter *et al.* [30] discussed. We present more on those concerns and implications in Sec. V B.

III. METHODS

We divided our approach to analyzing responses to the PMQ into four parts, as shown in Fig. 3. We describe the preprocessing that we did on the data in Sec. III A. After that preprocessing, we applied natural language processing (NLP) techniques, as described in Sec. III B. The results from NLP acted as the inputs to our machine learning system, which we describe in Sec. III C. Lastly, we describe the approaches and metrics we took to validate the machine learning output in Sec. III D.

A. Preprocessing

The first part of the process, called preprocessing, aimed to ensure that the human-analyzed dataset was consistent and clean.

To clean the hand-categorized data, we first removed any punctuation from the student responses and converted the text to lower case, so that words beginning with a capital letter would be treated the same as their lower case counterparts. We also removed all stop words from each response [33]. A stop word refers to the most common words in a language, such as *a*, *the*, and *of*. These words do not convey significant meaning, and would likely show up in a student's response regardless of which paradigm they employ. As there is no universally agreed upon list of stop words, we chose to use the list that comes in the Natural Language Toolkit (NLTK) package in PYTHON [28] to remove stop words from our responses.

Our preprocessing also included lemmatization. Often in natural language, the same word can be used in different forms, making it difficult for the computer to identify the

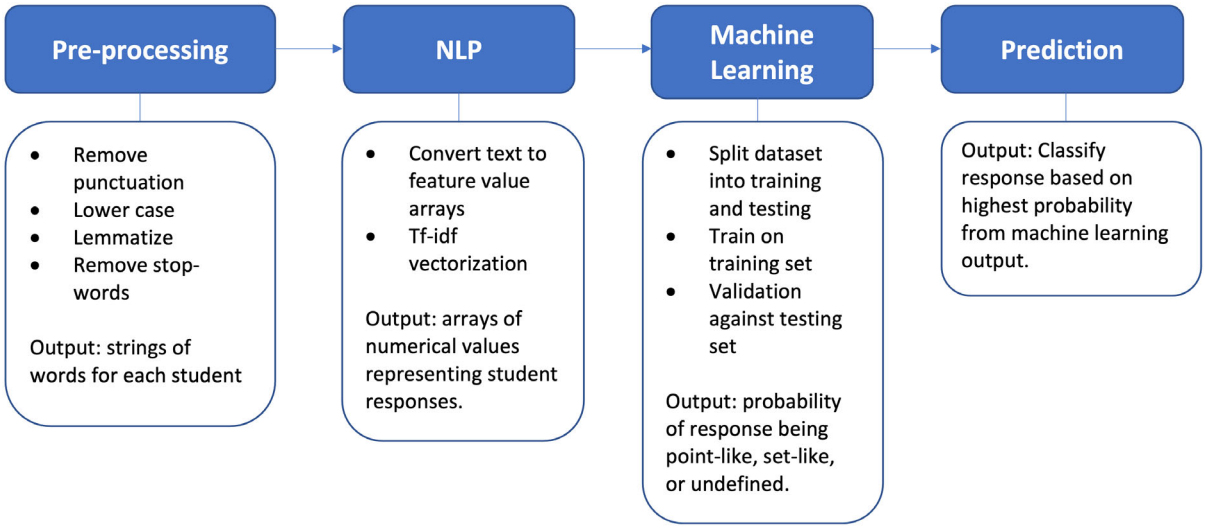

FIG. 3. A diagram of the process of analyzing student responses to the PMQ. The raw responses are preprocessed and then vectorized using NLP. These feature vectors are the input to the machine learning algorithm. The algorithm outputs probabilities that are used to predict in which category a response belongs.

common meaning across these forms. For example, a word can be used in different tenses or plural forms like *average*, *averaged*, and *averages*. Lemmatization is a technique for addressing this issue by returning all words to their lemma, or dictionary headword. In the above example, all instances of the word would be changed to *average*. We performed lemmatization on our data using the `lemmatize` function from NLTK [28].

B. Natural language processing

The next step was to convert the dataset of written responses into a vectorized representation consisting of numerical features. To convert student responses to feature vectors, we used a common NLP technique called term frequency, inverse document frequency (TF-IDF) vectorization [34]. In our case, a document refers to a single student's response to a PMQ probe, and a term refers to a preprocessed word that appears in a response. TF-IDF values each word proportionally to how many times it appears in a particular document, and inversely proportional to how many documents in which it appears. This approach results in words that are common in all responses having a low value, but words that appear many times in only a few responses having a high value. TF-IDF gives each word, w, a weight in a given document, d. The TF-IDF value is calculated using the following procedure. First, we find the fractional occurrence of a word, w, in a document, d:

$$\mathrm{TF}(w,d) = \frac{\text{occurrences of } w \text{ in } d}{\text{total number of words in } d}.$$

Here, $\mathrm{TF}(w,d)$ is this fractional occurrence. Thus, $\mathrm{TF}(w,d)$ is always less than 1 (except in the extreme case that a given w is the only word used in a document, d). $\mathrm{TF}(w,d)$ is then used to compute the inverse document frequency, $\mathrm{IDF}(w)$:

$$\mathrm{IDF}(w) = \log\left[\frac{N}{df(w)}\right],$$

where N is the total number of documents (student responses) in the data corpus and $df(w)$ is the number of documents that contain a given word (w). Because the number of documents containing w is less than the total N, the logarithm always produces a number between 0 and 1 (again, except in the extreme case where w appears in every d). Finally, we can compute the TF-IDF value for a given word appearing in the data by multiplying the fractional occurrence, $\mathrm{TF}(w,d)$, by the inverse frequency, $\mathrm{IDF}(w)$:

$$\text{TF-IDF}(w,d) = \mathrm{TF}(w,d) \times \mathrm{IDF}(w).$$

Because $\mathrm{TF}(w,d)$ and $\mathrm{IDF}(w)$ are positive fractions, TF-IDF vectorization produces a vector with elements bounded between 0 and 1 that represent each word in the data corpus.

In this work, since a feature array consisted of a value for every word used in the training data for that probe, excluding stop words, the RD, SMDS, and DMSS feature arrays were all slightly over 1000 elements long. However, for any given document, most of these feature values were zero, since only a handful of those words were used in an individual student response.

1. One-hot encoding of multiple choice

Because each PMQ response also contained a multiple choice answer that was relevant in interpreting the meaning

TABLE III. AUC of logistic regression model composed of features including a one-hot encoding of multiple choice (MC) and excluding a multiple-choice feature.

RD Probe			
	Setlike	Pointlike	Undefined
No MC	0.93	0.89	0.84
Including MC	0.94	0.91	0.88
SMDS Probe			
	Setlike	Pointlike	Undefined
No MC	0.94	0.94	0.66
Including MC	0.98	0.99	0.70
DMSS Probe			
	Setlike	Pointlike	Undefined
No MC	0.89	0.91	0.81
Including MC	0.91	0.92	0.82

of the response, we created a one-hot encoding (sometimes referred to as using *dummy variables*) of the multiple choice responses. A one-hot encoding, in this context, was an array of the same length as the number of multiple choice options a probe had. All values of this array were 0 except for the one corresponding to the student's choice; that array value was 1. For example, if a student answered choice B when the options were A, B, or C, that one-hot encoding would be $[0, 1, 0]$. We concatenated the one-hot encoding array to the feature vector of each response, thus including in the full feature vector information from a student's written explanation and from their multiple choice selection. Ultimately, the impact of including the multiple choice responses was minimal, but always better (as measured by AUC, Sec. III D) when included in the model as shown in Table III.

C. Machine learning

In order to create a supervised machine learning model, that is, one that learns on a set of precategorized data, we split our dataset into a *training set* and a *testing set* as recommended by Aiken *et al.* [35]. The *training set* consisted of data that the model (for this work: 80% of total) learned on using predetermined category labels to recognize patterns in the data. The *testing set* consisted of a smaller portion of the data (20% of total in this case) that the model classified without knowing the predetermined label, which it did after it had been trained on the training set. Using the predetermined labels in the testing set, we compared the model's predictions with these "true" labels to evaluate the model's performance.

We initially built several different types of machine learning models, including random forest [36], K nearest neighbors [37], and logistic regression. Our initial results from these three types of machine learning models showed that logistic regression was the most effective in this work, and thus we used that approach for the rest of our analysis.

Briefly, random forest methods involve an ensemble of decision trees, with each one making a series of decisions based on the values of an input feature vector. An individual decision tree can represent complex and irregular relationships, however it is prone to overfitting. An ensemble of such trees can mitigate that overfitting. By contrast, K nearest neighbor algorithms classify an input feature vector by defining a distance metric that calculates how far that vector is from other examples. The algorithm selects a group of K examples that are closest to the input vector, and uses their classifications to classify the input. While more straightforward and often easier to interpret than random forest approaches, K nearest neighbor algorithms can be overly sensitive to any peculiarities in the training data. By comparison, logistic regression is perhaps simpler than either of these other approaches, relying closely on linear regression techniques. We briefly describe logistic regression in the remainder of this section.

Logistic regression is a common approach to model processes that lead to a binary outcome ($Y = 1$ or $Y = 0$), In this model, the effect of a particular feature is modeled as a sigmoid:

$$\phi(z) = \frac{1}{1 + e^{-z}}.$$

In this model, z represents the product of the coefficient and the feature (i.e., $z = \beta X$). The resulting probability outcome $\phi(z)$ is modeled as a sigmoid, where above a certain threshold the probability of the outcome rises quickly and then saturates. We can generalize the definition of z for multiple features, i.e., multidimensional logistic regression. In the case of multidimensional logistic regression z then becomes a vector of feature values or coefficients β_i and the product of the data x_i. Additionally, we can cast the problem as seeking the probability, p, of finding the outcome of $Y = 1$. Mathematically this is given by

$$\log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n, \quad (1)$$

where $x_1, x_2, \ldots, x_n$ are the features and the β are the coefficients. Under this formula, logistic regression has a similar form as linear regression.

By rearranging, we can find the odds:

$$\text{odds}(x_1, x_2, \ldots, x_n) = \frac{p}{1-p} = b^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}, \quad (2)$$

where b is traditionally the natural base, e. From either of these formulations, we can determine the likelihood function from which the log likelihood can be determined.

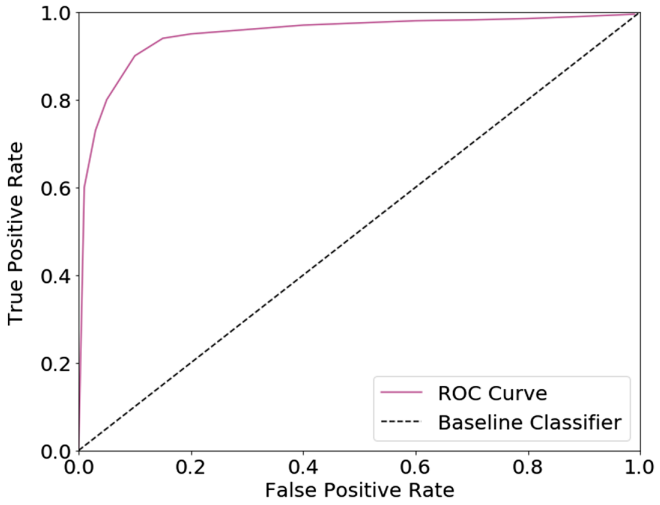

FIG. 4. An example ROC curve. This curve represents a model that is performing well, as it is to the upper left-hand corner. The plot suggests that there is a threshold of probability to classify something as the positive category that will result in a high true positive rate, and a corresponding low false positive rate. For comparison, the dashed line represents the expected ROC curve for a random classifier.

In most cases, a logistic regression model seeks to maximize this log likelihood using various forms of optimization. It is also possible to introduce penalization to the model where features that contribute little to the model are reduced to near zero (Ridge) or set to zero (Lasso). Such penalization is used when you may seek a reduced model—one with fewer input features. Typically, such odds ratios are reported as a measure of the influence of a given feature on the model.

D. Model validation

Before interpreting the classification results of our model, we measured how well our model was performing in a process of model validation. We employed a measure referred to as an ROC curve [38]. An example ROC curve is shown in Fig. 4. This curve is a plot the true positive rate, the proportion of responses that were classified as the "positive" category that actually were positive, versus the false positive rate, the proportion of responses that were classified as the "negative" category that were actually positive. The ROC curve shows this value for many different thresholds of sufficient probability to classify something as positive. ROC curves are useful in cases where there are unbalanced classes such as this case [39]. A curve that approaches the top left corner represents a model that achieves a high true positive rate with a low corresponding false positive rate. Such a situation would suggest a model that performs well. From plots of ROC curves, we measured quantitatively how well the model is performing by calculating the area under the curve (AUC). An AUC value closer to 1 represents a well-performing model. An AUC value of 0.5 corresponds to random guessing.

As another method to optimize and test the performance of our model, we used hyperparameters, parameters that are set before training that govern how the logistic regression algorithm behaves and converges. This approach is a standard one, which we have used in prior work [35,40,41]. Details on tuning hyperparameters can be found in those references. To find the optimal hyperparameters, we performed a grid search, constructing many models given predetermined options for each hyperparameter. These outputs determined which model was optimal. Our grid search returned the optimal regularization and maximum iteration values for our logistic regression model. Regularization helps the model from overfitting on the data and classifying based on small patterns seen only in the training set [42]. The maximum iteration value dictates the maximum number of weight adjustments the logistic regression algorithm can perform before converging.

After we set the hyperparameters, we assessed how our model compared to the baseline of random guessing. The resulting hyperparameters from the grid search included the regularization value of 1.0, the maximum number of iterations the model will perform when training is 500, and the tolerance for stopping criteria of 0.0001. Our search for hyperparameters used Monte Carlo validation where the data are randomly split (80% training and 20% testing) and resampled to develop a distribution of possible model performances. We use this approach to compare the quality of models with different hyperparameters and determine the quality mode to carry forward.

To determine how well our analysis was performing, we randomized the labels for all responses before constructing the model. We constructed 100 models with different randomizations of the labels, and one model with the correct labels. Since the labels were randomized, this method eliminated the true patterns in the data. Instead, we expected our model to have difficulty finding patterns that lead to accurate classifications. Thus, we would expect to see performance metrics much lower for the randomized data than for our true labels.

Lastly, we compared our model's output to a second human's categorization of the dataset. This allowed us to compare the level of interrater reliability between two humans, the original coder and the second one, to the level of reliability between our model and the first human. We used Cohen's kappa a measure of interrater reliability. This comparison indicated how close the performance of our automated system was to that of a human. All of our analysis codes are available online [43].

E. Noncharacteristic undefined scheme

To better represent the notion behind the *undefined* classification, we determined that a response should only be classified as undefined if it is not sufficiently setlike or

TABLE IV. Noncharacteristic undefined scheme threshold values. Each value represents how much the pointlike and setlike probabilities of a response must differ in order to be classified as pointlike or setlike, as opposed to undefined.

RD	SMDS	DMSS
0.353	0.682	0.366

pointlike, or if it is conceivably both setlike and pointlike. To implement this idea, we considered the decision probabilities of the logistic regression function for *setlike* and for *pointlike*. If the setlike probability was over a certain threshold, or if the pointlike probability was over that threshold, then the response was classified as such. However, if both probabilities were under the threshold, or if both were over it, then the response was classified as undefined. This approach avoided basing the classification of undefined on a set of characteristics of an undefined response, which would go against the notion of being undefined. Here, we call the approach of classifying *undefined* responses in this way the noncharacteristic undefined (NCU) scheme, and the previous approach of treating each classification separately the *one-vs-all* scheme.

More specifically, when a response was given a classification by the logistic regression algorithm, we identified the probability associated with each possible class for that response. The response was then assigned a score that indicated the difference between the pointlike probability and the setlike probability. By sorting the classified responses by this score, we can see how a classification of *undefined* given that score would change the true positive rate and false positive rate of our model. This allows us to construct an ROC curve from that true positive rate and false positive rate data.

The score thresholds that optimize this ROC curve for each probe are shown in Table IV. The number represents the amount that the pointlike and setlike probabilities must differ in order for a response to *not* be classified as *undefined*. For more details on the implementation of the NCU scheme, we point the reader to the open-source code on Github [43].

IV. RESULTS

A. Model evaluation

Table V shows the results from three of our initial models based on three different metrics: AUC, accuracy, and Cohen's kappa. We define accuracy as the proportion of responses our system classified correctly based on the true labels from the human coder. In each case, logistic regression performs similarly or better than the other three models, suggesting that it is the most effective model to use for this particular investigation.

TABLE V. Comparison of three machine learning models across the RD, SMDS, and DMSS probes of the PMQ. The three models are logistic regression, random forest, and *K* nearest neighbors. Comparison metrics are the AUC values from, respective, ROC curves, accuracy (the proportion of correctly classified responses), and Cohen's kappa values. Here, Cohen's kappa is comparing the human coder to the model coding.

RD Probe			
Model	Average AUC	Accuracy	Cohen's kappa
Logistic regression	0.907	0.778	0.634
Random forest	0.843	0.768	0.621
K neighbors	0.853	0.658	0.419
SMDS Probe			
Model	Average AUC	Accuracy	Cohen's kappa
Logistic regression	0.880	0.918	0.823
Random forest	0.893	0.909	0.774
K neighbors	0.900	0.867	0.641
DMSS Probe			
Model	Average AUC	Accuracy	Cohen's kappa
Logistic regression	0.897	0.802	0.643
Random forest	0.763	0.821	0.627
K neighbors	0.807	0.776	0.536

To further explore the effectiveness of the logistic regression model, we created 100 randomized data sets for each probe where the labels (i.e., point, set) were randomized. That is, for a given student response to a given probe, we randomized how their response would have been coded 100 times. We then built 100 models of these simulated data sets and computed the AUCs for these models. In Fig. 5, these randomized models (blue dots) for each probe are compared to our model of the data with the true labels (red dots). The AUC scores for the randomized data are near 0.5 (simple guessing), while, as expected, the real model is significantly higher.

We then turned to the ROC curves resulting from TF-IDF vectorization and the logistic regression algorithm. These ROC curves are shown in Fig. 6. The plot for each probe has three curves for the following reason. While an ROC curve compares true positives and false positives for a single classification, our system predicted three different categories of response. Thus, the positive category for the ROC curve could refer to any one of three categories. Thus, each curve in the plots in Fig. 6 represents a classification where the positive category is one of the three paradigm classifications, and the negative category is any of the other two classifications. This approach is known as a one-vs-all classification [44]. For example, the blue curves in Fig. 6 represent an ROC curve where the positive category is *setlike* and the negative category is *not setlike*. In this way, we actually measured three binary classifiers in order to

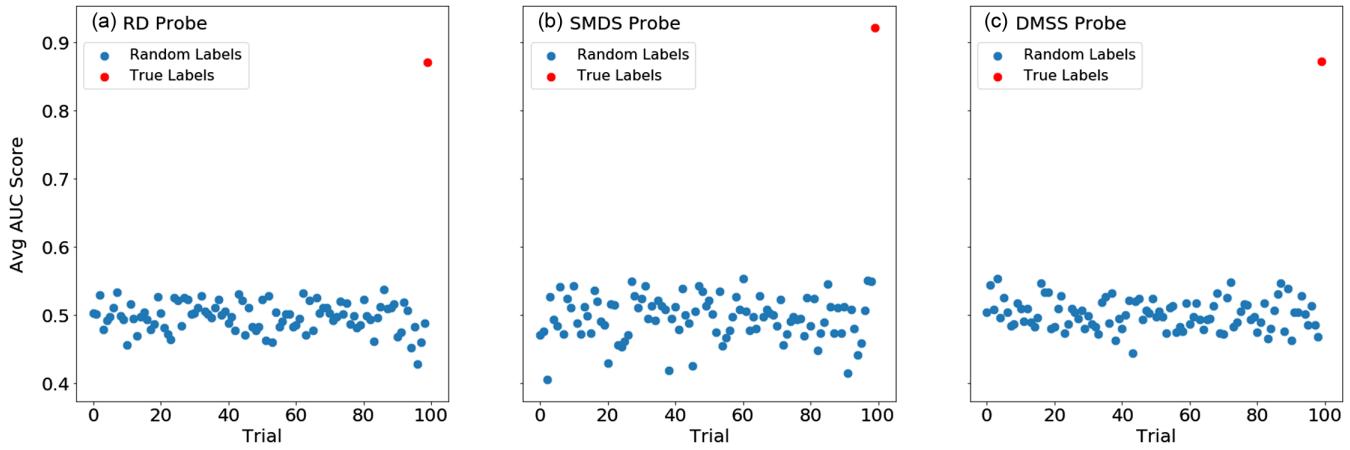

FIG. 5. Average AUC for 100 random trials and 1 true trial. For each probe, this graphic shows, in blue, the AUC (averaged over all three categories) produced when we randomize the input labels to the logistic regression model. The red data point is the average AUC when the correct labels are used on the training data, showing significant improvement and true pattern recognition.

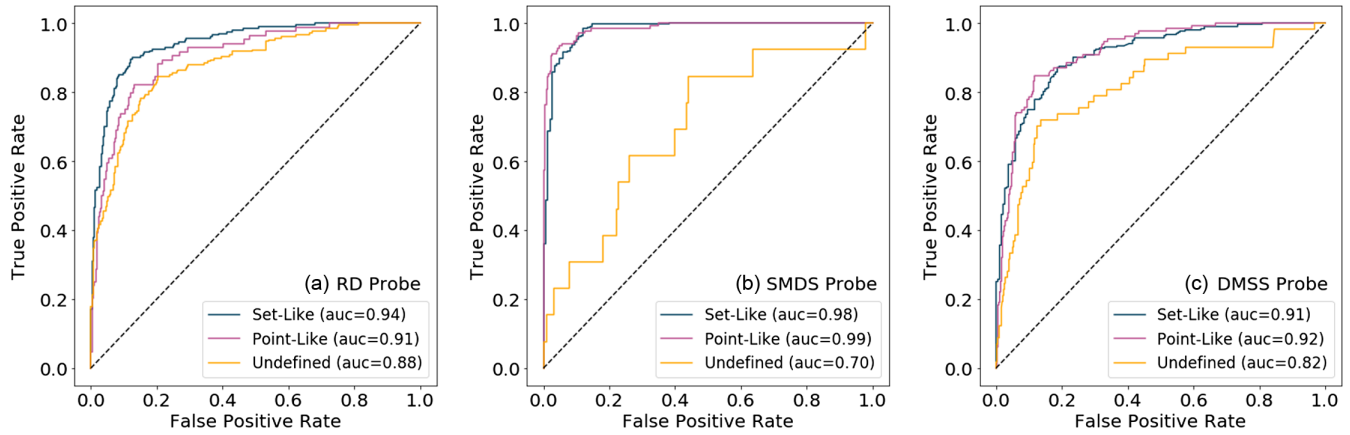

FIG. 6. ROC curves for the logistic regression classifier. After running the logistic regression algorithm on the training data for each probe, we plot our true positive rate and false positive rate from predicting classifications on the testing data. Using one-vs-all classification, each curve represents the classification of a student response as being of the labeled category, or not of the labeled category (e.g., a response being *setlike* or *not setlike*).

make ROC curves that can evaluate the system's performance in categorizing each different classification.

However, these ROC curves do not use the NCU scheme. The NCU scheme classifies the student responses in a manner closer to the intention of the original human categorization scheme, where undefined responses are simply defined as having indistinguishable point- and setlike attributes. To see how the NUC technique's performance compares to our previous one-vs-all scheme, we created an AUC comparison shown in Fig. 7.

Error bars represent the 95% confidence interval of the AUC scores of 500 random splits of the entire dataset into training (80% of the data) and testing (the remaining 20%).

The NCU scheme performs generally as well as the one-vs-all scheme, where undefined responses were treated as having their own characteristics. On the RD probe, the one-vs-all scheme performs better across all classes, although the difference is small. On SMDS, the one-vs-all scheme and NCU scheme do not perform noticeably differently on set- or pointlike, but the NCU scheme performs slightly better on undefined. For the DMSS probe, the one-vs-all scheme and NCU scheme are indistinguishable for set- and pointlike, but the system performs significantly better on undefined responses under the one-vs-all scheme.

Based on the comparisons of the two schemes and the fact that NCU more closely matches the process followed by a human coder, we choose the NCU scheme for the rest of the analysis. Next, we compared the NCU results from our model to that of a human categorization of the same data. We ran 100 different trials of the computational model, each performing slightly differently based on how we split the training and testing data, giving us a range of Cohen's kappa values instead of just one as seen in Fig. 8. Here, we see that for the RD and DMSS probes, the

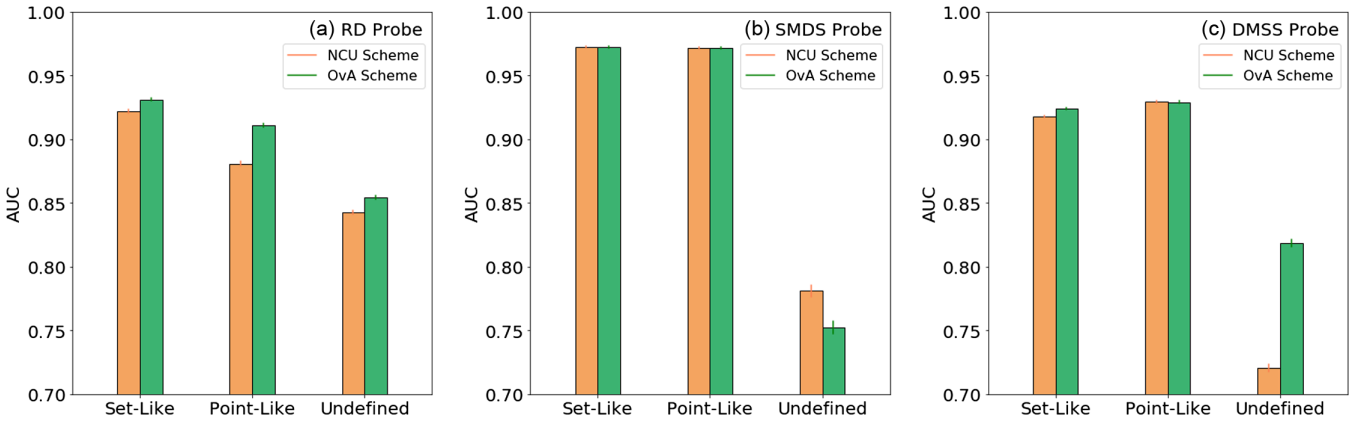

FIG. 7. AUC values and 95% confidence intervals (error bars) comparing the AUC of the NCU scheme to the one-vs-all (OvA) classification scheme. These results were determined from 500 different splits of the data into training and testing sets, and indicate whether the NCU scheme performs differently than the one-vs-all scheme.

Cohen's kappa value between two humans categorizing the same data is within the range of the Cohen's kappa value of our model and the human that originally categorized the data.

On the SMDS probe, the average Cohen's kappa score for our model and a single human coder is higher than that of the two humans that originally coded the data (Fig. 8). It might appear that the model is outperforming a human or over-performing generally. But, the data used by the model were coded fully by one human with the second coder comparing their work to the first [18]. Thus, our model is achieving high agreement with the single coder (Sec. III C).

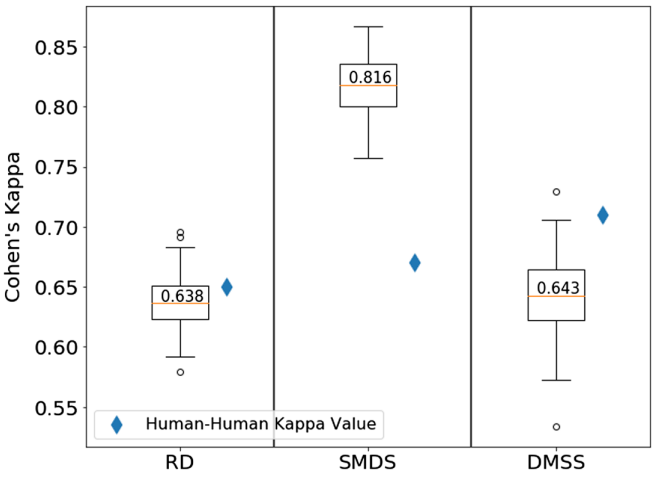

FIG. 8. Comparison of interrater reliability of the machine learning system and a human rater. Cohen's kappa value ranges are shown comparing the machine learning system to the human categorization. Diamonds represent the kappa values between the original human categorization and a different human categorization. Generally, the machine learning system achieved the same level of agreement with the original human rater as the other human.

B. Model explanation

To gain some insight into how our classifier was working, we investigated the features, or words, that had the largest magnitude of their feature coefficients. Figure 9 shows the coefficient, or relative weight, of some of the point- and setlike features for each probe. We plot the 10 features with the highest magnitude coefficients. In general, our model was fitting on words that are associated with the concept of measurement. The top features for the RD and DMSS probes consist mostly of words associated with discussions of measurement in experiments. The SMDS probe has slightly fewer such words, as words like *less* and *trial* appear in the top features. There are also large magnitude coefficients for multiple choice A and setlike responses, and multiple choice B and pointlike responses.

V. DISCUSSION

Across several metrics and approaches to measuring the performance of our system, we observe that it performs significantly better than the baseline of random guessing. This result is supported by the significantly higher AUC score of our trained system in Fig. 5, and by the various ROC curves in Fig. 6 that all lie far above the diagonal. These two results combined give us confidence that our system is learning meaningful patterns in the data.

The prevalence of words related to measurement and uncertainty in the high-magnitude-coefficient features in Fig. 9 give further confidence that the system is sensitive to the same things that are meaningful to a human coder.

The RD and DMSS probes generally fit on words that humans would associate with statistical measurement uncertainty, such as *average*, *range*, *distribution*, and *spread*, for setlike categorizations. Other words, such as *outlier*, *confirm*, *percent* (as in "percent agreement" or "percent error"), and *human* (as in "human error") are strongly weighted towards pointlike categorizations. Generally, these setlike words mirror the language that

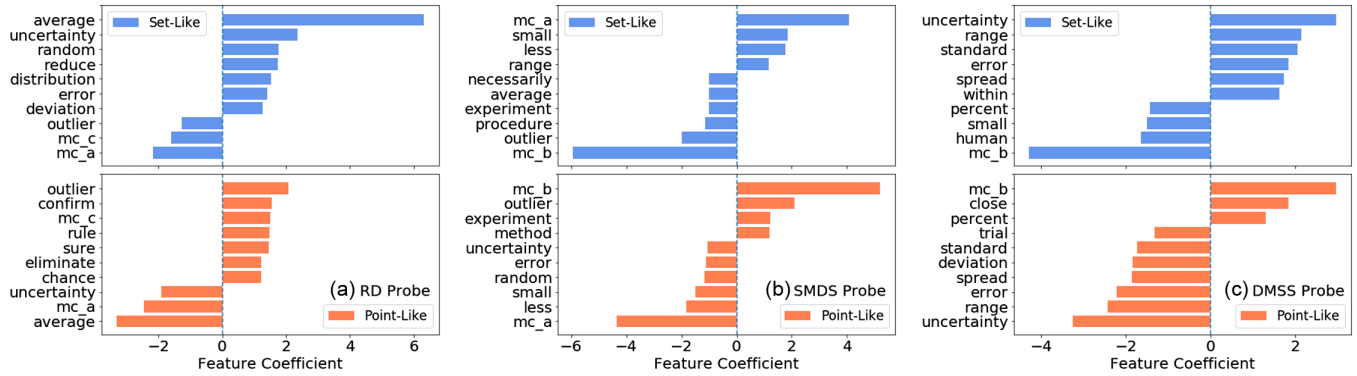

FIG. 9. The high-magnitude-coefficient features and their unpenalized beta coefficients for each probe. The vertical axis lists different features, or words, that are important in categorizing a student response. The horizontal axis shows their relative weight, or beta coefficient. The words mc_a, mc_b, and mc_c correspond to the selection of multiple choice answer A, B, and C, respectively.

lab instructors use when communicating about measurement uncertainty, whereas these pointlike words represent the ideas that lab instructors use when talking about concepts that hinder students' learning [45]. The correspondence between these high-magnitude coefficient features and instructors' language aligns with previous research on the PMQ that identified setlike reasoning as more expertlike [21].

On the other hand, the SMDS probe has more top features that are not necessarily relevant words in the context of measurement, like *process* and *less*. There is high weight put on the multiple choice features, though, as indicated by the features named *mc* followed by either *a*, *b*, or *c* depending on what the student chose. This high weighting of the multiple choice decision indicates that much of the classification of responses to this probe can be determined by the multiple choice response. For this probe, there is built-in paradigm knowledge in the choices themselves, which allows the system to put less emphasis on the students' written explanations. We note, however, that this strong relationship between the multiple choice response and the classification is unique to just the SMDS probe in our dataset.

For the other probes, the multiple choice response carried less information relative to the weighting of the written explanations.

We now turn to discussing the overall goal of creating a system that mirrors the behavior of a human coder. That goal motivated our use of the NCU scheme. We found that the NCU scheme, overall, performed as well as the first scheme we created. We therefore transitioned to the NCU classification scheme to more closely follow the approach to undefined responses in the original PMQ codebook and analysis.

Namely, when a human categorizes a response, they were not looking for specific traits and characteristics of an *undefined* category, like they were with a setlike or pointlike category. Rather, they determined that a response did not show evidence of setlike or pointlike thinking, or that it shows significant evidence of both. The NCU scheme reflects this approach. However, in particular categories for some probes, the results based on this method of classification do not always perform as well as previous models, particularly on the undefined category of the DMSS probe. If such results are of critical importance to a study, one could consider switching back to the original scheme. Nevertheless, the fact that we observe results overall that are on par with the original classification scheme suggests that the underlying model is detecting features and meaning in students' responses in a way aligned with the human approach.

When we turn to directly comparing our system's performance to that of another human, we find that our system matches the original coding as well as, or better than, another human coder. Specifically, in Fig. 8, we can see that the Cohen's kappa values between the two human raters on this dataset for the RD and DMSS probes line up well with the Cohen's kappa range of values between the machine learning model (here treated as a rater) and a human. This comparison indicates that there is as much interrater reliability between our model and a human as there is between two humans categorizing the same data on these two probes. On the SMDS probe, we see a kappa value for the machine learning system that is higher than the human-to-human kappa value on that probe. However, this comparison does not suggest that our system is performing better than a human in any meaningful sense. Rather, it merely indicates that our machine learning model agrees more with the original human that categorized the data than with the second human that categorized the data.

We have no reason to believe that the original human coder was inherently better or worse at interpreting students' responses than the second human coder. Thus, when it comes to understanding students' reasoning, the

fact that the system matched the first coder better than the second should not be taken to suggest that its results are of higher quality than any individual human, but rather that it is performing at the same level as a human would.

A. Limitations

Although the results we present here show promising evidence that a natural language processing system could be used to classify student open-response data, there are still some limitations of our model. Most of the model's failures can be seen in the *undefined* ROC curve of the SMDS probe in Fig. 6. This curve is the lowest performing (AUC = 0.70) out of any from the three probes. We speculate that this low performance can be attributed to mostly the lack of data from the undefined category for the SMDS probe. This lack of data means there is less for the model to learn on. If we had a more balanced dataset for the SMDS probe, we may see more consistent performance akin to the RD or DMSS probe. Because of the nature of the PMQ, however, it is possible that there will always be such an imbalance in the proportion of responses for our student population, as some PMQ probes may not elicit an undefined response from our students. From a survey design perspective, such a situation suggests that the SMDS probe is performing well, as it allows researchers to unambiguously characterize students' reasoning into the point or the set paradigm. In that case, an ideal dataset would be expected to have relatively few undefined responses, making the performance of the SMDS *unknown* ROC curve less of a concern.

A more fundamental limitation of our system is the establishment of ground truth. A machine learning model built on an objective ground truth will often outperform one built on a subjective ground truth. For example, a system built to classify handwritten digits has a higher ceiling of performance because there is little subjectivity as to the validity of the true training label. However, in our context, our ground truth is built on a human's categorization of the dataset, which may be different than another human's interpretation of the data. Our ground truth can only ever be a human's subjective categorization, as we can never fully and unambiguously know the inner thought processes of a student as they reason about measurement uncertainty. Therefore, our system can only ever be trained to be a model of particular human coder.

B. Outlook

We have demonstrated that we can train a NLP model to categorize open-ended responses to the Physics Measurement Questionnaire essentially as well as a pair of human coders. In so doing, we have taken some of the first steps to use NLP to analyze student work in physics. The fact that our model aligns well with human coding is promising for researchers and instructors alike. As the research continues, we might begin to use NLP to provide formative feedback to large classrooms of physics students by investigating data from digital open-response systems. NLP could be used to more efficiently understand how students' responses to open-ended physics questions are organized into common clusters in order to improve instruction. Some may suggest NLP could also, in some cases, replace the current approaches to assessing student learning by automatically analyzing and scoring student work. Here, we should be cautious.

NLP is not a panacea for physics instruction. In fact, like every quantitative approach to modeling data, it is subject to bias, both in terms of the data used to develop the model and the ways we might interpret the results. Therefore, we must be careful both with what data we are using to build our NLP model and with how we interpret the results. Nonetheless, we do see many relevant applications for NLP that the physics education research community could explore. As Science, Technology, Engineering, and Mathematics (STEM) programs have expanded, more students are enrolled in physics courses than in the past. In particular, enrollments in the typical two-semester introductory sequence have continued to grow [46,47]. With the constant pressure to reduce departmental budgets, there might be real pressure on some physics programs to increase student-to-instructor ratios. In confronting this reality, programs could be forced to reduce the time or energy available to provide good formative feedback to students on their progress. Here, NLP might be able to help physics instructors continue to deliver high quality instruction. For example, NLP could provide a first cut of the potential clusters of student responses that instructors can review to provide common formative feedback and individualized feedback as needed. That is, rather than having students in large-enrollment courses respond to conceptual multiple-choice questions, students could turn in typewritten long form answers that contain more depth and nuance than a close-form question. Such long-form answers could then be preprocessed with a NLP model, which an instructor could then further review. It is worth reiterating that the instructor would still need to exercise judgment with regard to any bias in their original data used to train the model, both in terms of how that bias could affect their model and in how they interpret the results.

Our example above shows one potential use case for NLP in physics education. We were careful to characterize it in two ways: (i) as a first cut, and (ii) as formative feedback. Our first point is important because NLP, especially as it stands currently, is not nuanced enough to completely replace feedback from a human instructor. The second is important because using a NLP model to assign grades to individual students is problematic. Especially if the NLP system is analyzing responses from

a research-based assessment instrument, as we did in this work, the content of the responses to such surveys should not be used to influence students' course grades [48]. Moreover, any NLP model is subject to statistical biases, which are the direct result of biases in the data on which that model was trained [49]. As an example, the majority of respondents studied here identified as white, as shown in Table I. As a result, the features and weightings in our NLP model would be expected to have a bias towards the written language used by white students, simply because we have significantly more data from white students. Thus, ample caution should be used when applying our model to a different population than the one on which it was trained.

Students who have difficulties expressing themselves in written form, students who are learning English as an additional language, and students with other dimensions of ability and identity could be negatively affected if responses from students sharing those identities are not well represented in the training dataset.

Overall, while we navigate potential research applications and use cases, the ethical considerations in using NLP should weigh heavily in our decisions, and we should be careful to understand and characterize the potential biases in our data that might lead to problematic results down the road.

VI. CONCLUSION

We have developed a NLP system that performs well in categorizing student responses to the PMQ across the three probes that we studied, both in comparison to a random baseline, but moreover in comparison to an independent human coder.

In particular, the interrater reliability between our system and the original human categorization showed significant agreement between the two on the RD and DMSS probes, and excellent agreement on the SMDS probe.

By performing this analysis automatically, our system drastically cuts down on the time and labor required to use the PMQ to study learning in a physics lab course.

In the future, one could use fundamentally different machine learning techniques like neural networks or unsupervised learning to classify open-ended written survey responses. These methods are a significant shift in the structure of our current model, and thus are outside the scope of the current investigation. Instead, we anticipate carrying this work forward by applying our model to PMQ data from a different student populations. This follow-up will allow us to see how the system's performance is affected by data coming from a different course environment, where the instructors may teach the material slightly differently, and students may have different backgrounds.

ACKNOWLEDGMENTS

We acknowledge Rajarshi Basak for valuable input and mentorship regarding NLP and machine learning. We acknowledge Robert Hobbs for producing the human-coded dataset that was central to this work. This work was supported by the National Science Foundation (Grant No. PHY-1734006) and by Michigan State's Lappan-Philips Foundation. This project has received support from the INTPART project of the Research Council of Norway (Grant No. 288125), the Olav Thon Foundation, and the Norwegian Agency for International Cooperation and Quality Enhancement in Higher Education (DIKU), which supports the Center for Computing in Science Education. The initial idea for this project was suggested by B. P.; further discussions around the idea occurred with H. J. L. and with J. M. A., J. W. wrote all of the code and performed all of the analysis, with the joint mentorship of B. P., J. M. A., M. D. C., and H. J. L., B. P. acted as the "second human" coder for interrater reliability comparison. J. W. wrote the initial drafts of this manuscript, which was finished by B. P. and H. J. L. All authors contributed to editing of the manuscript.

APPENDIX: LENGTH OF STUDENT RESPONSES

Table VI shows descriptive statistics concerning the distributions of the lengths of responses to the students' written explanations to the PMQ probes analyzed here. The differences between the means and the medians, in all cases, suggest that the distributions are skewed towards longer responses. Written responses, such as those used here, must be of sufficient length to carry enough meaning to by analyzed either by a human coder or a NLP system.

TABLE VI. The mean, median, and standard deviation (st. dev.) of the lengths of the written explanations to each probe. These lengths are calculated by the number of characters and by the number of words. The uncertainties on the means are 95% confidence intervals (CIs).

	Number of characters			Number of words		
	Mean ± CI	Median	St. dev.	Mean ± CI	Median	St. dev.
RD	101 ± 2	81	73	17.9 ± 0.4	14	13
SMDS	101 ± 2	84	74	18.2 ± 0.4	15	13
DMSS	96 ± 2	80	67	17.1 ± 0.4	14	12

[1] A. Madsen, S. B. McKagan, and E. C. Sayre, Resource Letter RBAI-1: Research-cased assessment instruments in physics and astronomy, Am. J. Phys. **85**, 245 (2017).

[2] A. Madsen, S. B. McKagan, E. C. Sayre, and C. A. Paul, Resource Letter RBAI-2: Research-based assessment instruments: Beyond physics topics, Am. J. Phys. **87**, 350 (2019).

[3] L. Ding, Theoretical perspectives of quantitative physics education research, Phys. Rev. Phys. Educ. Res. **15**, 020101 (2019).

[4] R. H. Nehm and H. Haertig, Human vs. computer diagnosis of students' natural selection knowledge: Testing the efficacy of text analytic software, J. Sci. Educ. Technol. **21**, 56 (2012).

[5] V. Kagklis, A. Karatrantou, M. Tantoula, C. T. Panagiotakopoulos, and V. S. Verykios, A learning analytics methodology for detecting sentiment in student fora: A case study in distance education, Eur. J. Open, Distance E-learning **18**, 74 (2015).

[6] R. Bajaj and V. Sharma, Smart education with artificial intelligence based determination of learning styles, Procedia Comput. Sci. **132**, 834 (2018).

[7] T. D. Ullmann, Automated analysis of reflection in writing: Validating machine learning approaches, Int. J. Artif. Intell. Educ. **29**, 217 (2019).

[8] A. Amigud, Cheaters on twitter: An analysis of engagement approaches of contract cheating services, Studies Higher Educ. **45**, 692 (2020).

[9] T. O. B. Odden, A. Marin, and M. D. Caballero, Thematic analysis of 18 years of physics education research conference proceedings using natural language processing, Phys. Rev. Phys. Educ. Res. **16**, 010142 (2020).

[10] T. O. B. Odden, A. Marin, and J. L. Rudolph, How has science education changed over the last 100 years? An analysis using natural language processing, Sci. Educ. **105**, 653 (2021).

[11] P. Wulff, D. Buschhüter, A. Westphal, A. Nowak, L. Becker, H. Robalino, M. Stede, and A. Borowski, Computer-based classification of preservice physics teachers' written reflections, J. Sci. Educ. Technol. **30**, 1 (2021).

[12] F. Lubben, B. Campbell, A. Buffler, and S. Allie, Point and set reasoning in practical science measurement by entering university freshmen, Sci. Educ. **85**, 311 (2001).

[13] B. Pollard and H. J. Lewandowski, Transforming a large introductory lab course: Impacts on views about experimental physics, in *Proceedings of the 2019 Physics Education Research Conference, Provo, UT* (AIP, New York, 2019).

[14] H. J. Lewandowski, D. R. Bolton, and B. Pollard, Initial impacts of the transformation of a large introductory lab course focused on developing experimental skills and expert epistemology, in *Proceedings of the 2018 Physics Education Research Conference, Washington, DC* (AIP, New York, 2018).

[15] H. J. Lewandowski, B. Pollard, and C. G. West, Using custom interactive video prelab activities in a large introductory lab course, in *Proceedings of the 2020 Physics Education Research Conference*, virtual conference (AIP, New York, 2020).

[16] B. Pollard, R. Hobbs, J. T. Stanley, D. R. Dounas-Frazer, and H. J. Lewandowski, Impact of an introductory lab course on students' understanding of measurement uncertainty, in *Proceedings of the 2017 Physics Education Research Conference, Cincinnati, OH* (AIP, New York, 2017), pp. 312–315.

[17] H. J. Lewandowski, R. Hobbs, J. T. Stanley, D. R. Dounas-Frazer, and B. Pollard, Student reasoning about measurement uncertainty in an introductory lab course, in *Proceedings of the 2017 Physics Education Research Conference, Cincinnati, OH* (AIP, New York, 2017), pp. 244–247.

[18] B. Pollard, R. Hobbs, D. R. Dounas-Frazer, and H. J. Lewandowski, Methodological development of a new coding scheme for an established assessment on measurement uncertainty in laboratory courses, in *Proceedings of the 2019 Physics Education Research Conference, Provo, UT* (AIP, New York, 2019).

[19] S. Allie, A. Buffler, B. Campbell, and F. Lubben, First-year physics students' perceptions of the quality of experimental measurements, Int. J. Sci. Educ. **20**, 447 (1998).

[20] A. Buffler, S. Allie, and F. Lubben, The development of first year physics students' ideas about measurement in terms of point and set paradigms, Int. J. Sci. Educ. **23**, 1137 (2001).

[21] A. Buffler, S. Allie, F. Lubben, and B. Campbell, Evaluation of a research-based curriculum for teaching measurement in the first year physics laboratory, in *Proceedings of the 4th Conference of the European Science Education Research Association, Noordwijkerhout, Netherlands, 2003* 09, 19 (2003).

[22] B. Campbell, F. Lubben, A. Buffler, and S. Allie, Teaching scientific measurement at university: Understanding students' ideas and laboratory curriculum reform, tool/images/281/people/buffler/physics_education/Monograph 2005.pdf **AJRMSTE**, 1 (2005).

[23] T. S. Volkwyn, S. Allie, A. Buffler, and F. Lubben, Impact of a conventional introductory laboratory course on the understanding of measurement, Phys. Rev. ST Phys. Educ. Res. **4**, 010108 (2008).

[24] B. Pollard, A. Werth, R. Hobbs, and H. J. Lewandowski, Impact of a course transformation on students' reasoning about measurement uncertainty, Phys. Rev. Phys. Educ. Res. **16**, 020160 (2020).

[25] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences* (L. Erlbaum Associates, Hillsdale, NJ, 1988).

[26] C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing* (MIT Press, Cambridge, MA, 1999).

[27] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit* (O'Reilly Media, Inc., Newton, MA, 2009).

[28] S. Bird, E. Loper, and E. Klein, *Natural Language Processing with Python* (O'Reilly Media Inc, Newton, MA, 2009).

[29] I. Feinerer and K. Hornik, tm: Text Mining Package (2020), r package version 0.7–8.

[30] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, Systematic review of research on artificial intelligence applications in higher education—where are the educators?, Int. J. Educ. Technol. Higher Educ. **16**, 39 (2019).

[31] R. H. Nehm and I. S. Schonfeld, Measuring knowledge of natural selection: A comparison of the cins, an open-response instrument, and an oral interview, J. Res. Sci. Teach. **45**, 1131 (2008).
[32] R. H. Nehm and I. S. Schonfeld, The future of natural selection knowledge measurement: A reply to Anderson *et al.* (2010), J. Res. Sci. Teach. **47**, 358 (2010).
[33] W. J. Wilbur and K. Sirotkin, The automatic identification of stop words, J. Inf. Sci. **18**, 45 (1992).
[34] J. Ramos *et al.*, Using TF-IDF to determine word relevance in document queries, in *Proceedings of the First Instructional Conference on Machine Learning* (Citeseer, 2003), Vol. 242, pp. 29–48.
[35] J. M. Aiken, R. D. Bin, H. Lewandowski, and M. D. Caballero, Framework for evaluating statistical models in physics education research, Phys. Rev. Phys. Educ. Res. **17**, 020104 (2021).
[36] E. Goel and E. Abhilasha, Random forest: A review, Int. J. Adv. Res. Comput. Sci. Sftwr. Engineer. **7**, 251 (2017).
[37] A. Kataria and M. Singh, A review of data classification using k-nearest neighbour algorithm, Int. J. Emerging Technol. Adv. Engin. **3**, 354 (2013).
[38] T. Fawcett, Roc graphs: Notes and practical considerations for researchers, Mach. Learn. **31**, 1 (2004).
[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Science and Business Media, New York, 2009).
[40] N. T. Young, G. Allen, J. M. Aiken, R. Henderson, and M. D. Caballero, Identifying features predictive of faculty integrating computation into physics courses, Phys. Rev. Phys. Educ. Res. **15**, 010114 (2019).
[41] J. M. Aiken, R. De Bin, M. Hjorth-Jensen, and M. D. Caballero, Predicting time to graduation at a large enrollment American university, PLoS One **15**, e0242334 (2020).
[42] F. Salehi, E. Abbasi, and B. Hassibi, The impact of regularization on high-dimensional logistic regression, arXiv:1906.03761.
[43] J. R. Wilson and B. Pollard, NLP PMQ Classification Notebooks (2020), https://github.com/Lewandowski-Labs-PER/eclass-public.
[44] R. Rifkin and A. Klautau, In defense of one-vs-all classification, J. Machine Learn. Res. **5**, 101 (2004).
[45] B. Pollard, R. Hobbs, R. Henderson, M. D. Caballero, and H. J. Lewandowski, Introductory physics lab instructors' perspectives on measurement uncertainty, Phys. Rev. Phys. Educ. Res. **17**, 010133 (2021).
[46] S. White and C. Langer Tesfaye, High school physics courses & enrollments (2014).
[47] S. Nicholson and P. J. Mulvey, Roster of physics departments with enrollment and degree data, 2019 (2020), https://www.aip.org/statistics/reports/roster-physics-departments-enrollment-and-degree-data-2020.
[48] A. Madsen, S. B. Mckagan, and E. C. Sayre, Best practices for administering concept inventories, Physics Teacher **55**, 530 (2017).
[49] R. J. Mooney, Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning, arXiv:cmp-lg/9612001.